



Delft University of Technology

Provable Privacy Advantages of Decentralized Federated Learning via Distributed Optimization

Yu, Wenrui; Li, Qiongxiu; Lopuhaa-Zwakenberg, Milan; Græsbøll Christensen, Mads; Heusdens, Richard

DOI

[10.1109/TIFS.2024.3516564](https://doi.org/10.1109/TIFS.2024.3516564)

Publication date

2025

Document Version

Final published version

Published in

IEEE Transactions on Information Forensics and Security

Citation (APA)

Yu, W., Li, Q., Lopuhaa-Zwakenberg, M., Græsbøll Christensen, M., & Heusdens, R. (2025). Provable Privacy Advantages of Decentralized Federated Learning via Distributed Optimization. *IEEE Transactions on Information Forensics and Security*, 20, 822-838. <https://doi.org/10.1109/TIFS.2024.3516564>

Important note

To cite this publication, please use the final published version (if applicable).

Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.

We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Provable Privacy Advantages of Decentralized Federated Learning via Distributed Optimization

Wenrui Yu^{ID}, Qiongxiu Li^{ID}, Member, IEEE, Milan Lopuhää-Zwakenberg^{ID},
 Mads Græsbøll Christensen^{ID}, Senior Member, IEEE,
 and Richard Heusdens^{ID}, Senior Member, IEEE

Abstract—Federated learning (FL) emerged as a paradigm designed to improve data privacy by enabling data to reside at its source, thus embedding privacy as a core consideration in FL architectures, whether centralized or decentralized. Contrasting with recent findings by Pasquini et al., which suggest that decentralized FL does not empirically offer any additional privacy or security benefits over centralized models, our study provides compelling evidence to the contrary. We demonstrate that decentralized FL, when deploying distributed optimization, provides enhanced privacy protection - both theoretically and empirically - compared to centralized approaches. The challenge of quantifying privacy loss through iterative processes has traditionally constrained the theoretical exploration of FL protocols. We overcome this by conducting a pioneering in-depth information-theoretical privacy analysis for both frameworks. Our analysis, considering both eavesdropping and passive adversary models, successfully establishes bounds on privacy leakage. In particular, we show information theoretically that the privacy loss in decentralized FL is upper bounded by the loss in centralized FL. Compared to the centralized case where local gradients of individual participants are directly revealed, a key distinction of optimization-based decentralized FL is that the relevant information includes differences of local gradients over successive iterations and the aggregated sum of different nodes' gradients over the network. This information complicates the adversary's attempt to infer private data. To bridge our theoretical insights with practical applications, we present detailed case studies involving logistic regression and deep neural networks. These examples demonstrate that while privacy leakage remains comparable in simpler models, complex models like deep neural networks exhibit lower privacy risks under decentralized FL. Extensive numerical tests further validate that decentralized FL is more resistant to privacy attacks, aligning with our theoretical findings.

Index Terms—Federated learning, privacy preservation, information theory, distribution optimization, ADMM, PDMM.

Received 12 July 2024; revised 26 October 2024 and 20 November 2024; accepted 29 November 2024. Date of publication 12 December 2024; date of current version 8 January 2025. The associate editor coordinating the review of this article and approving it for publication was Prof. Guowen Xu. (Wenrui Yu and Qiongxiu Li contributed equally to this work.) (Corresponding author: Qiongxiu Li.)

Wenrui Yu is with the CISPA Helmholtz Center for Information Security, 66123 Saarbrücken, Germany (e-mail: wenrui.yu@cispa.de).

Qiongxiu Li and Mads Græsbøll Christensen are with the Department of Electronic Systems, Aalborg University, 9220 Aalborg, Denmark (e-mail: qili@es.aau.dk; mgc@es.aau.dk).

Milan Lopuhää-Zwakenberg is with the Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7522 NB Enschede, The Netherlands (e-mail: m.a.lopuhää@utwente.nl).

Richard Heusdens is with Netherlands Defence Academy, 1781 AC Den Helder, The Netherlands, and also with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: r.heusdens@tudelft.nl).

Digital Object Identifier 10.1109/TIFS.2024.3516564

1556-6021 © 2024 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

I. INTRODUCTION

FEDERATED Learning (FL) enables collaborative model training across multiple participants/nodes/clients without directly sharing each node's raw data [1]. FL can operate on either a centralized/star topology or a decentralized topology, as shown in Figure 1 [2]. The prevalent centralized topology requires a central server that interacts with each and every node individually. The main procedure of a centralized FL protocol typically unfolds in three steps: 1) Nodes train local models based on their own private dataset and transmit model updates, such as gradients, to the server; 2) The server aggregates the local models to a global model and redistributes to the nodes; 3) Nodes update the local models based on the global model and send the model updates back to the server. The process is iteratively repeated until convergence. However, a centralized server is not always feasible due to its high communication demands and the need for universal trust from all nodes. In addition, it poses a risk of a single point of failure, making the network vulnerable to targeted attacks. As an alternative, decentralized FL circumvents these issues by facilitating direct data exchanges between (locally) connected nodes, thereby eliminating the need for a central server for model aggregation.

Decentralized FL protocols, also known as peer-to-peer learning protocols, fall into two main categories. The first involves average-consensus-based protocols. With these protocols, instead of sending model parameters to a central server, nodes collaborate together to perform model aggregation nodes in a distributed manner. The aggregation is typically done by partially averaging the local updates within a node's neighborhood. Examples of these protocols are the empirical methods where the aggregation is done using average consensus techniques such as gossiping SGD [3], D-PSGD [4], and variations thereof [5], [6]. The second category comprises protocols that are based on distributed optimization, referred to as optimization-based decentralized FL. These (iterative) methods directly formulate the underlying problem as a constrained optimization problem and employ distributed solvers like ADMM [7], [8], [9] or PDMM [10], [11], [12] to solve them. The constraints are formulated in such a way that, upon convergence, the learned models at all nodes are identical. Hence, there is no explicit separation between updating local models and the update of the global model, i.e., the three steps in centralized FL mentioned before are executed simultaneously.

Despite not directly sharing private data with servers or nodes, FL is shown vulnerable to privacy attacks as the exchanged information, such as gradients or weights, still

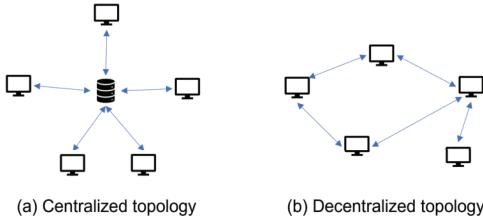


Fig. 1. Two topologies in federated learning.

poses a risk for privacy leakage. Existing work on privacy leakage predominantly focuses on the centralized case. A notable example is the gradient inversion attack [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], an iterative method for finding input data that produce a gradient similar to the gradient generated by the private data. Such attacks are based on the assumption that similar gradients are produced by similar data samples.

In exploring the privacy aspects of centralized versus decentralized FL, many works claim that the decentralized FL is more privacy-preserving than centralized FL without any privacy argument [23], [24], [25]. The main idea is that sensitive information, such as private data, model weights, and user states, can no longer be observed or controlled through a single server. However, recent empirical findings challenge such a claim, particularly for average-consensus-based decentralized FL protocols. It is shown in [26] that these protocols may not inherently offer better privacy protections over centralized FL, and might even increase susceptibility to privacy breaches. For instance, it has been shown that an arbitrary colluding client could potentially obtain the same amount of information as a central server in centralized FL when inverting input private data via gradients. In contrast, the privacy implications of optimization-based decentralized FL protocols have, to the best of our knowledge, been rarely investigated. This research gap is partly due to the fact that analytically tracking the privacy leakage in distributed algorithms, particularly over multiple iterations, is very challenging. The main difficulty lies in distinguishing and comprehending how information is correlated between these iterations.

A. Paper Contribution

In this paper, we take the first step to perform a theoretical privacy analysis of both centralized and decentralized FL frameworks by analyzing the information flow within the network. Our key contributions are summarized below:

- *Analytical privacy bounds of decentralized FL:* We conduct an information-theoretical privacy analysis of both centralized and decentralized FL protocols, using mutual information as a key metric. To the best of our knowledge, this is the first information-theoretical privacy analysis in this context. Notably, we derive two privacy bounds for the optimization-based decentralized FL and show that its privacy loss is upper bounded by the loss of centralized FL (Theorem 1 in Section V). We further exemplify the derived privacy gap through two applications, including logistic regression and Deep Neural Networks (DNNs).

- *Empirical validation through privacy attacks:* For DNN applications, we show that in the case of optimization-based decentralized FL, gradient inversion attacks can be applied to reconstruct the original input data, but the reconstruction performance is degraded compared to centralized FL due to the limited amount of information available to the adversary. A similar trend is also observed when evaluated using membership inference attacks. Overall, decentralized FL employing distributed optimization is shown to be less vulnerable to privacy attacks compared to centralized FL, consistent with our theoretical findings. This finding challenges the previous belief that decentralized FL offers no privacy advantages compared to centralized FL [26] (see Section VIII-E for a detailed explanation).

B. Outline and Notation

The paper is organized as follows. Section II reviews necessary fundamentals, and Section III introduces the involved metrics for quantifying privacy. Section IV introduces the optimization-based decentralized FL protocol. Section V analyzes the privacy of both centralized and decentralized FL protocols and states the main result. Section VI analyzes logistic regression example. Section VII, VIII analyze the application of DNNs. Conclusions are given in Section IX.

We use bold lowercase letters to denote vectors \mathbf{x} and bold uppercase letters for matrices \mathbf{X} . Calligraphic letters \mathcal{X} denotes sets. The i th entry of a vector \mathbf{x} is denoted \mathbf{x}_i . The superscript $(\cdot)^\top$ denotes matrix transposition. \mathbf{I} is used to denote the identity matrix of appropriate dimensions. $\mathbf{0}$ and $\mathbf{1}$ are the all-zero and all-one vectors. ∇ denotes the gradient. The value of the variable \mathbf{x} at iteration t is denoted as $\mathbf{x}^{(t)}$. We use $\|\cdot\|$ to indicate the ℓ_2 -norm and $\text{ran}(\cdot)$ and $\text{ker}(\cdot)$ to denote the range and kernel of their argument, respectively. For the sake of notational simplicity, we represent random variables using capital letters, regardless of whether its outcome is a scalar, a vector, or a matrix. $M(\mathbf{x}, i) := \mathbb{1}\{(\cdot, \mathbf{x}) \in \mathcal{D}\}$ is the indicator variable showing whether \mathbf{x} is in the dataset \mathcal{D} .

II. PRELIMINARIES

This section reviews the necessary fundamentals for the remainder of the paper.

A. Centralized FL

Without loss of generality, we focus on classification problems involving n nodes, each with its local dataset $\{(\mathbf{x}_{ik}, \ell_{ik}) : k = 1, \dots, n_i\}$, where $\mathbf{x}_{ik} \in \mathbb{R}^v$ represents an input sample, $\ell_{ik} \in \mathbb{R}$ is the associated label and n_i is the number of input samples. The dimension v of the data samples is application-dependent. Collecting the \mathbf{x}_{ik} s and ℓ_{ik} s, we define $\mathbf{x}_i = (\mathbf{x}_{i1}^\top, \dots, \mathbf{x}_{in_i}^\top)^\top$ and $\ell_i = (\ell_{i1}, \dots, \ell_{in_i})^\top$. Let $f_i(\mathbf{w}_i, (\mathbf{x}_i, \ell_i))$ denote the cost function of node i where $\mathbf{w}_i \in \mathbb{R}^u$ is the model weight to be learned from the input dataset (\mathbf{x}_i, ℓ_i) , whose dimension, again, depends on the application. In the remainder of the paper we will omit the (\mathbf{x}_i, ℓ_i) dependency for notational convenience when it is clear from the context and simply write $f_i(\mathbf{w}_i)$. A typical centralized FL protocol works as follows:

- 1) Initialization: at iteration $t = 0$, the central server randomly initializes the weights $\mathbf{w}_i^{(0)}$ for each node.
- 2) Local model training: at each iteration t , each user i first receives the model updates from the server and then computes its local gradient, denoted as $\nabla f_i(\mathbf{w}_i^{(t)})$, using its local data \mathbf{x}_i .
- 3) Model aggregation: the server collects these local gradients and performs aggregation to update the global model. The aggregation is often done by weighted averaging and typically uniform weights are applied, i.e., $\frac{1}{n} \sum_{i=1}^n \nabla f_i(\mathbf{w}_i^{(t)})$. Subsequently, each node i then updates its own model weight by

$$\mathbf{w}_i^{(t+1)} = \mathbf{w}_i^{(t)} - \frac{\mu}{n} \sum_{i=1}^n \nabla f_i(\mathbf{w}_i^{(t)}), \quad (1)$$

where μ is a constant controlling the convergence rate. The last two steps are repeated until the global model converges or until a predetermined stopping criterion is reached.

This algorithm is often referred to as the FedAvg [1].

B. Decentralized FL

Decentralized FL works for cases where a trusted centralized server is not available. In such cases, it works on a so-called distributed network which is often modeled as an undirected graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{1, 2, \dots, n\}$ representing the node set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ representing the edge set. $\mathcal{N}_i = \{j \mid (i, j) \in \mathcal{E}\}$ denotes the set of neighboring nodes of node i . In this decentralized setup, each node i can only communicate with its neighboring nodes $j \in \mathcal{N}_i$, facilitating peer-to-peer communication without any centralized coordination.

1) *Average Consensus-Based Approaches:* The model aggregation step requires all nodes' local gradients. Many decentralized FL protocols work by deploying distributed average consensus algorithms to compute the average of local gradients, i.e., computing $\frac{1}{n} \sum_{i=1}^n \nabla f_i(\mathbf{w}_i^{(t)})$ in Eq. (1) without any centralized coordination. Example average consensus algorithms are gossip [27] and linear iterations [28], which allow peer-to-peer communication over distributed networks.

The common decentralized FL often works similarly to the FedAvg algorithm, except for the step of model aggregation. For instance, D-PSGD [4], [29] uses gossip averaging with neighbors to implement the aggregation, i.e.,

$$\mathbf{w}_i^{(t+1)} = \mathbf{w}_i^{(t)} - \frac{\mu}{d_i} \sum_{j \in \mathcal{N}_i} \nabla f_j(\mathbf{w}_j^{(t)}), \quad (2)$$

where $d_i = |\mathcal{N}_i|$ is the degree of node i .

2) *Distributed Optimization-Based Approaches:* The goal of optimization-based decentralized FL is to collaboratively learn a global model, given the local datasets $\{(\mathbf{x}_i, \mathbf{t}_i) : i \in \mathcal{V}\}$, without any centralized coordination. The underlying problem can be posed as a constrained optimization problem given by

$$\begin{aligned} & \min_{\{\mathbf{w}_i : i \in \mathcal{V}\}} \sum_{i \in \mathcal{V}} f_i(\mathbf{w}_i), \\ & \text{subject to } \forall (i, j) \in \mathcal{E} : \mathbf{B}_{ilj} \mathbf{w}_i + \mathbf{B}_{jli} \mathbf{w}_j = \mathbf{0}, \end{aligned} \quad (3)$$

where \mathbf{B}_{ilj} and \mathbf{B}_{jli} define linear edge constraints. To ensure all nodes share the same model at convergence (consensus

constraints) we have $\mathbf{B}_{ilj} = -\mathbf{B}_{jli} = \pm \mathbf{I}$. In the following, we will use the convention that $\mathbf{B}_{ilj} = \mathbf{I}$ if $i < j$ and $\mathbf{B}_{ilj} = -\mathbf{I}$.

In what follows we will refer to centralized FL as CFL. While decentralized FL encompasses both average consensus-based and distributed optimization-based approaches (recall Section II-B), for simplicity, we will use the abbreviation DFL to specifically refer to the optimization-based decentralized FL as it is our main focus. We will differentiate between the two methods in contexts where such distinction is necessary to avoid confusion.

3) *Distributed Optimizers:* Given the optimization problem Eq. (3), many distributed optimizers, notably ADMM [30] and PDMM [10], [11] have been proposed. From a monotone operator theory perspective [11], [31], ADMM can be seen as a $\frac{1}{2}$ -averaged version of PDMM, allowing both to be analyzed within the same theoretical framework. Due to the averaging, ADMM is generally slower than PDMM, assuming it converges. Both ADMM and PDMM solve the optimization problem Eq. (3) iteratively, with the update equations for node i given by:

$$\mathbf{w}_i^{(t)} = \arg \min_{\mathbf{w}_i} \left(f_i(\mathbf{w}_i) + \sum_{j \in \mathcal{N}_i} z_{ilj}^{(t)} \mathbf{B}_{ilj} \mathbf{w}_i + \frac{\rho d_i}{2} \mathbf{w}_i^2 \right), \quad (4)$$

$$\forall j \in \mathcal{N}_i : z_{jli}^{(t+1)} = (1 - \theta) z_{jli}^{(t)} + \theta (z_{ilj}^{(t)} + 2\rho \mathbf{B}_{ilj} \mathbf{w}_i^{(t)}), \quad (5)$$

where ρ is a constant controlling the rate of convergence. The parameter $\theta \in (0, 1]$ controls the operator averaging with $\theta = \frac{1}{2}$ (Peaceman-Rachford splitting) yielding ADMM and $\theta = 1$ (Douglas-Rachford splitting) leading to PDMM. z is called auxiliary variable having entries indicated by z_{ilj} and z_{jli} , held by node i and j , respectively, related to edge $(i, j) \in \mathcal{E}$. Eq. (4) updates the local variables (weights) \mathbf{w}_i , whereas Eq. (5) represents the exchange of data in the network through the auxiliary variables z_{jli} .

C. Threat Models

We consider two types of adversary models: the eavesdropping and the passive (also known as honest-but-curious) adversary model. While eavesdropping can typically be addressed through channel encryption [32], it remains a pertinent concern in our context. This relevance stems from the nature of iterative algorithms, where communication channels are utilized repeatedly, continuously encrypting each and every message incurs high communication overhead. Therefore, in our framework, we assume that network communication generally occurs over non-secure channels, except for the initial network setup phase, details of which will be discussed later (see Section IV). The passive adversary consists of a number of colluding nodes, referred to as corrupt nodes, which comply with the algorithm instructions but utilize the received information to infer the private input data of the other so-called honest nodes. Consequently, the adversary has access to the following information: (a) all information gathered by the corrupt nodes, and (b) all messages transmitted over unsecured (i.e., non-encrypted) channels.

III. PRIVACY EVALUATION

When quantifying privacy, there are mainly two types of metrics: 1) empirical evaluation which assesses the

susceptibility of the protocol against established privacy attacks, and 2) information-theoretical metrics which offer a robust theoretical framework independent of empirical attacks. In this paper, we first evaluate privacy via an information-theoretical metric and then deploy empirical attacks to validate our theoretical results.

A. Information-Theoretical Privacy Metric

Among the information-theoretical metrics, popular ones include for example 1) ϵ -differential privacy [33], [34] which guarantees that the posterior guess of the adversary relating to the private data is only slightly better (quantified by ϵ) than the prior guess; 2) mutual information [35] which quantifies statistically how much information about the private data is revealed given the adversary's knowledge. In this paper, we choose mutual information as the information-theoretical privacy metric. The main reasons are the following. Mutual information has been proven effective in measuring privacy losses in distributed settings [36], and has been applied in various applications [37], [38], [39], [40], [41], [42], [43]. Secondly, mutual information is intrinsically linked to ϵ -differential privacy (see [44] for more details) and is more feasible to realize in practice [45], [46].

1) *Fundamentals of Mutual Information*: Given two (discrete) random variables X and Y , the mutual information $I(X; Y)$ between X and Y is defined as

$$I(X; Y) = H(X) - H(X|Y), \quad (6)$$

where $H(X)$ represents the Shannon entropy of X and $H(X|Y)$ is the conditional Shannon entropy, assuming they exist.¹ It follows that $I(X; Y) = 0$ when X and Y are independent, indicating that Y carries no information about X . Conversely, $I(X; Y)$ is maximal when Y and X share a one-to-one correspondence.

Denote \mathcal{V}_h and \mathcal{V}_c as the set of honest and corrupt nodes, respectively. Let \mathcal{O} denote the set of information obtained by the adversary. Hence, the privacy loss, measured by the mutual information between the private data \mathbf{x}_i of honest node $i \in \mathcal{V}_h$ and the knowledge available to the adversary, is given by

$$I(X_i; \mathcal{O}). \quad (7)$$

In the following, we use \mathcal{O}_{CFL} and \mathcal{O}_{DFL} to denote the sets of information that adversaries can obtain in the CFL and DFL scenarios, respectively.

B. Empirical Evaluation via Privacy Attacks

To complement our theoretical analysis, we incorporate empirical privacy attacks to validate our findings. In machine learning, based on the nature of the disclosed private data, privacy breaches typically fall into three types: membership inference [47], [48], [49], the property inference [48], [50] and the input reconstruction attack [51], [52], [53], [54], where the revealed information is membership (whether a particular data sample belongs to the training dataset or not), properties of the input such as age and gender, and the input training

¹In cases of continuous random variables we substitute both entropies by the differential entropy, thus $I(X; Y) = h(X) - h(X|Y)$.

data itself, respectively. Given that Eq. (7) measures how much information about the input training data is revealed, we align our empirical evaluation by mainly focusing on input reconstruction attacks. In FL, the gradient inversion attack has been extensively studied for its effectiveness in reconstructing input samples.

1) *Gradient Inversion Attack*: The gradient inversion attack typically works by iteratively refining an estimate of the private input data to align with the observed gradients generated by such data. For each node's local dataset (\mathbf{x}_i, ℓ_i) , the goal of the adversary is to recover the input data (\mathbf{x}_i, ℓ_i) based on the observed gradient. A typical setup is given by [13]:

$$(\mathbf{x}'^*, \ell'^*) = \arg \min_{\mathbf{x}'_i, \ell'_i} \|\nabla f_i(\mathbf{w}_i, (\mathbf{x}'_i, \ell'_i)) - \nabla f_i(\mathbf{w}_i, (\mathbf{x}_i, \ell_i))\|^2, \quad (8)$$

and many variants thereof are proposed [14], [15], [16], [17], [18], [19], [20], [21], [22]. To evaluate the quality of reconstructed inputs, we use the widely adopted structural similarity index measure (SSIM) [55] to measure the similarity between the reconstructed images and true inputs. The SSIM index ranges from -1 to 1 , where ± 1 signifies perfect resemblance and 0 indicates no correlation.

Analytical Label Recovery via Local Gradient: While it appears that both the input data \mathbf{x}_i and its label ℓ_i in Eq. (8) require reconstruction through optimization, existing work normally assumes that the label is already known. This is because the label can often be analytically inferred from the shared gradients [14], [49]. The main reason is as follows. Consider a classification task where the neural network has L layers and is trained with cross-entropy loss. Assume $n_i = 1$ for simplicity (one data sample at each node). Let $\mathbf{y} = (y_1, \dots, y_C)$ denote the outputs (logits), where y_i is the score (confidence) predicted for the i th class. With this, the cross-entropy loss over one-hot labels is given by

$$f_i(\mathbf{w}_i) = -\log \left(\frac{e^{y_i}}{\sum_j e^{y_j}} \right) = \log \left(\sum_j e^{y_j} \right) - y_i, \quad (9)$$

where $\log(\cdot)$ denotes the natural logarithm. Let $\mathbf{w}_{i,L,c}$ denote the weights in the output layer L corresponding to output y_c . The gradient of $f_i(\mathbf{w}_i)$ with respect to $\mathbf{w}_{i,L,c}$ can then be expressed as [14]:

$$\nabla' f_i(\mathbf{w}_{i,L,c}) \triangleq \frac{\partial f_i(\mathbf{w}_i)}{\partial \mathbf{w}_{i,L,c}} = \frac{\partial f_i(\mathbf{w}_i)}{\partial y_c} \frac{\partial y_c}{\partial \mathbf{w}_{i,L,c}} = g_c \mathbf{a}_{L-1}, \quad (10)$$

where \mathbf{a}_{L-1} is the activation at layer $L-1$ and g_c is the gradient of the cross entropy Eq. (9) with respect to logit c :

$$g_c = \frac{e^{y_c}}{\sum_j e^{y_j}} - \delta_{c,\ell_i}, \quad (11)$$

where δ_{c,ℓ_i} is the Kronecker-delta, defined as $\delta_{c,\ell_i} = 1$ when $c = \ell_i$ and $\delta_{c,\ell_i} = 0$ otherwise. Consequently, $g_c < 0$ for $c = \ell_i$ and $g_c > 0$ otherwise. Since the activation \mathbf{a}_{L-1} is independent of the class index c , the ground-truth label ℓ_i can be inferred from the shared gradients since $\nabla'^\top f_i(\mathbf{w}_{i,L,c}) \nabla' f_i(\mathbf{w}_{i,L,c}) = g_{\ell_i} g_{\ell_i} \|\mathbf{a}_{L-1}\|^2 < 0$ for $c \neq \ell_i$ and positive only for $c = \ell_i$. When dealing with $n_i > 1$, recovering labels becomes more challenging, yet feasible approaches are available. One example approach shown in [49] leverages the

fact that the gradient magnitude is proportional to the label frequency in untrained models. Hence, in the centralized FL case, label information can often be deduced from the shared gradients thus improving both the efficiency and accuracy of the reconstructed input \mathbf{x}'^* when compared to the real private input \mathbf{x} ; [14]. While for the decentralized FL protocol, we will show that the label information cannot be analytically computed for certain cases, thereby inevitably decreasing both the efficiency and reconstruction quality (see details in Remark 4).

2) *Membership Inference Attack*: The goal of membership inference attacks is to determine whether a specific data sample is part of the training set of a particular model. We deployed the gradient-based membership inference attacks proposed in [56] which are tailored for FL and have demonstrated superior performance compared to prior approaches such as the so-called loss-based [57] and modified entropy-based approaches [58]. The gradient-based approach uses cosine similarity between model updates and instance gradients, expressed as follows:

$$\begin{aligned} M(\mathbf{x}', i) \\ = \sum_{l' \in \mathbb{R}} \mathbb{1} \{ \text{cosim}(\nabla f_i(\mathbf{w}_i, (\mathbf{x}', l')), \nabla f_i(\mathbf{w}_i, (\mathbf{x}_i, \ell_i))) \geq \gamma \}. \end{aligned}$$

The latter one uses the indicator shown as

$$\begin{aligned} M(\mathbf{x}', i) = 1 \{ \|\nabla f_i(\mathbf{w}_i, (\mathbf{x}_i, \ell_i))\|_2^2 \\ - \left\| \nabla f_i(\mathbf{w}_i, (\mathbf{x}_i, \ell_i)) - \sum_{l \in \mathbb{R}} \nabla f_i(\mathbf{w}_i, (\mathbf{x}_i, \ell_i)) \right\|_2^2 > 0 \}. \end{aligned}$$

The corresponding results are presented in Section VIII-C and VIII-D.

IV. DFL USING DISTRIBUTED OPTIMIZERS

This section introduces distributed solvers considered in this work, explains pivotal convergence properties relevant to subsequent privacy analyses, and gives details of the decentralized protocol using distributed optimization techniques.

A. Differential A/PDMM

The optimality condition for Eq. (4) is given by²

$$\mathbf{0} = \nabla f_i(\mathbf{w}_i^{(t)}) + \sum_{j \in \mathcal{N}_i} \mathbf{B}_{i|j} z_{i|j}^{(t)} + \rho d_i \mathbf{w}_i^{(t)}. \quad (12)$$

Given that the adversary can eavesdrop all communication channels, by inspection of Eq. (12), transmitting the auxiliary variables $z_{j|i}$ would expose the gradient $\nabla f_i(\mathbf{w}_i^{(t)})$, as weight $\mathbf{w}_i^{(t)}$ can be determined from Eq. (5). Encrypting $z_{j|i}^{(t)}$ at every iteration would address this, albeit at prohibitive computational expenses. To circumvent this, only initial values $z_{j|i}^{(0)}$ are securely transmitted and $\Delta z_{j|i}^{(t+1)} = z_{j|i}^{(t+1)} - z_{j|i}^{(t)}$ being unencrypted in subsequent iterations [59], [60]. Consequently, upon receiving $\Delta z_{j|i}^{(t+1)}$, $z_{j|i}^{(t+1)}$ is reconstructed as

$$z_{j|i}^{(t+1)} = z_{j|i}^{(t)} + \Delta z_{j|i}^{(t+1)} = \sum_{\tau=1}^{t+1} \Delta z_{j|i}^{(\tau)} + z_{j|i}^{(0)}. \quad (13)$$

²Note that ADMM can also be applied to non-differentiable problems where the optimality condition can be expressed in terms of subdifferentials: $0 \in \partial f_i(\mathbf{w}_i^{(t)}) + \sum_{j \in \mathcal{N}_i} \mathbf{B}_{i|j} z_{i|j}^{(t)} + \rho d_i \mathbf{w}_i^{(t)}$.

Let t_{\max} denote the maximum number of iteration and denote $\mathcal{T} = \{0, 1, \dots, t_{\max}\}$. Hence, eavesdropping only uncovers

$$\{\Delta z_{j|i}^{(t+1)} : (i, j) \in \mathcal{E}, t \in \mathcal{T}\}, \quad (14)$$

and $z_{j|i}^{(t+1)}$ remains undisclosed unless $z_{j|i}^{(0)}$ is known.

B. DFL Using Differential A/PDMM

ADMM is guaranteed to converge to the optimal solution for arbitrary convex, closed and proper (CCP) objective functions f_i , whereas PDMM will converge in the case of differentiable and strongly convex functions [11]. Recently, it has been shown that these solvers are also effective when applied to non-convex problems like training DNNs [12]. Note that for complex non-linear applications such as training DNNs, although exact solutions of Eq. (4) are usually unavailable, convergence analysis of approximated solutions has been extensively investigated. For instance, it is shown in [12] that PDMM, using quadratic approximations, achieves good performance for non-convex tasks such as training DNNs. Moreover, convergence guarantees with quantized variable transmissions are investigated in [61].

Details of DFL using differential A/PDMM solvers are summarized in Algorithm 1. Note that at the initialization it requires that each node randomly initialize $z_{i|j}^{(0)}$ from independent distributions having variance σ_z^2 and sends it to neighbor $j \in \mathcal{N}_i$ via secure channels, also referred to as the subspace perturbation technique [60], [62]. The core concept involves introducing noise into the auxiliary variable z to obscure private data from potential exposure, while the convergence of \mathbf{w} is not affected. To explain this idea, consider Eq. (5) in a compact form:

$$z^{(t+1)} = (1 - \theta)z^{(t)} + \theta (\mathbf{P}z^{(t)} + 2c\mathbf{P}\mathbf{C}\mathbf{w}^{(t)}), \quad (15)$$

where $\mathbf{C} = [\mathbf{B}_+^\top, \mathbf{B}_-^\top]^\top$ and \mathbf{B}_+ and \mathbf{B}_- contains the positive and negative entries of \mathbf{B} , respectively. Additionally, \mathbf{P} is a permutation matrix that interchanges the upper half rows and lower half rows of the matrix it multiplies, leading to $\mathbf{P}\mathbf{C} = [\mathbf{B}_-^\top, \mathbf{B}_+^\top]^\top$. Denote $\Psi = \text{ran}(\mathbf{C}) + \text{ran}(\mathbf{P}\mathbf{C})$, its orthogonal complement is denoted by $\Psi^\perp = \text{ker}(\mathbf{C}^\top) \cap \text{ker}((\mathbf{P}\mathbf{C})^\top)$. Let Π_Ψ represent the orthogonal projection. We can then decompose z into components within Ψ and Ψ^\perp as $z^{(t)} = z_{\Psi}^{(t)} + z_{\Psi^\perp}^{(t)}$. Note that the component $z_{\Psi^\perp}^{(t)}$ is not null, requiring that the number of edges should be no smaller than the number of nodes. This condition is, however, not met in CFL with a star topology. Thus even though we deploy ADMM or PDMM for CFL, it would not give any privacy benefit.

It has been proven in [59] that

$$z_{\Psi^\perp}^{(t)} = \frac{1}{2} (z_{\Psi^\perp}^{(0)} + \mathbf{P}z_{\Psi^\perp}^{(0)}) + \frac{1}{2} (1 - 2\theta)^t (z_{\Psi^\perp}^{(0)} - \mathbf{P}z_{\Psi^\perp}^{(0)}).$$

Thus, for a given graph structure and θ , $z_{\Psi^\perp}^{(t)}$ depends solely on the initialization of the auxiliary variable $z^{(0)}$. Consequently, if $z_{i|j}^{(0)}$ is not known by the adversary, so does $z_{i|j}^{(t)}$ for subsequent iterations. This is key that privacy advantages can be provided when compared to centralized FL (in Remark 1 we will analyze the privacy loss for CFL when applying a similar trick).

Algorithm 1 Decentralized FL via A/PDMM

Each node i randomly initializes $z_{ilj}^{(0)}$ from independent distributions having variance σ_Z^2 and sends to neighbor $j \in \mathcal{N}_i$ via secure channels.

for $t = 0, 1, \dots$ **do**

for each node $i \in \mathcal{V}$ **in parallel do**

$\mathbf{w}_i^{(t)} = \arg \min_{\mathbf{w}_i} (f_i(\mathbf{w}_i) + \sum_{j \in \mathcal{N}_i} z_{ilj}^{(t)\top} \mathbf{B}_{ilj} \mathbf{w}_i + \frac{\rho d_i}{2} \mathbf{w}_i^2)$

for each $j \in \mathcal{N}_i$ **do**

$z_{jli}^{(t+1)} = (1 - \theta) z_{jli}^{(t)} + \theta (z_{ilj}^{(t)} + 2\rho \mathbf{B}_{ilj} \mathbf{w}_i^{(t)})$

$\Delta z_{jli}^{(t+1)} = z_{jli}^{(t+1)} - z_{jli}^{(t)}$

end for

end for

for each $i \in \mathcal{V}, j \in \mathcal{N}_i$ **do**

Node $_j \leftarrow \text{Node}_i(\Delta z_{jli}^{(t+1)})$

end for

for each $i \in \mathcal{V}, j \in \mathcal{N}_i$ **do**

$z_{jli}^{(t+1)} = z_{jli}^{(t)} + \Delta z_{jli}^{(t+1)}$

end for

end for

V. PRIVACY ANALYSIS

In this section, we conduct the comparative analysis of privacy loss in both CFL and DFL protocols, specifically focusing on the FedAvg algorithm and the decentralized approach introduced in Algorithm 1. For simplicity, we will primarily consider the case $\theta = 1$, i.e., PDMM, but the results can be readily extended to arbitrary $\theta \in (0, 1]$.

A. Privacy Loss of CFL

In CFL, the transmitted messages include the initial model weights $\mathbf{w}_j^{(0)}$, and the local gradients $\nabla f_j(\mathbf{w}_j^{(t)})$ at all iterations $t \in \mathcal{T}$ of all nodes $j \in \mathcal{V}$. Hence, by inspection of Eq. (1), we conclude that knowledge of local gradients and initial weights $\mathbf{w}_j^{(0)}$ is sufficient to compute all updated model weights $\mathbf{w}_j^{(t)}$ at every $t \in \mathcal{T}$. Hence, the eavesdropping adversary has the following knowledge

$$\{\nabla f_j(\mathbf{w}_j^{(t)}), \mathbf{w}_j^{(t)}\}_{j \in \mathcal{V}, t \in \mathcal{T}}. \quad (16)$$

The passive adversary, on the other hand, has the following knowledge

$$\{\mathbf{x}_j, \mathbf{w}_j^{(t)}, \nabla f_j(\mathbf{w}_j^{(t)})\}_{j \in \mathcal{V}, t \in \mathcal{T}}. \quad (17)$$

Combining both sets, the privacy loss, quantified by the mutual information between the private data \mathbf{x}_i and the knowledge available to the adversary (as in Eq. (7)), is given by

$$\begin{aligned} & I(X_i; \mathcal{O}_{\text{CFL}}) \\ &= I(X_i; \{X_j\}_{j \in \mathcal{V}}, \{\nabla f_j(\mathbf{w}_j^{(t)}), \mathbf{w}_j^{(t)}\}_{j \in \mathcal{V}, t \in \mathcal{T}}). \end{aligned} \quad (18)$$

Remark 1: We could securely transmit the initialized model weights $\mathbf{w}_j^{(0)}$ in CFL, analogous to the initial auxiliary variable $z_{jli}^{(0)}$ in DFL. However, such secure transmission would not reduce the privacy loss in Eq. (18). The main reason is that at convergence all local models will be identical, i.e., $\mathbf{w}_j^{(t_{\max})} = \mathbf{w}_k^{(t_{\max})}$ for $(j, k) \in \mathcal{E}$. Thus, as long as there is one

corrupt node, the passive adversary has knowledge of all $\mathbf{w}_j^{(t_{\max})}$ s. By inspecting Eq. (1) we can see that the difference $\mathbf{w}_j^{(t+1)} - \mathbf{w}_j^{(t)}$ at every iteration is known, and thus $\mathbf{w}_j^{(0)}$ for all $j \in \mathcal{V}$.

B. Privacy Loss of DFL

By inspection of Algorithm 1, the eavesdropping adversary can intercept all messages transmitted along non-secure channels, thus having access to:

$$\{\Delta z_{jli}^{(t+1)}\}_{(j,l) \in \mathcal{E}, t \in \mathcal{T}}. \quad (19)$$

For any edge in the network, the transmitted information will be known by the passive adversary as long as one end node is corrupt. Accordingly, we define $\mathcal{E}_h = \{(j, k) \in \mathcal{V}_h \times \mathcal{V}_h\}$, $\mathcal{E}_c = \mathcal{E} \setminus \mathcal{E}_h$ as the set of honest and corrupt edges, respectively. Given that the passive adversary can collect all information obtained by the corrupt nodes, by inspecting Algorithm 1 it thus has the knowledge of $\{\mathbf{x}_j\}_{j \in \mathcal{V}_c} \cup \{z_{jli}^{(0)}, \Delta z_{jli}^{(t+1)}\}_{(j,l) \in \mathcal{E}_c, t \in \mathcal{T}}$. Combining this with the eavesdropping knowledge in Eq. (19) we conclude that the adversary has the following knowledge:

$$\{\mathbf{x}_j\}_{j \in \mathcal{V}_c} \cup \{z_{jli}^{(0)}\}_{(j,l) \in \mathcal{E}_c} \cup \{\Delta z_{jli}^{(t+1)}\}_{(j,l) \in \mathcal{E}, t \in \mathcal{T}}.$$

The information loss of an honest node $i \in \mathcal{V}_h$'s private data is thus given by

$$\begin{aligned} & I(X_i; \mathcal{O}_{\text{DFL}}) \\ &= I(X_i; \{X_j\}_{j \in \mathcal{V}_c}, \{Z_{jli}^{(0)}\}_{(j,l) \in \mathcal{E}_c}, \{\Delta Z_{jli}^{(t+1)}\}_{(j,l) \in \mathcal{E}, t \in \mathcal{T}}), \end{aligned} \quad (20)$$

We first give some initial results on the information that can be deduced by the adversary.

Proposition 1: Let $\mathcal{G}_h = (\mathcal{V}_h, \mathcal{E}_h)$ be the subgraph of \mathcal{G} after eliminating all corrupt nodes. Let $\mathcal{G}_{h,1}, \dots, \mathcal{G}_{h,k_h}$ denote the components of \mathcal{G}_h and let $\mathcal{V}_{h,k}$ be the vertex set of $\mathcal{G}_{h,k}$. Without loss of generality, assume the honest nodes i belong to the first honest component, i.e., $i \in \mathcal{V}_{h,1}$. The adversary has the following knowledge about node $i \in \mathcal{V}_{h,1}$:

i) Noisy local gradients:

$$\forall t \in \mathcal{T} : \nabla f_i(\mathbf{w}_i^{(t)}) + \sum_{k \in \mathcal{V}_{h,1}} \mathbf{B}_{ik} z_{ik}^{(0)} \quad (21)$$

ii) Difference of local gradients:

$$\forall t \in \mathcal{T} : \nabla f_i(\mathbf{w}_i^{(t+1)}) - \nabla f_i(\mathbf{w}_i^{(t)}), \quad (22)$$

iii) The aggregated sum of local gradients in honest component $\mathcal{G}_{h,1}$:

$$\forall t \in \mathcal{T} : \sum_{j \in \mathcal{V}_{h,1}} \nabla f_j(\mathbf{w}_j^{(t)}). \quad (23)$$

Proof: See Appendix A. □

By inspecting Eq. (23), the sum of gradients reveals less information than the local gradient itself in contrast to the CFL case. We now proceed to present the main result of this paper. In particular, we will show that the information loss of DFL is dependent on the variance of initialized $z^{(0)}$, i.e., σ_Z^2 . Notably, in the special case where $\sigma_Z^2 = 0$, all local gradients will be exposed similar to the case of centralized FL. In contrast, when σ_Z^2 approaches infinity, the term Eq. (21) contains

no information about the private data or the local gradient, theoretically leading to $I(X_i; \nabla f_i(W_i^{(t)}) + \sum_{k \in \mathcal{N}_{i,h}} \mathbf{B}_{ilk} Z_{ilk}^{(0)}) = 0$. More specifically, we have the following privacy bounds.

Theorem 1 (Privacy bounds of DFL): We have

$$\begin{aligned} I(X_i; \mathcal{O}_{\text{CFL}}) &\stackrel{(a)}{\geq} I(X_i; \mathcal{O}_{\text{DFL}}) \\ &= I(X_i; \{X_j\}_{j \in \mathcal{V}_c}, \{Z_{j|k}^{(0)}\}_{(j,k) \in \mathcal{E}_c}, \{W_j^{(t)}\}_{j \in \mathcal{V}, t \in \mathcal{T}}, \\ &\quad \{\nabla f_j(W_j^{(t)})\}_{j \in \mathcal{V}_c, t \in \mathcal{T}}, \{Z_{j|k}^{(0)} - Z_{k|j}^{(0)}\}_{(j,k) \in \mathcal{E}_h}, \\ &\quad \left\{ \nabla f_j(W_j^{(t)}) + \sum_{k \in \mathcal{N}_{j,h}} \mathbf{B}_{ilk} Z_{ilk}^{(0)} \right\}_{j \in \mathcal{V}_h, t \in \mathcal{T}}) \end{aligned} \quad (24)$$

$$\begin{aligned} &\stackrel{(b)}{\geq} I(X_i; \{X_j\}_{j \in \mathcal{V}_c}, \{W_j^{(t)}\}_{j \in \mathcal{V}, t \in \mathcal{T}}, \{\nabla f_j(W_j^{(t)})\}_{j \in \mathcal{V}_c, t \in \mathcal{T}}, \\ &\quad \{\nabla f_j(W_j^{(t+1)}) - \nabla f_j(W_j^{(t)})\}_{j \in \mathcal{V}_h, t \in \mathcal{T}}, \\ &\quad \left\{ \sum_{j \in \mathcal{V}_{h,l}} \nabla f_j(W_j^{(t)}) \right\}_{1 \leq l \leq k_h, t \in \mathcal{T}}), \end{aligned} \quad (25)$$

where we have equality in (a) if $\sigma_Z^2 = 0$, and equality in (b) if $\sigma_Z^2 \rightarrow \infty$.

Proof: See Appendix B. \square

Hence, by inspecting the lower bound we can see that except for the knowledge of the corrupt nodes, i.e., $\{\mathbf{x}_j\}_{j \in \mathcal{V}_c}, \{\mathbf{w}_j^{(t)}\}_{j \in \mathcal{V}_c, t \in \mathcal{T}}, \{f_j(\mathbf{w}_j^{(t)})\}_{j \in \mathcal{V}_c, t \in \mathcal{T}}$, the revealed information includes the model weights of all honest nodes $\{\mathbf{w}_j^{(t)}\}_{j \in \mathcal{V}_h, t \in \mathcal{T}}$, gradient differences of each honest node over successive iterations $\{\nabla f_j(\mathbf{w}_j^{(t+1)}) - \nabla f_j(\mathbf{w}_j^{(t)})\}_{j \in \mathcal{V}_h, t \in \mathcal{T}}$ (note that $\cup_{1 \leq l \leq k_h} \mathcal{V}_{h,l} = \mathcal{V}_h$), and the sum of local gradients of the honest nodes $\{\sum_{j \in \mathcal{V}_{h,l}} \nabla f_j(\mathbf{w}_j^{(t)})\}_{1 \leq l \leq k_h, t \in \mathcal{T}}$ in each component.

Regarding the feasibility of the lower bound, we have the following remark.

Remark 2: It might seem impractical that the lower bound in Theorem 1 requires that the variance σ_Z^2 approaches infinity. For practical applications, however, like DNNs, a relatively small variance is already sufficient to make the leaked information in the noisy gradients negligible compared to gradient differences Eq. (22) and the gradient sum Eq. (23) (we will verify this claim in Section VII-C).

C. Privacy Gap Between CFL and DFL

Theorem 1 shows that the privacy loss in DFL is either less than or equal to that in CFL. This naturally leads to key questions: under what circumstances does this equality or inequality hold? To analyze the privacy gap we have the following result, showing that the privacy gap between DFL and CFL is dependent on how much more information about the private data X_i can the local gradients reveal given the knowledge of DFL, e.g., gradient differences and the gradient sum.

Corollary 1: Privacy gap between CFL and DFL If the lower bound Eq. (25) is achieved, for an honest node $i \in \mathcal{V}_h$'s private data, the privacy gap between CFL and DFL is given by

$$\begin{aligned} I(X_i; \mathcal{O}_{\text{CFL}}) - I(X_i; \mathcal{O}_{\text{DFL}}) &= I(X_i; \{\nabla f_j(W_j^{(t)})\}_{j \in \mathcal{V}_h, t \in \mathcal{T}} | \{X_j\}_{j \in \mathcal{V}_c}, \{W_j^{(t)}\}_{j \in \mathcal{V}, t \in \mathcal{T}}, \\ &\quad \{\nabla f_j(W_j^{(t+1)}) - \nabla f_j(W_j^{(t)})\}_{j \in \mathcal{V}_h, t \in \mathcal{T}}, \end{aligned}$$

$$\left\{ \sum_{j \in \mathcal{V}_{h,l}} \nabla f_j(W_j^{(t)}) \right\}_{1 \leq l \leq k_h, t \in \mathcal{T}} \) \quad (26)$$

Proof: See Appendix D. \square

Remark 3: The privacy gap, as defined in Eq. (26), depends on the number of corrupt nodes and narrows notably in the extreme case where only one honest node remains. By inspecting Eq. (26), we can see that the privacy gap is intrinsically linked to the sum of gradients from honest nodes within each honest component, specifically, $\{\sum_{j \in \mathcal{V}_{h,l}} \nabla f_j(W_j^{(t)})\}_{1 \leq l \leq k_h, t \in \mathcal{T}}$. As the number of honest nodes diminishes, the specificity of information conveyed by individual node gradients increases, consequently reducing the privacy gap. In the most extreme scenario, where only one node is honest, i.e., $\mathcal{V}_h = \{i\}$, the privacy gap reduces to zero since Eq. (26) = $I(X_i; \{\nabla f_i(W_i^{(t)})\}_{t \in \mathcal{T}} | \{\nabla f_i(W_i^{(t)})\}_{t \in \mathcal{T}}) = 0$.

In the following sections, we delve into two distinct cases to further investigate the privacy gap. First, we examine a straightforward logistic regression example, where it's possible to analytically calculate the privacy loss. In this instance, we find no discernible privacy gap. Next, we shift our focus to a more conventional application within FL: DNNs. Through extensive empirical analysis, we observe a notable privacy gap between CFL and DFL. This gap highlights DFL's reduced susceptibility to privacy attacks when compared to CFL. The code for this project is available on GitHub at <https://github.com/Wenrui-Yu/DFL-and-Privacy>

VI. LOGISTIC REGRESSION

Logistic regression is widely adopted in various applications and serves as a fundamental building block for complex applications. We will first give analytical derivations and then demonstrate numerical results.

A. Theoretical Analysis

Consider a logistic model with model parameters $\mathbf{w}_i \in \mathbb{R}^v$ (weights) and $b_i \in \mathbb{R}$ (bias) where each node has a local dataset $\{(\mathbf{x}_{ik}, \ell_{ik}) : k = 1, \dots, n_i\}$, where $\mathbf{x}_{ik} \in \mathbb{R}^v$ is an input sample, $\ell_{ik} \in \{0, 1\}$ is the associated label. In addition, let $y_{ik} = \mathbf{w}_i^T \mathbf{x}_{ik} + b_i$ denote the output of the model given the input \mathbf{x}_{ik} . Note that the bias term can be included in the weight vector. Here we explicitly separate the bias from the true network weights as it will lead to more insight into how to reconstruct the input data from the observed gradient information. Correspondingly, z_{ilj} is also separated into $z_{w,ilj}$ and $z_{b,ilj}$. With this, the loss (log-likelihood) function has the form

$$\begin{aligned} f_i(\mathbf{w}_i, b_i) &= - \sum_{k=1}^{n_i} \left(\ell_{ik} \log \frac{1}{1 + e^{-y_{ik}}} \right. \\ &\quad \left. + (1 - \ell_{ik}) \log \frac{e^{-y_{ik}}}{1 + e^{-y_{ik}}} \right). \end{aligned} \quad (27)$$

With this, Eq. (22) becomes

$$\begin{aligned} \frac{\partial f_i^{(t+1)}}{\partial \mathbf{w}_i} - \frac{\partial f_i^{(t)}}{\partial \mathbf{w}_i} &= \sum_{k=1}^{n_i} \left(\frac{1}{1 + e^{-y_{ik}^{(t+1)}}} - \frac{1}{1 + e^{-y_{ik}^{(t)}}} \right) \mathbf{x}_{ik} \\ &= - \sum_{j \in \mathcal{N}_i} \mathbf{B}_{ilj} \Delta z_{w,ilj}^{(t)} + \rho d_i(\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(t+1)}), \end{aligned} \quad (28)$$

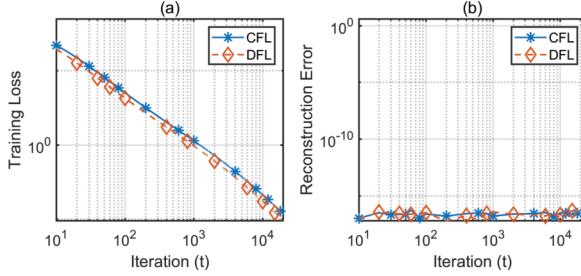


Fig. 2. Privacy comparisons of centralized and decentralized logistic regression. (a) Training loss and (b) Reconstruction error of input data as a function of iteration number (t) using CFL (blue color) and DFL (red color).

and

$$\begin{aligned} \frac{\partial f_i^{(t+1)}}{\partial b_i} - \frac{\partial f_i^{(t)}}{\partial b_i} &= \sum_{k=1}^{n_i} \left(\frac{1}{1 + e^{-y_{ik}^{(t+1)}}} - \frac{1}{1 + e^{-y_{ik}^{(t)}}} \right) \\ &= - \sum_{j \in \mathcal{N}_i} \mathbf{B}_{ilj} \Delta z_{b,ilj}^{(t)} + \rho d_i (b_i^{(t)} - b_i^{(t+1)}), \end{aligned} \quad (29)$$

where all terms in the RHS are known by the adversary as the differences of the local model $\mathbf{w}_i^{(t+1)} - \mathbf{w}_i^{(t)}$ can be determined from $\Delta z_{b,ilj}^{(t+1)} - \Delta z_{b,ilj}^{(t)}$ by considering two successive z updates of Eq. (5). As a special case where $n_i = 1$, Eq. (28) is just a scaled version of \mathbf{x}_{ik} where the scaling is given by Eq. (29). Hence, with gradient difference, we can analytically compute \mathbf{x}_{ik} as

$$\mathbf{x}_{ik} = \frac{- \sum_{j \in \mathcal{N}_i} \mathbf{B}_{ilj} \Delta z_{b,ilj}^{(t)} + \rho d_i (\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(t+1)})}{- \sum_{j \in \mathcal{N}_i} \mathbf{B}_{ilj} \Delta z_{b,ilj}^{(t)} + \rho d_i (b_i^{(t)} - b_i^{(t+1)})}.$$

Hence, in this case, one gradient difference at an arbitrary iteration is sufficient to reveal all information about the private data, i.e., $I(X_i; \nabla f_j(W_j^{(t+1)}) - \nabla f_j(W_j^{(t)})) = I(X_i; X_i)$ which is maximum. Thus, there is no privacy gap between CFL and DFL, i.e., Eq. (26) = 0. For the case of $n_i > 1$, the input \mathbf{x}_{ik} can also be reconstructed by searching for solutions that fit for given observations, i.e., Eq. (28) and Eq. (29) across iterations.

B. Convergence Behavior

To validate the theory presented above, we consider a toy example of a random geometric graph of $n = 60$ nodes randomly distributed in the unit cube having a communication radius $r = \sqrt{2 \log n / n}$ to ensure connectivity with high probability [63]. Detailed settings can be found in the Appendix E. In Figure 2(a) we demonstrate the convergence performances of both centralized and decentralized protocols, i.e., the loss Eq. (27), averaged over all nodes, as a function of the iteration t . We can see that both methods have a similar convergence rate.

C. Privacy Gap Between CFL and DFL

To evaluate the performance of input reconstruction, we define the reconstruction error as the average Euclidean distance between the reconstructed samples, denoted as $\hat{\mathbf{x}}_{ik}$, and the original data samples \mathbf{x}_{ik} given by $\frac{1}{n} \sum_{i=1}^n \|\hat{\mathbf{x}}_{ik} - \mathbf{x}_{ik}\|_2$. The corresponding errors are plotted as a function of iteration number in Figure 2(b). We can see that, the reconstruction

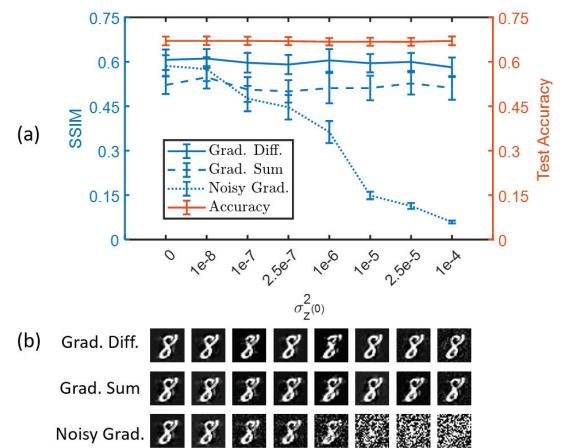


Fig. 3. (a) Averaged SSIM of reconstructed inputs by inverting noisy gradients, gradient differences, and the gradient sum (blue lines) and test accuracy (red line) for different variances of initialized auxiliary variable $z^{(0)}$: $\sigma_z^2 = 0, 10^{-8}, 10^{-7}, 2.5 \times 10^{-7}, 10^{-6}, 10^{-5}, 2.5 \times 10^{-5}$ and 10^{-4} . (b) Sample examples of reconstructed inputs for each case.

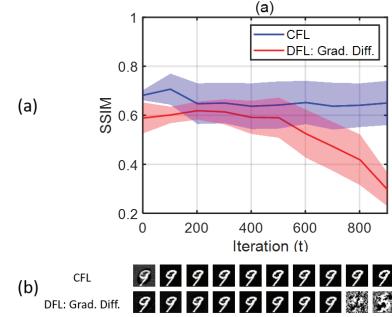


Fig. 4. Performance of reconstructed inputs via inverting gradients (CFL) and gradient differences (DFL) in terms of iterations t : (a) Averaged SSIM (solid lines) of all reconstructed inputs along with the corresponding standard derivation (shadows), (b) sample examples of reconstructed inputs at iteration number $t = 1, 100, \dots, 900$.

error of DFL, using gradient differences, has the same level of error as the CFL case across all iterations. Hence, we conclude that for the logistic regression example, there is no privacy gap between CFL and DFL, i.e., Eq. (26) = 0, aligning with our theoretical result.

VII. DEEP NEURAL NETWORKS I: CONVERGENCE AND GRADIENT INVERSION ATTACKS

For DNNs, it is challenging to give an analytical analysis like the above logistic regression example. We resort to empirical assessments of privacy leakage through privacy attacks. As we shall see, the empirical evaluation results are consistent with our theoretical claims.

A. Convergence Behavior

To test the performance of the introduced DFL protocol, we make comparison between the training process of CFL and DFL. The detailed settings and results are shown in Appendix E. Notably, the performance of the decentralized protocol closely aligns with that of the centralized approach.

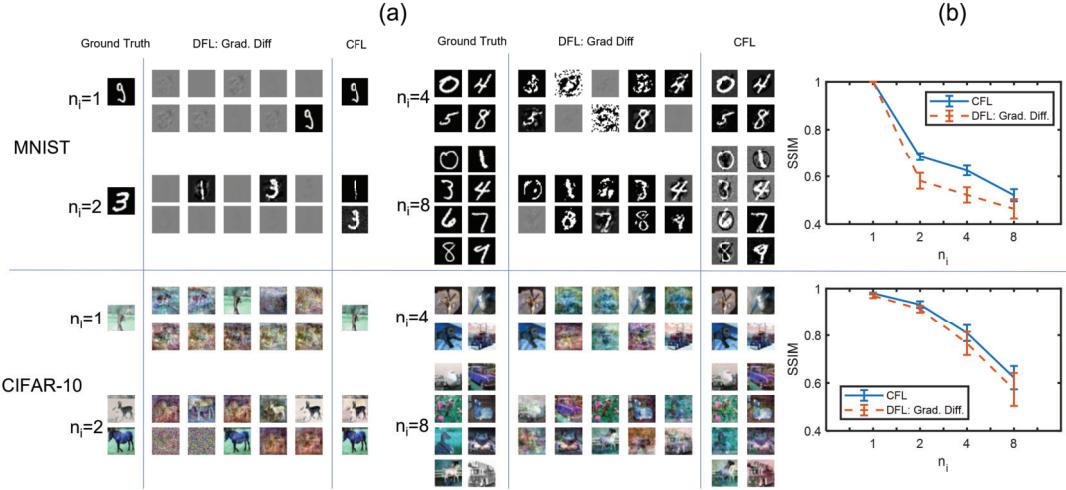


Fig. 5. Performance comparisons of CFL and DFL via inverting inputs from gradient differences: (a) Samples images of ground truth and reconstructed inputs, (b) SSIM comparisons of all reconstructed inputs for different batch size $n_i = 1, 2, 4, 8$ using two datasets MNIST (top) and CIFAR-10 (bottom), respectively.

This aligns with previous research findings [12], which suggests that decentralized protocols perform comparably to centralized ones, particularly in scenarios with independently and identically distributed data. The subsequent section will focus on evaluating their privacy via gradient inversion attack and membership inference attack, highlighting the privacy advantages inherent in DFL.

B. Gradient Inversion Attack in DFL

Gradient inversion attacks, which aim to reconstruct data from gradients, are applied here to assess vulnerabilities in the model. Recall that in Proposition 1 we identified three types of information directly related to the honest node's local gradient: noisy gradients Eq. (21), gradient differences Eq. (22), and the gradient sum Eq. (23). Similar to the traditional gradient inversion attack described in Eq. (8), inputs can be inverted from them as well. As an example with the gradient sum Eq. (23) the corresponding optimization problem can be formulated as

$$\{(\mathbf{x}'_i^*, \boldsymbol{\ell}'_i^*)\}_{i \in \mathcal{V}_h} = \arg \min_{\mathbf{x}'_i, \boldsymbol{\ell}'_i} \left\| \sum_{i \in \mathcal{V}_{h,1}} \nabla f_i(\mathbf{w}_i^{(t)}, (\mathbf{x}'_i, \boldsymbol{\ell}'_i)) - \sum_{i \in \mathcal{V}_{h,1}} \nabla f_i(\mathbf{w}_i^{(t)}, (\mathbf{x}_i, \boldsymbol{\ell}_i)) \right\|^2, \quad (30)$$

Although the gradient inversion attack in the decentralized case looks similar to the traditional centralized case, there are some important differences, in particular with respect to label recovery and fidelity of reconstructed inputs, discussed in the following remarks.

Remark 4: Analytical label recovery is no longer applicable given gradient differences Eq. (22). With gradient differences, Eq. (10) explained in Section III-B.1 becomes

$$\nabla f(\mathbf{w}_{i,L,c}^{(t)}) - \nabla f(\mathbf{w}_{i,L,c}^{(t+1)}) = g_c^{(t)} \mathbf{a}_{L-1}^{(t)} - g_c^{(t+1)} \mathbf{a}_{L-1}^{(t+1)} = \left(\frac{e^{y_c^{(t)}}}{\sum_j e^{y_j^{(t)}}} - \delta_{c,\ell_i} \right) \mathbf{a}_{L-1}^{(t)} - \left(\frac{e^{y_c^{(t+1)}}}{\sum_j e^{y_j^{(t+1)}}} - \delta_{c,\ell_i} \right) \mathbf{a}_{L-1}^{(t+1)}.$$

Hence, if $c \neq \ell_i$, then both $g_c^{(t)} < 0$ and $g_c^{(t+1)} < 0$ so that we cannot use the sign information of g_c to recover the correct label. Therefore, the adversary needs to consider all labels to find out the best fit, which inevitably increases the computation overhead and degrades the fidelity of the reconstructed inputs (see Section VIII-A.1 for numerical validations).

Remark 5: Bigger component size $|\mathcal{V}_{h,1}|$ in the gradient sum Eq. (23) will make it more challenging to invert inputs. This is due to the fact that Eq. (23) is related to all data samples of all honest nodes in $\mathcal{V}_{h,1}$, thus inverting input from Eq. (30) is analogous to increase the batchsize or the total number of data samples in Eq. (8) of the centralized case. It has been empirically shown in many works [13], [15], [16] that increasing batchsize will degrade the fidelity of reconstructed inputs severely. We will validate this result in Section VIII-A.2.

C. Optimum Attack Strategy

To test the performance of gradient inversion attack, we consider a random geometric graph with $n = 50$ nodes. Each node randomly selects n_i data samples from the corresponding dataset and uses a two-layer multilayer perceptron (MLP) to train the local model. For simplicity, in the following experiments, we set $n_i = 2$, unless otherwise specified. We use the approach proposed in DLG [13] to conduct the gradient inversion attack (see Appendix F for results of using the cosine similarity-based approach proposed in [15]).

1) Noisy Gradients vs. Gradient Differences vs. The Gradient Sum: As highlighted in Remark 2, the noisy gradient term vanishes in the lower bound Eq. (25) if the variance $\sigma_Z^2 \rightarrow \infty$. This raises an important question: in practice how large should σ_Z^2 be to ensure that the effectiveness of reconstructing inputs from noisy gradients is inferior to that of other variables, namely gradient differences and the gradient sum. To investigate this, Figure 3(a) presents a comparative analysis of input reconstruction performances using noisy gradients, gradient differences, and the gradient sum (considering two honest nodes in the component, i.e., $|\mathcal{V}_{h,1}| = 2$), along with the test accuracies of learn models for different choices using

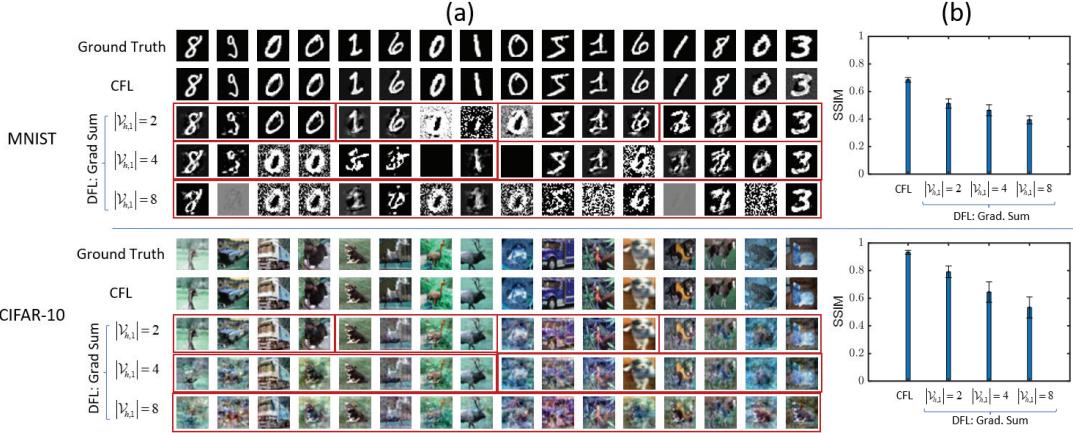


Fig. 6. Performance comparisons of CFL and DFL via inverting inputs from the gradient sum: (a) Sample images of ground truth and reconstructed inputs, (b) SSIM of all reconstructed samples for three different sizes of honest component $|V_{h,1}| = 2, 4, 8$ using two datasets MNIST (top) and CIFAR-10 (bottom), respectively. Wherein the red box indicates that the corresponding samples are from the same component.

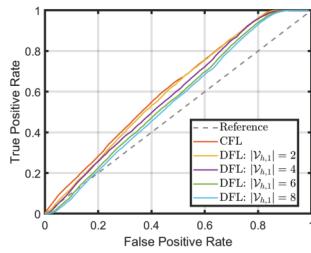


Fig. 7. Membership inference attack results comparisons of CFL and DFL with gradient sum using the gradient-based approach [56].

the MNIST dataset. The results indicate that, without loss of test performance, the effectiveness of inverting inputs from noisy gradient degrades very fast as σ_Z^2 increases, performing worse than inverting inputs using gradient differences and the gradient sum even at a very low variance of $\sigma_Z^2 = 10^{-5}$. This is further illustrated in plot (b) where the ground truth digit 8 can hardly be recognized when inverting noisy gradient for variance $\sigma_Z^2 \geq 10^{-5}$. This suggests that in practical scenarios, a small variance σ_Z^2 is sufficient to ensure the lower bound is attached. As a consequence, in what follows we will evaluate the reconstruction performances via inverting gradient differences and the gradient sum for the case of DFL.

2) *Optimum Attack Iteration*: Given that both CFL and DFL protocols are iterative processes consisting of numerous iterations, in principle the gradient inversion attack can deploy all iterations' information for inverting the input samples. To identify the most effective attack strategy for such an attack, we first explore which iteration yields the most effective results for the gradient inversion attack. In Figure 4(a) we demonstrate the attack performance, quantified by the averaged SSIM of all reconstructed samples (illustrated by solid lines) along with their standard derivation (shown as shadows) of CFL and DFL using gradient differences as a function of iteration number t using the MNIST dataset. We can see that the SSIM of the centralized case consistently surpasses those in the decentralized case throughout all iterations. Unlike the centralized case where the reconstruction performance remains relatively stable across iterations, the reconstruction

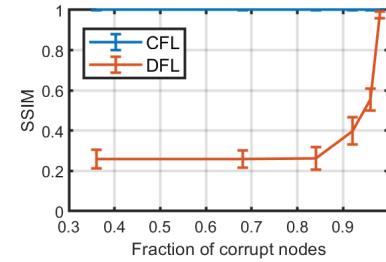


Fig. 8. Performance comparisons of CFL and DFL in terms of the portion of corrupted nodes.

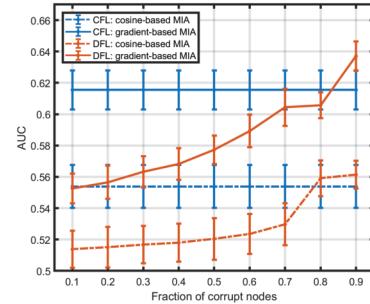


Fig. 9. Performance comparisons of CFL and DFL in terms of the portion of corrupted nodes via membership inference attacks.

performances of the decentralized case degrade significantly in later iterations. This trend aligns with the expectation that gradient differences will approach zero at convergence thereby containing very little information.

To visualize this phenomenon, Figure 4(b) displays illustrative examples of both cases at every 100 iterations. The SSIM results are in agreement with these visual examples; for instance, in the later iterations of the decentralized case, the ground truth digit 9 can hardly be recognized. Hence, the effectiveness of gradient inversion attacks is more pronounced in the initial iterations. Note that for both CFL and DFL we observe a slightly better performance at the early stage. This might be due to the fact that at early iterations the local models contain more information about the local datasets while at later

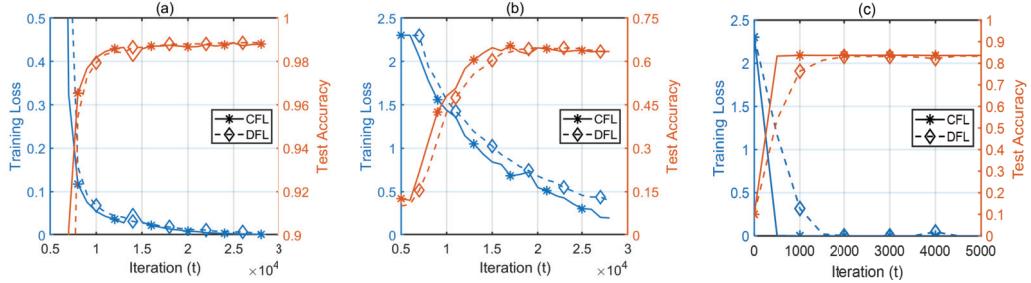


Fig. 10. Training loss and test accuracy in terms of iteration number of both CFL and DFL for two datasets: (a) LeNet (MNIST); (b) LeNet (CIFAR-10); (c) VGG-11 (CIFAR-10).

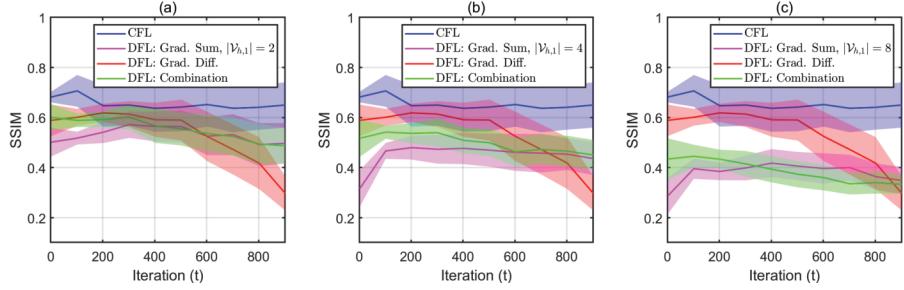


Fig. 11. Performance comparisons of CFL and DFL via a combination strategy using MNIST dataset: SSIM of all reconstructed samples of CFL and DFL by combining gradient differences with the gradient sum for three different sizes of honest component $|\mathcal{V}_{h,1}| = 2, 4, 8$ ((a)-(c), respectively).

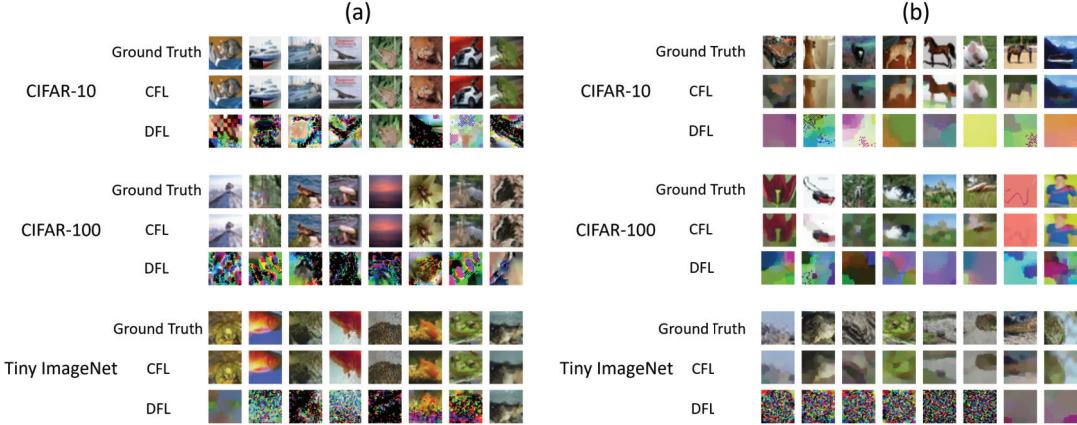


Fig. 12. Reconstructed inputs in CFL and DFL using cosine similarity based gradient inversion attack [15] with (a)VGG-11 and (b)AlexNet architecture and three datasets (CIFAR-10, CIFAR-100 and tiny ImageNet). Label information in DFL is assumed to be known.

iterations the models are more fitted to the global dataset. For this reason, for the forthcoming Figure 5 and 6, we will focus on exploiting gradients from early iterations to execute the gradient inversion attacks.

VIII. DEEP NEURAL NETWORKS II: PRIVACY GAP

We now evaluate the privacy gap between CFL and DFL using different settings and privacy attacks.

A. Evaluating the Privacy Gap Between CFL and DFL Using Gradient Inversion Attack

1) *Inverting Gradient Differences Eq. (22)*: In Remark 4 we showed that using gradient differences labels cannot be analytically computed, unlike in CFL. To assess the impacts of this on the performance of input reconstructions, Figure 5(a)

showcases examples of reconstructed inputs for both the MNIST and CIFAR-10 dataset for four different batch-sizes, i.e., $n_i = 1, 2, 4, 8$. Notable disparities exist in the reconstructed samples of CFL and DFL cases. More specifically, compared to the centralized case, reconstructing inputs via gradient differences is less efficient and yields lower quality. The inefficiency is due to the fact that the adversary needs to iterate through all possible labels to identify the best fit. Furthermore, the quality is compromised due to the inherently reduced information in gradient differences compared to the full gradients. To evaluate the quality of reconstructed inputs, we chose the best-fit samples among iterated inputs for each batch and computed their averaged SSIM. The comparison results are presented in Figure 5(b), illustrating the quality of reconstructed inputs of CFL are consistently better than that in DFL for all batchsizes n_i . This comparison underlines the

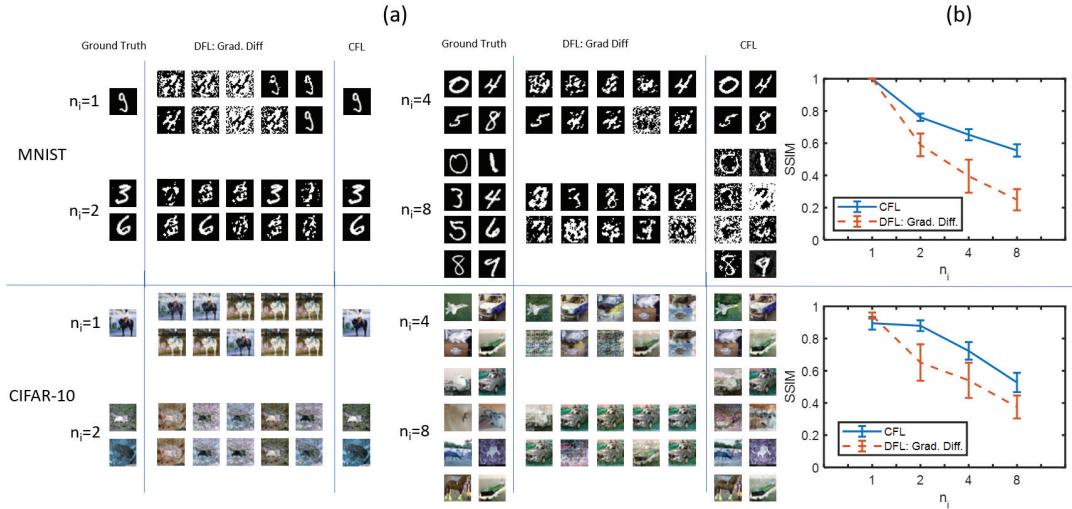


Fig. 13. Performance comparisons of CFL and DFL via inverting inputs from gradient differences using cosine similarity based gradient inversion attack [15]: (a) Samples images of ground truth and reconstructed inputs, (b) SSIM comparisons of all reconstructed inputs for different batch size $n_i = 1, 2, 4, 8$, respectively.

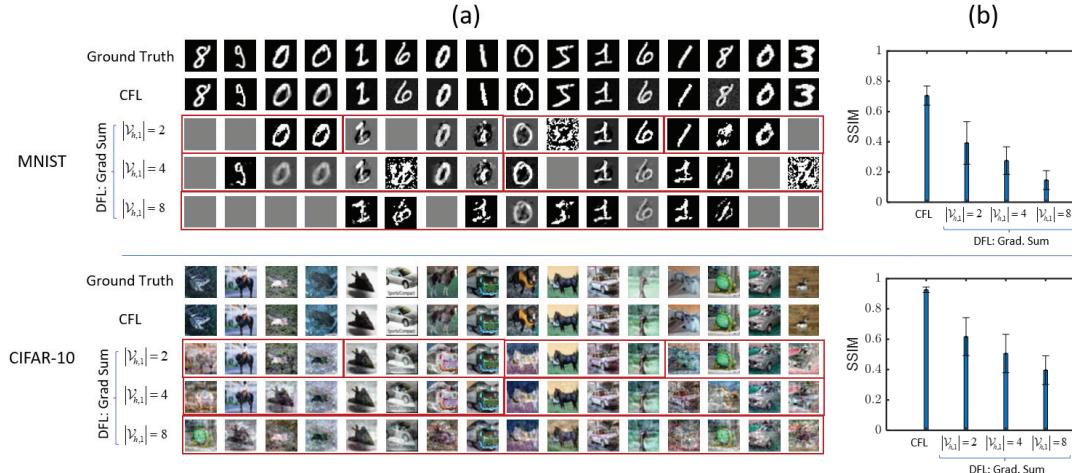


Fig. 14. Performance comparisons of CFL and DFL via inverting inputs from gradient sums using cosine similarity based gradient inversion attack [15]: (a) Samples images of ground truth and reconstructed inputs, (b) SSIM of all reconstructed samples for three different sizes of honest component $|\mathcal{V}_{h,1}| = 2, 4, 8$, respectively.

inherent challenges in reconstructing high-quality inputs from gradient differences in DFL.

2) *Inverting the Gradient Sum Eq. (23):* When inverting inputs from the gradient sum, it is intuitive that the accuracy of reconstructed inputs tends to diminish as the size of the honest component $|\mathcal{V}_{h,1}|$ increases, analogous to the situation of increasing batchsize. This relationship is clearly illustrated in Figure 6 for the MNIST dataset, showing a direct correlation between the size of the honest component and the precision of the reconstructed inputs. It is important to note that while obtaining label information becomes more challenging in DFL (given the gradient sum), in this comparison we assume that the label information is known a prior for both CFL and DFL. However, even with this assumption, the reconstruction performances of DFL are still notably inferior to that of CFL.

3) *Combining Both Gradient Differences and the Gradient Sum:* Since both gradient differences and the gradient

sum are available in DFL, one natural question to ask is if combining them together will improve the attack performances. We demonstrate the comparison results in Figure 11 in Appendix F, showing the SSIM of reconstructed inputs of CFL and DFL across varying iteration numbers for three different sizes. The results suggest that combining gradient differences and gradient sum does not amplify the reconstruction performances. Still, the attack performances of CFL are in general better than DFL, consistently across all iterations.

B. Evaluating Privacy Gap With Additional Gradient Inversion Attacks, Model Architectures and Datasets

In addition to the MNIST and CIFAR-10 dataset and MLP network, we further evaluated the privacy gap using two additional datasets (CIFAR-100 and Tiny ImageNet), two additional model architectures (VGG-11 and AlexNet), and two additional gradient inversion attacks: the cosine



Fig. 15. Reconstructed inputs of CIFAR-10 dataset in CFL and DFL via generative model-based gradient inversion attack [64] with (a)VGG-11 and (b)AlexNet. Label information in DFL is assumed to be known.

similarity-based method proposed in [15] (see Figure 12, Figure 13 and Figure 14) and the generative model-based method proposed in [64] (see Figure 15). The results indicate that the performance of DFL is generally lower than that of CFL, which aligns with the findings using DLG [13], as shown in Figure 5 and Figure 6. These results corroborate our theoretical analysis and suggest that the gap between CFL and DFL is consistent across different gradient inversion attacks, datasets, and model architectures.

C. Evaluating Privacy Gap Using Membership Inference Attacks

Apart from gradient inversion attacks, it is also interesting to see if there is a privacy gap between CFL and DFL when evaluated using membership inference attacks (MIAs). Recall that MIAs aim to test whether specific data points were used in training. The attack results and detailed experimental settings are demonstrated in Figure 7. Similar to the case of using gradient inversion attacks, we can see that CFL leaks more membership information compared to DFL.

D. Privacy Gap Reduces With More Corrupt Nodes

We now validate the result presented in Remark 3. As shown in Figure 8 when using MNIST datasets, the privacy gap between CFL and DFL predictably narrows as the number of corrupt nodes increases when using gradient inversion attacks. Notably, in the case where there is only a single honest node, the attack performance in DFL converges with that of CFL, effectively closing the privacy gap. As for the case of using membership inference attacks, we observe a similar tendency (see Figure 9). Hence, these results are consistent with our theoretical findings in Remark 3, thereby confirming the validity of our analytical approach.

Overall, we conclude that compared to the CFL case, the DFL protocol is less vulnerable to privacy attacks including gradient inversion attacks and membership inference attacks.

E. Related Work

The recent work [26] highlights that prior comparisons between the privacy implications of CFL and DFL either lack

empirical evidence or fail to comprehensively explore privacy arguments. Consequently, [26] stands as the only direct and relevant benchmark for our analysis, which contends that DFL offers no privacy advantages over CFL. However, our study provides a new perspective and challenges this conclusion. We hypothesize that these discrepancies may be attributed to the following factors:

- 1) **Decentralization Techniques:** The decentralization techniques employed are significantly different. While [26] utilizes average consensus-based decentralization methods (refer to Section II-B.1), our study focuses on distributed optimization techniques (see Section II-B.2). The former approach separates local and global model updates similar to CFL, whereas the latter integrates these updates into a joint optimization process.
- 2) **Threat Models:** [26] examines both passive and active adversary models. In contrast, our study considers eavesdropping and passive adversary models, offering a different perspective on potential security risks.
- 3) **Theoretical and Empirical Results:** While [26] confines its findings to empirical data, primarily evaluating privacy risks through membership inference attacks, our research provides a comprehensive analysis includes both information-theoretical analysis and empirical validations. These validations encompass gradient inversion attacks as well as membership inference attacks.

Given these significant methodological divergences, direct comparisons between our findings and those of [26] are inherently challenging. This highlights the necessity for more detailed and extensive investigations. We advocate for continued research in this domain to unravel the complex dynamics that contribute to these divergent outcomes and to deepen our understanding of privacy mechanisms within FL frameworks.

IX. CONCLUSION

In this paper, we showed that DFL through distributed optimization inherently provides privacy advantages compared to CFL, particularly in complex settings like neural networks. We conducted a detailed analysis of the information flow within the network across various iterations, establishing both upper

and lower bounds for the information loss. The bounds indicate that privacy leakage in DFL is consistently less than or equal to that in CFL. We further exemplified our results through two standard applications: logistic regression and training a DNN. In the case of simple logistic regression, we observed that the privacy leakage in both CFL and DFL are identical. However, in more complex scenarios like training DNNs, the privacy loss as measured by the gradient inversion attack is markedly higher in CFL than in DFL. As expected, the privacy gap between CFL and DFL is more pronounced in the presence of numerous honest nodes. Extensive experimental results substantiated our findings.

APPENDIX A PROOF OF PROPOSITION 1

With Eq. (13), the optimality condition Eq. (12) can be expressed as $\forall i \in \mathcal{V}_h$:

$$\begin{aligned} & - \left(\sum_{j \in \mathcal{N}_{i,c}} \mathbf{B}_{ilj} z_{ilj}^{(t)} + \sum_{j \in \mathcal{N}_{i,h}} \mathbf{B}_{ilj} \sum_{\tau=1}^t \Delta z_{ilj}^{(\tau)} + \rho d_i \mathbf{w}_i^{(t)} \right) \\ & = \nabla f_i(\mathbf{w}_i^{(t)}) + \sum_{j \in \mathcal{N}_{i,h}} \mathbf{B}_{ilj} z_{ilj}^{(0)}. \end{aligned} \quad (31)$$

Since all terms in the LHS of Eq. (31) are known by the adversary, the noisy gradient (RHS of Eq. (31)) is known by the adversary, proving claim *i*). As for Eq. (22), since the noise term the RHS of Eq. (31) does not depend on t , the difference

$$\nabla f(\mathbf{w}_i^{(t+\ell)}) - \nabla f(\mathbf{w}_i^{(t)}),$$

is known for any $\ell \geq 1$, hence proving claim *ii*). Moreover, by summing up the RHS of Eq. (31) over all honest nodes in $\mathcal{V}_{h,1}$, we have

$$\begin{aligned} & \sum_{j \in \mathcal{V}_{h,1}} \left(\nabla f_j(\mathbf{w}_j^{(t)}) + \sum_{k \in \mathcal{N}_{j,h}} \mathbf{B}_{jlk} z_{jlk}^{(0)} \right) \\ & = \sum_{j \in \mathcal{V}_{h,1}} \nabla f_j(\mathbf{w}_j^{(t)}) + \sum_{(j,k) \in \mathcal{E}_{h,1}} \mathbf{B}_{jlk} (z_{jlk}^{(0)} - z_{klj}^{(0)}), \end{aligned} \quad (32)$$

since $\mathbf{B}_{klj} = -\mathbf{B}_{jlk}$.

The difference between two successive z updates of Eq. (5) is given by

$$\Delta z_{jli}^{(t+1)} - \Delta z_{jli}^{(t)} = 2\rho \mathbf{B}_{ilj} (\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(t-1)}). \quad (33)$$

Hence, given the fact that at convergence we have $\mathbf{w}_j^{(t_{\max})} = \mathbf{w}_k^{(t_{\max})}$ for all $(j, k) \in \mathcal{E}$, the adversary has knowledge of all $\mathbf{w}_j^{(t)}, j \in \mathcal{V}, t \in \mathcal{T}$. Moreover, again from Eq. (5), we have

$$\begin{aligned} \Delta z_{jli}^{(t+1)} & = z_{jli}^{(t)} - z_{jli}^{(t)} + 2\rho \mathbf{B}_{ilj} \mathbf{w}_i^{(t)} \\ & = \sum_{\tau=1}^t (\Delta z_{jli}^{(\tau)} - \Delta z_{jli}^{(\tau)}) + z_{jli}^{(0)} - z_{jli}^{(0)}, \end{aligned} \quad (34)$$

showing that knowing the $\Delta z_{jli}^{(t+1)}$ s is equivalent to knowing the differences $z_{jli}^{(0)} - z_{jli}^{(0)}$ for all $(i, j) \in \mathcal{E}$.

Hence, the last term in the RHS of Eq. (32) is known to the adversary, the sum of gradients is thus known which completes the proof of claim *iii*).

APPENDIX B PROOF OF THEOREM 1

We have

$$\begin{aligned} & I \left(X_i; \{X_j\}_{j \in \mathcal{V}_c}, \left\{ Z_{jlk}^{(0)} \right\}_{(j,k) \in \mathcal{E}_c}, \left\{ \Delta Z_{jlk}^{(t+1)} \right\}_{(j,k) \in \mathcal{E}, t \in \mathcal{T}} \right) \\ & \stackrel{(a)}{=} I \left(X_i; \{X_j\}_{j \in \mathcal{V}_c}, \left\{ Z_{jlk}^{(0)} \right\}_{(j,k) \in \mathcal{E}_c}, \left\{ W_j^{(t)}, \right. \right. \\ & \quad \left. \left. Z_{jlk}^{(0)} - Z_{klj}^{(0)} \right\}_{(j,k) \in \mathcal{E}, t \in \mathcal{T}} \right) \\ & \stackrel{(b)}{=} I \left(X_i; \{X_j\}_{j \in \mathcal{V}_c}, \left\{ Z_{jlk}^{(0)} \right\}_{(j,k) \in \mathcal{E}_c}, \left\{ W_j^{(t)}, \right. \right. \\ & \quad \left. \left. \nabla f_j(W_j^{(t)}) + \sum_{k \in \mathcal{N}_{j,h}} B_{jlk} Z_{jlk}^{(0)}, Z_{jlk}^{(0)} - Z_{klj}^{(0)} \right\}_{(j,k) \in \mathcal{E}, t \in \mathcal{T}} \right) \\ & \stackrel{(c)}{=} I \left(X_i; \{X_j\}_{j \in \mathcal{V}_c}, \left\{ Z_{jlk}^{(0)} \right\}_{(j,k) \in \mathcal{E}_c}, \left\{ W_j^{(t)} \right\}_{j \in \mathcal{V}_h, t \in \mathcal{T}} \right. \\ & \quad \left\{ \nabla f_j(W_j^{(t)}) \right\}_{j \in \mathcal{V}_c, t \in \mathcal{T}}, \left\{ Z_{jlk}^{(0)} - Z_{klj}^{(0)} \right\}_{(j,k) \in \mathcal{E}_h} \\ & \quad \left\{ \nabla f_j(W_j^{(t)}) + \sum_{k \in \mathcal{N}_{j,h}} B_{jlk} Z_{jlk}^{(0)} \right\}_{j \in \mathcal{V}_h, t \in \mathcal{T}} \end{aligned}$$

where (a) follows from Eq. (33) and Eq. (34), and (b) follows from Eq. (31); (c) follows from Eq. (31) and the fact that all terms in (b) are sufficient to compute all terms in (c) and vice versa. Hence, the proof of Eq. (24) is now complete. Clearly, when $\sigma_Z^2 = 0$, the above (c) reduces to Eq. (18), thereby showing that the privacy loss in DFL is upper bounded by the loss of CFL.

Before proving the lower bound, we first present the following lemma necessary for the coming proof.

Lemma 1: Let X_1, \dots, X_n and R_1, \dots, R_n be independent random variables, and let $g(\cdot)$ be an arbitrary function. If they satisfy $\forall i: I(X_i; g(X_i) + R_i) = 0$. Then

$$\begin{aligned} & \forall i: I \left(X_i; g(X_1) + R_1, \dots, g(X_n) + R_n, \sum_{j=1}^n R_j \right) \\ & = I \left(X_i; \sum_{j=1}^n g(X_j) \right). \end{aligned}$$

Proof: See Appendix C. □

When $\sigma_Z^2 \rightarrow \infty$, Eq. (24) becomes

$$\begin{aligned} & \stackrel{(a)}{=} I \left(X_i; \{X_j\}_{j \in \mathcal{V}_c}, \{W_j^{(t)}\}_{j \in \mathcal{V}, t \in \mathcal{T}}, \right. \\ & \quad \left. \{\nabla f_j(W_j^{(t)})\}_{j \in \mathcal{V}_c, t \in \mathcal{T}}, \{Z_{jlk}^{(0)} - Z_{klj}^{(0)}\}_{(j,k) \in \mathcal{E}_h}, \right. \\ & \quad \left. \left\{ \nabla f_j(W_j^{(t)}) + \sum_{k \in \mathcal{N}_{j,h}} B_{jlk} Z_{jlk}^{(0)} \right\}_{j \in \mathcal{V}_h, t \in \mathcal{T}} \right) \\ & \stackrel{(b)}{=} I \left(X_i; \{X_j\}_{j \in \mathcal{V}_c}, \{W_j^{(t)}\}_{j \in \mathcal{V}, t \in \mathcal{T}}, \{\nabla f_j(W_j^{(t)})\}_{j \in \mathcal{V}_c, t \in \mathcal{T}}, \right. \\ & \quad \left. \{\nabla f_j(W_j^{(t+1)}) - \nabla f_j(W_j^{(t)})\}_{j \in \mathcal{V}_h, t \in \mathcal{T}}, \right. \\ & \quad \left. \left\{ \sum_{j \in \mathcal{V}_{h,l}} \nabla f_j(W_j^{(t)}) \right\}_{1 \leq l \leq k_h, t \in \mathcal{T}} \right) \end{aligned}$$

where (a) uses the fact that $\sigma_Z^2 \rightarrow \infty$, i.e., $\{Z_{jlk}^{(0)}\}_{(j,k) \in \mathcal{E}_c}$ is asymptotically independent of all other terms; (b) uses

Lemma 1 and the fact that for all honest components $\mathcal{G}_{h,1}, \dots, \mathcal{G}_{h,k_h}$ the last term in Eq. (32) is known: the gradient of honest nodes can be seen as the term $g(X_i)$'s and $Z_{j|k}^{(0)} - Z_{k|j}^{(0)}$'s can be seen as noise R_i 's in Lemma 1. Hence, proof of lower bound is complete.

APPENDIX C PROOF OF LEMMA 1

We first present the following equality:

$$\begin{aligned} & \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ 0 & 1 & \cdots & 1 & 1 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & 1 \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} g(X_1) + R_1 \\ g(X_2) + R_2 \\ \vdots \\ g(X_n) + R_n \\ -\sum_{i=1}^n R_i \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=1}^n g(X_i) \\ \sum_{i=2}^n g(X_i) - R_1 \\ \vdots \\ X_n - \sum_{i=1}^{n-1} R_i \\ -\sum_{i=1}^n R_i \end{bmatrix}. \end{aligned} \quad (35)$$

Thus

$$\begin{aligned} & I\left(X_i; g(X_1) + R_1, \dots, g(X_n) + R_n, \sum_{j=1}^n R_j\right) \\ & \stackrel{(a)}{=} I\left(X_i; \sum_{j=1}^n g(X_j), \sum_{j=2}^n g(X_j) - R_1, \dots, g(X_n) \right. \\ & \quad \left. - \sum_{j=1}^{n-1} R_j, \sum_{j=1}^n R_j\right) \\ &= I\left(X_i; \sum_{j=1}^n g(X_j)\right). \end{aligned}$$

where (a) follows from the fact that the linear map in Eq. (35) is bijective; by observing the linear map we note that the difference of the k 'th and $(k+1)$ 'th rows in RHS of Eq. (35) is $g(X_k) + R_k$. This difference is independent of all X_i 's and the k 'th row of Eq. (35), thus $X_i \rightarrow \sum_{j=1}^n g(X_j) \rightarrow \sum_{j=2}^n g(X_j) - R_1 \rightarrow \dots \rightarrow g(X_n) - \sum_{j=1}^{n-1} R_i \rightarrow \sum_{j=1}^n R_j$ forms a Markov chain. Which establishes the second equality, thereby completing the proof.

APPENDIX D PROOF OF COROLLARY 1

$$\begin{aligned} & I(X_i; \mathcal{O}_{\text{CFL}}) - I(X_i; \mathcal{O}_{\text{DFL}}) \\ & \stackrel{(a)}{=} I(X_i; \mathcal{O}_{\text{CFL}}, \mathcal{O}_{\text{DFL}}) - I(X_i; \mathcal{O}_{\text{DFL}}) \\ & \stackrel{(b)}{=} I(X_i; \mathcal{O}_{\text{CFL}}, \mathcal{O}_{\text{DFL}} | \mathcal{O}_{\text{DFL}}) \\ &= I(X_i; \mathcal{O}_{\text{CFL}} | \mathcal{O}_{\text{DFL}}) \\ & \stackrel{(c)}{=} I(X_i; \{\nabla f_j(W_j^{(t)})\}_{j \in \mathcal{V}_h, t \in \mathcal{T}} | \{X_j\}_{j \in \mathcal{V}_c}, \{W_j^{(t)}\}_{j \in \mathcal{V}, t \in \mathcal{T}}, \\ & \quad \{\nabla f_j(W_j^{(t)})\}_{j \in \mathcal{V}_c, t \in \mathcal{T}}, \{\nabla f_j(W_j^{(t+1)}) - \nabla f_j(W_j^{(t)})\}_{j \in \mathcal{V}_h, t \in \mathcal{T}}, \\ & \quad \left\{ \sum_{j \in \mathcal{V}_{h,l}} \nabla f_j(W_j^{(t)}) \right\}_{1 \leq l \leq k_h, t \in \mathcal{T}}) \\ & \stackrel{(d)}{=} I(X_i; \{\nabla f_j(W_j^{(t)})\}_{j \in \mathcal{V}_h, t \in \mathcal{T}} | \{X_j\}_{j \in \mathcal{V}_c}, \{W_j^{(t)}\}_{j \in \mathcal{V}, t \in \mathcal{T}}, \end{aligned}$$

$$\{\nabla f_j(W_j^{(t+1)}) - \nabla f_j(W_j^{(t)})\}_{j \in \mathcal{V}_h, t \in \mathcal{T}},$$

$$\left\{ \sum_{j \in \mathcal{V}_{h,l}} \nabla f_j(W_j^{(t)}) \right\}_{1 \leq l \leq k_h, t \in \mathcal{T}})$$

where (a) holds as $I(X_i; \mathcal{O}_{\text{CFL}}) = I(X_i; \mathcal{O}_{\text{CFL}}, \mathcal{O}_{\text{DFL}})$ since all terms in Eq. (18) of CFL are sufficient to compute all terms in Eq. (25) in DFL; (b) and (c) follow from the definition of conditional mutual information; (d) holds as X_j and $W_j^{(t)}$ can determine $\nabla f_j(W_j^{(t)})$;

APPENDIX E CONVERGENCE BEHAVIOR

A. Setting of Logistic Regression

We assume each node i holds $n_i = 1$ data sample $\mathbf{x}_{ik} \in \mathbb{R}^2$ and binary labels $\ell_{ik} \in \{0, 1\}$. For two labels, the input training samples are randomly drawn from a unit variance Gaussian distribution having mean $\mu_0 = (-1, -1)^\top$ ($\ell_{ik} = 0$) and mean $\mu_1 = (1, 1)^\top$ ($\ell_{ik} = 1$), respectively. We utilize PDMM for the decentralized protocol, i.e., $\theta = 1$ in Algorithm 1, and the convergence parameter ρ is set as 0.4. A single-step gradient descent with learning rate $\mu = 0.1$ is employed for updating Eq. (4). With this, the bound Eq. (28) becomes

$$\begin{aligned} \frac{\partial f_i^{(t+1)}}{\partial \mathbf{w}_i} - \frac{\partial f_i^{(t)}}{\partial \mathbf{w}_i} &= \sum_{k=1}^{n_i} \left(\frac{1}{1 + e^{-\mathbf{y}_{ik}^{(t+1)}}} - \frac{1}{1 + e^{-\mathbf{y}_{ik}^{(t)}}} \right) \mathbf{x}_{ik} \\ &= - \sum_{j \in \mathcal{N}_i} \mathbf{B}_{i,j} \Delta z_{w,i,j}^{(t)} + \frac{1}{\mu} (\mathbf{w}_i^{(t)} - \mathbf{w}_i^{(t+1)}) \\ &+ \left(\rho d_i - \frac{1}{\mu} \right) (\mathbf{w}_i^{(t-1)} - \mathbf{w}_i^{(t)}), \end{aligned} \quad (36)$$

and similar modification follows for Eq. (29). As for the CFL protocol, the same learning rate is applied for a fair comparison, i.e., μ in Eq. (1) is also set to 0.4.

B. Setting and Convergence Behavior of DNNs

To test the performance of the introduced DFL protocol, we generated a random geometric graph with $n = 10$ nodes. LeNet architecture [65] is used for training and two datasets, the MNIST [66] and CIFAR-10 [67] datasets, are used for evaluation. MNIST and CIFAR-10 contain 60,000 images of size 28×28 and 50,000 images of size 32×32 from 10 classes, respectively. We randomly split each dataset into 10 folds, allocating each node one fold. For the decentralized protocol, we also use PDMM with the convergence parameter ρ set to 0.4. We use the quadratic approximation technique [12], [68] to solve the sub-problems approximately with $\mu = \frac{1}{30}$. For the centralized protocol, the constant μ in Eq. (1) is also set to $\mu = \frac{1}{30}$. In addition, we also test VGG-11 [69] architecture of CIFAR-10 dataset with $n = 8$ nodes, where we use inner iteration to solve the sub-problems and set the inner learning rate as 0.05. In Figure 10 we demonstrate the training loss (in blue) and test accuracy (in red) for both protocols, applied on MNIST and CIFAR-10 datasets, respectively.

APPENDIX F SUPPLEMENTARY ATTACK RESULTS

Experimental Setting of Membership Inference Attack: Following similar settings in [56], we randomly select 4000

samples in Purchase dataset [50] as the training sets and distribute them to $n = 10$ nodes. For our experiments, we adopt the fully connected neural network architecture proposed by [56], comprising four perceptron layers: the first layer (FC1) contains 512 neurons, followed by the second layer (FC2) with 256 neurons, the third layer (FC3) with 128 neurons, and the final layer (FC4) with 100 neurons.

REFERENCES

- [1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.
- [2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.
- [3] P. H. Jin, Q. Yuan, F. Iandola, and K. Keutzer, “How to scale distributed deep learning?,” 2016, *arXiv:1611.04581*.
- [4] X. Lian, C. Zhang, H. Zhang, C. Hsieh, W. Zhang, and J. Liu, “Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017, pp. 5330–5340.
- [5] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, “D²: Decentralized training over decentralized data,” in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4848–4856.
- [6] C. Hu, J. Jiang, and Z. Wang, “Decentralized federated learning: A segmented gossip approach,” 2019, *arXiv:1908.07782*.
- [7] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, “D-ADMM: A communication-efficient distributed algorithm for separable optimization,” *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2718–2723, May 2013.
- [8] W. Li, Y. Liu, Z. Tian, and Q. Ling, “Communication-censored linearized ADMM for decentralized consensus optimization,” *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 6, pp. 18–34, 2020.
- [9] H. Chen, Y. Ye, M. Xiao, M. Skoglund, and H. V. Poor, “Coded stochastic ADMM for decentralized consensus optimization with edge computing,” *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5360–5373, Apr. 2021.
- [10] G. Zhang and R. Heusdens, “Distributed optimization using the primal-dual method of multipliers,” *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 4, no. 1, pp. 173–187, Mar. 2018.
- [11] T. W. Sherson, R. Heusdens, and W. B. Kleijn, “Derivation and analysis of the primal-dual method of multipliers based on monotone operator theory,” *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 5, no. 2, pp. 334–347, Jun. 2019.
- [12] K. Niwa, N. Harada, G. Zhang, and W. B. Kleijn, “Edge-consensus learning: Deep learning on P2P networks with nonhomogeneous data,” in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, Aug. 2020, pp. 668–678.
- [13] L. Zhu, Z. Liu, and S. Han, “Deep leakage from gradients,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.
- [14] B. Zhao, K. R. Mopuri, and H. Bilen, “IDLG: Improved deep leakage from gradients,” 2020, *arXiv:2001.02610*.
- [15] J. G. H. Bauermeister, H. Dröge, and M. Moeller, “Inverting gradients—how easy is it to break privacy in federated learning?,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 16937–16947.
- [16] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, “See through gradients: Image batch recovery via gradinversion,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 16337–16346.
- [17] F. Boenisch, A. Dziedzic, R. Schuster, A. S. Shamsabadi, I. Shumailov, and N. Papernot, “When the curious abandon honesty: Federated learning is not private,” 2021, *arXiv:2112.02918*.
- [18] J. Geng et al., “Improved gradient inversion attacks and defenses in federated learning,” *IEEE Trans. Big Data*, vol. 10, no. 6, pp. 839–850, Dec. 2024.
- [19] W. Wei et al., “A framework for evaluating gradient leakage attacks in federated learning,” 2020, *arXiv:2004.10397*.
- [20] H. Yang, M. Ge, K. Xiang, and J. Li, “Using highly compressed gradients in federated learning for data reconstruction attacks,” *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 818–830, 2023.
- [21] Z. Zhao, M. Luo, and W. Ding, “Deep leakage from model in federated learning,” 2022, *arXiv:2206.04887*.
- [22] J. Xu, C. Hong, J. Huang, L. Y. Chen, and J. Decouchant, “AGIC: Approximate gradient inversion attack on federated learning,” in *Proc. 41st Int. Symp. Rel. Distrib. Syst. (SRDS)*, 2022, pp. 12–22.
- [23] K. Yuan, Q. Ling, and W. Yin, “On the convergence of decentralized gradient descent,” *SIAM J. Optim.*, vol. 26, no. 3, pp. 1835–1854, Jan. 2016.
- [24] H. Cheng et al., “Towards decentralized deep learning with differential privacy,” in *Proc. Int. Conf. Cloud Comput.* Springer, 2019, pp. 130–145.
- [25] T. Vogels et al., “RelaySum for decentralized deep learning on heterogeneous data,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, Jan. 2021, pp. 28004–28015.
- [26] D. Pasquini, M. Raynal, and C. Troncoso, “On the (in)security of peer-to-peer decentralized machine learning,” in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2023, pp. 418–436.
- [27] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.
- [28] A. Olshevsky and J. Tsitsiklis, “Convergence speed in distributed consensus and averaging,” *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 33–55, 2009.
- [29] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. U. Stich, “A unified theory of decentralized SGD with changing topology and local updates,” in *Proc. Int. Conf. Mach. Learn.*, vol. 1, Jul. 2020, pp. 5381–5393.
- [30] S. Boyd et al., “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [31] E. K. Ryu and S. Boyd, “Primer on monotone operator methods,” *Appl. Comput. Math.*, vol. 15, no. 1, pp. 3–43, 2016.
- [32] D. Dolev, C. Dwork, O. Waarts, and M. Yang, “Perfectly secure message transmission,” *J. Assoc. Comput. Mach.*, vol. 40, no. 1, pp. 17–47, 1993.
- [33] C. Dwork, “Differential privacy,” in *Proc. Int. Colloq. Autom. Lang. Program.*, 2006, pp. 1–12.
- [34] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Proc. Theory Cryptogr. Conf.*, 2006, pp. 265–284.
- [35] T. M. Cover and J. A. Tomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.
- [36] Q. Li, J. S. Gundersen, R. Heusdens, and M. G. Christensen, “Privacy-preserving distributed processing: Metrics, bounds, and algorithms,” *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2090–2103, 2021.
- [37] M. Lopuhä-Zwakenberg, B. Skorić, and N. Li, “Information-theoretic metrics for local differential privacy protocols,” 2019, *arXiv:1910.07826*.
- [38] Q. Li, M. Coutino, G. Leus, and M. G. Christensen, “Privacy-preserving distributed graph filtering,” in *Proc. 28th Eur. Signal Process. Conf. (EUSIPCO)*, Jan. 2021, pp. 2155–2159.
- [39] S. Yagli, A. Dytso, and H. V. Poor, “Information-theoretic bounds on the generalization error and privacy leakage in federated learning,” in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, May 2020, pp. 1–5.
- [40] Y. Bu, S. Zou, and V. V. Veeravalli, “Tightening mutual information-based bounds on generalization error,” *IEEE J. Sel. Areas Commun.*, vol. 1, no. 1, pp. 121–130, May 2020.
- [41] Q. Li, J. S. Gundersen, M. Lopuhä-Zwakenberg, and R. Heusdens, “Adaptive differentially quantized subspace perturbation (ADQSP): A unified framework for privacy-preserving distributed average consensus,” *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 1780–1793, 2024.
- [42] Y. Liu, X. Zhu, J. Wang, and J. Xiao, “A quantitative metric for privacy leakage in federated learning,” in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, May 2021, pp. 3065–3069.
- [43] F. Mo, A. Borovykh, M. Malekzadeh, H. Haddadi, and S. Demetriadou, “Layer-wise characterization of latent information leakage in federated learning,” 2020, *arXiv:2010.08762*.
- [44] P. Cuff and L. Yu, “Differential privacy as a mutual information constraint,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 43–54.
- [45] M. Gtz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke, “Publishing search logs—A comparative study of privacy guarantees,” *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 3, pp. 520–532, Mar. 2012.
- [46] A. Haeberlen, B. C. Pierce, and A. Narayan, “Differential privacy under fire,” in *Proc. 20th USENIX Conf. Secur.*, Aug. 2011, p. 33.
- [47] M. Xu and X. Li, “Subject property inference attack in collaborative learning,” in *Proc. 12th Int. Conf. Intell. Human-Mach. Syst. Cybern. (IHMSC)*, vol. 1, 2020, pp. 227–231.

- [48] L. Melis, C. Song, E. D. Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 691–706.
- [49] A. Wainakh et al., "User-level label leakage from gradients in federated learning," 2021, *arXiv:2105.09369*.
- [50] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 3–18.
- [51] Z. He, T. Zhang, and R. B. Lee, "Model inversion attacks against collaborative inference," in *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, Dec. 2019, pp. 148–162.
- [52] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 2512–2520.
- [53] Z. Yang, J. Zhang, E.-C. Chang, and Z. Liang, "Neural network inversion in adversarial setting via background knowledge alignment," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 225–240.
- [54] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 253–261.
- [55] W. Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [56] J. Li, N. Li, and B. Ribeiro, "Effective passive membership inference attacks in federated learning against overparameterized models," in *Proc. 11th Int. Conf. Learn. Represent.*, 2022, pp. 1–28.
- [57] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *Proc. IEEE 31st Comput. Secur. Found. Symp. (CSF)*, Jul. 2018, pp. 268–282.
- [58] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," in *Proc. USENIX Secur. Symp.*, 2021, pp. 2615–2632.
- [59] Q. Li, R. Heusdens, and M. G. Christensen, "Communication efficient privacy-preserving distributed optimization using adaptive differential quantization," *Signal Process.*, vol. 194, May 2022, Art. no. 108456.
- [60] Q. Li, R. Heusdens, and M. G. Christensen, "Privacy-preserving distributed optimization via subspace perturbation: A general framework," *IEEE Trans. Signal Process.*, vol. 68, pp. 5983–5996, 2020.
- [61] J. A. G. Jonkman, T. Sherson, and R. Heusdens, "Quantisation effects in distributed optimisation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 3649–3653.
- [62] Q. Li, R. Heusdens, and M. G. Christensen, "Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2020, pp. 5895–5899.
- [63] J. Dall and M. Christensen, "Random geometric graphs," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 66, no. 1, Jul. 2002, Art. no. 016121.
- [64] J. Jeon, J. Kim, K. Lee, S. Oh, and J. Ok, "Gradient inversion with generative image prior," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, Jan. 2021, pp. 29898–29908.
- [65] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [66] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
- [67] K. Alex and H. Geoffrey, "Learning multiple layers of features from tiny images," 2009.
- [68] M. O'Connor, G. Zhang, W. B. Kleijn, and T. D. Abhayapala, "Function splitting and quadratic approximation of the primal-dual method of multipliers for distributed optimization over graphs," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 4, no. 4, pp. 656–666, Dec. 2018.
- [69] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.