Personalized Graph Federated Learning With Differential Privacy

Francois Gauthier, *Member, IEEE*, Vinay Chakravarthi Gogineni, *Senior Member, IEEE*, Stefan Werner, *Fellow, IEEE*, Yih-Fang Huang, *Life Fellow, IEEE*, and Anthony Kuh, *Life Fellow, IEEE*

***bstract*—This paper presents a personalized graph federated learning (PGFL) framework in which distributedly connected servers and their respective edge devices collaboratively learn device or cluster-specific models while maintaining the privacy of every individual device. The proposed approach exploits similarities among different models to provide a more relevant experience for each device, even in situations with diverse data distributions and disproportionate datasets. Furthermore, to ensure a secure and efficient approach to collaborative personalized learning, we study a variant of the PGFL implementation that utilizes differential privacy, specifically zero-concentrated differential privacy, where a noise sequence perturbs model exchanges. Our mathematical analysis shows that the proposed privacy-preserving PGFL algorithm converges to the optimal cluster-specific solution for each cluster in linear time. It also reveals that exploiting similarities among clusters could lead to an alternative output whose distance to the original solution is bounded and that this bound can be adjusted by modifying the algorithm s hyperparameters. Further, our analysis shows that the algorithm ensures local differential privacy for all clients in terms of zero-concentrated differential privacy. Finally, the effectiveness of the proposed PGFL algorithm is showcased through numerical experiments conducted in the context of regression and classification tasks using some of the National Institute of Standards and Technology s (NIST s) datasets, namely, MNIST, and MedMNIST.**

***Index Terms*—Differential privacy, federated learning, graph federated architecture, personalized learning, zero-concentrated differential privacy.**

Manuscript received 5 June 2023; revised 16 September 2023; accepted 15 October 2023. Date of publication 23 October 2023; date of current version 1 November 2023. This work was supported by the Research Council of Norway. An earlier version of this work appears in the Asilomar conference on signals, systems, and computers, Pacific Grove, USA, Nov. 2022 [DOI: 10.1109/IEEECONF56349.2022.10051979]. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Wee Peng Tay. *Corresponding author: Francois Gauthier.)*

Francois Gauthier is with the Department of Electronic Systems, Norwegian University of Science and Technology, 7034 Trondheim, Norway (e-mail: francois.gauthier@ntnu.no).

Vinay Chakravarthi Gogineni was with the Norwegian University of Science and Technology, 7034 Trondheim, Norway. He is now with SDU Applied AI and Data Science, the Maersk Mc-Kinney Moller Institute, University of Southern Denmark, 5230 Odense, Denmark (e-mail: vigo@mmmi.sdu.dk).

Stefan Werner is with the Department of Electronic Systems, Norwegian University of Science and Technology, 7034 Trondheim, Norway, and also with the Department of Information and Communications Engineering, Aalto University, 00076 Aalto, Finland (e-mail: stefan.werner@ntnu.no).

Yih-Fang Huang is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: huang@nd.edu).

Anthony Kuh is with the Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, HI 96822 USA (e-mail: kuh@hawaii.edu).

Digital Object Identifier 10.1109/TSIPN.2023.3325963

I. Introduction

THE rise of internet-of-things (IoT) and cyber-physical systems has led to exponential growth in data collection from distributed devices. However, transferring this massive amount of data to a centralized processing point for inference and decision-making is often impractical due to resource constraints and privacy concerns. To overcome these challenges, distributed learning that features on-device processing is an attractive alternative. Such distributed learning enables efficient data analysis without moving the raw data out of the edge devices. Federated learning (FL) is a distributed learning framework that facilitates collaborative model training across edge devices or clients without exposing the underlying data [2], [3], [4]. In particular, using its own local data, each client refines a global model shared by a server and subsequently transmits the updated model back to the server, which then aggregates all updated client models before sending an update back to clients for further refinements.

To date, research on FL mostly uses a single-server architecture, which is susceptible to communication and computation bottlenecks and vulnerabilities. It also scales poorly with the number and with geographical dispersion of participating clients. To address these concerns, some alternatives to the single-server architecture have been proposed, see, e.g., [5], [6], [7], [8]. Examples of those alternatives include client-edge-server hierarchical learning [6] and the graph federated architecture [5], [8]. In client-edge-server hierarchical learning, edge servers perform partial aggregation with their associated clients and communicate their results to a single cloud server that performs the global aggregation. However, using a single cloud server is susceptible to bottlenecks and can only accommodate a limited number of edge servers. In contrast, the graph federated architecture uses a server network in which each server aggregates the information from its associated clients and shares its model with its neighbors. Therefore, the graph federated architecture is highly scalable with the number of clients and easier to implement, thanks to its distributed nature.

One of the main challenges in FL is data heterogeneity, which means there can be substantial differences in the underlying statistical distributions among clients' data [9], [10], [11]. Consequently, a unique globally shared model can be inadequate for such settings, and personalized models must be learned instead [12], [13], [14]. For example, autonomous vehicles need to maintain vehicle-specific models of their highly dynamic environment while collaborating with nearby vehicles and/or smart city IoT devices [10]. This requirement can be met by personalized FL, where clients, or groups of clients (clusters), learn client- or cluster-specific models [15], [16], [17]. These personalized models typically share some similarities [18]. As

an example, the environment of an autonomous vehicle could be shared with other connected objects. Leveraging the similarities between cluster-specific models can, therefore, improve performance [18], [19], a process known as inter-cluster learning, which is particularly useful when some clients or clusters have insufficient data [20], [21].

Personalized FL has received considerable attention lately due to its ability to improve learning performance in settings where clients are required to observe device-specific behaviors, see, e.g., [18], [20], [21], [22], [23], [24]. It is used in many applications such as healthcare, electrical load forecasting, biometrics, drone swarms, and autonomous vehicles [10], [11], [25], [26], [27]. However, all those works are limited to single-server cases. For example, although [8] extends personalized FL to a multi-server architecture, it assumes that all the clients associated with a given server learn the same model. Under this assumption, each server maintains a single model trained via conventional FL and the model is refined by communicating with other servers about their models. However, the general case where each distributed server needs to enable the learning of personalized models and collaborate with its neighbors to refine those is yet to be studied.

In the context of graph FL, many devices take part in the training process. Thus ensuring the privacy and security of client data is crucial. The risk of eavesdropping attacks on the client-server channels increases with the number of devices in the system, and not all devices can be trusted. Even if data is not explicitly shared among clients, repeated message exchanges could reveal sensitive information to curious devices or external eavesdroppers [28], [29]. In order to reduce this risk, differential privacy (DP) has been introduced to protect client privacy by ensuring that the inclusion or exclusion of an individual data sample does not significantly affect the algorithm output. In other words, DP limits the ability of attackers to infer information from individual data samples by adding controlled noise to the data before sharing it with the server [30], [31], [32], [33], [34]. To improve the privacy-accuracy trade-off, conventional (ϵ, δ)-DP has been relaxed into concentrated DP (CDP) in [35], which has been further relaxed to zero-concentrated DP (zCDP) [36]. The zCDP is easier to analyze and offers a tighter equivalence with (ϵ, δ)-DP. Furthermore, dynamic DP is well-suited for iterative implementations, as it allows the privacy budget to be adjusted dynamically based on the number of iterations [37]. For those reasons, this paper considers dynamic zCDP in the graph FL architecture, where the privacy of client data is of utmost importance. By employing dynamic zCDP, clients perturb their local model estimates with a noise sequence of known variance that decreases progressively during the learning process. This process ensures privacy without compromising model accuracy.

This manuscript tackles the general case of personalized graph federated learning (PGFL) in conventional and privacy-preserving ways. Specifically, we consider a multi-server architecture with distributed clients grouped into clusters (of similar learning tasks), irrespective of their associated servers, for the decentralized training of cluster-specific personalized models. The proposed algorithms, within the considered PGFL architecture, leverage similarities between clusters to mitigate data scarcity and improve learning performance. The local training in the proposed framework uses the alternating direction method of multipliers (ADMM), well-suited for distributed applications [38], [39], [40] and demonstrating fast, often linear [41], [42], convergence. The main contributions of this manuscript are summarized in the following.

- A PGFL framework is proposed to improve learning performance in a distributed learning setting. Our approach employs inter-cluster learning to improve the accuracy of local models by leveraging information from other clusters. The graph FL problem is formulated as a constrained optimization problem and solved in a distributed manner using ADMM.
- We design a privacy-preserving variant of the PGFL algorithm, where clients perturb their local models to achieve local differential privacy using the zCDP framework. The privacy loss is quantified per iteration as well as throughout the computation.
- Mathematical analysis is given to show that the privacy-preserving implementation of the PGFL algorithm converges to the optimal solution for each cluster in linear time. Additionally, our analysis shows that utilizing inter-cluster learning leads to an alternative solution whose distance to the original solution is bounded and that the bound depends on cluster similarity and can be adjusted with hyperparameter selection.

The paper is organized as follows: Section II presents the problem formulation and the PGFL algorithm along with its zCDP variant. Sections III and IV are dedicated to the convergence and privacy analyses of the proposed algorithm. In Section V, we demonstrate the effectiveness of the algorithm through a series of experiments involving regression and classification tasks. Section VI concludes the paper. The following table contains the mathematical symbols used throughout the paper.

$\mathbf{I}$	Identity matrix
$\mathbf{0}$	Null vector
$\langle \boldsymbol{a}, \boldsymbol{b} \rangle$	Inner product between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$
$(\cdot)^\mathsf{T}$	Transpose operator
$\mathbb{E}[\cdot]$	Statistical expectation operator
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathcal{U}(a, b)$	Uniform distribution on an interval (a, b)
$\nabla f(\cdot)$	Gradient of a function $f(\cdot)$
$\|\| \cdot \|\|_2^2$	Euclidean norm
$\sum \prod$	Sum and product operators
$\bigcap \bigcup$	Intersection and union operators
$\mathcal{A} \backslash a$	Exclusion of the element a from the set $\mathcal{A}$

II. Problem Formulation and Proposed Method

The proposed PGFL framework solves a personalized optimization problem in a graph federated architecture and utilizes the similarities among clusters to enhance learning performance. For this purpose, we consider a distributed network that consists of S servers with a total of K clients. The server network is modeled as an undirected graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$, where $\mathcal{S}$ is the set of servers and $\mathcal{E}$ is the set of edges so that two servers s and p can communicate if and only if $(s, p) \in \mathcal{E}$. The set of neighbors to a server s is denoted by $\mathcal{N}_s$, and it contains s. We denote $\mathcal{N}_s^- = \mathcal{N}_s \backslash s$. Each server s is associated with a set of clients, denoted $\mathcal{C}_s$, with $\bigcup_{s \in \mathcal{S}} \mathcal{C}_s = \mathcal{C}$ and $\mathcal{C}_s \bigcap \mathcal{C}_p = \emptyset, \forall s \neq p$. Every client $k \in \mathcal{C}$ has access to a local dataset $\mathcal{D}_k$ of cardinality $|\mathcal{D}_k| = D_k$, which is composed of a data matrix $\mathbf{X}_k = [\mathbf{x}_{k,1} \dots \mathbf{x}_{k,D_k}]$, where $\mathbf{x}_{k,i}, i \in \{1, \dots, D_k\}$ is a vector

of size L, and a response vector $\mathbf{y}_k = [y_{k,1}, \ldots, y_{k,D_k}]$ that is subject to white observation noise. Each client $k \in \mathcal{C}$ aims to learn a personalized, client-specific model $\mathbf{w}_k$.

The learning task for each client is defined by the set $\{\mathcal{D}_k, \ell_k\}$, which represents its local data and loss function. All clients connected to distributed servers, regardless of their associated servers, are grouped into Q clusters. These clusters are formed by clients with similar learning tasks, such as $\mathcal{F}$-similar tasks [43], with the aim of collectively learning a shared model. It is assumed that there is a degree of relationship among the learning tasks across clusters, which can manifest in various ways. For example, clusters may share the same loss and regularizer functions while having different data distributions, or they may have the same data distribution but distinct objective functions. For instance, in healthcare, clusters can represent various patient diagnostics, independent of their respective associated hospitals, with a hospital functioning akin to a server. We denote the set of clusters as $\mathcal{Q} = \{1, \ldots, Q\}$. The clients belonging to a specific cluster $q \in \mathcal{Q}$ form the set $\mathcal{C}_{(q)}$ aiming to learn the model $\mathbf{w}^*_{(q)}$. Additionally, the set of clients associated with server s within cluster q is denoted as $\mathcal{C}_{s,(q)}$, with $\mathcal{C}_{s,(q)} = \mathcal{C}_s \bigcap \mathcal{C}_{(q)}$.

A. Personalized Graph Federated Learning

To address task variations, personalized (cluster) models are preferable. However, despite their differences, the underlying relationship among tasks, or equivalently, clusters, can still be exploited in decentralized learning. Here, we consider a modified regularized empirical risk minimization problem to leverage cluster similarities. For this purpose, we introduce an additional regularizer function that enforces similarity among the cluster-specific personalized models. This additional regularizer function corresponds to inter-cluster learning and is controlled by the inter-cluster learning parameter $\tau \in (0, 1)$. The resulting optimization problem for a cluster q is formulated as:

$$\min_{\mathbf{w}_{q)}} \sum_{k \in \mathcal{C}_{q)}} \frac{1}{D_k} \sum_{i=1}^{D_k} \ell_k(\mathbf{x}_{k,i}, y_{k,i}; \mathbf{w}_{(q)}) + R(\mathbf{w}_{(q)}) + \tau \sum_{r \in \mathcal{Q} \setminus q} ||\mathbf{w}_{(r)} - \mathbf{w}_{(q)}||_2^2, \tag{1}$$

where $\ell_k(\,)$, $R(\,)$, and denote the client loss function, the global regularizer function, and the regularization parameter, respectively. The larger the τ value is, the more the similarities among cluster-specific personalized models are exploited.

The centralized optimization problem above relies on the global variable $\mathbf{w}_{(q)}$. In a multi-server architecture, the servers maintain local cluster-specific models and communicate with their neighbors to reach a consensus for each cluster. The equivalent distributed optimization problem for cluster q, is given by

$$\min_{\{\mathbf{w}_{s\ q)}\}} \quad \sum_{q \in \mathcal{Q}} \left(\sum_{k \in \mathcal{C}_{s\ q)}} \frac{1}{D_k} \sum_{i=1}^{D_k} \ell_k(\mathbf{x}_{k,i}, y_{k,i}; \mathbf{w}_{s,(q)}) + R(\mathbf{w}_{s,(q)}) + \tau \sum_{r \in \mathcal{Q} \setminus q} \sum_{p \in \mathcal{N}_s} ||\mathbf{w}_{p,(r)} - \mathbf{w}_{s,(q)}||_2^2 \right),$$

$$\text{s.t.} \quad \mathbf{w}_{s,(q)} = \mathbf{z}_{s,p,(q)}, \mathbf{w}_{p,(q)} = \mathbf{z}_{s,p,(q)}; \quad \forall (s,p) \in \mathcal{E}, \forall q \in \mathcal{Q}, \tag{2}$$

where $\mathbf{w}_{s,(q)}$ denotes the model for the cluster q connected to server s and consensus is enforced by the cluster-specific auxiliary variables $\{\mathbf{z}_{s,p,(q)}; \forall (s,p) \in \mathcal{E}, \forall q \in \mathcal{Q}\}$. From (2), the augmented Lagrangian with penalty parameter ρ can be derived as

$$\begin{aligned} &\mathcal{L}_{\rho,q}(\mathcal{V}_q, \mathcal{M}, \mathcal{Z}) \\ &= \sum_{s \in \mathcal{S}} \Bigg[\sum_{k \in \mathcal{C}_{s\ q)}} \frac{\ell_k(\mathbf{X}_k, \mathbf{y}_k; \mathbf{w}_{s,(q)})}{D_k} + R(\mathbf{w}_{s,(q)}) \\ &+ \tau \sum_{r \in \mathcal{Q} \setminus q} \sum_{p \in \mathcal{N}_s} ||\mathbf{w}_{p,(r)} - \mathbf{w}_{s,(q)}||_2^2 \\ &+ \sum_{p \in \mathcal{N}_s^-} \left(\boldsymbol{\mu}_{s,p} (\mathbf{w}_{s,(q)} - \mathbf{z}_{s,p,(q)}) + \boldsymbol{\psi}_{s,p} (\mathbf{w}_{p,(q)} - \mathbf{z}_{s,p,(q)}) \right. \\ &\left. + \frac{\rho}{2} \sum_{p \in \mathcal{N}_s^-} \left(||\mathbf{w}_{s,(q)} - \mathbf{z}_{s,p,(q)}||_2^2 + ||\mathbf{w}_{p,(q)} - \mathbf{z}_{s,p,(q)}||_2^2 \right) \right], \end{aligned} \tag{3}$$

with the set of primal variables $\mathcal{V}_q = \{\mathbf{w}_{s,(q)}; s \in \mathcal{S}\}$, Lagrange multipliers $\mathcal{M} = (\{\boldsymbol{\mu}_{s,p}\}, \{\boldsymbol{\psi}_{s,p}\})$, and auxiliary variables $\mathcal{Z} = \{\mathbf{z}_{s,p,(q)}\}$. Given that the Lagrange multipliers are initialized to zero, using the Karush-Kuhn-Tucker conditions of optimality and setting $\boldsymbol{\psi}_s = 2 \sum_{p \in \mathcal{N}_s^-} \boldsymbol{\psi}_{s,p}$, it can be shown that the Lagrange multipliers $\boldsymbol{\mu}_{s,p}$ and the auxiliary variables $\mathcal{Z}$ are eliminated [44]. From (3), it is possible to derive the local update steps of the ADMM for clients and servers. For client $k \in \mathcal{C}_{s,(q)}$, the primal and dual updates are given by

Client primal update:

$$\mathbf{w}_k^{(n)} = \arg\min_{\mathbf{w}} \frac{1}{D_k} \ell_k(\mathbf{X}_k, \mathbf{y}_k; \mathbf{w}) + \frac{1}{|\mathcal{C}_s|} R(\mathbf{w}) - \left\langle \boldsymbol{\varphi}_k^{(n-1)}, \mathbf{w} - \mathbf{w}_{s,(q)}^{(n-1)} \right\rangle + \frac{\rho}{2} \left|\left| \mathbf{w} - \mathbf{w}_{s,(q)}^{(n-1)} \right|\right|_2^2, \tag{4}$$

Client dual update:

$$\boldsymbol{\varphi}_k^{(n)} = \boldsymbol{\varphi}_k^{(n-1)} + \rho \left(\mathbf{w}_{s,(q)}^{(n)} - \mathbf{w}_k^{(n)} \right), \tag{5}$$

where the superscript n denotes the iteration number. Further, the primal and dual updates for a server $s \in \mathcal{S}$ are given by:

Server primal update:

$$\begin{aligned} \mathbf{w}_{s,(q)}^{(n)} = \frac{1}{1 + \tau^{(n)} + \rho |\mathcal{N}_s^-|} \Bigg[&\frac{1}{|\mathcal{C}_{s,(q)}|} \sum_{k \in \mathcal{C}_{s\ q)}} \mathbf{w}_k^{(n)} \\ &- \frac{1}{\rho |\mathcal{C}_{s,(q)}|} \sum_{k \in \mathcal{C}_{s\ q)}} \boldsymbol{\varphi}_k^{(n-1)} \\ &- \frac{1}{2} \boldsymbol{\psi}_{q,s}^{(n-1)} + \frac{\rho}{2} \sum_{p \in \mathcal{N}_s^-} \left(\mathbf{w}_{s,(q)}^{(n-1)} - \mathbf{w}_{p,(q)}^{(n-1)} \right) \\ &+ \tau^{(n)} \frac{1}{Q-1} \frac{1}{|\mathcal{N}_s|} \sum_{r \in \mathcal{Q} \setminus q} \sum_{p \in \mathcal{N}_s} \mathbf{w}_{p,(r)}^{(n-1)} \Bigg], \end{aligned} \tag{6}$$

Server dual update:

$$\boldsymbol{\psi}_{q,s}^{(n)} = \boldsymbol{\psi}_{q,s}^{(n-1)} + \rho \sum_{p \in \mathcal{N}_s^-} \left(\mathbf{w}_{p,(q)}^{(n)} - \mathbf{w}_{s,(q)}^{(n)} \right), \quad (7)$$

where $\tau^{(n)}$, the inter-cluster learning parameter, is iteration-dependent. Since inter-cluster learning may degrade performance toward the end of the computation, it may be necessary for $\tau^{(n)}$ to follow a decreasing sequence.

The computation in (6) performs local aggregation (first two lines), inter-server aggregation (third line), and inter-cluster learning (fourth line) in a single step. This presents the major drawback of using the models of the previous iteration for inter-server aggregation, i.e., $\mathbf{w}_{p,(q)}^{(n-1)}$, and inter-cluster learning, i.e., $\mathbf{w}_{p,(r)}^{(n-1)}$ [18], [45]. A multi-step mechanism addresses this issue by replacing the primal and dual updates of the server as follows:

Server aggregation:

$$\widetilde{\mathbf{w}}_{s,(q)}^{(n)} = \frac{1}{|\mathcal{C}_{s,(q)}|} \sum_{k \in \mathcal{C}_s \quad q)} \mathbf{w}_k^{(n)} - \frac{1}{\rho |\mathcal{C}_{s,(q)}|} \sum_{k \in \mathcal{C}_s \quad q)} \boldsymbol{\varphi}_k^{(n-1)}. \quad (8)$$

Inter-server aggregation:

$$\widehat{\mathbf{w}}_{s,(q)}^{(n)} = \frac{1}{|\mathcal{N}_s|} \sum_{p \in \mathcal{N}_s} \widetilde{\mathbf{w}}_{p,(q)}^{(n)}. \quad (9)$$

Inter-cluster learning:

$$\mathbf{w}_{s,(q)}^{(n)} = \left(1 - \tau^{(n)} \right. \widehat{\mathbf{w}}_{s,(q)}^{(n)} + \frac{\tau^{(n)}}{Q-1} \sum_{r \in \mathcal{Q} \setminus q} \widehat{\mathbf{w}}_{s,(r)}^{(n)}. \quad (10)$$

The above multi-step mechanism has two main advantages. First, performing server aggregation prior to inter-server aggregation enables the servers to maintain models composed of the last available client estimates. Second, the fact that inter-cluster learning is performed at the end of the multi-step mechanism ensures that model similarities are leveraged evenly; that is, the same weight is given to any two clients' estimates within the server neighborhood. The resulting PGFL algorithm is summarized in Algorithm 1.

Algorithm 1: PGFL.

Initialization: $\mathbf{w}_k^{(0)} =$ and $\mathbf{w}_{s,(q)}^{(0)} = 0, \forall k, q, s$
– *Procedure at client* $k \in \mathcal{C}_s$ –
For iteration $n = 1, 2, \ldots$
  Update $\mathbf{w}_k^{(n)}$ as in (4)
  Share $\mathbf{w}_k^{(n)}$ and $\boldsymbol{\varphi}_k^{(n-1)}$ with server s
  Receive $\mathbf{w}_{s,(q)}^{(n)}$ from server s
  Update $\boldsymbol{\varphi}_k^{(n)}$ as in (5)
EndFor
– *Procedure at server* s –
For iteration $n = 1, 2, \ldots$
  Receive $\{\widetilde{\mathbf{w}}_k^{(n)}, \boldsymbol{\varphi}_k^{(n-1)}; \forall k \in \mathcal{C}_s\}$
  Update $\widetilde{\mathbf{w}}_{s,(q)}^{(n)}$ as in (8)
  Share $\widetilde{\mathbf{w}}_{s,(q)}^{(n)}, \forall q$ with each server p in $\mathcal{N}_s^-$
  Receive $\widetilde{\mathbf{w}}_{p,(q)}^{(n)}, \forall q$ from each server p in $\mathcal{N}_s^-$
  Aggregate $\widehat{\mathbf{w}}_{s,(q)}^{(n)}$ as in (9)
  Compute $\mathbf{w}_{s,(q)}^{(n)}$ as in (10)
  Share $\mathbf{w}_{s,(q)}^{(n)}$ with clients in $\mathcal{C}_s$
EndFor

B. Privacy Preservation in PGFL

This section presents a privacy-preserving variant of the PGFL algorithm that uses dynamic zero-concentrated differential privacy to protect the participants' data.

Zero-concentrated differential privacy is defined as follows.

Definition: A randomized mechanism M satisfies ϕ-zCDP if, for any two neighboring datasets $\mathcal{D}$, $\mathcal{D}'$ differing in only one data sample, we have

$$D \ (M(\mathcal{D}) || M(\mathcal{D}')) \leqslant \phi \ , \forall \ \in (1, +\infty), \quad (11)$$

where D ( ) denotes the -Rényi divergence between the distributions $M(\mathcal{D})$ and $M(\mathcal{D}')$.

The motivation for choosing zCDP over conventional (ϵ, δ)-DP is that, like CDP, it offers improved accuracy for identical privacy loss in the worst-case scenario, where an eavesdropper aggregates all the exchanged messages [35], [36]. We have the option to use either CDP or zCDP to preserve privacy in the proposed algorithm, but for simplicity, we choose to use zCDP.

Since the proposed PGFL algorithm is iterative in nature, it is crucial to control privacy protection at every iteration and consider the privacy leakage for the entire learning process. For this purpose, we adjust the privacy protection dynamically per iteration, as developed in [37], to control the total privacy leakage of the algorithm throughout the computation. In practice, instead of sharing the exact local estimate $\mathbf{w}_k^{(n)}$, a client k shares with its server at iteration n the perturbed estimate $\widetilde{\mathbf{w}}_k^{(n)}$, given by

$$\widetilde{\mathbf{w}}_k^{(n)} = \mathbf{w}_k^{(n)} + \boldsymbol{\xi}_k^{(n)}, \quad (12)$$

where the perturbation noise follows a Gaussian mechanism, $\boldsymbol{\xi}_k^{(n)} \sim \mathcal{N}(\ , \delta_k^{2(n)} \mathbf{I})$, with $\delta_k^{2(n)}$ being the variance of the perturbation noise at iteration n.

In the context of dynamic zCDP, privacy protection is governed by $\phi_k^{(0)}$ and ζ. The parameter $\phi_k^{(0)}$ represents the initial privacy leakage, indicating the desired level of privacy at the start of the algorithm. On the other hand, $\zeta \in (0, 1)$ denotes the exponential decay factor of the noise variance, dynamically adjusting the iteration-specific privacy budget as the computation takes place. As shown later in Section IV, the privacy parameter at iteration n, $\phi_k^{(n)}$, is inversely proportional to the variance of the perturbation noise, $\delta_k^{2(n)}$. Here, for each client, $k \in \mathcal{C}$, the initial variance $\delta_k^{2(0)}$ is fixed, and subsequently, the variance at iteration n is updated according to the relationship $\delta_k^{2(n)} = \zeta \delta_k^{2(n-1)}$. This recursive update ensures a decreasing privacy budget as the algorithm progresses.

The server aggregation (8) and client dual update (5) are affected by the noise perturbation (12). The server aggregation becomes

$$\widetilde{\mathbf{w}}_{s,(q)}^{(n)} = \frac{1}{|\mathcal{C}_{s,(q)}|} \sum_{k \in \mathcal{C}_s \quad q)} \widetilde{\mathbf{w}}_k^{(n)} - \frac{1}{\rho |\mathcal{C}_{s,(q)}|} \sum_{k \in \mathcal{C}_s \quad q)} \boldsymbol{\varphi}_k^{(n-1)}, \quad (13)$$

Algorithm 2: Privacy-Preserving PGFL.

Initialization: $\mathbf{w}_k^{(0)} =$ and $\mathbf{w}_{s,(q)}^{(0)} =$, $\forall k, q, s$
– Procedure at client $k \in \mathcal{C}_s$ –
For iteration $n = 1, 2, \ldots$
  Update $\mathbf{w}_k^{(n)}$ as in (4)
  Perturb $\mathbf{w}_k^{(n)}$ into $\widetilde{\mathbf{w}}_k^{(n)}$ as in (12)
  Share $\widetilde{\mathbf{w}}_k^{(n)}$ and $\boldsymbol{\varphi}_k^{(n-1)}$ with server s
  Receive $\mathbf{w}_{s,(q)}^{(n)}$ from server s
  Update $\boldsymbol{\varphi}_k^{(n)}$ as in (5) using $\widetilde{\mathbf{w}}_{s,(q)}^{(n)}$ and $\widetilde{\mathbf{w}}_k^{(n)}$.
EndFor
– Procedure at server s –
For iteration $n = 1, 2, \ldots$
  Receive $\{\widetilde{\mathbf{w}}_k^{(n)}, \boldsymbol{\varphi}_k^{(n-1)}; \forall k \in \mathcal{C}_s\}$
  Update $\widetilde{\mathbf{w}}_{s,(q)}^{(n)}$ as in (13)
  Share $\widetilde{\mathbf{w}}_{s,(q)}^{(n)}, \forall q$ with each server p in $\mathcal{N}_s^-$
  Receive $\widetilde{\mathbf{w}}_{p,(q)}^{(n)}, \forall q$ from each server p in $\mathcal{N}_s^-$
  Aggregate $\widehat{\mathbf{w}}_{s,(q)}^{(n)}$ as in (9)
  Compute $\mathbf{w}_{s,(q)}^{(n)}$ as in (10)
  Share $\mathbf{w}_{s,(q)}^{(n)}$ with clients in $\mathcal{C}_s$
EndFor

and in the client dual update, we substitute $\mathbf{w}_{s,(q)}^{(n)}$ with $\widetilde{\mathbf{w}}_{s,(q)}^{(n)}$ and $\mathbf{w}_k^{(n)}$ with $\widetilde{\mathbf{w}}_k^{(n)}$.

The resulting privacy-preserving algorithm is summarized in Algorithm 2. In the following sections, we provide a detailed study of the privacy protection and convergence properties of the proposed privacy-preserving PGFL algorithm.

III. Convergence Analysis

This section studies the convergence behavior of the proposed privacy-preserving PGFL algorithm. Sections III-A and III-B study the algorithm without inter-cluster learning and show that it converges to the optimal solution of (2) with $\tau = 0$ in linear time. Section III-C then shows the impact of inter-cluster learning. In particular, we show that although inter-cluster learning leads to a different convergence point than intra-cluster learning, the distance between these two points is bounded by a function of the task dissimilarity and the inter-cluster learning parameter sequence. Moreover, we show that this bound can be used to design the inter-cluster learning parameter sequence to achieve a desired convergence point under mild assumptions on cluster similarity, allowing for greater accuracy control in personalized learning while leveraging the task similarity for faster convergence and improved performance.

A. Problem Reformulation

We consider the server update steps with $\tau^{(n)} = 0$. Then, the minimization problem for the client $k \in \mathcal{C}_{s,(q)}$ becomes

$$\min_{\mathbf{w}_k} \frac{1}{D_k}\ell_k(\mathbf{X}_k, \mathbf{y}_k; \mathbf{w}_k) + \frac{1}{|\mathcal{C}_s|}R(\mathbf{w}_k)$$
$$\text{s.t. } \mathbf{w}_k = \widehat{\mathbf{w}}_{s,(q)}, \tag{14}$$

where $\widehat{\mathbf{w}}_{s,(q)}$ is the result of inter-server aggregation (9), defined as the average model for cluster q in $\mathcal{N}_s$. To simplify the analysis, we reformulate (14) as

$$\min_{\mathbf{w}_k} f_k(\mathbf{w}_k)$$
$$\text{s.t. } \mathbf{w}_k = \mathbf{e}_{k,l}, \mathbf{w}_l = \mathbf{e}_{k,l}, \forall l \in \sum_{p \in \mathcal{N}_s} \mathcal{C}_{p,(q)}, \tag{15}$$

where $f_k(\mathbf{w}_k)$ is given by

$$f_k(\mathbf{w}_k) = \frac{1}{D_k}\ell_k(\mathbf{X}_k, \mathbf{y}_k; \mathbf{w}_k) + \frac{1}{|\mathcal{C}_s|}R(\mathbf{w}_k), \tag{16}$$

and the auxiliary variables $\{\mathbf{e}_{k,l}\}, \forall k, l \in \sum_{p \in \mathcal{N}_s} \mathcal{C}_{p,(q)}$ enforce consensus. To reformulate (15) further, we introduce the following:

$$\begin{aligned}
\mathbf{w} &= [\mathbf{w}_1, \ldots, \mathbf{w}_k, \ldots \mathbf{w}_{|\mathcal{C}|}] \ , \\
\widetilde{\mathbf{w}} &= [\widetilde{\mathbf{w}}_1, \ldots, \widetilde{\mathbf{w}}_k, \ldots \widetilde{\mathbf{w}}_{|\mathcal{C}|}] \ = \mathbf{w} + \boldsymbol{\xi} \\
\boldsymbol{\varphi} &= [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_k, \ldots, \boldsymbol{\varphi}_{|\mathcal{C}|}] \ , \\
F(\mathbf{w}) &= \sum_{k \in \mathcal{C}} f_k(\mathbf{w}_k),
\end{aligned} \tag{17}$$

where $\boldsymbol{\xi}$ is the concatenation of the noise added to the local models to ensure privacy. In addition, we introduce the vector $\mathbf{e} \in \mathbb{R}^{2Md}$ concatenating the vectors $\mathbf{e}_{k,l}, \mathbf{e}_{l,k}, \forall (k,l) \in \{1, \ldots, K\} : k \neq l$, where d is the dimension of the models and M is the number of constraints in (15). We can then reformulate (15) as

$$\min_{\mathbf{w}} F(\mathbf{w})$$
$$\text{s.t. } \mathbf{A}\mathbf{w} + \mathbf{B}\mathbf{e} = \ . \tag{18}$$

where $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2]$ and $\mathbf{B} = [-\mathbf{I}_{2Md}, -\mathbf{I}_{2Md}]$. The matrices $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{2Md \times |\mathcal{C}|d}$ are composed of $d \times d$-sized blocks. Given a couple of connected clients (k, l), their associated auxiliary variable $\mathbf{e}_{k,l}$, and its corresponding index in $\mathbf{e}$, q; the blocks $\big(\mathbf{A}_1 \ _{q,k}$ and $\big(\mathbf{A}_2 \ _{q,l}$ are equal to the identity matrix $\mathbf{I}_d$, all other blocks are null.

From the above definitions, one can express $\sum_{k\ l \in} \|\mathbf{w}_k - \mathbf{e}_{k,l}\|^2 + \|\mathbf{w}_l - \mathbf{e}_{k,l}\|^2 = \|\mathbf{A}\mathbf{w} + \mathbf{B}\mathbf{e}\|^2$ and, for $\in \mathbb{R}^{4Md}$, $\sum_{k \in \mathcal{C}} \sum_{l \in \mathcal{N}_k} (\langle \mathbf{w}_k - \mathbf{e}_{k,l}, \ _q \rangle + \langle \mathbf{w}_l - \mathbf{e}_{k,l}, \ _{2e+q} \rangle) = \langle \mathbf{A}\mathbf{w} + \mathbf{B}\mathbf{e}, \ \rangle$.

Therefore, the Lagrangian can be rewritten as

$$\mathcal{L}_\rho(\mathcal{V}_q, \mathcal{M}) = F(\mathbf{w}) + \langle \mathbf{A}\mathbf{w} + \mathbf{B}\mathbf{e}, \ \rangle + \frac{\rho}{2}\|\mathbf{A}\mathbf{w} + \mathbf{B}\mathbf{e}\|^2 . \tag{19}$$

B. Convergence Proof

We make the following assumptions to continue the analysis.

Assumption 1: The functions $f_k(\ ), k \in \{1, \ldots, K\}$, are convex and smooth.

Using (19), and under Assumption 1, the steps of the PGFL algorithm without inter-cluster learning can be expressed as follows:

$$\begin{aligned}
\nabla F(\mathbf{w}^{(n+1)}) + \mathbf{A} \ ^{(n)} + \rho \mathbf{A} \ \big(\mathbf{A}\mathbf{w}^{(n+1)} + \mathbf{B}\mathbf{e}^{(n)} \ &= 0, \\
\mathbf{B} \ ^{(n)} + \rho \mathbf{B} \ \big(\mathbf{A}\widetilde{\mathbf{w}}^{(n+1)} + \mathbf{B}\mathbf{e}^{(n+1)} \ &= 0, \\
^{(n+1)} - \ ^{(n)} + \rho\big(\mathbf{A}\widetilde{\mathbf{w}}^{(n+1)} + \mathbf{B}\mathbf{e}^{(n+1)} \ &= 0.
\end{aligned} \tag{20}$$

Similarly to [41], we introduce the following to simplify (20):

$$\begin{aligned} \mathbf{H}_+ &= \mathbf{A}_1 + \mathbf{A}_2, & \mathbf{H}_- &= \mathbf{A}_1 - \mathbf{A}_2, \\ \mathbf{L}_+ &= \frac{1}{2}\mathbf{H}_+\mathbf{H}_+, & \mathbf{L}_- &= \frac{1}{2}\mathbf{H}_-\mathbf{H}_-, \\ &= \mathbf{H}_-\mathbf{w}, & \mathbf{M} &= \frac{1}{2}(\mathbf{L}_+ + \mathbf{L}_-). \end{aligned}$$

Then, as derived in [41, Section II.B], (20) becomes

$$\begin{aligned} \nabla F(\mathbf{w}^{(n+1)}) + \quad^{(n)} + 2\rho\mathbf{M}\mathbf{w}^{(n+1)} - \rho\mathbf{L}_+\widetilde{\mathbf{w}}^{(n)} = 0, \\ \quad^{(n+1)} - \quad^{(n)} - \rho\mathbf{L}_-\widetilde{\mathbf{w}}^{(n+1)} = 0. \end{aligned} \tag{21}$$

As in [46, Lemma 1], the equations in (21) can be combined to obtain

$$\begin{aligned} \mathbf{w}^{(n+1)} = \frac{\mathbf{M}^{-1}\nabla F(\mathbf{w}^{(n+1)})}{2\rho} + \frac{\mathbf{M}^{-1}\mathbf{L}_+\widetilde{\mathbf{w}}^{(n)}}{2} \\ - \frac{\mathbf{M}^{-1}\mathbf{L}_-}{2}\sum_{s=0}^{n}\widetilde{\mathbf{w}}^{(s)}. \end{aligned} \tag{22}$$

Similarly to [46], by introducing the following:

$$\begin{aligned} \mathbf{Q} &= \sqrt{\mathbf{L}_-/2}, & \mathbf{r}^{(n)} &= \sum_{s=0}^{n}\mathbf{Q}\widetilde{\mathbf{w}}^{(s)}, \\ \mathbf{q}^{(n)} &= \begin{pmatrix}\mathbf{r}^{(n)} \\ \widetilde{\mathbf{w}}^{(n)}\end{pmatrix}, & \mathbf{G} &= \begin{bmatrix}\rho\mathbf{I} & 0 \\ 0 & \rho\mathbf{L}_+/2\end{bmatrix}, \end{aligned}$$

(22) can be reformulated using [46, Lemma 2] as

$$\begin{aligned} \frac{\nabla F(\mathbf{w}^{(n+1)})}{\rho} + 2\mathbf{Q}\mathbf{r}^{(n+1)} \\ + \mathbf{L}_+\big(\mathbf{w}^{(n+1)} - \widetilde{\mathbf{w}}^{(n)} \quad = 2\mathbf{M}\boldsymbol{\xi}^{(t+1)}. \end{aligned} \tag{23}$$

Theorem 1: Under Assumption 1, if $\tau^{(n)} = \tau = 0, \forall n$, the proposed PGFL algorithm converges to the optimal solution of (2) in linear time for each cluster.

Proof: Under Assumption 1, $F(\mathbf{w})$ is convex and smooth by composition and, therefore, differentiable. Using [47, Lemma 6] and [47, Theorem V] with a convex and smooth function $F(\mathbf{w})$ demonstrates that the proposed PGFL algorithm, without inter-cluster learning ($\tau = 0$), converges to the optimal solution of (2) in linear time for any given cluster.

C. Impact of Inter-Cluster Learning

In situations with limited data, as demonstrated in Section V, employing inter-cluster learning ($\tau \neq 0$) can enhance performance compared to $\tau = 0$. This section establishes an upper bound on the disparity between the resulting cluster-specific personalized models obtained in scenarios with and without inter-cluster learning. It is worth noting that this bound can be controlled by properly choosing the sequence $\tau(n)$.

To do so, it is necessary to reformulate the client primal update using Assumption 1. The primal update for client $k \in \mathcal{C}_{s,(q)}$ is expressed as follows:

$$\begin{aligned} \mathbf{w}_k^{(n+1)} = \arg\min_{\mathbf{w}} f_k(\mathbf{w}) - \left\langle \boldsymbol{\varphi}_k^{(n)}, \mathbf{w} - \mathbf{w}_{s,(q)}^{(n)} \right\rangle \\ + \frac{\rho}{2}||\mathbf{w} - \mathbf{w}_{s,(q)}^{(n)}||^2, \end{aligned} \tag{24}$$

which, under Assumption 1, is equivalent to

$$\nabla f_k(\mathbf{w}_k^{(n+1)}) - \boldsymbol{\varphi}_k^{(n)} + \rho\Big(\mathbf{w}_k^{(n+1)} - \mathbf{w}_{s,(q)}^{(n)}\Big) = 0. \tag{25}$$

Further reformulation leads to the following:

$$\mathbf{w}_k^{(n+1)} = \mathbf{w}_{s,(q)}^{(n)} + \frac{1}{\rho}\boldsymbol{\varphi}_k^{(n)} - \frac{1}{\rho}\nabla f_k(\mathbf{w}_k^{(n+1)}). \tag{26}$$

By replacing $\mathbf{w}_k^{(n+1)}$ with (26) in (8), we obtain

$$\widehat{\mathbf{w}}_{s,(q)}^{(n)} = \frac{1}{|\mathcal{N}_s|}\sum_{p\in\mathcal{N}_s}\frac{1}{|\mathcal{C}_{p,(q)}|}\sum_{k\in\mathcal{C}_{p\quad q)}}\Big(\mathbf{w}_{p,(q)}^{(n-1)} - \frac{1}{\rho}\nabla f_k(\mathbf{w}_k^{(n)})\Big). \tag{27}$$

Next, we investigate the effect of inter-cluster learning by comparing the performance of models obtained using the PGFL algorithm with and without inter-cluster learning. We shall prove that the difference between the resulting models is bounded and depends on both the inter-cluster learning parameter and the similarity of models between clusters.

Theorem 2: Given a sufficiently large penalty parameter ρ, for all iterations, server $s \in \mathcal{S}$ and cluster $q \in \mathcal{Q}$, the impact of inter-cluster learning after n iterations is bounded by

$$\Big[\Big|\Big|\bar{\mathbf{w}}_{s,(q)}^{(n)} - \mathbf{w}_{s,(q)}^{(n)}\Big|\Big|_2^2\Big] \leqslant \sum_{i=1}^{n}\left(\prod_{j=i+1}^{n}\big(1-\tau^{(j)}\quad\right)\tau^{(i)}\eta, \tag{28}$$

where the expectation is taken with respect to the privacy-related noise added in (12) and the data observation noise, $\bar{\mathbf{w}}_{s,(q)}^{(n)}$ denotes the model obtained by the algorithm without inter-cluster learning, and η is the maximum cluster model distance, defined as:

$$\eta = \max_{q,r\in\mathcal{Q}}\Big|\Big|\mathbf{w}_{(q)}^* - \mathbf{w}_{(r)}^*\Big|\Big|_2^2, \tag{29}$$

with the models $\mathbf{w}_{(q)}^*, q \in \mathcal{Q}$ being the cluster-specific solutions of (2) with $\tau = 0$.

Proof: We prove this theorem by induction. With initial values $\mathbf{w}_{s,(q)}^{(0)} =$ and $\bar{\mathbf{w}}_{s,(q)}^{(0)} =$, one can write.

$$\begin{aligned} \mathbf{w}_{s,(q)}^{(1)} &= \big(1-\tau^{(1)}\ \ \widehat{\mathbf{w}}_{s,(q)}^{(1)} + \frac{\tau^{(1)}}{Q-1}\sum_{r\in\mathcal{Q}\setminus q}\widehat{\mathbf{w}}_{s,(r)}^{(1)}, \\ \bar{\mathbf{w}}_{s,(q)}^{(1)} &= \frac{1}{|\mathcal{N}_s|}\sum_{p\in\mathcal{N}_s}\frac{1}{|\mathcal{C}_{p,(q)}|}\sum_{k\in\mathcal{C}_{p\quad q)}}\Big(\bar{\mathbf{w}}_{p,(q)}^{(0)} - \frac{1}{\rho}\nabla f_k\left(\bar{\mathbf{w}}_k^{(1)}\right)\Big), \end{aligned} \tag{30}$$

where, given that $\bar{\mathbf{w}}_{p,(q)}^{(0)} = \mathbf{w}_{p,(q)}^{(0)}$ and $\bar{\mathbf{w}}_k^{(0)} = \mathbf{w}_k^{(0)}$, and using (27), we have $\widehat{\mathbf{w}}_{s,(q)}^{(1)} = \bar{\mathbf{w}}_{s,(q)}^{(1)}$. Hence,

$$\bar{\mathbf{w}}_{s,(q)}^{(1)} - \mathbf{w}_{s,(q)}^{(1)} = \frac{\tau^{(1)}}{Q-1}\sum_{r\in\mathcal{Q}\setminus q}\Big(\bar{\mathbf{w}}_{s,(q)}^{(1)} - \widehat{\mathbf{w}}_{s,(r)}^{(1)}\Big). \tag{31}$$

Taking the expectation with respect to the privacy-related and observation noises, we can express this difference as a function of the inter-cluster learning parameter and the maximum cluster model distance.

$$[||\bar{\mathbf{w}}_{s,(q)}^{(1)} - \mathbf{w}_{s,(q)}^{(1)}||_2^2] \leqslant \tau^{(1)}\eta. \tag{32}$$

Further, we assume that (28) is satisfied for all iterations up to iteration $n-1$. For iteration n, we have

$$\mathbf{w}_{s,(q)}^{(n)} = \big(1-\tau^{(n)}\ \widehat{\mathbf{w}}_{s,(q)}^{(n)} + \frac{\tau^{(n)}}{Q-1}\sum_{r\in\mathcal{Q}\setminus q}\widehat{\mathbf{w}}_{s,(r)}^{(n)},$$

$$\bar{\mathbf{w}}_{s,(q)}^{(n)} = \frac{1}{|\mathcal{N}_s|}\sum_{p\in\mathcal{N}_s}\frac{1}{|\mathcal{C}_{p,(q)}|}\sum_{k\in\mathcal{C}_{p\ \ q)}}\Big(\bar{\mathbf{w}}_{p,(q)}^{(n-1)} - \frac{1}{\rho}\nabla f_k(\bar{\mathbf{w}}_k^{(n)})\Big), \tag{33}$$

where $\widehat{\mathbf{w}}_{s,(q)}^{(n)} \neq \bar{\mathbf{w}}_{s,(q)}^{(n)}$ since

$$\widehat{\mathbf{w}}_{s,(q)}^{(n)} = \frac{1}{|\mathcal{N}_s|}\sum_{p\in\mathcal{N}_s}\frac{1}{|\mathcal{C}_{p,(q)}|}\sum_{k\in\mathcal{C}_{p\ \ q)}}\Big(\mathbf{w}_{p,(q)}^{(n-1)} - \frac{1}{\rho}\nabla f_k(\mathbf{w}_k^{(n)})\Big). \tag{34}$$

The difference is given by

$$\bar{\mathbf{w}}_{s,(q)}^{(n)} - \mathbf{w}_{s,(q)}^{(n)} = \big(1-\tau^{(n)}\ \Big(\bar{\mathbf{w}}_{s,(q)}^{(n)} - \widehat{\mathbf{w}}_{s,(q)}^{(n)}\Big) + \frac{\tau^{(n)}}{Q-1}\sum_{r\in\mathcal{Q}\setminus q}\Big(\bar{\mathbf{w}}_{s,(q)}^{(n)} - \widehat{\mathbf{w}}_{s,(r)}^{(n)}\Big), \tag{35}$$

with

$$\bar{\mathbf{w}}_{s,(q)}^{(n)} - \widehat{\mathbf{w}}_{s,(q)}^{(n)} = \frac{1}{|\mathcal{N}_s|}\sum_{p\in\mathcal{N}_s}\frac{1}{|\mathcal{C}_{p,(q)}|}\sum_{k\in\mathcal{C}_{p\ \ q)}}\Big(\bar{\mathbf{w}}_{p,(q)}^{(n-1)} - \mathbf{w}_{p,(q)}^{(n-1)} - \frac{1}{\rho}\nabla f_k(\bar{\mathbf{w}}_k^{(n)}) + \frac{1}{\rho}\nabla f_k(\mathbf{w}_k^{(n)})\Big). \tag{36}$$

We note that the expectation of $||\bar{\mathbf{w}}_{p,(q)}^{(n-1)} - \mathbf{w}_{p,(q)}^{(n-1)}||_2^2$ with respect to the privacy-related and observation noises is identical for all servers. Therefore, since (28) is satisfied for iteration $n-1$ for all servers, given a sufficiently large penalty parameter ρ, and taking the expectation with respect to the privacy-related and observation noises, we have

$$||\bar{\mathbf{w}}_{s,(q)}^{(n)} - \widehat{\mathbf{w}}_{s,(q)}^{(n)}||_2^2 \leqslant \ ||\bar{\mathbf{w}}_{s,(q)}^{(n-1)} - \mathbf{w}_{s,(q)}^{(n-1)}||_2^2. \tag{37}$$

Combining (35) and (37), we will have

$$||\bar{\mathbf{w}}_{s,(q)}^{(n)} - \mathbf{w}_{s,(q)}^{(n)}||_2^2 \leqslant (1-\tau^{(n)})\ \ ||\bar{\mathbf{w}}_{s,(q)}^{(n-1)} - \mathbf{w}_{s,(q)}^{(n-1)}||_2^2 + \frac{\tau^{(n)}}{Q-1}\sum_{r\in\mathcal{Q}\setminus q}\ ||\bar{\mathbf{w}}_{s,(q)}^{(n)} - \widehat{\mathbf{w}}_{s,(r)}^{(n)}||_2^2, \tag{38}$$

which, using the maximum cluster model distance, yields

$$||\bar{\mathbf{w}}_{s,(q)}^{(n)} - \mathbf{w}_{s,(q)}^{(n)}||_2^2 \leqslant \big(1-\tau^{(n)}\ \ \ ||\bar{\mathbf{w}}_{s,(q)}^{(n-1)} - \mathbf{w}_{s,(q)}^{(n-1)}||_2^2 + \tau^{(n)}\eta. \tag{39}$$

Given (28) for iteration $n-1$, we have

$$||\bar{\mathbf{w}}_{s,(q)}^{(n)} - \mathbf{w}_{s,(q)}^{(n)}||_2^2 \leqslant \big(1-\tau^{(n)}\ \sum_{i=1}^{n-1}\left(\prod_{j=i+1}^{n-1}(1-\tau^{(j)})\right)\tau^{(i)}\eta + \tau^{(n)}\eta,$$

$$\leqslant \sum_{i=1}^{n}\left(\prod_{j=i+1}^{n}(1-\tau^{(j)})\right)\tau^{(i)}\eta. \tag{40}$$

That is, (28) is satisfied for iteration n.

By the principle of induction, (28) is satisfied for all iterations, server $s\in\mathcal{S}$ and cluster $q\in\mathcal{Q}$.

Corollary: If $\tau^{(i)}=0, \forall i<n$ and $\tau^{(n)}\neq 0$, the impact of a single iteration of inter-cluster learning is bounded by

$$||\bar{\mathbf{w}}_{s,(q)}^{(n)} - \mathbf{w}_{s,(q)}^{(n)}||_2^2 \leqslant \tau^{(n)}\eta, \tag{41}$$

where $\bar{\mathbf{w}}_{s,(q)}^{(n)}$ denotes a model obtained without inter-cluster learning, η is as defined in Theorem 2, and the expectation is taken with respect to the privacy-related and observation noises.

Theorem 2 bounds the difference in the resulting models with and without inter-cluster learning. Combining Theorems 1 and 2, the resulting models obtained by the algorithms are guaranteed to reside within a neighborhood of the optimal solution of (2) with $\tau=0$. The size of this neighborhood can be adjusted by selecting the sequence $\tau^{(n)}$. When ample data is available, the algorithm converges to a satisfactory solution within this neighborhood. However, in cases of limited data, the solution of (2) with $\tau=0$ may be inadequate. In such situations, inter-cluster learning becomes crucial, allowing the proposed algorithm to achieve higher accuracy, as demonstrated in Section V. By exploiting inter-cluster learning, the algorithm effectively overcomes the limitations imposed by scarce data, leading to improved performance.

IV. Privacy Analysis

This section focuses on quantifying the local privacy protection provided by the proposed PGFL algorithm. To achieve this, we begin by calculating the l_2-norm sensitivity, which quantifies the variation in output resulting from a change in an individual data sample. Once we have established the l_2-norm sensitivity, we proceed to adjust the noise variance added to the primal variables, ensuring satisfactory protection.

Definition: The l_2-norm sensitivity is defined by

$$\Delta_{k,2}^{(n)} = \max_{\mathcal{D}_k,\mathcal{D}_l}\left\|\mathbf{w}_{k,\mathcal{D}_k}^{(n)} - \mathbf{w}_{k,\mathcal{D}_l}^{(n)}\right\| \tag{42}$$

where $\mathbf{w}_{k,\mathcal{D}_k}^{(n)}$ and $\mathbf{w}_{k,\mathcal{D}_l}^{(n)}$ denote the local primal variables obtained from two neighboring data sets $\mathcal{D}_k$ and $\mathcal{D}_l$, which differ in only one data sample.

Assumption 3: The functions $\ell_k(\ )$, $k\in\mathcal{C}$, have bounded gradients. That is, for $k\in\mathcal{C}$ there exists a constant C_k such that $||\nabla\ell_k(\ )||\leqslant C_k$.

Lemma 1: Under Assumption 3, the l_2-norm sensitivity for a client k is given by

$$\Delta_{k,2}^{(n)} = \max_{\mathcal{D}_k,\mathcal{D}_l}||\mathbf{w}_{k,\mathcal{D}_k}^{(n)} - \mathbf{w}_{k,\mathcal{D}_l}^{(n)}|| = \frac{2C_k}{\rho D_k}. \tag{43}$$

Proof: We consider two neighboring data sets for a client k, $\mathcal{D}_k$ and $\mathcal{D}_l$, both of cardinality D_k. For simplicity, we assume that they differ on the last data sample. We denote $\mathbf{w}_{k,\mathcal{D}_k}^{(n)}$ the model obtained using the initial data set, and $\mathbf{w}_{k,\mathcal{D}_l}^{(n)}$ the model obtained using the alternative data set. Those are obtained,

according to (4), by:

$$\mathbf{w}_{k,\mathcal{D}_k}^{(n)} = \arg\min_{\mathbf{w}} \frac{1}{D_k}\sum_{i=1}^{D_k}\ell_k(\mathbf{x}_{k,i},y_{k,i};\mathbf{w}) + \frac{1}{|\mathcal{C}_s|}R(\mathbf{w}) - \left\langle \boldsymbol{\varphi}_k^{(n-1)}, \mathbf{w}-\mathbf{w}_{s,(q)}^{(n-1)}\right\rangle + \frac{\rho}{2}||\mathbf{w}-\mathbf{w}_{s,(q)}^{(n-1)}||^2,$$

$$\mathbf{w}_{k,\mathcal{D}_l}^{(n)} = \arg\min_{\mathbf{w}} \frac{1}{|\mathcal{C}_s|}R(\mathbf{w}) + \frac{1}{D_k}\left(\sum_{i=1}^{D_k-1}\ell_k(\mathbf{x}_{k,i},y_{k,i};\mathbf{w}) + \ell_k(\mathbf{x}'_{k,D_k},y'_{k,D_k};\mathbf{w})\right) - \left\langle \boldsymbol{\varphi}_k^{(n-1)}, \mathbf{w}-\mathbf{w}_{s,(q)}^{(n-1)}\right\rangle + \frac{\rho}{2}||\mathbf{w}-\mathbf{w}_{s,(q)}^{(n-1)}||^2.$$

Using (26), that we recall:

$$\mathbf{w}_k^{(n)} = \mathbf{w}_{s,(q)}^{(n-1)} + \frac{1}{\rho}\boldsymbol{\varphi}_k^{(n-1)} - \frac{1}{\rho}\nabla f_k(\mathbf{w}_k^{(n)}), \tag{44}$$

we can derive:

$$||\mathbf{w}_{k,\mathcal{D}_k}^{(n)} - \mathbf{w}_{k,\mathcal{D}_l}^{(n)}|| = \left\|\frac{1}{\rho D_k}(\nabla\ell_k(\mathbf{x}_{k,D_k},y_{k,D_k};\mathbf{w}_k) - \nabla\ell_k(\mathbf{x}'_{k,D_k},y'_{k,D_k};\mathbf{w}_k))\right\|, \tag{45}$$

which, under Assumption 3, provides a value for the l_2-norm sensitivity:

$$\max_{\mathcal{D}_k,\mathcal{D}_l}||\mathbf{w}_{k,\mathcal{D}_k}^{(n)} - \mathbf{w}_{k,\mathcal{D}_l}^{(n)}|| = \frac{2C_k}{\rho D_k}. \tag{46}$$

With the l_2-norm sensitivity, we can establish the relation between the noise variance added in (12) and the privacy parameter $\phi_k^{(n)}$ as well as prove the privacy guarantee of the algorithm in terms of zCDP.

Theorem 3: Under Assumption 3, PGFL satisfies dynamic $\phi_k^{(n)}$-zCDP with the relation between the privacy parameter and the perturbation noise variance given by

$$\delta_k^{2(n)} = \frac{^{2(n)}_{k,2}}{2\phi_k^{(n)}}. \tag{47}$$

Proof: For any client k and iteration n, the perturbed primal update is obtained with (12). That is, it is equivalent to $\widetilde{\mathbf{w}}_k^{(n)} \sim \mathcal{N}(\mathbf{w}_k^{(n)}, \delta_k^{2(n)}\mathbf{I})$. The result in [36, Proposition 6], states that a sensitivity- query q releasing an output $\mathcal{N}(q(x),\delta^2)$ from an input x satisfies $(\ ^2/2\delta^2) - zCDP$. Thus, the PGFL algorithm satisfies the dynamic $\phi_k^{(n)}$-zCDP with $\phi_k^{(n)} = \frac{^{2\ n)}_{k\ 2}}{2\delta_k^{2\ n)}}$.

Theorem 3 gives the relationship between the noise perturbation variance and the privacy protection at a given iteration. Since the proposed algorithm is iterative in nature and models are exchanged several times with the servers, one should consider the total privacy loss throughout the learning process. To this aim, we establish the following theorem.

Theorem 4: Under Assumption 3 and for a final iteration N, the PGFL algorithm satisfies ϕ_k^{total}-zCDP throughout the entire computation for each client k, with ϕ_k^{total} given by

$$\phi_k^{\text{total}} = \sum_{n=1}^{N}\phi_k^{(n)}. \tag{48}$$

Proof: This theorem results from the use of [36, Lemma 7] N times over.

V. Numerical Simulations

This section illustrates the performance of the proposed PGFL algorithm for solving regression and classification tasks.

A. Experiments for Regression

We consider a graph federated network consisting of $|\mathcal{S}| = 10$ servers, each having access to $|\mathcal{C}_s| = 15$ clients, for a total of $|\mathcal{C}| = 150$ clients. The set of servers and their communication channels form a random connected graph where the average node degree is three. Each client has access to a random number of noisy data samples between $D_k = 2$ and $D_k = 9$, each composed of a vector $\mathbf{x}_{k,i}$ of dimension $d = 60$ and a response scalar $y_{k,i}$. Doing so, each cluster is globally observable but not locally at any given client or set $\mathcal{C}_s, s \in \mathcal{S}$. The servers implement random scheduling of clients to reduce the communication load [48]. In particular, at every global iteration, each server randomly selects a subset of three clients to participate in the learning process.

The clients of the network are randomly split between $Q = 3$ clusters. Clients of a given cluster solve the ridge regression problem with data generated from an original model $\mathbf{w}_{(q)}^*$, obtained with $\mathbf{w}_{(q)}^* = \mathbf{w}_0^* + \gamma\mathbf{w}_0^*$ with $\gamma \sim \mathcal{U}(-0.15, 0.15)$, where $\mathbf{w}_0^*$ is a base model. In doing so, the learning tasks of different clusters share the same objective functions but have different, albeit related, data distributions. The loss and regularizer functions are given by

$$\ell_k(\mathbf{X}_k,\mathbf{y}_k;\mathbf{w}_k) = ||\mathbf{y}_k - \mathbf{X}_k\mathbf{w}_k||^2, \quad R(\mathbf{w}_k) = ||\mathbf{w}_k||^2. \tag{49}$$

Performance is evaluated by computing the normalized mean squared deviation (NMSD) of the local models with respect to the corresponding cluster-specific original model used to generate the data, $\mathbf{w}_{(q)}^*$ for $k \in \mathcal{C}_{(q)}$. It is given by:

$$\gamma^{(n)} = \frac{1}{|\mathcal{C}|}\sum_{q=1}^{|\mathcal{Q}|}\sum_{k\in\mathcal{C}_{\ q)}}\frac{\left\|\mathbf{w}_k^{(n)} - \mathbf{w}_{(q)}^*\right\|_2^2}{\left\|\mathbf{w}_{(q)}^*\right\|_2^2}, \tag{50}$$

where the result is averaged over several Monte Carlo iterations. The proposed algorithm is compared with various existing algorithms. The ClusterFL algorithm, defined in [49], implements conventional personalized FL with inter-cluster learning. For a fair comparison, the ClusterFL algorithm has been modified to leverage similarity among tasks in the same manner as the PGFL algorithm. The GFL algorithm, defined in [5], implements single-task graph FL in a privacy-preserving manner. To ensure a fair comparison, the ClusterFL and GFL algorithms have been modified to ensure privacy in the same manner as the PGFL algorithm. Furthermore, the algorithms are set to observe the same initial convergence rate whenever possible. For most

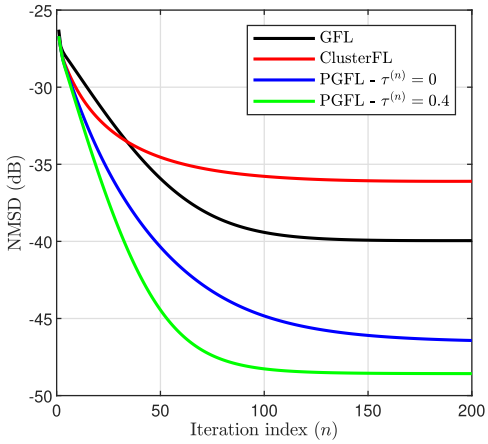

Fig. 1. Learning curves of the PGFL algorithm with a fixed inter-cluster learning parameter and the FedAvg algorithm, without client scheduling or privacy.

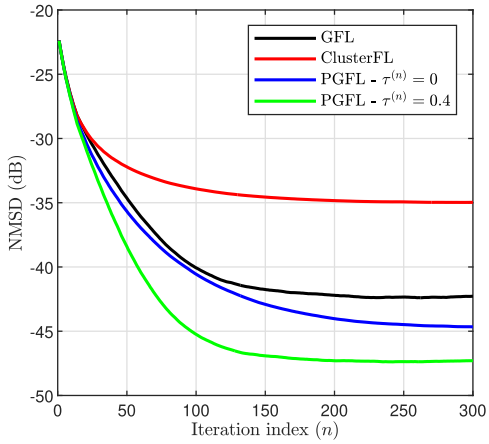

Fig. 2. Learning curves of the PGFL algorithm with a fixed inter-cluster learning parameter and the FedAvg algorithm, considering client scheduling and without privacy.

experiments, the learning curves are displayed as plots of the NMSD versus the iteration index.

We first consider an ideal setting wherein all algorithms are evaluated without privacy considerations ($\boldsymbol{\xi}^{(n)} = \quad, \forall n$)) and client scheduling. In this scenario, the inter-cluster parameter $\tau^{(n)}$ of the PGFL algorithm was kept fixed throughout the learning, specifically, $\tau^{(n)} = 0$ and $\tau^{(n)} = 0.4$. Fig. 1 shows the learning curves for the GFL, ClusterFL, and PGFL algorithms. The results illustrate the superiority of the proposed PGFL algorithm over GFL, as cluster-specific learning tasks benefit significantly from personalized models tailored to each cluster. We also see that incorporating inter-cluster learning results in improved convergence speed and steady-state accuracy. Furthermore, the performance of the ClusterFL algorithm is notably poor in this setting, emphasizing the importance of using the graph federated architecture when data is scarce. Leveraging the model similarities improves learning speed and accuracy by compensating for data scarcity. In addition, isolated servers whose clients lack sufficient data to achieve satisfactory accuracy independently reinforce the necessity of the graph federated architecture.

Next, we modify the setting to incorporate client scheduling and evaluate the aforementioned algorithms with reduced communication load. Fig. 2 shows the learning curves for the GFL, ClusterFL, and PGFL algorithms with client scheduling. In this figure and the ones below, 3 clients out of 15 are randomly selected to participate by each server at every iteration, reducing the communication load by 80% for every algorithm. We observe that the PGFL algorithm exhibits slower convergence and higher steady-state NMSD when utilizing client scheduling. And we note that GFL performs better with client scheduling. The performance degradation for the PGFL algorithm is due to the lower client participation resulting in a smaller quantity of data being utilized. The better performance of GFL in this setting is due to the imbalance of cluster representation in the universal model, which benefits the participating clients on average.

Finally, we evaluate the aforementioned algorithms in a setting with client scheduling and privacy protection. All of the algorithms utilize zCDP with the noise perturbation presented in (12) and the parameters $\phi_k^{(0)} = 0.001, \forall k$ and $\zeta = 0.99$.

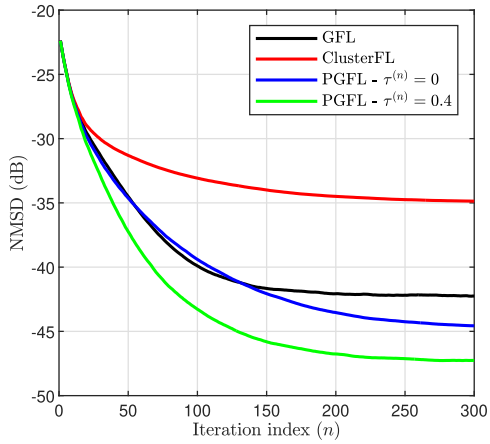

Fig. 3. Learning curves of the PGFL algorithm with a fixed inter-cluster learning parameter and the FedAvg algorithm, considering client scheduling and privacy.

Hence, all the algorithms satisfy ϕ_k^{final}-zCDP throughout the computation with $\phi_k^{\text{final}} = 0.095, \forall k$. Fig. 3 shows the learning curves for the GFL, ClusterFL, and PGFL algorithms with client scheduling and privacy. We observe that the noise perturbation associated with differential privacy significantly reduces the convergence speed of all the simulated algorithms. However, we note that the NMSD after 300 iterations is nearly identical to the one in Fig. 2. This behavior is explained by the use of zCDP, in which the variance of the noise perturbation starts high and decreases linearly throughout the learning process.

Further, we illustrate the importance of carefully choosing the value of the inter-cluster learning parameter. In Fig. 4, we simulated the proposed PGFL algorithm for various fixed $\tau^{(n)}$ values and displayed the NMSD after 200 iterations. For instance, the NMSD for $\tau^{(n)} = 0.4$ corresponds to the result obtained in Fig. 3. This figure confirms that inter-cluster learning has the potential to increase learning performance by alleviating data scarcity, as the PGFL algorithm achieves lower NMSD with $\tau^{(n)} \in (0.1, 0.5)$ than with $\tau^{(n)} = 0$. It also shows that the

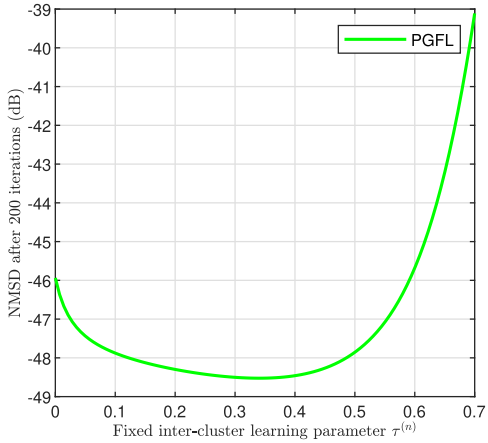

Fig. 4. NMSD after 200 iterations vs. fixed inter-cluster learning parameter τ) values for the PGFL algorithm with client scheduling and privacy.

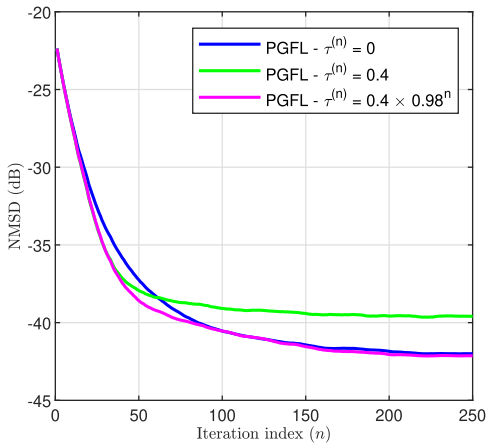

Fig. 5. Learning curves of the PGFL algorithm with fixed and time-varying inter-cluster learning parameter τ) in a setting with low cluster similarity, considering client scheduling and privacy.

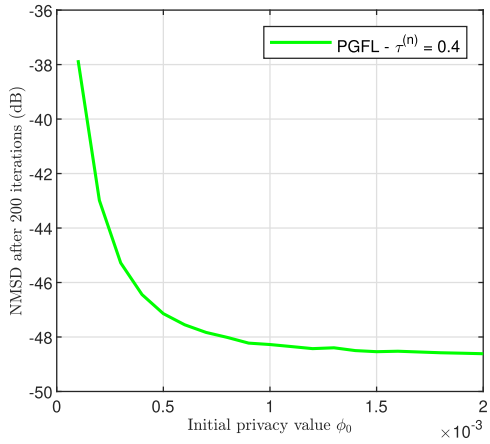

Fig. 6. Privacy-accuracy trade-off of the PGFL algorithm for ϕ_0 with a fixed inter-cluster learning parameter, considering client scheduling.

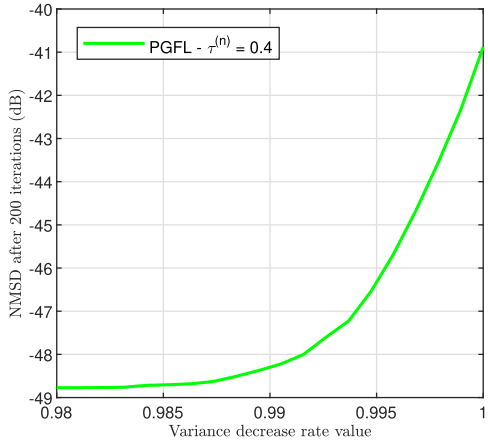

Fig. 7. Privacy-accuracy trade-off of the PGFL algorithm for with a fixed inter-cluster learning parameter, considering client scheduling.

inter-cluster learning parameter must be carefully selected, as a value too large for the setting leads to performance degradation.

We then illustrate an alternative use of inter-cluster learning. For this experiment, the difference between the data distribution of the different clusters has been increased. Precisely, the datasets were simulated with the models obtained by $\mathbf{w}_{(q)} = \mathbf{w}_0 + \gamma \mathbf{w}_0$ with $\gamma \sim \mathcal{U}(-0.5, 0.5)$. The learning curves are presented in Fig. 5. We observed that, because of the higher cluster dissimilarity, inter-cluster learning degrades steady-state NMSD; this is observed in the learning curves for PGFL with $\tau^{(n)} = 0$ and $\tau^{(n)} = 0.4$. However, by mitigating data scarcity within a cluster, inter-cluster learning improves the initial convergence rate. To benefit from an improved initial convergence rate and avoid steady-state performance degradation, it is possible to reduce the inter-cluster learning parameter progressively. Doing so, the PGFL algorithm with time-varying $\tau^{(n)} = 0.4 \times 0.98^n$ has the same initial convergence rate as the PGFL algorithm with fixed $\tau = 0.4$ and attains near-identical steady-state NMSD as the PGFL algorithm with fixed $\tau = 0$.

Finally, we study the impact of privacy protection on the steady-state NMSD of the PGFL algorithm. Fig. 6 shows the NMSD after 200 iterations versus the initial value of the privacy parameter ϕ_0 for a decaying rate of $\zeta = 0.99$. Note that, as seen in Theorem 3, a lower value of ϕ_0 ensures more privacy. We observe that for smaller values of ϕ_0, the steady-state NMSE of the PGFL algorithm is higher. In fact, a lower total privacy loss bound leads to higher perturbation noise variance and diminishes the learning performance of the algorithm. Similarly, Fig. 7 shows the NMSD after 200 iterations versus the variance decrease rate ζ for an initial privacy value of $\phi_0 = 0.001$. The lower the decrease rate, the faster the privacy protection weakens, and the lower the steady-state NMSE of the algorithm as more information is exchanged among clients. On the other hand, a decrease rate close to 1 ensures better privacy protection but comes at the cost of lower accuracy.

B. *Experiments for Classification on the MNIST Dataset*

The following experiments were conducted on the MNIST handwritten digits dataset [50]. In those experiments, the learning tasks of the clients associated with different clusters share

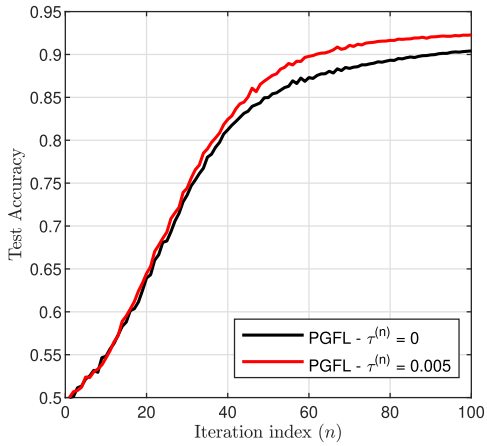

Fig. 8. Test accuracy curve of the PGFL algorithm with a fixed inter-cluster learning parameter on MNIST, considering client scheduling and privacy, with low cluster similarity.

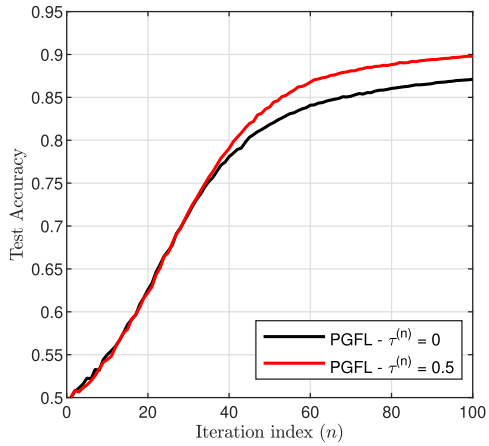

Fig. 9. Test accuracy curve of the PGFL algorithm with a fixed inter-cluster learning parameter on MNIST, considering client scheduling and privacy, with high cluster similarity.

the same data but have different, related, objective functions. The structure of the server network, as well as the number of clients per server, are identical to the experiments for regression. In the following experiments, the clients of a given cluster use the ADMM for logistic regression to differentiate between two classes. The loss function for the logistic regression is given by

$$\log[\ell_k(\mathbf{X}_k, \mathbf{y}_k; \mathbf{w}_k)] = \frac{-1}{D_k} \sum_{i=1}^{D_k} \Big(y_{k,i} \log[y'_{k,i}] + (1 - y_{k,i}) \log[1 - y'_{k,i}] \Big), \quad (51)$$

with

$$y'_{k,i} = \frac{1}{1 + \exp(-\mathbf{w}_k \mathbf{x}_{k,i})}. \quad (52)$$

We simulated the PGFL algorithm in the context of classification with client scheduling, privacy, a fixed inter-cluster learning parameter $\tau^{(n)} = \tau = 0.4$, and without inter-cluster learning $\tau^{(n)} = 0$. Fig. 8 shows the test accuracy versus iteration index in a setting where the clients of a given cluster must differentiate between two classes composed of a single digit. Each client receives between $D_k = 2$ and $D_k = 4$ data samples composed of two MNIST images. The clients of cluster 1 have access to images of the digits $\{1\}$ and $\{8\}$. The clients of clusters 2 and 3 have access to images of the digits $\{1\}$ and $\{9\}$, and $\{7\}$ and $\{8\}$, respectively. Given that the clients of different clusters must differentiate between different digits, the similarity between the learning task is limited. Nevertheless, we observe that inter-cluster learning does improve the accuracy of the PGFL algorithm in this setting.

Further, we modified the setting so that the clusters exhibit more similarity. Fig. 9 shows the test accuracy versus iteration index in a setting where the clients of a given cluster must differentiate between two classes composed of triplets of digits. Each client receives between $D_k = 6$ and $D_k = 12$ data samples, each composed of two triplets of MNIST images. The clients of cluster 1 must differentiate between the classes $\{1, 2, 3\}$ and $\{6, 7, 8\}$, the clients of cluster 2 between $\{1, 2, 3\}$ and $\{7, 8, 9\}$,

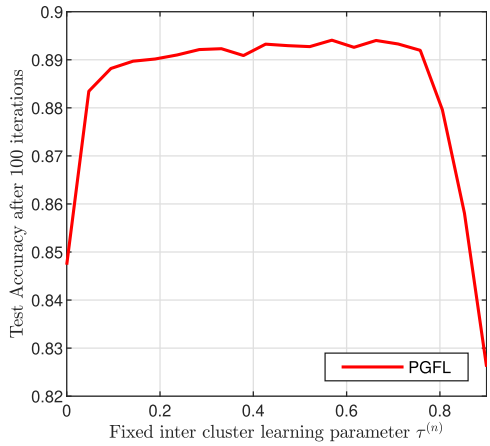

Fig. 10. Test accuracy of the PGFL algorithm on MNIST after 100 iterations vs. fixed inter-cluster learning parameter τ), considering client scheduling and privacy.

and the clients of cluster 3 between $\{1, 2, 3\}$ and $\{6, 8, 9\}$. We observe that, in this setting, inter-cluster learning significantly improves the accuracy of the PGFL algorithm.

Finally, we utilize the previous setting and evaluate the impact of the value of the inter-cluster learning parameter $\tau^{(n)}$ on the accuracy achieved by the PGFL algorithm in the context of classification. Fig. 10 displays the accuracy achieved by the PGFL algorithm after 100 iterations versus the value of the inter-cluster learning parameter in the context of the classification task of Fig. 9. We observe that, in this setting where the similarity among the learning tasks is high, medium and large fixed values for $\tau^{(n)}$ lead to significant accuracy improvement. However, very large values lead to performance degradation, similar to Fig. 4.

C. Experiments for Classification on the MedMNIST Dataset

To demonstrate the proposed method of utilizing inter-cluster learning to palliate data scarcity and improve learning performance in real-life applications, two experiments are

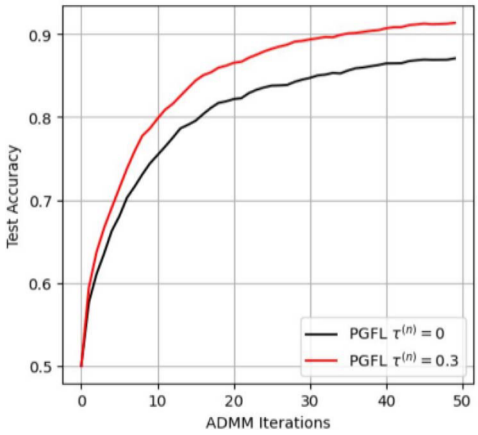

Fig. 11. Test accuracy curve of the PGFL algorithm with a fixed inter-cluster learning parameter on MedMNIST, considering privacy, with high cluster similarity.

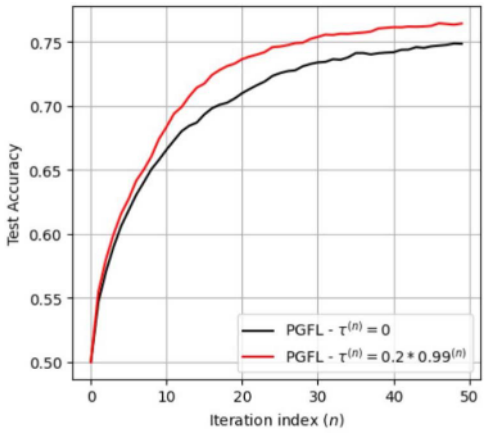

Fig. 12. Test accuracy curve of the PGFL algorithm with a varying inter-cluster learning parameter on MedMNIST, considering privacy, with low cluster similarity.

conducted on the OrganAMNIST dataset, part of the biomedical MedMNIST dataset [51]. The OrganAMNIST dataset contains lightweight images of 11 different organs labeled by type. It comprises more than 58000 data samples split into training, validation, and testing data. We use the proposed method to improve classification accuracy in the following setting. The server network and the loss function are identical to previous experiments; however, only three clients are associated with each server, each client having access to two data samples. In both experiments, clients of a given cluster are tasked with differentiating between two types of organs. Different clusters are associated with different pairs of organs, and inter-cluster learning is utilized to improve classification accuracy by leveraging the similarity between some of the organs.

In the first experiment, the three clusters are given similar learning tasks. In particular, one of the elements of each pair of organs is identical. Cluster 1 differentiates between the right lung and the left lung, cluster 2 between the liver and the left lung, and cluster 3 between the right kidney and the left lung. Fig. 11 shows the test accuracy versus iteration index. We observe that a large amount of inter-cluster learning leads to significantly improved performances, increasing classification accuracy by about 5 .

In the next experiment, the learning tasks associated with each cluster are less similar than in the previous experiment. They share only the vague shape of the classified organs. Cluster 1 differentiates between the spleen and the left lung, cluster 2 between the left kidney and the bladder, and cluster 3 between the right kidney and the right lung. Due to the lower cluster similarity, we utilize a decaying inter-cluster learning parameter to preserve steady-state accuracy. Fig. 12 shows the test accuracy versus iteration index. We observe that a medium decay rate of the inter-cluster learning parameter can improve the learning speed, boosting classification accuracy by about 2 .

VI. Conclusion

This paper proposed a framework for personalized graph federated learning in which distributed servers collaborate with each other and their respective clients to learn cluster-specific personalized models. The proposed framework leverages the similarities among clusters to improve learning speed and alleviate data scarcity. Further, this framework is implemented with the ADMM as a local learning process and with local zero-concentrated differential privacy to protect the participants' data from eavesdroppers. Our mathematical analysis showed that this algorithm converges to the exact optimal solution for each cluster in linear time and that utilizing inter-cluster learning leads to an alternative output whose distance to the original solution is bounded by a value that can be adjusted with the inter-cluster learning parameter sequence. Finally, numerical simulations showed that the proposed method is capable of leveraging the graph federated architecture and the similarity between the clusters' learning tasks to improve learning performance.

References

[1] F. Gauthier, V. C. Gogineni, S. Werner, Y.-F. Huang, and A. Kuh, "Clustered graph federated personalized learning," in *Proc. IEEE Asilomar Conf. Signals Syst. Comput.*, 2022, pp. 744–748.
[2] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.
[3] 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.
[4] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, Jun. 2020.
[5] E. Rizk and A. H. Sayed, "A graph federated architecture with privacy preserving learning," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun.*, 2021, pp. 131–135.
[6] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–6.
[7] Y. Sarcheshmehpour, M. Leinonen, and A. Jung, "Federated learning from Big Data over networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 3055–3059.
[8] V. C. Gogineni, S. Werner, Y.-F. Huang, and A. Kuh, "Decentralized graph federated multitask learning for streaming data," in *Proc. IEEE Annu. Conf. Inf. Sci. Syst.*, 2022, pp. 101–106.
[9] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Comput. Ind. Eng.*, vol. 149, 2020, Art. no. 106854.
[10] B. Yang et al., "Edge intelligence for autonomous driving in 6G wireless system: Design challenges and solutions," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 40–47, Apr. 2021.

[11] S. Boll and J. Meyer, "Health-X dataLOFT: A sovereign federated cloud for personalized health care services," *IEEE MultiMedia*, vol. 29, no. 1, pp. 136–140, Jan.-Mar. 2022.
[12] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access,Mar. 28, 2022, doi: 10.1109/TNNLS.2022.3160699.
[13] C. T. Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with Moreau envelopes," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 21394–21405.
[14] C. J. Felix, J. Ye, and A. Kuh, "Personalized learning using kernel methods and random fourier features," in *Proc. Int. Joint Conf. Neural Netw.*, 2022, pp. 1–5.
[15] V. C. Gogineni, S. Werner, F. Gauthier, Y.-F. Huang, and A. Kuh, "Personalized online federated learning for IoT/CPS: Challenges and future directions," *IEEE Internet Things Mag.*, vol. 5, no. 4, pp. 78–84, Dec. 2022.
[16] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," 2020, *arXiv:2002.07948*.
[17] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," 2020, *arXiv:2003.13461*.
[18] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4129–4144, Aug. 2014.
[19] V. C. Gogineni and M. Chakraborty, "Improving the performance of multitask diffusion APA via controlled inter-cluster cooperation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 3, pp. 903–912, Mar. 2020.
[20] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," in *Proc. Int. Conf. Neural Info. Process. Syst.*, 2020, pp. 19586–19597.
[21] D. Caldarola, M. Mancini, F. Galasso, M. Ciccone, E. Rodolá, and B. Caputo, "Cluster-driven graph federated learning over multiple domains," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2749–2758.
[22] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3710–3722, Aug. 2021.
[23] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*. 2017, pp. 4427–4437.
[24] R. Li, F. Ma, W. Jiang, and J. Gao, "Online federated multitask learning," in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 215–220.
[25] A. Taïk and S. Cherkaoui, "Electrical load forecasting using edge computing and federated learning," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–6.
[26] F.-Z. Lian, J.-D. Huang, J.-X. Liu, G. Chen, J.-H. Zhao, and W.-X. Kang, "FedFV: A personalized federated learning framework for finger vein authentication," *Mach. Intell. Res.*, vol. 20, pp. 683–696, 2023.
[27] S. Wang, S. Hosseinalipour, M. Gorlatova, C. G. Brinton, and M. Chiang, "UAV-assisted online machine learning over multi-tiered networks: A hierarchical nested personalized federated learning approach," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 2, pp. 1847–1865, Jun. 2023.
[28] A. M. G. Salem, A. Bhattacharyya, M. Backes, M. Fritz, and Y. Zhang, "Updates-leak: Data set inference and reconstruction attacks in online learning," in *Proc. 29th USENIX Secur. Symp.*, 2020, pp. 1291–1308.
[29] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3454–3469, 2020.
[30] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Conf. Theory Cryptogr.*, 2006, pp. 265–284.
[31] J. Cortés, G. E. Dullerud, S. Han, J. L. Ny, S. Mitra, and G. J. Pappas, "Differential privacy in control and network systems," in *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 4252–4272.
[32] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, pp. 211–407, Aug. 2014.
[33] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3454–3469, 2020.
[34] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1376–1385.
[35] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," 2016, *arXiv:1603.01887*.
[36] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Proc. Theory Cryptogr. Conf.*, 2016, pp. 635–658.
[37] T. Zhang and Q. Zhu, "A dual perturbation approach for differential private ADMM-Based distributed empirical risk minimization," in *Proc. ACM Workshop Artif. Intell. Secur.*, 2016, pp. 129–137.
[38] S. Zhou and G. Y. Li, "Federated learning via inexact ADMM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9699–9708, Aug. 2023.
[39] Y. Chen, R. S. Blum, and B. M. Sadler, "Communication efficient federated learning via ordered ADMM in a fully decentralized setting," in *Proc. Int. Conf. Inf. Sci. Syst.*, 2022, pp. 96–100.
[40] S. Yue, J. Ren, J. Xin, S. Lin, and J. Zhang, "Inexact-ADMM based federated meta-learning for fast and continual edge learning," in *Proc. Int. Symp. Theory Algorithmic Found. Protocol Des. Mobile Netw. Mobile Comput. Assoc. Comput. Mach.*, 2021, pp. 91–100.
[41] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, Apr. 2014.
[42] J. Ding, X. Zhang, M. Chen, K. Xue, C. Zhang, and M. Pan, "Differentially private robust ADMM for distributed machine learning," in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 1302–1311.
[43] S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," in *Proc. Conf. Learn. Theory Kernel Workshop*, 2003, pp. 567–580.
[44] G. B. Giannakis, Q. Ling, G. Mateos, and I. D. Schizas, *Splitting Methods in Communication, Imaging, Science, and Engineering*. Berlin, Germany: Springer, Jan. 2017, pp. 461–497.
[45] V. C. Gogineni and M. Chakraborty, "Diffusion affine projection algorithm for multitask networks," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2018, pp. 201–206.
[46] Q. Li, B. Kailkhura, R. Goldhahn, P. Ray, and P. K. Varshney, "Robust decentralized learning using ADMM with unreliable agents," 2017, doi: 10.1109/TSP.2022.3178655.
[47] J. Ding, Y. Gong, M. Pan, and Z. Han, "Optimal differentially private ADMM for distributed machine learning," Feb. 2019*arXiv:1901.02094*.
[48] 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.
[49] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G. Xing, "ClusterFL: A similarity-aware federated learning system for human activity recognition," in *Proc. 19th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2021, pp. 54–66.
[50] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[51] J. Yang et al., "MedMNIST v2-A large-scale lightweight benchmark for 2D and 3D biomedical image classification," *Sci. Data*, vol. 10, no. 1, 2023, Art. no. 41.

Francois Gauthier (Member, IEEE) received the B.Sc. and M.Sc. degrees in mathematics and computer science from the École Nationale Supérieure d'Informatiques et de Mathématiques Appliquées de Grenoble, Saint-Martin-d'Hs, France, in 2015 and 2018. He is currently working toward the Ph.D. degree with the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. His research interests include federated learning, differential privacy, communication efficiency, personalized learning, and reinforcement learning.

Vinay Chakravarthi Gogineni (Senior Member, IEEE) received the bachelor's degree in electronics and communication engineering from Jawaharlal Nehru Technological University, Anantapur, India, in 2005, the master's degree in communication engineering from Vellore Institute of Technology, Vellore, India, in 2008, and the Ph.D. degree in electronics and electrical communication engineering from the Indian Institute of Technology Kharagpur, Kharagpur, India, in 2019. He is currently an Assistant Professor with SDU Applied AI and Data Science, The Maersk Mc-Kinney Moller Institute, Copenhagen, Denmark, University of Southern Denmark, Odense, Denmark. Prior to this, he was a Postdoctoral Research Fellow with NTNU and Simula, Norway. From 2008 to 2011, he was with a couple of MNCs in India. His research interests include deep learning, decentralized machine learning, geometric deep learning, and their application in healthcare and the internet-of-things. He was the recipient of the ERCIM Alain Bensoussan Fellowship in 2019 and the Best Paper Award at APSIPA ASC-2021, Tokyo, Japan. He is also a member of the Editorial Board for the IEEE SENSORS JOURNAL.

Stefan Werner (Fellow, IEEE) received the M.Sc. degree in electrical engineering from the Royal Institute of Technology, Stockholm, Sweden, in 1998, and the D.Sc. degree (Hons.) in electrical engineering from the Signal Processing Laboratory, Helsinki University of Technology, Espoo, Finland, in 2002. He is currently a Professor with the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, the Director of IoT@NTNU, and Adjunct Professor with Aalto University, Espoo, Finland. He was a Visiting Melchor Professor with the University of Notre Dame, Notre Dame, IN, USA, during the summer of 2019 and an Adjunct Senior Research Fellow with the Institute for Telecommunications Research, University of South Australia, Adelaide, SA, Australia, from 2014 to 2020. He held an Academy Research Fellowship, funded by the Academy of Finland, from 2009 to 2014. His research interests include adaptive and statistical signal processing, wireless communications, and security and privacy in cyber-physical systems. He is a member of the Editorial Boards for the *EURASIP Journal of Signal Processing* and IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS.

Yih-Fang Huang (Life Fellow, IEEE) received the B.S.E.E. degree from National Taiwan University, Taipei, Taiwan, in 1976, the M.S.E.E. degree from the University of Notre Dame, Notre Dame, IN, USA, in 1980, and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, USA, in 1981 and 1982, repectively. He is currently a Professor of electrical engineering and Special Advisor to the Dean of the College of Engineering. He was the Chair of Notre Dame's Electrical Engineering Department from 1998 to 2006, and was the Senior Associate Dean for Education and Undergraduate Programs for the College of Engineering from 2013 to 2023. His research interests include statistical and adaptive signal processing and employs principles in mathematical statistics to solve signal detection and estimation problems that arise in various applications, including wireless communications, distributed sensor networks, smart electric power grid. Dr. Huang was the recipient of the Golden Jubilee Medal of the IEEE Circuits and Systems Society in 1999. He also was the Vice President during 1997–1998 and was a Distinguished Lecturer for the same society during 2000–2001. He received the Toshiba Fellowship and was Toshiba Visiting Professor at Waseda University, Tokyo, Japan. From April to July 2007, he was a Visiting Professor with the Munich University of Technology, Munich, Germany. In 2007, Dr. Huang was awarded the Fulbright-Nokia scholarship for lectures/research at Helsinki University of Technology in Finland. He was the lead Guest Editor for a Special Issue on Signal Processing in Smart Electric Power Grid of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, December 2014. At the University of Notre Dame, he received Presidential Award in 2003, the Electrical Engineering Department's Outstanding Teacher Award in 1994 and in 2011, the Rev. Edmund P. Joyce, CSC Award for Excellence in Undergraduate Teaching in 2011, and the Engineering College's Outstanding Teacher of the Year Award in 2013. In Spring 1993, he was appointed Honorary Professor in the College of Electrical Engineering and Computer Science at National Chiao-Tung University, Hsinchu, Taiwan, in 2014. Dr. Huang is a Fellow of the AAAS.

Anthony Kuh (Life Fellow, IEEE) received the B.S. degree in electrical engineering and computer Science with the University of California, Berkeley in 1979, the M.S. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1980, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 1987. He was with AT&T Bell Laboratories and has been on the Faculty with the Department of Electrical and Computer Engineering, University of Hawai'i, Honolulu, HI, USA, since 1986. He is currently a Professor and previously was Department Chair. His research interests include neural networks and machine learning, adaptive signal processing, sensor networks, and renewable energy and smart grid applications. He was the recipient of the National Science Foundation (NSF) Presidential Young Investigator Award. He is currently serving as a Program Director for NSF with the Electrical, Communications, and Cyber Systems (ECCS) Division working in the Energy, Power, Control, and Network (EPCN) Group. He served the IEEE Signal Processing Society as a member of the Board of Governors as a Regional Director-at-Large Regions1-6, as the Senior Editor for IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, and as a member of the Awards Board. He was also the President of the Asia Pacific Signal and Information Processing Association (APSIPA).