# Decentralized Graph Federated Multitask Learning for Streaming Data

Vinay Chakravarthi Gogineni*, Stefan Werner*, Yih-Fang Huang†, Anthony Kuh‡
*Dept. of Electronic Systems, Norwegian University of Science and Technology-NTNU, Norway
†Dept. of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA
‡Dept. of Electrical and Computer Engineering, University of Hawaii, Hawaii, USA
E-mails: {vinay.gogineni, stefan.werner}@ntnu.no, huang@nd.edu, kuh@hawaii.edu

***Abstract*—In federated learning (FL), multiple clients connected to a single server train a global model based on locally stored data without revealing their data to the server or other clients. Nonetheless, the current FL architecture is highly vulnerable to communication failures and computational bottlenecks at the server. In response, a recent work proposed a multi-server federated architecture, namely, a graph federated learning architecture (GFL). However, existing work assumes a fixed amount of data at clients and the training of a single global model. This paper proposes a decentralized online multitask learning algorithm based on GFL (O-GFML). Clients update their local models using continuous streaming data while clients and multiple servers can train different but related models simultaneously. Furthermore, to enhance the communication efficiency of O-GFML, we develop a partial-sharing-based O-GFML (PSO-GFML). The PSO-GFML allows participating clients to exchange only a portion of model parameters with their respective servers during a global iteration, while non-participating clients update their local models if they have access to new data. In the context of kernel regression, we show the mean convergence of the PSO-GFML. Experimental results show that PSO-GFML can achieve competitive performance with a considerably lower communication overhead than O-GFML.**

***Index Terms*—Graph federated architecture, multitask learning, kernel regression, random Fourier features**

## I. Introduction

Federated learning (FL) [1]–[3] is an increasingly popular distributed learning framework that pushes computation to the edge devices and learns globally shared models from locally stored data. Two particular features, namely, statistical and system heterogeneity, distinguish FL from typical distributed learning. Relevant to statistical heterogeneity, FL aims to learn a global model from non-IID and imbalanced client data [4], [5]. As for system heterogeneity, many clients participating in FL will have different storage, computational, and communication capacities [6]–[8].

In every global iteration of FL, the participating clients have to communicate the model back and forth with the server for client-side local updates and server-side aggregation. The classical single-server-based FL architecture is vulnerable to communication and computation bottlenecks at the server-side. The client-edge-cloud hierarchical FL algorithm addresses this issue by performing partial model aggregation at multiple edge servers that communicate with a cloud server for final aggregation [9]. Recently, a more realistic FL framework, namely, graph federated learning (GFL) [10], has been proposed. GFL comprises several interconnected servers, each associated with its own set of clients. In such a system, a graph can represent the connections between servers.

Many FL approaches, including GFL, assume a fixed amount of training data on each client, which is impractical for many real-life scenarios, e.g., in sensor networks, the internet of things, and wireless communications. In reality, clients may receive new data or a data stream during the training [11]–[13]. On the other hand, since the nodes have access to the data with a distinct distribution, client-specific models or cluster-specific models (a model for a group of clients) can be learned from client data, and certain relations may exist among these models. However, the existing federated multitask learning (FMTL) [14], [15] works are derived in the context of classical single-server-based architectures. Finally, the communication overhead associated with GFL framework in the context of streaming data has not been explored in the literature.

To address the above challenges, this paper proposes an online graph federated multitask learning (O-GFML) algorithm in the context of kernel regression. In every global iteration of the proposed O-GFML, all servers share a copy of the global model with a random subset of their clients. The selected clients use the global model and local data to perform the nonlinear regression task in situ using random Fourier features (RFF) based kernel least mean squares (KLMS) [16], [17] algorithm and send the updated local models back to their respective servers. After aggregating the received models, the servers share them with neighboring servers. The servers then run a clustered multitask diffusion-type algorithm [18], [19] to perform multitask learning in a collaborative fashion. To improve the communication efficiency of O-GFML, we adopt partial-sharing-based communication [20], [21] and derive partial-sharing-based online graph federated multitask learning (PSO-GFML). Furthermore, we establish mean convergence of PSO-GFML under certain conditions. Finally, we demonstrate the performance of the O-GFML and the PSO-GFML through numerical experiments on synthetic non-IID data. Our results

This work was supported by the Research Council of Norway.

show that the PSO-GFML achieves competitive performance at very low communication overhead compared to O-GFML.

## II. Problem Formulation and Algorithm

In this section, we present graph federated architecture-based decentralized online multitask learning in the context of kernel regression. In particular, we consider a scenario where $P$ servers are grouped into $Q$ clusters connected to each other. Further, each server $p$, for $p = 1, 2, \ldots, P$, is connected to its own set of $K$ geographically distributed clients, as illustrated in Fig. 1.

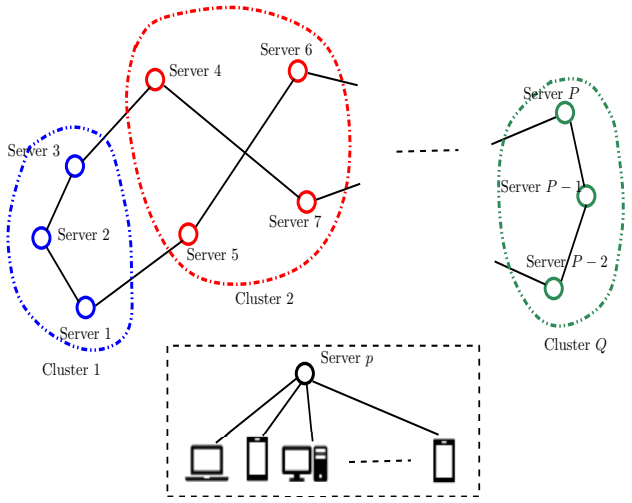


Fig. 1. Graph federated multitask learning architecture.

At every time instance $n$, every client $k$, connected to the server $p$, has access to a continuous streaming signal $x_{p,k,n}$ and associated desired outputs $y_{p,k,n}$ that are assumed to be related by the following model:

$$y_{p,k,n} = f_p(\mathbf{x}_{p,k,n}) + \nu_{p,k,n}, \tag{1}$$

where $f_p(\cdot)$ is a continuous nonlinear model to be estimated collaboratively at server $p$ using locally stored client data, $\mathbf{x}_{p,k,n} = [x_{p,k,n}, x_{p,k,n-1}, \cdots, x_{p,k,n-L+1}]^{\mathrm{T}}$ and $\nu_{p,k,n}$ are the local data vector of size $L \times 1$ and the observation noise, receptively. The servers that are grouped in the same cluster $\mathcal{C}_q$, $q = 1, 2, \ldots, Q$, estimate the same nonlinear model $f_p(\cdot)$, implying $f_p(\cdot) = f_{\mathcal{C}_q}(\cdot)$ for $p \in \mathcal{C}_q$. Furthermore, the neighboring clusters carry out different but related estimation tasks, implying $f_{\mathcal{C}_q} \sim f_{\mathcal{C}_{q'}}$ if clusters $\mathcal{C}_q$ and $\mathcal{C}_{q'}$ are connected. In the following, we use the notation $\mathcal{C}(p)$ to denote the cluster to which the server $p$ belongs, i.e., $\mathcal{C}(p) \in \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_Q\}$.

The cluster-specific nonlinear models can be estimated by solving the following optimization problem:

$$\min_{\mathbf{w}_{\mathcal{C}_1}, \ldots, \mathbf{w}_{\mathcal{C}_Q}} \left( \begin{array}{l} \sum\limits_{p=1}^{P} \frac{1}{K} \sum\limits_{k=1}^{K} \mathcal{J}_{p,k}(\mathbf{w}_{\mathcal{C}(p)}) \\ +\eta \sum\limits_{p=1}^{P} \sum\limits_{r \in \mathcal{N}_p \setminus \mathcal{C}(p)} \rho_{pr} \|\mathbf{w}_{\mathcal{C}(p)} - \mathbf{w}_{\mathcal{C}(r)}\|_2^2 \end{array} \right), \tag{2}$$

where $\mathcal{J}_{p,k}(\mathbf{w}_{\mathcal{C}(p)})$ is the local objective function of client $k$ that is connected to the server $p$, and $\eta > 0$ is the regularization strength parameter. The symbol $\mathcal{N}_p$ denotes the neighborhood of the server $p$ including itself and the symbol $\setminus$ is the set difference operator. The coefficients $\rho_{pr}$ adjust the regularizer strength between servers $p$ and $r$. These coefficients are non-negative and the matrix $\boldsymbol{\rho}$ with $[\boldsymbol{\rho}]_{p,r} = \rho_{pr}$ is an asymmetric right-stochastic matrix. The regularizer promotes the similarities between the models that are in different clusters. The local objective function at client $k$ that is connected to server $p$ is defined as

$$\mathcal{J}_{p,k}(\mathbf{w}_{\mathcal{C}(p)}) = \mathbb{E}\left[|y_{p,k,n} - \hat{y}_{p,k,n}|^2\right], \tag{3}$$

with $\hat{y}_{p,k,n} \approx \mathbf{w}_{\mathcal{C}(p)}^{\mathrm{T}} \mathbf{z}_{p,k,n}$, where $\mathbf{w}_{\mathcal{C}(p)} \in \mathbb{R}^D$ is the linear representation of the function $f_p(\cdot)$ in the random Fourier features (RFF) space [16] and $\mathbf{z}_{p,k,n}$ is the mapping of $\mathbf{x}_{p,k,n}$ into the RFF space $\mathbb{R}^D$.

To solve problem (2), each server, with its clients, runs the online federated averaging (Online-Fed) algorithm [12], [13]. Thereafter, the servers, amongst themselves, run a clustered multitask diffusion type algorithm [18], [19] to learn multiple models in a collaborative fashion. More formally, in each global iteration $n$, server $p$ selects a subset of clients and shares the global model $\mathbf{w}_{p,n}$ with them. Thereafter, the selected clients $\forall k \in \mathcal{S}_{p,n}$ ($\mathcal{S}_{p,n}$ is a set of selected client indices in global iteration $n$) run a stochastic gradient descent to solve the local optimization problem $\mathcal{J}_{p,k}(\mathbf{w}_{\mathcal{C}(p)})$ as follows:

$$\mathbf{w}_{p,k,n+1} = \mathbf{w}_{p,n} + \mu\, \mathbf{z}_{p,k,n}\, \epsilon_{p,k,n}, \tag{4}$$

where $\mu$ is the learning rate and $\epsilon_{p,k,n} = y_{p,k,n} - \mathbf{w}_{p,n}^{\mathrm{T}} \mathbf{z}_{p,k,n}$. These clients communicate the updated local models to their respective server $p$. Then, server $p$ aggregates the received updated models as

$$\boldsymbol{\psi}'_{p,n+1} = \frac{1}{|\mathcal{S}_{p,n}|} \sum_{k \in \mathcal{S}_{p,n}} \mathbf{w}_{p,k,n+1}, \tag{5}$$

where $|\mathcal{S}_{p,n}|$ denotes the cardinality of $\mathcal{S}_{p,n}$. Then, the servers diffuse their estimates to their neighbors and perform inter-cluster cooperation, and intra-cluster cooperation as given below.

**Inter-cluster Cooperation**:

$$\boldsymbol{\psi}_{p,n+1} = \boldsymbol{\psi}'_{p,n+1} + \eta \sum_{r \in \mathcal{N}_p \setminus \mathcal{C}(p)} \rho_{pr} \left(\boldsymbol{\psi}'_{r,n+1} - \boldsymbol{\psi}'_{p,n+1}\right) \tag{6a}$$

**Intra-cluster Cooperation**:

$$\mathbf{w}_{p,n+1} = \sum_{r \in \mathcal{N}_p \cap \mathcal{C}(p)} c_{rp} \boldsymbol{\psi}_{r,n+1}, \tag{6b}$$

where the combiner coefficients $c_{rp}$ are non-negative and the matrix $\mathbf{C}$ with $[\mathbf{C}]_{r,p} = c_{rp}$ is a left-stochastic matrix that defines the combining weights of intra-cluster servers. The steps (4)-(8) together are referred to here as the online graph federated multitask learning (O-GFML) algorithm.

### A. Communication-efficient Online Graph Federated Multi-task Learning Algorithm

Since clients often operate on limited power, it is essential to reduce the amount of communication overhead between clients and servers. In this paper, we propose to employ the principles of partial-sharing [20], [21] to derive a communication-efficient version of O-GFML, namely, partial-sharing-based O-GFML (PSO-GFML). A diagonal selection matrix $\mathbf{S}_{p,k,n}$, containing $M$ ones and $D-M$ zeros on the principal diagonal, specifies which of the model parameters to be exchanged. In designing the selection matrices, we consider coordinated and uncoordinated partial-sharing schemes, which are particular cases of the sequential and stochastic partial-sharing-based communication families.

In coordinated partial-sharing-based communication, every server assigns the same initial selection matrices to its respective clients, i.e., $\mathbf{S}_{p,1,0} = \mathbf{S}_{p,2,0} = \cdots = \mathbf{S}_{p,K,0} = \mathbf{S}_0$, $\forall p = 1, 2, \ldots, P$. All clients thus exchange the same portion of local model parameters with their respective servers. In contrast, the servers assign random initial selection matrices to clients in uncoordinated partial-sharing-based communication, i.e., $\mathbf{S}_{p,1,0} \neq \mathbf{S}_{p,2,0} \neq \cdots \neq \mathbf{S}_{p,K,0}$. For the current global iteration $n$, the entry selection matrix $\mathbf{S}_{p,k,n}$ can be obtained by simply shifting $\mathbf{S}_{p,k,n-1}$ right by $\tau$, where $\tau$ is the circular shift variable. As each entry is exchanged $M$ times in $D$ iterations, the chance of updating a model parameter with the server is $\frac{M}{D}$. By using selection matrices, the steps (4) and (5) in O-GFML can be alternatively expressed as

$$\mathbf{w}_{p,k,n+1} = \mathbf{S}_{p,k,n}\mathbf{w}_{p,n} + (\mathbf{I}_D - \mathbf{S}_{p,k,n})\mathbf{w}_{p,n} + \mu\, \mathbf{z}_{p,k,n}\, \epsilon_{p,k,n}, \tag{7a}$$

with $\epsilon_{p,k,n} = y_{p,k,n} - (\mathbf{S}_{p,k,n}\mathbf{w}_{p,n} + (\mathbf{I}_D - \mathbf{S}_{p,k,n})\mathbf{w}_{p,n})^{\mathrm{T}}\mathbf{z}_{p,k,n}$ and

$$\boldsymbol{\psi}'_{p,n+1} = \frac{1}{|\mathcal{S}_{p,n}|}\sum_{k\in\mathcal{S}_{p,n}} \mathbf{S}_{p,k,n+1}\mathbf{w}_{p,k,n+1} + (\mathbf{I}_D - \mathbf{S}_{p,k,n+1})\mathbf{w}_{p,k,n+1}. \tag{7b}$$

Due to partial-sharing-based communication, however, the server $p$ does not have access to the entire model parameter vector of its participating clients during the aggregation step. Similarly, the participating clients do not have access to the entire model parameter vectors of their respective server. Consequently, they will substitute their previous model parameters for the unknown portions, i.e., participating clients use $(\mathbf{I}_D - \mathbf{S}_{p,k,n})\mathbf{w}_{p,k,n}$ in place of $(\mathbf{I}_D - \mathbf{S}_{p,k,n})\mathbf{w}_{p,n}$ and their corresponding server uses $(\mathbf{I}_D - \mathbf{S}_{p,k,n+1})\mathbf{w}_{p,n}$ in place of $(\mathbf{I}_D - \mathbf{S}_{p,k,n+1})\mathbf{w}_{p,k,n+1}$. Furthermore, the partial-sharing-based O-GFML (PSO-GFML) still permits non-participating clients to perform local updates as long as they have access to the new data. The proposed PSO-GFML is summarized in **Algorithm** 1.

**Algorithm 1:** **`PSO-GFML`**. $M$ clusters, $P$ servers, $K$ clients, learning rate $\mu$, set of all clients $\mathcal{S}$, and circular shift variable $\tau$.

**Initialization**: Server models $\mathbf{w}_{p,0}$, local model $\mathbf{w}_{p,k,0}$, RFF space dimension $D$ and selection matrices $\mathbf{S}_{p,k,0}$, $\forall k \in \mathcal{S}$ and $p = 1, \ldots, P$

**For** $n = 1$ to $N$

**For** $p = 1$ to $P$

Every server $p$ randomly selects a subset $\mathcal{S}_{p,n}$ of $K$ clients and communicate $\mathbf{S}_{p,k,n}\mathbf{w}_{p,n}$ to them,

**Client Local Update**:

**If** $k \in \mathcal{S}_n$

$$\mathbf{w}'_{p,k,n+1} = \mathbf{S}_{p,k,,n}\mathbf{w}_{p,n} + (\mathbf{I}_D - \mathbf{S}_{p,k,n})\mathbf{w}_{p,k,n}$$
$$\epsilon_{p,k,n} = y_{p,k,n} - (\mathbf{w}'_{p,k,n+1})^{\mathrm{T}}\mathbf{z}_{p,k,n},$$
$$\mathbf{w}_{p,k,n+1} = \mathbf{w}'_{p,k,n+1} + \mu\, \mathbf{z}_{p,k,n}\, \epsilon_{p,k,n},$$

**Else**

$$\epsilon_{p,k,n} = y_{p,k,n} - \mathbf{w}^{\mathrm{T}}_{p,k,n}\mathbf{z}_{p,k,n},$$
$$\mathbf{w}_{p,k,n+1} = \mathbf{w}_{p,k,n} + \mu\, \mathbf{z}_{p,k,n}\, \epsilon_{p,k,n},$$

**EndIf**

The clients $\forall k \in \mathcal{S}_{p,n}$ communicate $\mathbf{S}_{p,k,n+1}\mathbf{w}_{p,k,n+1}$ to the server, where $\mathbf{S}_{p,k,n+1} = \text{circshift}(\mathbf{S}_{p,k,n}, \tau)$,

**Aggregation at the Server**:

The server updates the global model as

$$\boldsymbol{\psi}'_{p,n+1} = \frac{1}{|\mathcal{S}_{p,n}|}\sum_{k\in\mathcal{S}_{p,n}} \mathbf{S}_{p,k,n+1}\mathbf{w}_{p,k,n+1} + (\mathbf{I}_D - \mathbf{S}_{p,k,n+1})\mathbf{w}_{p,n}.$$

**EndFor**

**Inter-cluster Cooperation**:

$$\boldsymbol{\psi}_{p,n+1} = \boldsymbol{\psi}'_{p,n+1} + \eta \sum_{r\in\mathcal{N}_p\setminus\mathcal{C}(p)} \rho_{pr}\left(\boldsymbol{\psi}'_{r,n+1} - \boldsymbol{\psi}'_{p,n+1}\right)$$

**Intra-cluster Cooperation**:

$$\mathbf{w}_{p,n+1} = \sum_{r\in\mathcal{N}_p\cap\mathcal{C}(p)} c_{rp}\boldsymbol{\psi}_{r,n+1},$$

**EndFor**

## III. Convergence Analysis

In this section, we examine the mean convergence of PSO-GFML. Before proceeding to the analysis, we define the global optimal extended model parameter vector $\mathbf{w}_e^\star = \text{col}\{\mathbf{1}_{K+1} \otimes \mathbf{w}^\star_{\mathcal{C}(1)}, \ldots, \mathbf{1}_{K+1} \otimes \mathbf{w}^\star_{\mathcal{C}(P)}\}$, estimated global extended model parameter vector $\mathbf{w}_{e,n} = \text{col}\{\mathbf{w}_{1,n}, \mathbf{w}_{1,1,n}, \ldots, \mathbf{w}_{1,K,n}, \ldots, \mathbf{w}_{P,n}, \mathbf{w}_{P,1,n}, \ldots, \mathbf{w}_{P,K,n}\}$, extended input data matrix $\mathbf{Z}_{e,n} = \text{blockdiag}\{\mathbf{0}, \mathbf{z}_{1,1,n}, \ldots, \mathbf{z}_{1,K,n}, \ldots, \mathbf{0}, \mathbf{z}_{P,1,n}, \ldots, \mathbf{z}_{P,K,n}\}$ and extended observation noise vector $\boldsymbol{\nu}_{e,n} = \text{col}\{\mathbf{0}, \nu_{1,1,n}, \ldots, \nu_{1,K,n}, \ldots, \mathbf{0}, \nu_{P,1,n}, \ldots, \nu_{P,K,n}\}$, where $\text{col}\{\cdot\}$ and

blockdiag$\{\cdot\}$ represent column-wise stacking and block diagonalization operators, respectively. The symbol $\mathbf{1}_{K+1}$ is a $(K+1)\times 1$ column vector with each element taking the value one. From the above definitions, we can write

$$\begin{aligned}\mathbf{y}_{e,n} &= \text{col}\{0, y_{1,1,n}, \ldots, y_{1,K,n}, \ldots, 0, y_{P,1,n}, \ldots, y_{P,K,n}\} \\ &= \mathbf{Z}_{e,n}^{\mathrm{T}}\mathbf{w}_e^{\star} + \boldsymbol{\nu}_{e,n},\end{aligned} \tag{9}$$

and

$$\begin{aligned}\boldsymbol{\epsilon}_{e,n} &= \text{col}\big\{0, \epsilon_{1,1,n}, \ldots, \epsilon_{1,K,n}, \ldots, 0, \epsilon_{P,1,n}, \ldots, \epsilon_{P,K,n}\big\} \\ &= \mathbf{y}_{e,n} - \mathbf{Z}_{e,n}^{\mathrm{T}}\boldsymbol{\mathcal{A}}_{\mathbf{S},n}\mathbf{w}_{e,n},\end{aligned} \tag{10}$$

with

$$\boldsymbol{\mathcal{A}}_{\mathbf{S},n} = \text{blockdiag}\{\boldsymbol{\mathcal{A}}_{\mathbf{S},1,n}, \boldsymbol{\mathcal{A}}_{\mathbf{S},2,n}, \ldots, \boldsymbol{\mathcal{A}}_{\mathbf{S},P,n}\}, \tag{11}$$

where $\boldsymbol{\mathcal{A}}_{\mathbf{S},p,n}$, for $p = 1, 2, \ldots, P$ is given by

$$\boldsymbol{\mathcal{A}}_{\mathbf{S},p,n} = \begin{bmatrix} \mathbf{I}_D & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ a_{p,1,n}\mathbf{S}_{p,1,n} & \begin{pmatrix}\mathbf{I}_D - \\ a_{p,1,n}\mathbf{S}_{p,1,n}\end{pmatrix} & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{p,K,n}\mathbf{S}_{p,K,n} & \mathbf{0} & \mathbf{0} & \ldots & \begin{pmatrix}\mathbf{I}_D - \\ a_{p,K,n}\mathbf{S}_{p,K,n}\end{pmatrix} \end{bmatrix}. \tag{12}$$

In the above, $a_{p,k,n} = 1$ if the client $k \in \mathcal{S}_n$, and zero otherwise. Using these definitions, the global recursion of PSO-GFML can be stated as

$$\mathbf{w}_{e,n+1} = \boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_{\mathbf{S},n+1}\big(\boldsymbol{\mathcal{A}}_{\mathbf{S},n}\mathbf{w}_{e,n} + \mu\, \mathbf{Z}_{e,n}\, \boldsymbol{\epsilon}_{e,n}\big), \tag{13}$$

with

$$\begin{aligned}\boldsymbol{\mathcal{C}} &= \mathbf{C}^{\mathrm{T}} \otimes \mathbf{I}_{D(K+1)}, \\ \boldsymbol{\mathcal{P}} &= \mathbf{I}_{PD(K+1)} - (\boldsymbol{\rho} \otimes \mathbf{I}_{D(K+1)}),\end{aligned} \tag{14}$$

and

$$\boldsymbol{\mathcal{B}}_{\mathbf{S},n+1} = \text{blockdiag}\{\boldsymbol{\mathcal{B}}_{\mathbf{S},1,n+1}, \boldsymbol{\mathcal{B}}_{\mathbf{S},2,n+1}, \ldots, \boldsymbol{\mathcal{B}}_{\mathbf{S},P,n+1}\}, \tag{15}$$

where $\boldsymbol{\mathcal{B}}_{\mathbf{S},p,n+1}$, for $p = 1, 2, \ldots, P$ is given by

$$\boldsymbol{\mathcal{B}}_{\mathbf{S},p,n+1} = \begin{bmatrix} \begin{pmatrix}\mathbf{I}_D - \\ \sum\limits_{k\in\mathcal{S}_n}\frac{a_{p,k,n}}{|\mathcal{S}_{p,n}|}\mathbf{S}_{p,k,n+1}\end{pmatrix} & \frac{a_{p,1,n}}{|\mathcal{S}_{p,n}|}\mathbf{S}_{p,1,n+1} & \ldots & \frac{a_{p,K,n}}{|\mathcal{S}_{p,n}|}\mathbf{S}_{p,K,n+1} \\ \mathbf{0} & \mathbf{I}_D & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{I}_D \end{bmatrix}. \tag{16}$$

We make the following assumptions to establish the convergence condition for PSO-Fed:

**A1**: At each client $k$ that connected to a server $p$, the input signal vector $\mathbf{z}_{p,k,n}$ is drawn from a wide-sense stationary multivariate random sequence with correlation matrix $\mathbf{R}_{p,k} = \mathrm{E}[\mathbf{z}_{p,k,n}\mathbf{z}_{p,k,n}^{\mathrm{T}}]$.

**A2**: The noise process $\nu_{p,k,n}$ is assumed to be zero-mean i.i.d. and independent of all input and output data,

**A3**: At each client $k$ that connected to a server $p$, the model parameter vector is taken to be independent of input signal vector.

**A4**: The selection matrices $\mathbf{S}_{p,k,n}$ are assumed to be independent of any other data; in addition, $\mathbf{S}_{p,k,n}$ and $\mathbf{S}_{r,l,m}$ are independent, for all $p \neq r$, $k \neq l$ and $m \neq n$.

Denoting $\widetilde{\mathbf{w}}_{e,n} = \mathbf{w}_e^{\star} - \mathbf{w}_{e,n}$, and utilizing the fact that $\mathbf{w}_e^{\star} = \boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_{\mathbf{S},n+1}\boldsymbol{\mathcal{A}}_{\mathbf{S},n}\mathbf{w}_e^{\star}$ (since $\boldsymbol{\mathcal{C}}$ is doubly-stochastic and $\boldsymbol{\mathcal{P}}$, $\boldsymbol{\mathcal{B}}_{\mathbf{S},n+1}$, and $\boldsymbol{\mathcal{A}}_{\mathbf{S},n}$ are right-stochastic, one can easily prove this result), then from (13), $\widetilde{\mathbf{w}}_{e,n+1}$ can be recursively expressed as

$$\begin{aligned}\widetilde{\mathbf{w}}_{e,n+1} =& \boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_{\mathbf{S},n+1}\big(\mathbf{I} - \mu\mathbf{Z}_{e,n}\mathbf{Z}_{e,n}^{\mathrm{T}}\big)\boldsymbol{\mathcal{A}}_{\mathbf{S},n}\widetilde{\mathbf{w}}_{e,n} \\ &- \mu\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_{\mathbf{S},n+1}\mathbf{Z}_{e,n}\boldsymbol{\nu}_{e,n}.\end{aligned} \tag{17}$$

Applying expectation $\mathbb{E}[\cdot]$ on both sides of (17) and using assumptions **A1**-**A4**, we obtain

$$\mathbb{E}[\widetilde{\mathbf{w}}_{e,n+1}] = \boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{P}}\mathbb{E}[\boldsymbol{\mathcal{B}}_{\mathbf{S},n+1}]\big(\mathbf{I} - \mu\boldsymbol{\mathcal{R}}_e\big)\mathbb{E}[\boldsymbol{\mathcal{A}}_{\mathbf{S},n}]\mathbb{E}[\widetilde{\mathbf{w}}_{e,n}], \tag{18}$$

where $\boldsymbol{\mathcal{R}}_e = \text{blockdiag}\{\mathbf{0}, \mathbf{R}_{1,1}, \ldots, \mathbf{R}_{1,K}, \ldots, \mathbf{0}, \mathbf{R}_{P,1}, \ldots, \mathbf{R}_{P,K}\}$. Since $\boldsymbol{\mathcal{C}}\mathbf{1}_{PD(K+1)}] = 1$, $\boldsymbol{\mathcal{P}}\mathbf{1}_{PD(K+1)}] = 1$, $\mathbb{E}[\boldsymbol{\mathcal{A}}_{\mathbf{S},n}\mathbf{1}_{PD(K+1)}] = 1$ and $\mathbb{E}[\boldsymbol{\mathcal{B}}_{\mathbf{S},n+1}\mathbf{1}_{PD(K+1)}] = 1$, from (18), one can see that $\mathrm{E}\big[\widetilde{\mathbf{w}}_{e,n}\big]$ converges under $|1 - \mu\lambda_i(\mathbf{R}_{p,k})| < 1$, $\forall p, k, i$, where $\lambda_i(\cdot)$ is the $i$th eigenvalue of its argument matrix. After solving the above convergence condition, we finally have following first-order convergence condition:

$$0 < \mu < \frac{2}{\max\limits_{i,p,k}\{\lambda_i(\mathbf{R}_{p,k})\}}. \tag{19}$$

## IV. Numerical Simulations

In this section, experimental results are presented to examine the performance of the proposed O-GFL and PSO-GFL. For this, we consider a GFL architecture consisting of 10 servers that are grouped into $Q = 3$ clusters: $\mathcal{C}_1 = \{1, 2, 3\}, \mathcal{C}_2 = \{4, 5, 6, 7\}$, and $\mathcal{C}_3 = \{8, 9, 10\}$, and each server connected with $K = 50$ clients of its own. At every client $k$ that is connected to server $p$, synthetic non-IID input signal $x_{p,k,n}$ and corresponding observed output are generated so that they are related via the following model:

$$\begin{aligned}f_{\mathcal{C}_q}(\mathbf{x}_{p,k,n}) =& \sqrt{x_{p,k,n,1}^2 + \gamma_{q,1}\sin^2(\pi\, x_{p,k,n,4})} \\ &+ \big(\gamma_{q,2} - \gamma_{q,3}\exp(-x_{p,k,n,2}^2)\big)x_{p,k,n,3} + \nu_{p,k,n},\end{aligned} \tag{20}$$

with $\gamma_{q,1} \in \{0.75, 0.8, 0.85\}$, $\gamma_{q,2} \in \{0.85, 0.8, 0.75\}$, and $\gamma_{q,3} \in \{0.55, 0.5, 0.45\}$. The input signal at each client $x_{p,k,n}$ was generated by driving a first-order autoregressive (AR) model: $x_{p,k,n} = \theta_{p,k}\, x_{p,k,n-1} + \sqrt{1-\theta_{p,k}^2}\, u_{p,k,n}$, $\theta_{p,k} \in$

$\mathcal{U}(0.2, 0.9)$, where $u_{p,k,n}$ was drawn from a Gaussian distribution $\mathcal{N}(\mu_{p,k}, \sigma^2_{u_{p,k}})$, with $\mu_{p,k} \in \mathcal{U}(-0.2, 0.2)$ and $\sigma^2_{u_{p,k}} \in \mathcal{U}(0.2, 1.2)$, respectively (where $\mathcal{U}(\cdot)$ indicates the uniform distribution). The observation noise $\nu_{p,k,n}$ was taken as zero mean i.i.d. Gaussian with variance $\sigma^2_{\nu_{p,k}} \in \mathcal{U}(0.005, 0.03)$. Using a Cosine feature function, $x_{p,k,n}$ was mapped into the 200-dimensional RFF space. The same learning rate of 0.75 was used for each client in all simulated algorithms. Each server $p$ implemented uniform random selection procedure to select $|\mathcal{S}_{p,n}| = 4$ clients in every global iteration $n$. The coefficients $\rho_{pr}$ and $c_{rp}$ are set similar to [18]. Average mean-square error (MSE) on test data was considered to be performance metric, which is given by

$$\text{Testing MSE} = \frac{1}{P}\sum_{p=1}^{P}\frac{1}{N_{\text{test}}}\|\mathbf{y}_{p,\text{test}} - \mathbf{Z}^{\text{T}}_{p,\text{test}}\mathbf{w}_{p,n}\|_2^2, \quad (21)$$

where $\{\mathbf{Z}_{p,\text{test}}, \mathbf{y}_{p,\text{test}}\}$ is the test dataset corresponding to server $p$ ($N_{\text{test}}$ examples in total) covering all clients data. We simulated O-GFML and PSO-GFML to perform the above mentioned nonlinear regression task. The latter was also simulated for various values of $M$ (number of parameters exchanged between server and and its respective clients). In order to obtain the learning curves (i.e., testing MSE in dB against the global iteration index $n$), we average the results over 500 independent experiments. The resulting plots are displayed in Figs. 2a and 2b for coordinated and uncoordinated partial-sharing schemes, respectively.

From Fig. 2, the following observations can be made:

1) The O-GFML and the PSO-GFML are able estimate multiple models efficiently. When $M$ is small (say, 1), the PSO-GFML performs poorly compared to the O-GFML. However, as $M$ increases to higher values (say, 4 and 40), its performance improves . In summary, PSO-GFML exhibits the same performance as that of the O-GFML when $M \geq 40$.
2) As compared to O-GFML, PSO-GFML has a lower communication cost as $M << D$. When $M = 40$, PSO-GFML behaves the same as O-GFML, but only consumes $\frac{1}{5}$ of its communication load ($D$ = 200).
3) For very small values of $M$ (say, 1 in our experiment), the coordinated partial-sharing scheme shows superior performance (i.e., initial convergence) over the uncoordinated partial-sharing scheme. By allowing the server to aggregate the same entries of the local model parameter vectors, the coordinated scheme preserves the connectedness of clients. Both schemes perform equally well for large values of $M$ (say, $\geq 5$ in our experiment).

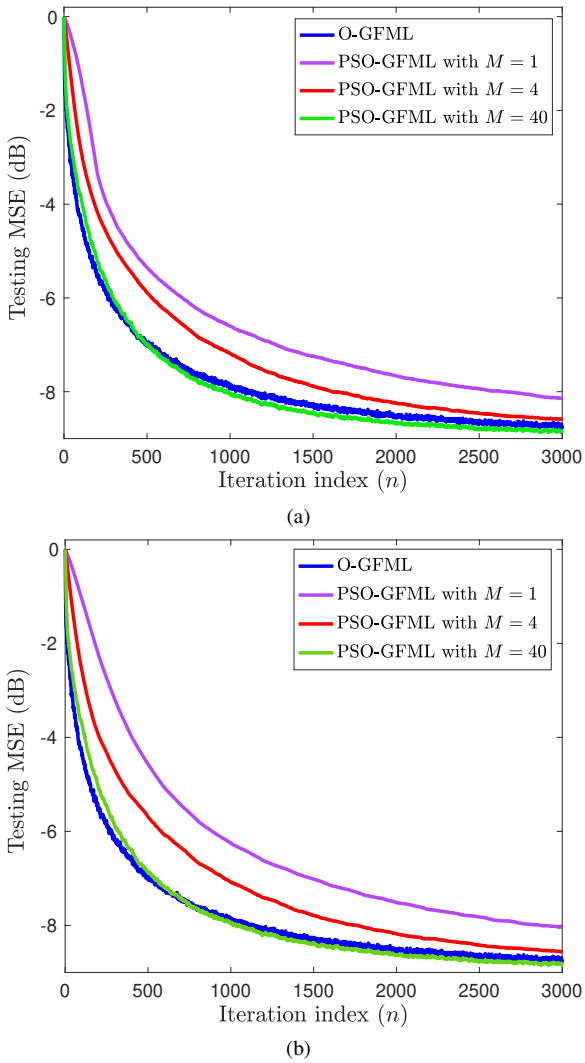


Fig. 2. Performance of O-GFML and PSO-GFML: (a). Coordinated partial-sharing. (b). Uncoordinated partial-sharing.

## V. Conclusions

A decentralized online multitask learning algorithm based on GFL (O-GFML) has been proposed. The O-GFML allows clients to update their models using continuous streaming data and train separate but related multiple models. Furthermore, a partial-sharing-based O-GFML (PSO-GFML) was derived to enhance the communication efficiency of O-GFML. Participating clients exchange only a portion of model parameters with their respective servers during a global iteration of PSO-GFML, while non-participating clients update their local models if they have access to new data. In the context of kernel regression, the performance of the O-GFML and PSO-GFML has been evaluated. the mean convergence of PSO-GFML has been established. The experimental results have shown that both coordinated and uncoordinated PSO-GFML algorithms exhibit estimation performance comparable to O-GFML's with reduced communication costs.

## References


[1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," . *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 2157-6904, Feb. 2019.

[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] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031-2063, 2020.

[4] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Proc. Advances in Neural Info. Process. Syst.*, 2017, pp. 4424–4434.

[5] F. Sattler, S. Wiedemann, K. -R. Müller and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400-3413, Sep. 2020.

[6] Y. Zhou, Q. Ye and J. Lv, "Communication-efficient federated learning with compensated overlap-FedAvg," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 01, pp. 192-205, 2022.

[7] Y. Chen, Y. Ning, M. Slawski and H. Rangwala, "Asynchronous online federated learning for edge devices with non-IID data," in *Proc. IEEE Int. Conf. Big Data*, 2020, pp. 15-24.

[8] 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.

[9] L. Liu, J. Zhang, S. H. Song and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in Proc. IEEE Int. Conf. Commun., 2020, pp. 1-6.

[10] E. Rizk and A. H. Sayed, "A graph federated architecture with privacy preserving learning," in Proc. IEEE Int. Workshop on Signal Process. Advances in Wireless Commun., 2021, pp. 131-135.

[11] W. U. Bajwa, V. Cevher, D. Papailiopoulos and A. Scaglione, "Machine learning from distributed, streaming Data," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 11-13, May 2020.

[12] A. Kuh, "Real time kernel learning for sensor networks using principles of federated learning," in *Proc. IEEE Int. Conf. Asia Pacific Signal and Info. Process. Assoc.*, 2021.

[13] V. C. Gogineni, S. Werner, Y-F. Haung and A. Kuh "Communication-efficient online federated learning framework for nonlinear regression," in arXiv:2110.06556, 2021, [online]. Available: arXiv: 2110.06556.

[14] V. Smith, C. K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Advances in Neural Info. Process. Syst.*, Long Beach, CA, USA, Dec. 2017.

[15] R. Li, F. Ma, W. Jiang and J. Gao, "Online federated multitask learning," *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 215-220.

[16] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2007, pp. 1177-1184.

[17] V. C. Gogineni, V. R. M. Elias, W. A. Martins and S. Werner, "Graph diffusion kernel LMS using random Fourier features," in Proc. Asilomar Conf. on Signals, Syst., and Comput., 2020, pp. 1528-1532.

[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, "Diffusion affine projection algorithm for multitask networks," in *Proc. IEEE Int. Conf. Asia Pacific Signal and Info. Process. Assoc.*, Honolulu, 2018, pp. 201-206.

[20] R. Arablouei, S. Werner, Y. F. Huang and K. Doğançay, "Distributed least mean-square estimation with partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 472–484, Jan. 2013.

[21] R. Arablouei,K. Doğançay, S. Werner and Y. F. Huang, "Adaptive distributed estimation based on recursive least-squares and partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3510–3522, Jul. 2014.