Energy Efficient Collaborative Federated Learning Design: A Graph Neural Network based Approach

Nuocheng Yang*, Sihua Wang*†, Mingzhe Chen‡, Christopher G. Brinton§, and Changchuan Yin*

*Beijing Laboratory of Advanced Information Network, Beijing University of Posts and Telecommunications, Beijing, China, Emails: {YangNuoCheng,sihuawang,ccyin}@bupt.edu.cn.
†State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China.
‡Department of Electrical and Computer Engineering and the Institute for Data Science and Computing, University of Miami, Coral Gables, FL USA, Email: mingzhe.chen@miami.edu.
§ School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, Email: cgb@purdue.edu.

GLOBECOM 2023 - 2023 IEEE Global Communications Conference | 979-8-3503-1090-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/GLOBECOM54140.2023.10437172

***Abstract*—In this paper, we consider the design of an energy efficient collaborative federated learning (CFL) methodology where devices exchange their local FL parameters with a subset of their neighbors without reliance on a parameter server. In the considered model, mobile devices implement the designed CFL to train their local FL models using their own datasets over a realistic wireless network. Due to the limited wireless resources and user movements, each device may not be able to transmit its FL parameters with all neighboring devices. Therefore, each device must select a subset of devices to share its FL parameters and optimize the transmit power. This problem is formulated as an optimization problem, whose goal is to minimize CFL training energy consumption while satisfying the delay and CFL training loss requirements. To solve this problem, a two-stage solution is proposed. At the first stage, a graph neural network (GNN) based algorithm is proposed, which enables each device to individually determine the subset of devices to transmit FL parameters using its neighboring devices' location and connection information. Compared to standard iterative algorithms that need to iteratively optimize device connections and transmit power, the proposed GNN based method can directly obtain the optimal device connections without iterative optimization. Given the optimal device connections, at the second stage, each device can directly obtain the optimal transmit power. Simulation results show that the proposed algorithm can decrease energy consumption by up to 46% compared to the algorithm where each device will directly connect to its first and second nearest neighbors.**

***Keywords*—Collaborative federated learning, energy consumption optimization, graph neural network.**

I. Introduction

Federated learning (FL) [1], [2] enables devices to cooperatively train a machine learning (ML) model without data exchange thus improving data privacy. However, standard FL requires devices to transmit their FL models to a parameter server which may limit its application scenarios due to limited wireless resources. To address this problem, a novel FL framework that combines the principles of collaborative learning with federated learning, called collaborative federated learning (CFL), has been proposed [3]. In particular, CFL enables devices to collaboratively train an ML model via exchanging their local FL model parameters with their neighboring device without the reliance on a parameter server. Compared to standard FL, CFL enables more devices to join FL training. For example, the devices that cannot directly connect to the parameter server due to limited wireless resources can connect to their neighboring devices so as to participate in the CFL framework. However, the implementation of CFL over wireless networks presents several challenges including CFL learning efficiency optimization, device connection optimization, and distributed gradient update method design.

Recently, a number of works such as [3]–[7] have studied the use of distributed methods for CFL performance optimization. The authors in [3] introduced a CFL framework based on the lazy Metropolis method to update devices' local FL models and introduced the challenges of deploying the designed CFL framework over wireless networks. The authors in [4] designed a device connection scheme based on the expander graph to reduce the total number of communication rounds required to reach CFL convergence. The authors in [5] studied jointly optimized network topology and FL model compression to improve CFL convergence speed. The authors in [6] introduced several FL model update approaches such as the Metropolis update and the lazy Metropolis update, to allow devices to share their model parameters with only their neighboring devices but find a global ML model. The authors in [7] developed a distributed gradient descent algorithm to improve the CFL convergence rate. However, most of the CFL algorithms in [3]–[7] designed FL parameter update methods based on a static network topology where device connections will not change over time. In consequence, these works [3]–[7] did not consider how network topology affects CFL training thus resulting in redundant CFL parameter transmission and increasing communication overhead. Meanwhile, these works [3]–[7] did not consider the impacts of network dynamics such as devices' mobility patterns on CFL training, which will limit the applications of these CFL algorithms.

The main contribution of this paper is a novel fully distributed CFL framework that jointly considers the device connection scheme and device transmit power. Our key contributions include:

- We propose a novel CFL framework in which wireless mobile devices collaboratively train an ML model via exchanging their FL parameters with their neighboring devices. In the considered model, due to the limited energy and bandwidth resources, each device must select

This work was supported in part by Beijing Natural Science Foundation under Grant L223027, in part by the National Natural Science Foundation of China under Grants 61629101 and 61671086, and 111 Project under Grant B17007, and in part by the U.S. National Science Foundation under grant CNS-2212565 and CNS-2146171.

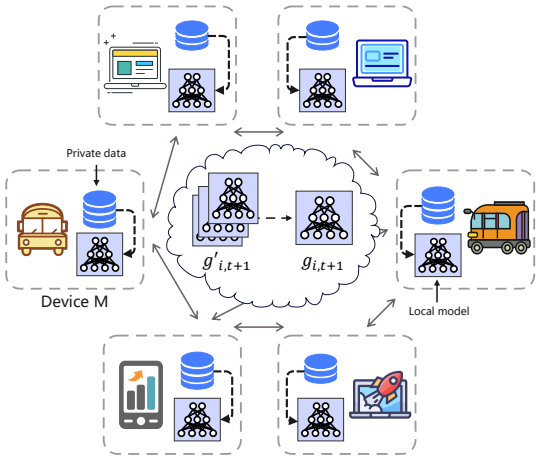

Fig. 1. Illustration of the considered CFL model.

a subset of devices per FL iteration to exchange their FL parameters aiming to minimize CFL training loss. Therefore, it is necessary to select the devices that each device shares its FL parameters and the transmit power of FL parameter transmission. This problem is formulated as an optimization problem aiming to minimize the energy consumption of CFL training while satisfying the delay and FL training loss requirements by jointly determining the device connections and transmit power of FL parameter transmission.

- To solve the formulated problem, we proposed a solution that consists of two stages. At the first stage, a graph neural network (GNN) [8] based algorithm is proposed, which enables each device to individually determine the subset of devices that needs to transmit FL parameters using its neighboring devices' location and connection information. Compared to standard iterative algorithms that need to iteratively optimize device connections and transmit power, the proposed GNN based method can directly obtain the optimal device connections without iterative optimization. Given the optimal device connections, at the second stage, each device can directly obtain the optimal transmit power.

Simulation results show that, compared to a fully distributed algorithm where each device will directly connect to its first and second nearest neighbors, the proposed GNN algorithm can reduce energy consumption and training loss by up to 46% and 2%, respectively.

II. System Model and Problem Formulation

Consider a wireless network that consists of a set $\mathcal{M}$ of M mobile user that collaboratively train a supervised learning model, as shown in Fig. 1. We assume that each device i has N_i training data samples, and each training data sample n consists of an input feature vector $\boldsymbol{x}_{i,n} \in \mathbb{R}^{N_{\mathrm{I}} \times 1}$ and a corresponding label vector $\boldsymbol{y}_{i,n} \in \mathbb{R}^{N_{\mathrm{O}} \times 1}$. The objective of the training is to minimize the global loss function over all data samples, i.e.,

$$F(\boldsymbol{g}) = \min_{\boldsymbol{g}} \frac{1}{N} \sum_{i=1}^{M} \sum_{n=1}^{N_i} f(\boldsymbol{g}_i, \boldsymbol{x}_{i,n}, \boldsymbol{y}_{i,n}), \tag{1}$$

where $\boldsymbol{g} = [\boldsymbol{g}_1, ..., \boldsymbol{g}_i, ..., \boldsymbol{g}_M]$, $\boldsymbol{g}_i \in \mathbb{R}^{V \times 1}$ is a vector of the local FL model parameters of device i with $N = \sum_{i=1}^{M} N_i$ being the total number of training data samples across devices. $f(\boldsymbol{g}_i, \boldsymbol{x}_{i,n}, \boldsymbol{y}_{i,n})$ is a loss function (e.g., squared error) that measures the accuracy of the generated local FL model $\boldsymbol{g}_i$ in building a relationship between the input vector $\boldsymbol{x}_{i,n}$ and the output vector $\boldsymbol{y}_{i,n}$.

Different from existing works that require all devices to transmit FL parameters to a central controller for model aggregation, here, we propose a novel FL that does not rely on a central controller [3]. In particular, the designed FL training process is explained as follows:

1) Each device randomly initializes its local model.
2) Each device updates its local model using its collected data samples.
3) Each device exchanges the updated local model with a subset of devices that are selected by itself.
4) Each device aggregates the FL models received from other devices to generate its FL model.

Steps 2)-4) are repeated until the optimal vector $\boldsymbol{g}$ is found. Next, we will explain the training process of designed FL mathematically.

A. CFL Training Process

A backward propagation (BP) algorithm based on stochastic gradient descent (SGD) is used to update the local FL parameters of each device, which is given by

$$\boldsymbol{g}'_{i,t+1} = \boldsymbol{g}_{i,t} - \lambda \sum_{n \in \mathcal{N}_{i,t}} \frac{\partial f(\boldsymbol{g}_{i,t}, \boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{g}_{i,t}}. \tag{2}$$

where $\boldsymbol{g}'_{i,t+1}$ is the updated local model at iteration t, λ is the learning rate, and $\mathcal{N}_{i,t}$ is the subset of training data samples (i.e., minibatch) selected from the training dataset $\mathcal{N}_i$ of device i at iteration t.

Given the updated local model, each device i exchanges $\boldsymbol{g}'_{i,t+1}$ with its selected devices and aggregate their FL model parameters, which can be given by [6]

$$\boldsymbol{g}_{i,t+1}(\boldsymbol{u}_{i,t}) = \boldsymbol{g}'_{i,t+1} + \sum_{j \in \mathcal{M}} \frac{u_{j,i,t}}{\max\{||\boldsymbol{u}_{i,t}||, ||\boldsymbol{u}_{j,t}||\}} (\boldsymbol{g}'_{j,t+1} - \boldsymbol{g}'_{i,t+1}), \tag{3}$$

where $\boldsymbol{u}_{i,t} = [u_{i,1,t}, ..., u_{i,j,t}, ..., u_{i,M,t}]$ is a vector of FL transmission index of device i with $u_{i,j,t} = 1$ implying that device i will exchange its local FL model with device j at iteration t, and $u_{i,j,t} = 0$ otherwise. $||\boldsymbol{u}_{i,t}|| = \sum_{j \in \mathcal{M}} u_{i,j,t}$ is the number of devices that will transmit FL parameters to device i.

B. Mobility Model

The mobility of each device is modeled by a random walk model. In particular, at each CFL iteration, each device can stay at the current location or move to four directions: a) up, b) down, c) left, and d) right. The probability of each device's movement at each time slot t is expressed by a vector $\boldsymbol{\xi}_{i,t} = [\xi_{i,t,0}, \xi_{i,t,1}, \xi_{i,t,2}, \xi_{i,t,3}, \xi_{i,t,4}]$. where $\xi_{i,t,0}$ is the probability that device i stays at the current location at time t while $\xi_{i,t,1}$, $\xi_{i,t,2}$, $\xi_{i,t,3}$, and $\xi_{i,t,4}$, represent, respectively, the probability that device i moves up, down, left, and right at time t. The speed of device i is v_i. The location of each device i at the time t is captured by a vector $\boldsymbol{\phi}_{i,t} = [\phi_{i,t,1}, \phi_{i,t,2}]$. We assume that the duration of each time slot t is Δt. The location of each device i at the time $t+1$ is:

$$\boldsymbol{\phi}_{i,t+1}=\begin{cases}[\phi_{i,t,1}, \phi_{i,t,2}] \text{ ,with probability } \xi_{i,t,0},\\ [\phi_{i,t,1}, \phi_{i,t,2} + v_i\Delta t] \text{ ,with probability } \xi_{i,t,1},\\ [\phi_{i,t,1}, \phi_{i,t,2} - v_i\Delta t] \text{ ,with probability } \xi_{i,t,2},\\ [\phi_{i,t,1} + v_i\Delta t, \phi_{i,t,2}] \text{ ,with probability } \xi_{i,t,3},\\ [\phi_{i,t,1} - v_i\Delta t, \phi_{i,t,2}] \text{ ,with probability } \xi_{i,t,4}.\end{cases} \tag{4}$$

C. CFL Model Transmission Process

We adopt an orthogonal frequency division multiple access (OFDMA) transmission scheme. Let B be the bandwidth that device can use for model parameter transmission and $p_{i,j,t}$ be the transmit power of device i. The data rate of device i transmitting FL parameters to device j is

$$c_{i,j,t}(\boldsymbol{u}_{i,t}, \boldsymbol{\phi}_t, B, p_{i,j,t}) = \frac{B}{||\boldsymbol{u}_{i,t}||}\log\left(1+\frac{p_{i,j,t}h_{i,j,t}\left(\boldsymbol{\phi}_t\right)}{\sigma_N^2}\right), \tag{5}$$

where $h_{i,j,t} = \rho_{i,j,t}r_{i,j,t}^{-2}$ is the channel gain between device i and j with $\rho_{i,j,t}$ is the Rayleigh fading parameters, $r_{i,j,t}$ is the distance between device i and j. σ_N^2 represents the variance of additive white Gaussian noise. Based on (5), the time of device i transmitting FL model parameters to device j is

$$l_{i,j,t}(\boldsymbol{u}_{i,t}, \boldsymbol{\phi}_t, B, p_{i,j,t}) = \frac{A}{c_{i,j,t}(\boldsymbol{u}_{i,t}, \boldsymbol{\phi}_t, B, p_{i,j,t})}, \tag{6}$$

where A is the size of the CFL model parameters, which is assumed to be equal for each device.

The energy consumption of device i transmitting its FL parameters to device j can be expressed as

$$E_{i,j,t}\left(\boldsymbol{u}_{i,t}, \boldsymbol{\phi}_t, B, p_{i,j,t}\right) = p_{i,j,t}l_{i,j,t}(\boldsymbol{u}_{i,t}, \boldsymbol{\phi}_t, B, p_{i,j,t}). \tag{7}$$

Given (7), the energy consumption of device i updating its local FL model at iteration t is

$$E_{i,t}(\boldsymbol{u}_{i,t}, \boldsymbol{\phi}_t, B, \boldsymbol{p}_{i,t}) = \sum_{j=1}^{M} u_{i,j,t}E_{i,j,t}(\boldsymbol{u}_{i,t}, \boldsymbol{\phi}_t, B, p_{i,j,t}), \tag{8}$$

where $\boldsymbol{p}_{i,t} = [p_{i,1,t}, ..., p_{i,j,t}, ..., p_{i,M,t}]$ is the transmit power vector of device i at time slot t.

D. Problem Formulation

Our goal is to minimize the total energy consumption under the constraints of latency and CFL training loss. The optimization problem is formulated as

$$\min_{\boldsymbol{U},\boldsymbol{P}} \sum_{t=1}^{T}\sum_{i=1}^{M} E_{i,t}\left(\boldsymbol{u}_{i,t}, B, \boldsymbol{\phi}_t, \boldsymbol{p}_{i,t}\right), \tag{9}$$

$$\text{s.t. } l_{i,j,t}(\boldsymbol{u}_{i,t}, \boldsymbol{\phi}_t, B, p_{i,j,t}) \leqslant \Gamma, \forall i, j \in \mathcal{M}, \forall t \in \mathcal{T}, \tag{9a}$$

$$F\left(\boldsymbol{g}\left(\boldsymbol{u}_T\right)\right) < F^*, \tag{9b}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_t, ...\boldsymbol{u}_T]^\top$ is the FL model transmission matrix, $\boldsymbol{P} = [\boldsymbol{p}_1, ..., \boldsymbol{p}_t, ...\boldsymbol{p}_T]^\top$ is the transmit power matrix. Here, $\boldsymbol{p}_t = [\boldsymbol{p}_{1,t}, ..., \boldsymbol{p}_{i,t}, ..., \boldsymbol{p}_{M,t}]$ is the transmit power vector for each device at time slot t. Γ is the maximum FL model transmission delay per iteration allowed by the network operator. (9a) is a constraint on the FL model transmission delay per iteration, and (9b) is the CFL training loss requirement. The problem in (9) is challenging to solve by traditional algorithms due to the following reasons. First, FL model transmission matrix $\boldsymbol{U}$ and transmit power matrix $\boldsymbol{P}$ are dependent and hence iterative algorithms with high complexity must be used to solve problem (9). Meanwhile, these iterative algorithms require each device to collect the information (i.e., locations) of all devices in the network find the globally optimal solution for problem (9), which will further increase the communication overhead and complexity of solving problem (9). Second, in (9), device locations will change over time. Therefore, traditional iterative algorithms must re-execute for different network topology, which may lead to additional communication and computational overhead. To overcome these challenges, we propose a fully distributed algorithm based on GNNs which enables each device to require only its neighboring devices' location information to determine the FL model transmission matrix $\boldsymbol{U}$ and transmit power $\boldsymbol{P}$.

III. Proposed Method

The proposed scheme of solving problem (9) consists of two stages. At the first stage, a GNN based algorithm is proposed [8], which enables each device i to individually determine the FL model transmission matrix $\boldsymbol{u}_i$ using its neighboring devices' location and connection information. Compared to standard iterative algorithms that need to iteratively optimize $\boldsymbol{u}_i$ and $\boldsymbol{P}$, the proposed GNN based method can directly obtain the optimal $\boldsymbol{u}_i^*$ without iterative optimization. Given $\boldsymbol{u}_i^*$, at the second stage, each device can directly obtain the optimal transmit power $\boldsymbol{p}_i^*$. Next, we will first introduce the components of the designed GNN-based algorithm. Then, the training method of the designed algorithm is explained. Finally, we explain the optimization of transmit power given the optimal FL model transmission matrix $\boldsymbol{U}^*$.

A. Components of the GNN-based Algorithm

As shown in Fig. 2, the proposed GNN-based algorithm that is implemented by each device consists of three components: a) input, b) hidden layer, and c) output layer, which are specified as follows:

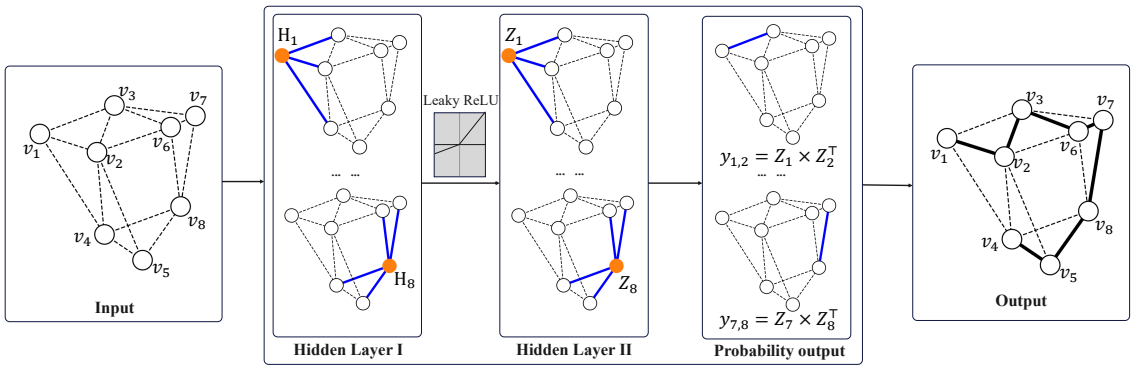

Fig. 2. Illustration of our proposed GNN-based algorithm.

1) Input: To determine the input of the GNN based method, we first construct a k nearest neighbor graph [9]. Here, we use a k nearest neighbor graph instead of a fully-connected graph for the input of the GNN. Let $\mathcal{L}^1(i,k)$ be the set of the first hop devices that can directly connect to device i. Let $\mathcal{L}^2(i,k)$ be the set of the second hop devices that can connect to device i via the first hop devices. Let $\mathcal{L}(i,k) = \mathcal{L}^1(i,k) \cup \mathcal{L}^2(i,k) \cup \{i\}$ with $|\mathcal{L}(i,k)|$ being the number of devices in $\mathcal{L}(i,k)$. Then, we define a matrix $\boldsymbol{R}(i,k) \in \mathbb{R}^{|\mathcal{L}(i,k)| \times |\mathcal{L}(i,k)|}$ to represent the connectivity among devices in $\mathcal{L}(i,k)$, where $r_{s,d}(i,k) \in \{0,1\}$ is an element of $\boldsymbol{R}(i,k)$ with $r_{s,d}(i,k) = 1$ indicating that device d is one of the k nearest devices of device s and $r_{s,d}(i,k) = 0$, otherwise. The input of the GNN model of device i is matrix $\boldsymbol{R}(i,k)$ and a vector of locations of the first and second hop devices of device i, which is represented by $\boldsymbol{\Phi}_i = \left[\boldsymbol{\phi}_{1,t}, ..., \boldsymbol{\phi}_{j,t}, ..., \boldsymbol{\phi}_{|\mathcal{L}(i,k)|,t}\right]^\top \in \mathbb{R}^{|\mathcal{L}(i,k)| \times 2}$.

2) Hidden layer: The hidden layer consists two graph convolutional networks layers (GCNs), which is expressed as

$$\begin{aligned} \boldsymbol{H} &= \zeta(\boldsymbol{D}^{-\frac{1}{2}} \tilde{\boldsymbol{R}}(i,k)\, \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{\Phi}_i \boldsymbol{\Theta}_0), \\ \boldsymbol{Z} &= \zeta(\boldsymbol{D}^{-\frac{1}{2}} \tilde{\boldsymbol{R}}(i,k)\, \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{H} \boldsymbol{\Theta}_1), \end{aligned} \tag{10}$$

where $\tilde{\boldsymbol{R}}(i,k) = \boldsymbol{R}(i,k) + \boldsymbol{I}$ and $\boldsymbol{I} = \text{diag}(1)$, $\boldsymbol{D} = \text{diag}\left(||\boldsymbol{r}_1||, ..., ||\boldsymbol{r}_i||, ..., ||\boldsymbol{r}_{|\mathcal{L}(i,k)|}||\right)$ with $\boldsymbol{r}_i$ being row i of $\boldsymbol{R}(i,k)$ [10]. $\boldsymbol{\Theta}_0 \in \mathbb{R}^{2 \times V_1}$ and $\boldsymbol{\Theta}_1 \in \mathbb{R}^{V_1 \times V_2}$ are GNN model parameters and $\zeta(\cdot)$ is the leaky rectified linear unit function, and V_1 and V_2 are output dimensions. $\boldsymbol{H} \in \mathbb{R}^{(|\mathcal{L}(i,k)|) \times V_1}$ is the output of the first GCN layer and $\boldsymbol{Z} \in \mathbb{R}^{(|\mathcal{L}(i,k)|) \times V_2}$ is the output of the second GCN layer.

3) Output Layer: The output of the network is a probability distribution $\boldsymbol{\mu}_i = \left[\mu_{i,1}, ..., \mu_{i,|\mathcal{L}^1(i,k)|}\right]$ of device i connecting to its first hop devices, where the probability $\mu_{i,j}$ of device i connect to device j can be expressed as

$$\mu_{i,j} = \frac{e^{\boldsymbol{z}_i \times \boldsymbol{z}_j^\top}}{\sum_{l=1}^{|\mathcal{L}^1(i,k)|} e^{\boldsymbol{z}_i \times \boldsymbol{z}_l^\top}}, \forall j \in \mathcal{L}^1(i,k), \tag{11}$$

where $\boldsymbol{z}_i$ is row i of $\boldsymbol{Z}$ in (10). Here, the prediction result $\boldsymbol{\mu}_i$ is determined not only by the FL model transmission matrix $\boldsymbol{U}$ but also the optimal transmit power vector $\boldsymbol{P}$ since the data labels used to train the GNN model are generated by the optimal $\boldsymbol{U}$ and $\boldsymbol{P}$.

Given the prediction $\boldsymbol{\mu}_i$, the next step is to determine the FL model transmission matrix $\boldsymbol{u}_i$ of each device i. First, each device i transmits its prediction result $\boldsymbol{\mu}_i$ to its neighboring devices. Then, each device i will use its received FL model transmission matrix to determine $\boldsymbol{\mu}_i$. We first define a set $\mathcal{S}_i = \left\{\mu_{i,1}, ..., \mu_{i,|\mathcal{L}^1(i,k)|}\right\}$ to represent the set of elements in $\boldsymbol{\mu}_i$ and $\mu_{i,n_{i,\max}} = \max\{x | x \in \mathcal{S}_i\}$ as the maximum connection probability among $\boldsymbol{\mu}_i$ with $n_{i,\max}$ being the device index. Similarly, we define $\mu_{i,n'_{i,\max}} = \max\left\{x | x \in \mathcal{S}(i) \text{ and } x \neq \mu_{i,n_{i,\max}}\right\}$ as the maximum connection probability among the elements in $\boldsymbol{\mu}_i$ except $\mu_{i,n_{i,\max}}$ with $n'_{i,\max}$ being the device index. Hence, element $u_{i,j}$ in FL model transmission matrix is determined by

$$u_{i,j} = \begin{cases} 1, & \text{if } j = n_{i,\max}, \\ 1, & \text{if } j = n'_{i,\max},\ \mu_{i,n_{i,\max}} < \mu_{n_{i,\max},i}, \\ & \text{and } n_{n_{i,\max},\max} = i, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where $n_{n_{i,\max},\max}$ is the index of device with the maximum connection probability in $\boldsymbol{\mu}_{n_{i,\max}}$. Here, $\boldsymbol{\mu}_{n_{i,\max}}$ represents the probability distribution of device $n_{i,\max}$ connecting to its first hop devices. From (12), we see that each device i can at most transmit FL parameters to two devices: 1) the device $n_{i,\max}$ with the highest connection probability and 2) the device $n'_{i,\max}$ with the second highest connection probability.

B. Training the Proposed GNN-based Model

Given the defined components, we next introduce the entire procedure of training the proposed GNN-based method. To minimize the difference between the predicted connections and the optimal connections, we use binary cross-entropy (BCE) as the loss function, which is expressed as

$$\begin{aligned} &\mathcal{J}(\boldsymbol{R}(i,k), \boldsymbol{\phi}_i, \boldsymbol{\Theta}_0, \boldsymbol{\Theta}_1) \\ &= \sum_{i=1}^{M} \sum_{j=1}^{|\mathcal{L}^1(i,k)|} -z_{i,j} \log \delta(\mu_{i,j}) - (1 - z_{i,j}) \log(1 - \delta(\mu_{i,j})), \end{aligned} \tag{13}$$

where $\delta(\cdot)$ is the sigmoid function, $z_{i,j} \in \{0,1\}$ is the label of connection with $z_{i,j} = 1$ indicating there is a connection

between device i and device j and $z_{i,j} = 0$, otherwise. To minimize the training loss in (13), we optimize $\boldsymbol{\Theta}_0$ and $\boldsymbol{\Theta}_1$ using the back-propagation algorithm with minibatch stochastic gradient descent (SGD) approach.

C. Optimization of Transmit Power Vector with Fixed FL Model Transmission Matrix

When the FL model transmission matrix $\boldsymbol{U}$ is determined, each device only needs to optimize its own transmit power vector $\boldsymbol{p}_{i,t}$. Given the optimized transmission matrix $\boldsymbol{U}$, the transmit power optimization problem for device i at time slot t can be rewritten as

$$\min_{\boldsymbol{p}_{i,t}} \sum_{j=1}^{M} E_{i,j,t}\left(\boldsymbol{\phi}_t, B, p_{i,j,t}\right), \tag{14}$$

$$\text{s.t. } l_{i,j,t}(\boldsymbol{\phi}_t, B, p_{i,j,t}) \leqslant \Gamma, \forall j \in \mathcal{L}^1\left(i,k\right). \tag{14a}$$

Based on (14), the optimal transmit power of each device i is given by the following lemma.

Lemma 1. The optimal transmit power $p_{i,j,t}$ of each device i transmiting its model to device j can be given by

$$p^*_{i,j,t} = \frac{u_{i,j,t}\sigma_N^2}{h_{i,j,t}\left(\boldsymbol{\phi}_t\right)}\left(2^{\frac{A||\boldsymbol{u}_i||}{B\Gamma}} - 1\right). \tag{15}$$

Proof. To prove Lemma 1, we first prove that energy consumption $E_{i,j,t}\left(\boldsymbol{\phi}_t, B, p_{i,j,t}\right)$ is an increasing function of variable $p_{i,j,t}$. Based on (5), (6), and (7), we have

$$E_{i,j,t}\left(\boldsymbol{\phi}_t, B, p_{i,j,t}\right) = \frac{A||\boldsymbol{u}_i||p_{i,j,t}}{B\log\left(1 + \frac{h_{i,j,t}(\boldsymbol{\phi}_t)p_{i,j,t}}{\sigma_N^2}\right)}. \tag{16}$$

The derivative of $E_{i,j,t}\left(\boldsymbol{\phi}_t, B, p_{i,j,t}\right)$ with respect to $p_{i,j,t}$ is given by

$$\begin{aligned}\frac{\partial E_{i,j,t}\left(\boldsymbol{\phi}_t, B, p_{i,j,t}\right)}{\partial p_{i,j,t}} &= \frac{A||\boldsymbol{u}_i||}{B\log\left(1 + \frac{h_{i,j,t}(\boldsymbol{\phi}_t)p_{i,j,t}}{\sigma_N^2}\right)} \\ &- \frac{Ah_{i,j,t}\left(\boldsymbol{\phi}_t\right)p_{i,j,t}||\boldsymbol{u}_i||}{B\sigma_N^2\log\left(1 + \frac{h_{i,j,t}(\boldsymbol{\phi}_t)p_{i,j,t}}{\sigma_N^2}\right)^2\left(1 + \frac{h_{i,j,t}(\boldsymbol{\phi}_t)p_{i,j,t}}{\sigma_N^2}\right)} \\ &= \frac{A||\boldsymbol{u}_i||}{Bw\left(p_{i,j,t}\right)\log\left(w\left(p_{i,j,t}\right)\right)^2} f\left(w\left(p_{i,j,t}\right)\right),\end{aligned} \tag{17}$$

where $w\left(p_{i,j,t}\right)$ and $f\left(w\left(p_{i,j,t}\right)\right)$ are short for $1 + \frac{h_{i,j,t}(\boldsymbol{\phi}_t)p_{i,j,t}}{\sigma_N^2}$ and $\left(w\left(p_{i,j,t}\right)\left(\log\left(w\left(p_{i,j,t}\right)\right) - 1\right) + 1\right)$, respectively. Obviously, $\frac{A||\boldsymbol{u}_i||}{Bw(p_{i,j,t})\log(w(p_{i,j,t}))^2}$ and $\frac{\partial w(p_{i,j,t})}{\partial p_{i,j,t}} = \frac{h_{i,j,t}(\boldsymbol{\phi}_t)}{\sigma_N^2}$ are always positive as $p_{i,j,t} \geq 0$. Hence, to prove that (17) is positive, we need to only prove that $f\left(w\left(p_{i,j,t}\right)\right)$ is positive when $w\left(p_{i,j,t}\right) \geq 1$. The derivative of $f\left(w\left(p_{i,j,t}\right)\right)$ with respect to $w\left(p_{i,j,t}\right)$ is given by

$$\frac{\partial f\left(w\left(p_{i,j,t}\right)\right)}{\partial w\left(p_{i,j,t}\right)} = \log\left(w\left(p_{i,j,t}\right)\right) + \frac{1}{\ln 2} - 1 > 0. \tag{18}$$

Obviously, $f\left(w\left(p_{i,j,t}\right)\right)$ is a monotonically increasing function when $w\left(p_{i,j,t}\right) \geq 1$. Therefore, $f\left(w\left(p_{i,j,t}\right)\right) \geq 0$ when $w\left(p_{i,j,t}\right) \geq 1$. Since $\frac{\partial E_{i,j,t}(\boldsymbol{\phi}_t, B, p_{i,j,t})}{\partial p_{i,j,t}}$ is always positive when $p_{i,j,t} \geq 0$, $E_{i,j,t}\left(\boldsymbol{\phi}_t, B, p_{i,j,t}\right)$ is a monotonically increasing function when $p_{i,j,t} \geq 0$. From the constraint (14a) we have $p_{i,j,t} \geq \frac{u_{i,j,t}\sigma_N^2}{h_{i,j,t}(\boldsymbol{\phi}_t)}\left(2^{\frac{A||\boldsymbol{u}_i||}{B\Gamma}} - 1\right)$. Hence, to minimize the transmission energy consumption, the optimal transmit power $p^*_{i,j,t}$ is $\frac{u_{i,j,t}\sigma_N^2}{h_{i,j,t}(\boldsymbol{\phi}_t)}\left(2^{\frac{A||\boldsymbol{u}_i||}{B\Gamma}} - 1\right)$. This completes the proof. □

Algorithm 1 GNN-based Algorithm for Connection Optimization

1: **Initialize** The FL model transmission matrix $\boldsymbol{U}$.
2: Devices obtain location information based on GPS data;
3: Each device i broadcasts its location information to devices in $\mathcal{L}\left(i,k\right)$;
4: Each device i establishes $\boldsymbol{R}\left(i,k\right)$ and $\boldsymbol{\Phi}_i$;
5: **for** device $i \in \mathcal{M}$ **do**
6: Aggregate the neighboring devices' information based on (10);
7: Calculate the output probability distribution $\boldsymbol{\mu}_i$ based on (11);
8: Each device i transmits $\boldsymbol{\mu}_i$ to its neighboring devices;
9: **end for**
10: Each device i determines $\boldsymbol{u}_i$ based on (12);
Output: The FL model transmission matrix $\boldsymbol{U}$.

TABLE I
SIMULATION PARAMETERS [11]

Parameters	Values	Parameters	Values
B	1 MHz	f	3.3GHz
σ_N^2	-174 dB	Γ	10ms
A	92 KB	ρ	1
V_1	5	V_2	10
M	8	k	3

IV. SIMULATION RESULTS

For our simulations, we consider a network with a circular area having a radius $r = 1000$ m and 8 uniformly distributed devices. The other parameters used in simulations are listed in Table I. We consider the use of FL for handwritten digit identifications based on the MNIST dataset [12]. For comparison, we consider two baselines. Baseline a) is an optimal solution, which is obtained by exhaustive search. Baseline b) is a decentralized solution, where each device directly connects to its first and second nearest neighboring devices with optimal transmit power.

In Fig. 3, we show how the energy consumption of CFL training changes as the number of devices varies. In this simulation, we only train the GNN using the data collected by the network with 8 devices. When the network has more or less than 8 devices, we directly use the GNN model trained by the data collected from the network with 8 devices. From Fig. 3, we can see that the proposed method achieves similar performance as the optimal solution. This is because the proposed algorithm enables each device to determine the FL model transmission matrix using its neighboring devices'

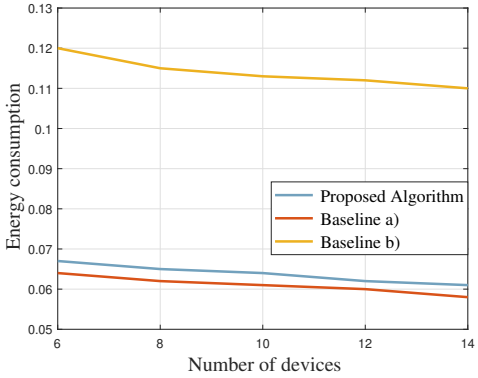

Fig. 3. Energy consumption vs. the number of devices with $k = 3$.

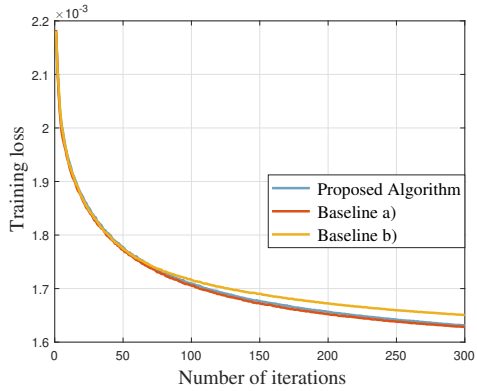

Fig. 4. Transmit loss vs. the number of iterations.

location and connection information thus optimizing FL model transmission matrix and transmit power. From this figure, we can also see that, as the number of devices increases, the average energy consumption of each device decreases. This is because, as the number of devices increases, the FL parameter transmission distance among the devices decreases. Fig. 3 also shows that compared to baseline b), the proposed algorithm can reduce energy consumption of CFL training by up to 46%. This is because the proposed algorithm can select an optimal subset of devices for FL parameter exchange. Fig. 3 also shows that the gap between the proposed algorithm and baseline a) does not change as the number of devices increases. This implies that the proposed algorithm trained in the network with 8 devices can directly be used for the networks with more or less than 8 devices.

Fig. 4 shows how the average CFL training loss changes as the number of iterations varies. From Fig. 4, we observe that, as the number of iterations increases, the average CFL training loss decreases. This is due to the fact that devices share local FL model parameters with their neighboring devices per iteration such that the proposed CFL converges. Fig. 4 also shows that the proposed algorithm can reduce training loss by up to 2% compared to baseline b). This is due to the fact that the proposed algorithm can optimize the FL model transmission matrix via analyzing the location and device connection information.

V. Conclusion

In this paper, we have developed a novel CFL framework that enables devices to collaboratively train a ML model via exchanging their local FL parameters with their neighboring device without the reliance on a parameter server. We have formulated an optimization problem, whose goal is to minimize CFL energy consumption while satisfying the delay and CFL training loss requirements. To solve this problem, we proposed a two-stage solution. At the first stage, we have proposed a GNN based algorithm which enables each device to individually determine its FL parameter transmission matrix using its neighboring devices' location and connection information. Given the optimal FL parameter transmission matrix, at the second stage, we have derived a close-form expression of the optimal transmit power. Simulation results have demonstrated that the proposed algorithm can significantly reduce CFL training energy consumption compared to the algorithm where each device will directly connect to its first and second nearest neighbors.

References

[1] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Guest editorial special issue on distributed learning over wireless edge networks-part II," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 2, pp. 445–448, Jan. 2022.

[2] M. Chen, Z. Yang, W. Saad, C. Yin, and H. V. Poorand S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, Oct. 2021.

[3] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Wireless communications for collaborative federated learning," *IEEE Communications Magazine*, vol. 58, no. 12, pp. 48–54, Dec. 2020.

[4] Y.-T. Chow, W. Shi, T. Wu, and W. Yin, "Expander graph and communication-efficient decentralized optimization," in *Proc. Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, Nov. 2016.

[5] L. Wang, Y. Xu, H. Xu, M. Chen, and L. Huang, "Accelerating decentralized federated learning in heterogeneous edge computing," *IEEE Transactions on Mobile Computing*, pp. 1–1, May. 2022.

[6] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, May 2018.

[7] G. Qu and N. Li, "Accelerated distributed nesterov gradient descent," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2566–2581, Aug. 2020.

[8] Y. Shen, J. Zhang, S. H. Song, and K. B. Letaief, "Graph neural networks for wireless communications: From theory to practice," *IEEE Transactions on Wireless Communications*, vol. 22, no. 5, pp. 3554–3569, 2023.

[9] M. Lee, G. Yu, and G. Y. Li, "Graph embedding-based wireless link scheduling with few training samples," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2282–2294, Dec. 2021.

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. International Conference on Learning Representations (ICLR)*, Toulon, France, May. 2017.

[11] S. Wang, M. Chen, Z. Yang, C. Yin, W. Saad, S. Cui, and H. V. Poor, "Distributed reinforcement learning for age of information minimization in real-time iot systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 501–515, Jan. 2022.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.