

Decentralized Federated Learning with Distributed Aggregation Weight Optimization

Zhiyuan Zhai, Xiaojun Yuan, *Senior Member, IEEE*, Xin Wang, *Fellow, IEEE*, and Geoffrey Ye Li, *Fellow, IEEE*

Abstract—Decentralized federated learning (DFL) is an emerging paradigm to enable edge devices collaboratively training a learning model using a device-to-device (D2D) communication manner without the coordination of a parameter server (PS). Aggregation weights, also known as mixing weights, are crucial in DFL process, and impact the learning efficiency and accuracy. Conventional design relies on a so-called central entity to collect all local information and conduct system optimization to obtain appropriate weights. In this paper, we develop a distributed aggregation weight optimization algorithm to align with the decentralized nature of DFL. We analyze convergence by quantitatively capturing the impact of the aggregation weights over decentralized communication networks. Based on the analysis, we then formulate a learning performance optimization problem by designing the aggregation weights to minimize the derived convergence bound. The optimization problem is further transformed as an eigenvalue optimization problem and solved by our proposed subgradient-based algorithm in a distributed fashion. In our algorithm, edge devices only need local information to obtain the optimal aggregation weights through local (D2D) communications, just like the learning itself. Therefore, the optimization, communication, and learning process can be all conducted in a distributed fashion, which leads to a genuinely distributed DFL system. Numerical results demonstrate the superiority of the proposed algorithm in practical DFL deployment.

Index Terms—Decentralized federated learning, distributed aggregation weight optimization, communication networks.

I. INTRODUCTION

Due to the unprecedented increase in local data generated by edge devices, there is a rising trend in developing deep learning applications at the network edge, which span many research areas, including image recognition [1] and natural language processing [2]. However, traditional machine learning (ML) approaches need to collect data from the edge devices for centralized training, which consumes large communication bandwidth and causes potential privacy concerns. Federated learning (FL) [3]–[7], a novel machine learning paradigm, is capable of addressing these issues by enabling edge devices to collaboratively train a global learning model under the coordination of a parameter server (PS). In FL, each device independently updates local model, such as model parameters or gradients, using its own datasets. These updates are then transmitted to the central PS, where the aggregated model is computed and broadcast back to the edge devices.

One important limitation of FL is its heavy dependence on the central PS for parameter aggregation, which leads to huge communication overheads and decreases fault tolerance. Furthermore, in scenarios, such as autonomous robotics and collaborative driving [8], FL may be unrealistic due to the lack of a central PS. Decentralized federated learning (DFL)

is a promising alternative to tackle these limitations. In DFL, the dependence on the PS is alleviated by allowing each device to maintain and refine its own local model, with model updates exchanged through device-to-device (D2D) communications. The concept of DFL is inherited from decentralized optimization, which can be dated back to the 1980s [9], with the foundational algorithms, such as dual averaging [10], alternating direction method of multipliers (ADMM) [11], and gradient descent [12] being pivotal. Recently, decentralized stochastic gradient descent (DSGD) [13], [14] has appeared as an innovative algorithm for tackling large-scale decentralized optimization challenges. DSGD guarantees optimal convergence under assumptions on convexity, gradient function, and network connectivity. This approach has been further adapted to tackle various network configurations. For instance, the technique in [15] combines quantization, sparsification, and local computations to mitigate communication overhead. Moreover, the algorithm in [16] incorporates local SGD updates, synchronous communication, and pairwise gossip in dynamically changing network topologies.

Aggregation weights, also known as mixing weights, have significant impacts on the learning performance of DFL. Since each device receives models from many other devices through D2D communication, these models must be aggregated with appropriate weight for training. When deploying DFL, aggregation weights are crucial to improve the training efficiency and learning accuracy [17], [18]. However, in a decentralized network, where DFL is deployed, aggregation weight design becomes very challenging due to the complexity of the network structure and the varying quality of D2D communication links. Many existing works, e.g., [19]–[24], have addressed this issue. However, all of them rely on the centralized approach and involve the collection of the local information from all devices. In these works, it is generally assumed that there exists a central entity (server or monitor) to gather the information from devices and link statistics, then optimize the aggregation weights accordingly. This approach fundamentally contradicts the spirit of decentralization in DFL design [25].

In this paper, we investigate distributed aggregation weight optimization, where edge devices with local information cooperatively optimize aggregation weights through D2D communications, just like the DFL learning process. By using this design, the entire optimization, communication, and learning process can be conducted in a distributed manner, which forms a fully decentralized DFL system. Specifically, we consider a DFL system where devices exchange models through D2D communication links and the quality of links are characterized

by the link reliability metric. For this scenario, we conduct a rigorous convergence analysis to quantitatively reveal the impact of aggregation weights on the learning performance over the communication networks. Based on this analysis, aggregation weights are optimized to minimize the convergence bound, which is a non-convex eigenvalue optimization problem. In general, such a problem is hard to be solved in a distributed fashion. However, by exploiting the problem structure, we are able to develop a distributed subgradient-based algorithm to solve the problem over the communication network. Simulation results confirm our convergence analysis and demonstrate the superiority of the proposed distributed algorithm. Furthermore, the proposed distributed algorithm can achieve similar performance as centralized method.

The rest of this paper is outlined as follows. Section II details the DFL learning model and the modeling of communication quality. Section III introduces the preliminary assumptions and derives a convergence bound. In Section IV, we formulate the performance optimization problem and propose a distributed algorithm to determine aggregation weights. Section V shows the simulation results, and the conclusion is provided in Section VI.

Notations: We use the notation $[M]$ to denote the set $\{i | 1 \leq i \leq M\}$ and use $\mathbb{R}$ to denote the sets of real numbers. Scalars are represented by regular letters, vectors by bold lowercase letters, and matrices by bold capital letters. Symbol $\odot$ denotes the Hadamard product (also known as the element-wise product) of two matrices of the same dimensions and $(\cdot)^T$ indicates the transpose operator. Symbol $\text{Diag}(\mathbf{A})$ converts a square matrix $\mathbf{A}$ into a diagonal matrix by retaining the diagonal elements and setting all off-diagonal elements to zero. Symbol $\lambda_i(\cdot)$ denotes the i -th largest eigenvalue of a matrix. The i -th entry of a vector $\mathbf{x}$ is denoted as x_i , and the (i, j) -th entry of a matrix $\mathbf{X}$ is denoted as x_{ij} . The l_2 -norm for vector and Frobenius norm $\|\cdot\|_F$ for matrix are both denoted as $\|\cdot\|$. The expectation operator is denoted by $\mathbb{E}$. $\mathbf{1}$ is used to denote a vector with all elements equal to 1 of appropriate dimension. The trace of a square matrix is denoted by $\text{Tr}(\cdot)$. The identity matrix is denoted by $\mathbf{I}$. The gradient of function f is denoted as $\nabla f(\cdot)$.

II. SYSTEM MODEL

We first describe the decentralized federated learning (DFL) model. In the DFL system, a set of M devices collaboratively conduct the training of a machine learning (ML) model, where the common objective is to minimize an empirical loss function, given by

$$f(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^M f_i(\mathbf{x}), \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^D$ represents the model and D denotes the dimensionality of the parameter space. For an arbitrary device i , the local loss function $f_i : \mathbb{R}^D \rightarrow \mathbb{R}$, is formulated as

$$f_i(\mathbf{x}) := \mathbb{E}_{\xi_i \in \mathcal{D}_i} F(\mathbf{x}, \xi_i), \quad (2)$$

where $\mathcal{D}_i$ is the local dataset on device i and $F(\mathbf{x}, \xi_i)$ is the loss function with respect to samples ξ_i .

TABLE I
SUMMARY OF MAIN NOTATIONS

Notation	Definition
$\mathbf{P}; p_{ij}$	Link reliability matrix; (i, j) -th element of $\mathbf{P}$
$\mathbf{S}^{(t)}; s_{ij}^{(t)}$	Transmission matrix at round t ; (i, j) -th element of $\mathbf{S}^{(t)}$
$\mathbf{W}; w_{ij}$	Aggregation weight matrix; (i, j) -th element of $\mathbf{W}$
$\widehat{\mathbf{W}}^{(t)}; \widehat{w}_{ij}^{(t)}$	Mixing matrix at round t ; (i, j) -th element of $\widehat{\mathbf{W}}^{(t)}$
$\overline{\mathbf{W}}; \overline{w}_{ij}$	Expectation of $\widehat{\mathbf{W}}^{(t)}$; (i, j) -th element of $\overline{\mathbf{W}}$
$\overline{\mathbf{W}}^2; \overline{w}_{ij}^2$	Expectation of $(\widehat{\mathbf{W}}^{(t)})^2$; (i, j) -th element of $\overline{\mathbf{W}}^2$
$\rho(\overline{\mathbf{W}})$	Second-largest eigenvalue (in magnitude) of $\overline{\mathbf{W}}$
$\rho(\overline{\mathbf{W}}^2)$	Second-largest eigenvalue (in magnitude) of $\overline{\mathbf{W}}^2$

In such a system, the devices compute the local model by minimizing their local loss function, and exchange the acquired model through D2D communications. Most of recent decentralized learning frameworks [13]–[16] also assume stable and reliable communication links. However, in practical networks, the communication system is prone to transmission distortion or failure. The practical conditions, such as channel fading, additive noise, path loss, and limited resources, make the D2D link unreliable. During each communication round, only a subset of the links can achieve successful communication. The quality of link can be quantified by a link reliability matrix $\mathbf{P} \in \mathbb{R}^{M \times M}$, which is invariant during the whole training process¹. In this matrix, the (i, j) -th element, p_{ij} , denotes the probability of successful transmission from the i -th device to the j -th device². Since a device does not need to communicate with itself, we have $p_{ii} = 0, \forall i \in [M]$ ³.

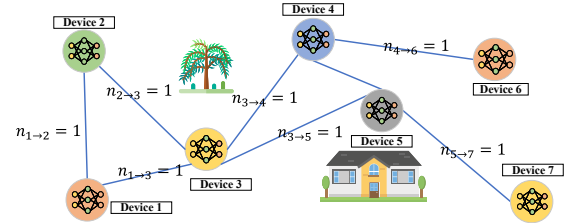


Fig. 1. An example of the links with successful transmission.

When the i -th device broadcasts its model, i.e., $\mathbf{x}_i \in \mathbb{R}^D$, to the rest of network, this model is received by other devices with some random distortion. The reception of $\mathbf{x}_i$ at the j -th device, denoted by $\mathbf{r}_{i \rightarrow j}$, is expressed as $\mathbf{r}_{i \rightarrow j} = n_{i \rightarrow j} \mathbf{x}_i$, where $n_{i \rightarrow j} = 1$ if the model of the i -th device is successfully received by the j -th device, and $n_{i \rightarrow j} = 0$ otherwise. Without loss of generality, we assume the reciprocity of the transmission link, that is $n_{i \rightarrow j} = n_{j \rightarrow i}, \forall i, j$. An example of

¹Note that our analytical framework can be directly applied to the situation where the transmission probability is dynamic, i.e., $\mathbf{P}$ changes over training rounds. In this paper, we focus on the static case to simplify our analysis.

²This link reliability model can be applied to decentralized network with any topology. For a specific topology where certain devices cannot communicate, we just need to set the corresponding reliability $p_{ij} = 0$. This probability model allows us to capture the stochastic and unreliable nature of D2D communication while keeping the optimization framework general and independent of a specific channel model [26].

³We can also assume $p_{ii} = 1$ and rewrite (3) as $\mathbf{x}_i^{(t+\frac{1}{2})} = \sum_{j=1}^M w_{ij} \hat{\mathbf{r}}_{j \rightarrow i}^{(t)}$ correspondingly. This does not hurt our theoretical analysis.

the links that achieve successful communication in one round is shown in Fig. 1, where the blue line represents the links with successful transmission.

We are now ready to introduce the training process of the DFL system. We adopt the stochastic gradient descent (SGD) technique [27] for local model training. The models of each device are updated in an iterative manner at each training round. In the t -th training round, the training process consists of the following steps.

- **Local computation:** Each device, say device i , computes its local stochastic gradient $\nabla F(\mathbf{x}_i^{(t)}, \xi_i^{(t)})$ by randomly selecting data $\xi_i^{(t)}$ from the local dataset $\mathcal{D}_i$, where $\mathbf{x}_i^{(t)}$ represents the model of device i at the t -th training round.
- **Model exchange:** Each device communicates with others to exchange the model parameters. Specifically, each device broadcasts the model to the rest of the network. If transmission fails in certain communication links, the receiving device cannot obtain the correct model. To mitigate this issue, the receiving device will substitute this lost value with its own model. Therefore, the received model of device i from device j is expressed as $\hat{\mathbf{r}}_{j \rightarrow i}^{(t)} = \mathbf{r}_{i \rightarrow j}^{(t)} + (1 - n_{i \rightarrow j}^{(t)})\mathbf{x}_i^{(t)}$. After that, each device calculates the weighted average of models as

$$\begin{aligned} \mathbf{x}_i^{(t+\frac{1}{2})} &= w_{ii}\mathbf{x}_i^{(t)} + \sum_{j=1, j \neq i}^M w_{ij}\hat{\mathbf{r}}_{j \rightarrow i}^{(t)} \\ &= \mathbf{x}_i^{(t)} + \sum_{j=1, j \neq i}^M w_{ij}n_{j \rightarrow i}^{(t)}(\mathbf{x}_j^{(t)} - \mathbf{x}_i^{(t)}), \end{aligned} \quad (3)$$

where w_{ij} represents the aggregation weight of the model from device j when device i performs model aggregation, and $\mathbf{x}_i^{(t+\frac{1}{2})}$ denotes the aggregated model at device i at the t -th training round.

- **Local update:** Using the aggregated model, $\mathbf{x}_i^{(t+\frac{1}{2})}$, each device updates its local model as

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t+\frac{1}{2})} - \lambda \nabla F(\mathbf{x}_i^{(t)}, \xi_i^{(t)}), \quad \forall i \in [M], \quad (4)$$

where $\lambda \in \mathbb{R}$ is the learning rate.

We view the above DFL training process from a global perspective. Define the concatenation of devices' models and stochastic gradients at round t as $\mathbf{X}^{(t)} \triangleq [\mathbf{x}_1^{(t)}, \dots, \mathbf{x}_M^{(t)}]$ and $\partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)}) \triangleq [\nabla F(\mathbf{x}_1^{(t)}, \xi_1^{(t)}), \dots, \nabla F(\mathbf{x}_M^{(t)}, \xi_M^{(t)})]$, respectively. Then, iterative update formula (4) can be rewritten in matrix form as

$$\mathbf{X}^{(t+1)} = \mathbf{X}^{(t)}\widehat{\mathbf{W}}^{(t)} - \lambda \partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)}), \quad (5)$$

where $\widehat{\mathbf{W}}^{(t)} \in \mathbb{R}^{M \times M}$ is the mixing matrix in the t -th round and is given by $\widehat{\mathbf{W}}^{(t)} = \mathbf{I} + \mathbf{W} \odot \mathbf{S}^{(t)} - \text{Diag}(\mathbf{W}\mathbf{S}^{(t)})$, and $\mathbf{S}^{(t)} \in \mathbb{R}^{M \times M}$ is the matrix representing the success of transmission at round t . If device i successfully transmits its model to device j , then $s_{ij}^{(t)} = 1$; otherwise, $s_{ij}^{(t)} = 0$. In particular, we have $s_{ii}^{(t)} = 0, \forall i$, and $s_{ij}^{(t)} = s_{ji}^{(t)}, \forall i, j$.

The randomness in the considered DFL system arises from two aspects: one is the randomness of the communication link, and the other is the randomness of the training samples. At

the t -th iteration, given the current model parameters $\mathbf{X}^{(t)}$ and the training samples $\boldsymbol{\xi}^{(t)}$, the expectation of (5) is

$$\mathbb{E}_{(\cdot|\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)})}\{\mathbf{X}^{(t+1)}\} = \mathbf{X}^{(t)}\overline{\mathbf{W}} - \lambda \partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)}), \quad (6)$$

where $\overline{\mathbf{W}}$ is the expectation of $\widehat{\mathbf{W}}^{(t)}$, i.e., $\overline{\mathbf{W}} = \mathbb{E}\{\widehat{\mathbf{W}}^{(t)}\}$, with its (i, j) -th entry given by

$$\overline{w}_{ij} = \begin{cases} w_{ij}p_{ij}, & i \neq j \\ 1 - \sum_{l=1, l \neq i}^M w_{il}p_{il}, & i = j \end{cases} \quad (7)$$

III. CONVERGENCE ANALYSIS

Our convergence analysis is based on the following assumptions.

Assumption 1. (Symmetric communication) We assume that the communication reliability from device i to device j is equal to that from device j to device i . That is, the link reliability matrix, $\mathbf{P}$, is a symmetric matrix, i.e., $\mathbf{P} = \mathbf{P}^T$.

Assumption 2. (Independent connections) The transmission reliability of different communication links (p_{ij} and p_{ji}) are considered to be the reliability of the same communication link) are independent.

Assumption 3. (Aggregation weights) Assume $\mathbf{W}$ is a symmetric doubly stochastic matrix, i.e., $\mathbf{W}^T = \mathbf{W}$, $\mathbf{W}\mathbf{1} = \mathbf{1}$. Define $\overline{\mathbf{W}^2} = \mathbb{E}\{(\widehat{\mathbf{W}}^{(t)})^2\}$ and $\rho(\overline{\mathbf{W}^2}) \triangleq \max\{\lambda_2(\overline{\mathbf{W}^2}), -\lambda_M(\overline{\mathbf{W}^2})\}$. We assume $\rho(\overline{\mathbf{W}^2}) < 1$.

Assumption 4. (Smoothness) Loss functions $f_1, \dots, f_M$ are all differentiable and their gradients $\nabla f_1(\cdot), \dots, \nabla f_M(\cdot)$ are Lipschitz continuous with parameter ω , i.e.,

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq \omega \|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^D, \forall i \in [M].$$

Assumption 5. (Bounded variance) The variances of stochastic gradient $\mathbb{E}\|\nabla F(\mathbf{x}, \xi_i) - \nabla f_i(\mathbf{x})\|^2$ and $\mathbb{E}\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|^2$ are bounded, i.e.,

$$\begin{aligned} \mathbb{E}\|\nabla F(\mathbf{x}, \xi_i) - \nabla f_i(\mathbf{x})\|^2 &\leq \alpha^2, \quad \forall \mathbf{x} \in \mathbb{R}^D, \forall i \in [M], \\ \mathbb{E}\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 &\leq \beta^2, \quad \forall \mathbf{x} \in \mathbb{R}^D, \end{aligned}$$

where α^2 denotes the upper limit of the variance of stochastic gradients among devices, and β^2 denotes the upper limit of the divergence of data distributions among devices. The expectation is taken over the randomness of local data sampling ξ_i in the first inequality and over the random selection of device index i (i.e., $i \sim \mathcal{U}([M])$) in the second inequality.

Assumptions 1 and 2 are related to the communication networks. Channel reciprocity ensures equal reliability for bidirectional transmissions over the same link, and hence Assumption 1 holds. Furthermore, Assumption 2 holds as long as the distance between devices significantly exceeds the carrier wavelength⁴ [21].

Assumptions 3-5 are widely adopted in research concerning decentralized stochastic optimization algorithms, e.g., [28],

⁴In some specific network structures, the link reliability matrix $\mathbf{P}$ may have dependent elements. Extending the analytical framework to account for these dependencies is beyond the scope of this paper but remains an important direction for future work.

[29], and [30]. From definitions, matrices $\widehat{\mathbf{W}}^{(t)}$, $\overline{\mathbf{W}}$ and $\overline{\mathbf{W}}^2$ are also shown to be symmetric doubly stochastic if aggregation weight matrix $\mathbf{W}$ is symmetric doubly stochastic. It is known that for a doubly stochastic matrix $\mathbf{W}$, $\lambda_1(\mathbf{W}) = 1$ and $|\lambda_i(\mathbf{W})| \leq 1, \forall i$. Assumption 3 requires that for all $i \neq 1$, $|\lambda_i(\mathbf{W})|$ must be strictly less than 1. This assumption is always true if the underlying graph of the communication network is connected and non-bipartite [31]. Assumption 4 is about the Lipschitz continuity property of the loss function. Assumption 5 guarantees that there is a limited disparity between the gradient of local sample-dependent loss function $\nabla F(\mathbf{x}, \xi_i)$ and the gradient of overall loss function $\nabla f(\mathbf{x})$. In practice, the assumption holds locally within the region visited by the algorithm, as the iterates remain bounded due to step-size control, Lipschitz continuity, and regularization. Based on the above assumptions, we can derive the following convergence theorem, proved in Appendix A.

Theorem 1. *Under Assumptions 1-5, with $\lambda < \frac{1-\sqrt{\rho(\overline{\mathbf{W}}^2)}}{6\sqrt{M}\omega}$, we have the following ergodic convergence bound*

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(t)} \mathbf{1}}{M} \right) \right\|^2 &\leq \frac{1}{(1/2 - 9G(\overline{\mathbf{W}}^2))} \\ &\times \left(\frac{f_0 - f^*}{\lambda T} + \frac{\lambda \omega \alpha^2}{2M} + \alpha^2 G(\overline{\mathbf{W}}^2) + 9\beta^2 G(\overline{\mathbf{W}}^2) \right) \end{aligned} \quad (8)$$

where $G(\overline{\mathbf{W}}^2) = \frac{M\lambda^2\omega^2}{(1-\sqrt{\rho(\overline{\mathbf{W}}^2)})^2 - 18M\lambda^2\omega^2}$, f_0 (or f^*) is initial (or optimal) value of the loss function, and the expectation is taken over the stochastic transmission and the randomness of data sampling.

Remark 1. *From Theorem 1, the ergodic bound consists of a decaying term proportional to $1/T$ and several constant terms related to λ , M , α , β , and $G(\overline{\mathbf{W}}^2)$. Therefore, the algorithm exhibits an $\mathcal{O}(1/T)$ transient convergence rate and converges to a neighborhood of the optimum determined by these constant terms. The neighborhood becomes smaller with larger M , smaller learning rate λ , lower gradient variance (α, β) , and better network mixing (smaller $G(\overline{\mathbf{W}}^2)$).*

Proposition 1. *The convergence bound in (8) is a monotonically increasing function with respect to $\rho(\overline{\mathbf{W}}^2)$.*

Proof. The convergence bound (8) can be abbreviated as $f(G(\overline{\mathbf{W}}^2)) = \frac{A+B G(\overline{\mathbf{W}}^2)}{1/2-9G(\overline{\mathbf{W}}^2)}$, where $A = \frac{f_0-f^*}{\lambda T} + \frac{\lambda \omega \alpha^2}{2M}$, $B = \alpha^2 + 9\beta^2$. The derivative of f with respect to $G(\overline{\mathbf{W}}^2)$ is $f'(G(\overline{\mathbf{W}}^2)) = \frac{1/2+9A}{(1/2-9G(\overline{\mathbf{W}}^2))^2} > 0$, which means that $f(G(\overline{\mathbf{W}}^2))$ monotonically increases with respect to $G(\overline{\mathbf{W}}^2)$. Furthermore, since $0 \leq \rho(\overline{\mathbf{W}}^2) < 1$, $G(\overline{\mathbf{W}}^2)$ also monotonically increases with respect to $\rho(\overline{\mathbf{W}}^2)$. Hence, the convergence bound in (8) is monotonically increasing with respect to $\rho(\overline{\mathbf{W}}^2)$. $\square$

From Proposition 1, aggregation weights $\mathbf{W}$ should be designed to minimize the value of $\rho(\overline{\mathbf{W}}^2)$ as much as possible, which provides a guideline for aggregation weight optimization.

IV. DISTRIBUTED WEIGHT OPTIMIZATION

In order to improve the learning performance of DFL, we shall design aggregation weight matrix $\mathbf{W}$ to minimize an objective function obtained from the convergence bound in (8). We will address this issue in this section.

A. Motivation of Distributed Optimization

Aggregation weights dictate how well models from various devices are combined, which in turn affects the system's capability to generalize and learn effectively. From Proposition 1, the aggregation weights should be chosen to minimize $\rho(\overline{\mathbf{W}}^2)$ in order for the DFL system to operate at its optimal performance.

In previous works [19]–[24], the aggregation weights are optimized based on a centralized approach, which involves the global information from all devices within the network and typically requires a central entity to collect local device states and link conditions. Since devices are often limited to device-to-device communications, distributed aggregation weight optimization is more practical than centralized approach.

B. Surrogate Objective Function

From Proposition 1, the convergence rate decreases monotonically with the increase of $\rho(\overline{\mathbf{W}}^2)$. To enhance the learning performance, we need to minimize $\rho(\overline{\mathbf{W}}^2)$, which however, is a nonlinear function of the second-order statistics, $\overline{\mathbf{W}}^2 = \mathbb{E}\{(\widehat{\mathbf{W}}^{(t)})^T (\widehat{\mathbf{W}}^{(t)})\}$. To design an effective decentralized algorithm, we replace $\rho(\overline{\mathbf{W}}^2)$ by a tractable surrogate objective function.

Our design is based on the following proposition, which is proved in Appendix B.

Proposition 2. *Assume a large-scale system where the number of devices approaches infinity, i.e., $M \rightarrow \infty$, $\rho(\overline{\mathbf{W}}) = \max\{\lambda_2(\overline{\mathbf{W}}), -\lambda_M(\overline{\mathbf{W}})\}$ can serve as a surrogate objective function of $\rho(\overline{\mathbf{W}}^2)$.*

Now we are ready to formulate the DFL learning performance optimization as follows.

$$\min_{\mathbf{W}} \quad \rho(\overline{\mathbf{W}}) = \max\{\lambda_2(\overline{\mathbf{W}}), -\lambda_M(\overline{\mathbf{W}})\} \quad (9a)$$

$$\text{s.t.} \quad \mathbf{W}^T = \mathbf{W}, \mathbf{W}\mathbf{1} = \mathbf{1}, \mathbf{W} \in [0, 1]^{M \times M}. \quad (9b)$$

Problem (9) is an eigenvalue optimization problem. The main challenge comes from the distributed solving restriction. We develop a distributed subgradient-based algorithm to address this problem subsequently.

C. Subgradient Analysis

Since $\overline{\mathbf{W}}$ is a doubly stochastic symmetric matrix with eigenvalue $\lambda_1(\overline{\mathbf{W}}) = 1$, problem (9) can be transformed to

$$\min_{\mathbf{W}} \quad \rho(\overline{\mathbf{W}}) = \max\{\lambda_2(\overline{\mathbf{W}}), -\lambda_M(\overline{\mathbf{W}})\} \quad (10a)$$

$$\text{s.t.} \quad \mathbf{W}^T = \mathbf{W}, \mathbf{W}\mathbf{1} = \mathbf{1}, \mathbf{W} \in [0, 1]^{M \times M}. \quad (10b)$$

We call the second largest (in magnitude) eigenvalue, $\rho(\bar{\mathbf{W}})$, the mixing rate of $\bar{\mathbf{W}}$. Since $\lambda_1(\bar{\mathbf{W}}) = 1$, we can express the second largest eigenvalue as

$$\lambda_2(\bar{\mathbf{W}}) = \sup\{\mathbf{u}^T \bar{\mathbf{W}} \mathbf{u} \mid \|\mathbf{u}\|_2 \leq 1, \mathbf{1}^T \mathbf{u} = 0\}. \quad (11)$$

As $\lambda_2(\bar{\mathbf{W}})$ is a point-wise supremum of a family of linear functions ($\mathbf{u}^T \bar{\mathbf{W}} \mathbf{u}$) of $\bar{\mathbf{W}}$, it is thus convex [32, Section 3.2.3]. Similarly, the negative of the smallest eigenvalue $-\lambda_M(\bar{\mathbf{W}})$ can be expressed as

$$-\lambda_M(\bar{\mathbf{W}}) = \sup\{-\mathbf{u}^T \bar{\mathbf{W}} \mathbf{u} \mid \|\mathbf{u}\|_2 \leq 1\}, \quad (12)$$

which is also convex. Therefore, $\rho(\bar{\mathbf{W}}) = \max\{\lambda_2(\bar{\mathbf{W}}), -\lambda_M(\bar{\mathbf{W}})\}$ is the point-wise maximum of two convex functions and hence it is convex. The subsequent discussion needs the following proposition.

Proposition 3. *A subgradient of $\rho(\bar{\mathbf{W}})$ is a symmetric matrix $\mathbf{G}$ that satisfies the following inequality*

$$\rho(\bar{\mathbf{W}}') \geq \rho(\bar{\mathbf{W}}) + \text{Tr}(\mathbf{G}(\bar{\mathbf{W}}' - \bar{\mathbf{W}})), \quad (13)$$

where $\bar{\mathbf{W}}'$ is an arbitrary symmetric doubly stochastic matrix, $\langle \cdot, \cdot \rangle$ represents the matrix inner product. When $\rho(\bar{\mathbf{W}}) = \lambda_2(\bar{\mathbf{W}})$ and $\mathbf{v}$ is the unit eigenvector corresponding to $\lambda_2(\bar{\mathbf{W}})$, the subgradient is given by $\mathbf{G} = \mathbf{v}\mathbf{v}^T$. Similarly, when $\rho(\bar{\mathbf{W}}) = -\lambda_M(\bar{\mathbf{W}})$ and $\mathbf{v}$ is a unit eigenvector corresponding to $\lambda_M(\bar{\mathbf{W}})$, we have $\mathbf{G} = -\mathbf{v}\mathbf{v}^T$.

Proof. We first consider the case, $\rho(\bar{\mathbf{W}}) = \lambda_2(\bar{\mathbf{W}})$, and $\mathbf{v}$ is the corresponding unit eigenvector. Since $\mathbf{1}$ is the eigenvector for eigenvalue 1 of matrix $\bar{\mathbf{W}}$, we have $\mathbf{v}^T \mathbf{1} = 0$. By using the variational characterization of the second largest eigenvalue λ_2 of matrix $\bar{\mathbf{W}}$ and $\bar{\mathbf{W}}'$, we have

$$\rho(\bar{\mathbf{W}}) = \lambda_2(\bar{\mathbf{W}}) = \mathbf{v}^T \bar{\mathbf{W}} \mathbf{v}, \quad (14a)$$

$$\rho(\bar{\mathbf{W}}') \geq \lambda_2(\bar{\mathbf{W}}') \geq \mathbf{v}^T \bar{\mathbf{W}}' \mathbf{v}. \quad (14b)$$

Subtracting the two sides of (14a) from those of (14b), we have

$$\begin{aligned} \rho(\bar{\mathbf{W}}') &\geq \rho(\bar{\mathbf{W}}) + \mathbf{v}^T (\bar{\mathbf{W}}' - \bar{\mathbf{W}}) \mathbf{v}, \\ &= \rho(\bar{\mathbf{W}}) + \text{Tr}(\mathbf{v}\mathbf{v}^T (\bar{\mathbf{W}}' - \bar{\mathbf{W}})). \end{aligned} \quad (15)$$

Hence, $\mathbf{G} = \mathbf{v}\mathbf{v}^T$ is a subgradient when $\rho(\bar{\mathbf{W}}) = \lambda_2(\bar{\mathbf{W}})$. Similarly, we can prove $\mathbf{G} = -\mathbf{v}\mathbf{v}^T$ when $\rho(\bar{\mathbf{W}}) = -\lambda_M(\bar{\mathbf{W}})$ and $\mathbf{v}$ is a unit eigenvector corresponding to $\lambda_M(\bar{\mathbf{W}})$. $\square$

Define matrices $\mathbf{E}_{ij}$, with entries $\mathbf{E}_{ij}(i, j) = \mathbf{E}_{ij}(j, i) = p_{ij}$, $\mathbf{E}_{ij}(i, i) = \mathbf{E}_{ij}(j, j) = -p_{ij}$, and zero entries everywhere else. Therefore, we can recast optimization problem (10) as

$$\min_{\mathbf{W}} \quad \rho \left(\mathbf{I} + \frac{1}{2} \sum_{i,j=1}^M w_{ij} \mathbf{E}_{ij} \right) \quad (16a)$$

$$\text{s.t.} \quad \mathbf{W}^T = \mathbf{W}, \mathbf{W}\mathbf{1} = \mathbf{1}, \mathbf{W} \in [0, 1]^{M \times M}. \quad (16b)$$

Denote $\mathbf{R}(\mathbf{W}) = \mathbf{I} + \frac{1}{2} \sum_{i,j=1}^M w_{ij} \mathbf{E}_{ij}$. In the subgradient method, we need to calculate the subgradient of the objective function $\rho(\mathbf{R}(\mathbf{W}))$ for a given feasible $\mathbf{W}$. If $\rho(\mathbf{R}(\mathbf{W})) = \lambda_2(\mathbf{R}(\mathbf{W}))$ and $\mathbf{v}$ is the corresponding unit eigenvector. From

Proposition 3, we have

$$\lambda_2(\mathbf{R}(\mathbf{W}')) \geq \lambda_2(\mathbf{R}(\mathbf{W})) + \sum_{i,j=1}^M (\mathbf{v}^T \mathbf{E}_{ij} \mathbf{v}) (w'_{ij} - w_{ij}), \quad (17)$$

so subgradient $g(\mathbf{W})$ is expressed as

$$g(\mathbf{W}) = \begin{bmatrix} \mathbf{v}^T \mathbf{E}_{11} \mathbf{v} & \mathbf{v}^T \mathbf{E}_{12} \mathbf{v} & \cdots & \mathbf{v}^T \mathbf{E}_{1M} \mathbf{v} \\ \mathbf{v}^T \mathbf{E}_{21} \mathbf{v} & \mathbf{v}^T \mathbf{E}_{22} \mathbf{v} & \cdots & \mathbf{v}^T \mathbf{E}_{2M} \mathbf{v} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}^T \mathbf{E}_{M1} \mathbf{v} & \mathbf{v}^T \mathbf{E}_{M2} \mathbf{v} & \cdots & \mathbf{v}^T \mathbf{E}_{MM} \mathbf{v} \end{bmatrix}, \quad (18)$$

and subgradient component $g(w_{ij})$ is

$$g(w_{ij}) = \mathbf{v}^T \mathbf{E}_{ij} \mathbf{v} = -p_{ij}(v_i - v_j)^2. \quad (19)$$

Similarly, if $\rho(\mathbf{R}(\mathbf{W})) = -\lambda_M(\mathbf{R}(\mathbf{W}))$ and $\mathbf{v}$ is the corresponding unit eigenvector, subgradient is given by

$$g(\mathbf{W}) = \begin{bmatrix} -\mathbf{v}^T \mathbf{E}_{11} \mathbf{v} & -\mathbf{v}^T \mathbf{E}_{12} \mathbf{v} & \cdots & -\mathbf{v}^T \mathbf{E}_{1M} \mathbf{v} \\ -\mathbf{v}^T \mathbf{E}_{21} \mathbf{v} & -\mathbf{v}^T \mathbf{E}_{22} \mathbf{v} & \cdots & -\mathbf{v}^T \mathbf{E}_{2M} \mathbf{v} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{v}^T \mathbf{E}_{M1} \mathbf{v} & -\mathbf{v}^T \mathbf{E}_{M2} \mathbf{v} & \cdots & -\mathbf{v}^T \mathbf{E}_{MM} \mathbf{v} \end{bmatrix}, \quad (20)$$

with the (i, j) -th subgradient component

$$g(w_{ij}) = -\mathbf{v}^T \mathbf{E}_{ij} \mathbf{v} = p_{ij}(v_i - v_j)^2. \quad (21)$$

Remark 2. From (19) and (21), for the subgradient of the aggregation weight between devices i and j , i.e., w_{ij} , we only need to know the link reliability information, i.e., p_{ij} and the (i, j) -th components of the unit eigenvector, i.e., v_i and v_j . This implies that if each device, say device i , knows its own link reliability $p_{ij}, \forall j$ and the eigenvector component v_i , then the subgradient can be computed in a distributed manner by using only local information.

D. Distributed Eigenvector Computation

In this subsection, we compute the eigenvector component of $\bar{\mathbf{W}}$ for the corresponding device in a distributed fashion, where device i is only aware of the i -th row of $\bar{\mathbf{W}}$ and can only communicate with its neighbors.

The problem of distributed computation of the top k eigenvectors of a symmetric weighted adjacency matrix of a graph is discussed in [33], the orthogonal iteration algorithm in [34] is adopted to the distributed environment for a QR decomposition based approach.

Since matrix $\bar{\mathbf{W}}$ is symmetric doubly stochastic with the largest eigenvalue 1 and the corresponding eigenvector $\mathbf{1}$, the orthogonal iterations take on a very simple form as summarized in Algorithm 1. We do not need to calculate any QR decomposition at device for orthogonalization.

Remark 3. Algorithm 1 can be performed in a distributed manner. In the initialization step, vector $\mathbf{v}(0)$ can be constructed by the way that each device generates a random scalar component. To obtain the i -th element of $\mathbf{v}(k+1)$ in the multiplication $\mathbf{v}(k+1) = \bar{\mathbf{W}}\mathbf{v}(k)$ step, the i -th device can fetch the corresponding components of $\mathbf{v}(k)$ from its neighbors

Algorithm 1 Distributed Orthogonal Iterations

1: **Initialization:** Random chosen vector $\mathbf{v}(0)$, iteration index $k = 1$, and number of iterations $K_{\max}$.
 2: **for** $k \leq K_{\max}$ **do**
 3: Update $\mathbf{v}(k+1) = \overline{\mathbf{W}}\mathbf{v}(k)$.
 4: Orthogonalize $\mathbf{v}(k+1) = \mathbf{v}(k+1) - \frac{1}{M}\mathbf{v}(k+1)^T\mathbf{1}\mathbf{1}$.
 5: Scale to vector $\mathbf{v}(k+1) = \mathbf{v}(k+1)/\|\mathbf{v}(k+1)\|$.
 6: **end for**
Output: $\mathbf{v}(k+1)$.

and aggregate them according to the i -th row of $\overline{\mathbf{W}}$. The orthogonalization and scaling steps involve the operation that sum over each element of $\mathbf{v}(k+1)$. This can be achieved through the distributed averaging method introduced in [35]. It can also be achieved by simply executing step 3 repeatedly, since $\overline{\mathbf{W}}$ is a doubly stochastic matrix where $\overline{\mathbf{W}}^\infty = \frac{1}{M}\mathbf{1}\mathbf{1}^T$.

Remark 4. According to [36, Section 7.3.1], obtained eigenvector $\mathbf{v}(k+1)$ in Algorithm 1 is naturally associated with the second largest (in magnitude) eigenvalue, i.e., $\rho(\overline{\mathbf{W}})$. Therefore, $\mathbf{v}(k+1)$ can be directly applied to the subgradient evaluation in Section IV-C (without the need to identify whether $\rho(\mathbf{R}(\mathbf{W})) = \lambda_2(\mathbf{R}(\mathbf{W}))$ or $\rho(\mathbf{R}(\mathbf{W})) = -\lambda_M(\mathbf{R}(\mathbf{W}))$).

E. Subgradient Projection Algorithm

In this subsection, we develop a distributed subgradient projection algorithm to solve (10) based on the analysis in Section IV-C and IV-D. The details of this algorithm is given in Algorithm 2.

Algorithm 2 Subgradient Projection Algorithm for (10)

1: **Initialization:** Channel reliability matrix $\mathbf{P}$, a feasible matrix $\mathbf{W}(0)$, iteration index $n = 1$, and number for iterations $J_{\max}$.
 2: **for** $n \leq J_{\max}$ **do**
 3: Compute unit eigenvector $\mathbf{v}(n)$ based on Algorithm 1.
 4: Compute subgradient $\mathbf{g}(\mathbf{W}(n))$ based on (19) and (21).
 5: Update $\mathbf{W}$ as $\mathbf{W}(n+1) = \mathbf{W}(n) - \gamma_n \mathbf{g}(\mathbf{W}(n))$.
 6: Project $\mathbf{W}$ onto the feasible set according to (22) and (26).
 7: **end for**
Output: $\mathbf{W}$.

We now show how to obtain the projected results with a distributed method. Since $\mathbf{W}$ is constrained to be symmetric, the projection method should be performed in a sequential manner. Each device sequentially calculates its projected aggregation weights.

Take device i as an example. The aggregation weights, $w_{ij}, \forall j < i$ should be equal to the projected result w_{ji} from the preceding device j to meet the symmetry requirement, i.e.,

$$w_{ij} = w_{ji}, \forall j < i. \quad (22)$$

Define $\mathbf{w}_i = [w_{ii+1}, w_{ii+2}, \dots, w_{iM}]^T \in \mathbb{R}^{m_i}$, $m_i = M - i$ and $l_i = 1 - \sum_{j=1}^{i-1} w_{ij}$. For aggregation weights $w_{ij}, \forall j > i$,

the projection result is obtained from the optimal conditions of the following problem.

$$\min_{\mathbf{q}} \quad \|\mathbf{q} - \mathbf{w}_i\|^2 \quad (23a)$$

$$\text{s.t.} \quad \mathbf{1}^T \mathbf{q} \leq l_i, \mathbf{q} \succeq 0. \quad (23b)$$

Problem (23) is a convex problem. By introducing Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^{m_i}$ for inequality constrain $\mathbf{q} \succeq 0$ and ν for equality constraint $\mathbf{1}^T \mathbf{q} - l_i = 0$, the Karush–Kuhn–Tucker (KKT) conditions for the optimal primal and dual variables $\mathbf{q}^*, \boldsymbol{\lambda}^*, \nu^*$ are

$$\mathbf{q}^* \succeq 0, \quad \mathbf{1}^T \mathbf{q}^* - l_i \leq 0, \quad (24a)$$

$$\boldsymbol{\lambda}^* \succeq 0, \quad \nu^* \geq 0, \quad (24b)$$

$$\boldsymbol{\lambda}_i^* \mathbf{q}_i^* = 0, \quad \nu^* (\mathbf{1}^T \mathbf{q}^* - l_i) = 0, \quad i = 1, \dots, m_i, \quad (24c)$$

$$2(\mathbf{q}_j^* - (\mathbf{w}_i)_j) - \boldsymbol{\lambda}_j^* + \nu^* = 0, \quad j = 1, \dots, m_i. \quad (24d)$$

By eliminating dual variables $\boldsymbol{\lambda}^*$, we obtain the equivalent optimality condition

$$\mathbf{q}^* \succeq 0, \quad \mathbf{1}^T \mathbf{q}^* - l_i \leq 0, \quad (25a)$$

$$\nu^* \geq 0, \quad \nu^* (\mathbf{1}^T \mathbf{q}^* - l_i) = 0, \quad (25b)$$

$$2(\mathbf{q}_j^* - (\mathbf{w}_i)_j) + \nu^* \mathbf{q}_j^* = 0, \quad j = 1, \dots, m_i, \quad (25c)$$

$$2(\mathbf{q}_j^* - (\mathbf{w}_i)_j) + \nu^* \geq 0, \quad j = 1, \dots, m_i. \quad (25d)$$

From (25d), if $\nu^* < 2(\mathbf{w}_i)_j$, we must have $\mathbf{q}_j^* > 0$. According to (25c), $\mathbf{q}_j^* = (\mathbf{w}_i)_j - \nu^*/2$. Otherwise, if $\nu^* \geq 2(\mathbf{w}_i)_j$, $\nu^* \geq 2(\mathbf{w}_i)_j - 2\mathbf{q}_j^*$ from (25d). Furthermore, it is necessary to have $\mathbf{q}_j^* = 0$ since $\mathbf{q}^* \succeq 0$ and the constraint (25c) holds. Therefore, we conclude

$$\mathbf{q}_j^* = \max \{0, (\mathbf{w}_i)_j - \nu^*/2\}, \quad j = 1, \dots, m_i. \quad (26)$$

Since $\mathbf{q}^*$ must satisfy $\mathbf{1}^T \mathbf{q}^* - l_i \leq 0$, we have $\sum_{j=1}^{m_i} \max \{0, (\mathbf{w}_i)_j - \nu^*/2\} \leq l_i$. Considering complementary slackness condition $\nu^* (\mathbf{1}^T \mathbf{q}^* - l_i) = 0$, we can obtain the solution of ν^* either at $\nu^* = 0$ or at ν^* satisfying $\sum_{j=1}^{m_i} \max \{0, (\mathbf{w}_i)_j - \nu^*/2\} = l_i$. Substituting ν^* into (26), we obtain the projected vector $\mathbf{q}^*$.

Remark 5. Our subgradient-based optimization framework is inspired by the distributed consensus optimization method in [37]. The proposed algorithm can be run in a distributed manner. In step 3, the unit eigenvector can be computed based on Algorithm 1, which is shown to be distributed in Remark 3. In step 4, since each device, say device i , is aware of its corresponding eigenvector component v_i , the subgradient of the i -th device's aggregation weights, i.e., $w_{ij}, \forall j$, can be computed by (19) and (21) with only local communications. In step 5, aggregation weights $w_{ij}, \forall j$ and corresponding subgradient components $g(w_{ij}), \forall j$ are held in each device, so each device is able to calculate the updated aggregation weights. Finally, in step 6, the projection method is sequentially carried out at each device based on (22) and (26), and hence this step can be performed distributedly.

Remark 6. The proposed algorithm requires prior knowledge of link reliability p_{ij} . In practice, this information can be estimated locally by each node based on its transmission history, for example, by computing the ratio of success-

fully acknowledged packets using acknowledgment (ACK) and negative acknowledgment (NACK) feedback messages. This provides a simple and fully distributed way to approximate p_{ij} without centralized assistance.

F. Convergence and Complexity Analysis

We first analyze the computational complexity of the algorithm by counting the number of multiplications and additions involved in Algorithm 2. For each device, the complexity of Step 3 is $K_{\max}M$, as it involves iterative updates over $K_{\max}$ rounds, each requiring operations with M elements. The complexity of Step 4 is $3M$ as it includes computations involving gradient updates for M elements with each accounting for 2 multiplications and 1 addition. Step 5 has a complexity of $2M$, corresponding to the updates of the aggregation weights, and Step 6 has a complexity of M , which accounts for the projection step. Since Algorithm 2 requires $J_{\max}$ iterations, the overall complexity for each device is $\mathcal{O}(J_{\max}(K_{\max} + 6)M)$, which simplifies to $\mathcal{O}(J_{\max}K_{\max}M)$. This demonstrates that the proposed algorithm has linear complexity with respect to the number of iterations. This property makes Algorithm 2 well-suited for deployment in distributed communication networks, where computational efficiency is critical.

According to [33], the distributed orthogonal iteration can compute the eigenvector in a decentralized manner with desired accuracy through local updates. Nevertheless, since Algorithm 1 relies on iterative numerical computation, approximation errors in the eigenvector are unavoidable, which lead to inexact subgradient evaluations in Algorithm 2. To verify that the proposed method remains stable under such approximation, we establish the following convergence result based on the framework of approximate subgradient methods [38], which is proved in Appendix C.

Proposition 4. Let $\mathbf{v}_r(n)$ be the exact eigenvector that lies in the eigenspace associated with $\rho(\bar{\mathbf{W}})$ and minimizes the distance to $\mathbf{v}(n)$. Assume that in iteration n , the eigenvector computed in Algorithm 1 satisfies $\|\mathbf{v}(n) - \mathbf{v}_r(n)\|_2^2 \leq \epsilon_n$, and that the stepsize is set to $\gamma_n = 1/n$. Then the generated sequence $\{\mathbf{W}(n)\}$ obeys

$$\liminf_n f(\mathbf{W}(n)) \leq f^* + \delta, \quad \delta = \limsup_n \epsilon_n,$$

where f^* is the optimal value of f and $\epsilon_n = \mathcal{O}(\epsilon_n)$.

In practice, the eigenvector accuracy, ϵ_n , decreases with the number of inner iterations $K_{\max}$ in Algorithm 1 [33]. Therefore, $K_{\max}$ can be increased until the resulting ϵ_n keeps $\epsilon_n = \mathcal{O}(\epsilon_n)$ below a desired tolerance while $J_{\max}$ specifies the total number of outer subgradient updates required to reach convergence.

V. NUMERICAL RESULTS

In this section we investigate the performance of the proposed distributed optimization algorithm over decentralized communication networks.

A. Experiment Settings

We conduct the image classification task on the MNIST dataset [39]. From the original dataset, we utilize 20,000 samples for training and 10,000 samples for validation. We implement the heterogeneous data splitting scheme described in [40]. Since the MNIST dataset comprises 10 classes, we divide the devices into 10 equally sized groups, with each group assigned dataset from a specific class. For the network configuration, we train a convolutional neural network (CNN) consisting of two 5×5 convolutional layers, with 10 and 20 links, respectively, each followed by 2×2 max pooling layers. This is followed by a batch normalization layer, a fully connected layer with 50 units and ReLU activation, and a final softmax output layer. The network comprises a total of 21,880 parameters. The cross-entropy loss function is utilized for training. We train this model with an NVIDIA RTX 3060Ti GPU.

For the communication setup, we generate a geometric random graph to represent the network. Specifically, we randomly place $M = 40$ devices within a 1×1 square unit area. The probability of successful communication decays with the distance between the corresponding devices, given by $p_{ij} = p_{ji} = \exp(-rd_{ij}^v)$, where d_{ij} denotes the distance between devices i and j , and r and v are adjustable parameters⁵. Based on this, we generate every realization $n_{i \rightarrow j}^{(t)}$ by sampling from a Bernoulli distribution with the success probability p_{ij} . As for the optimization parameters, we set $K_{\max} = 10^4$, $J_{\max} = 10^4$, and step size $\gamma = 0.01$. The results are averaged over 40 Monte Carlo trials.

B. Validation of Surrogate Function

To start with, we conduct experiments to analyze the impact of surrogate objective function $\rho(\bar{\mathbf{W}})$ on system performance. To obtain various $\rho(\bar{\mathbf{W}})$, we use the convex optimization tool CVXPY [41] to randomly generate matrices $\bar{\mathbf{W}}$ with different $\rho(\bar{\mathbf{W}})$.

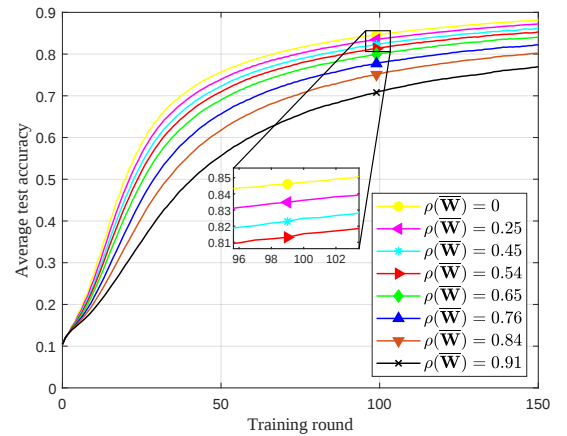


Fig. 2. Average test accuracy versus training round.

⁵ When a device is far from all others, i.e., d_{ij} is large for all j , the corresponding link reliabilities p_{ij} approach zero, which represents a device disconnection scenario.

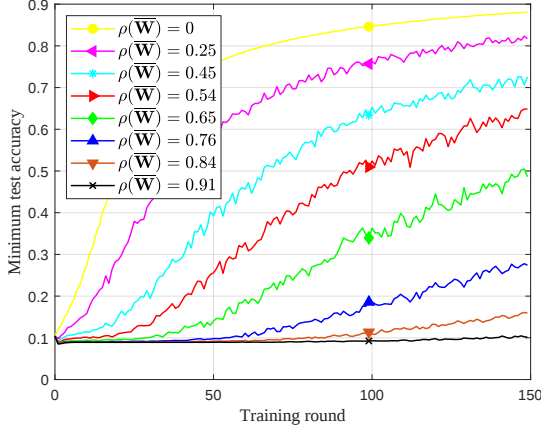


Fig. 3. Minimum test accuracy versus training round.

In Fig. 2 and 3, we show the average test accuracy (the average accuracy of all devices) and the minimum test accuracy (the minimum accuracy among all devices) for different $\rho(\bar{\mathbf{W}})$ over 150 training rounds. From the figure, the learning accuracy gradually decreases with the increase in $\rho(\bar{\mathbf{W}})$, exhibiting a monotonic relationship. This validates the precision of the surrogate objective function derived in Section IV-B. The case of $\rho(\bar{\mathbf{W}}) = 0$, with its minimum accuracy and average accuracy remaining consistent, performs best in both figures. This is because $\rho(\bar{\mathbf{W}}) = 0$ corresponds to the scenario where $\mathbf{W} = \frac{1}{M}\mathbf{11}^T$ and $\mathbf{P} = \mathbf{11}^T$, which is a fully connected network with reliable communication links. In this scenario, each device can obtain the model aggregated from all devices (i.e., global model). This setting is equivalent to traditional federated learning with error-free transmission [42].

Furthermore, the performance gap in the average accuracy with varying $\rho(\bar{\mathbf{W}})$ is relatively small, whereas this gap becomes significant in the minimum accuracy. This demonstrates that for DFL deployment with larger $\rho(\bar{\mathbf{W}})$, the discrepancies between the models of different devices are huge. Therefore, it implies that $\rho(\bar{\mathbf{W}})$ has a substantial impact on the learning and consensus performance and emphasizes the importance of the optimization.

C. Convergence of Proposed Algorithm

To validate the convergence performance of the proposed distributed algorithm, we show objective value $\rho(\bar{\mathbf{W}})$ in each subgradient iteration of Algorithm 2 and compare with the value obtained by centralized optimization algorithm. In the centralized algorithm, which assumes a central entity to collect information from all edge devices and solve problem (10) to optimize aggregation weight matrix $\mathbf{W}$. Since (10) is a convex problem, the obtained result from centralized algorithm is the global optimal solution.

We fix the device positions and vary the values of parameters r and v to acquire results under different link reliability matrix $\mathbf{P}$. Fig. 4 and Fig. 5 show the cumulative probability function of link reliability matrix $\mathbf{P}$ and the objective value of each iteration of proposed algorithm, respectively. In the proposed algorithm, we take aggregation weights $\mathbf{W} = \frac{1}{M}\mathbf{11}^T$

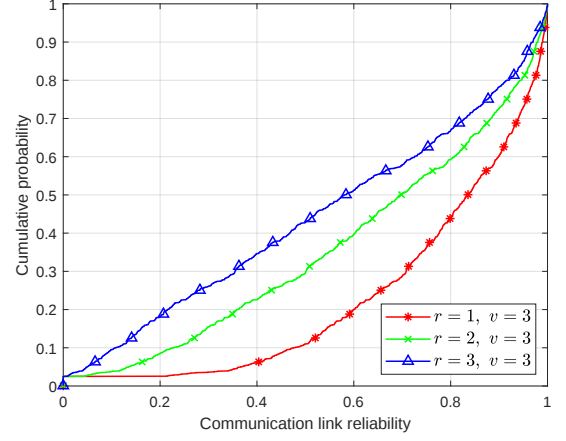


Fig. 4. Cumulative probability versus distance.

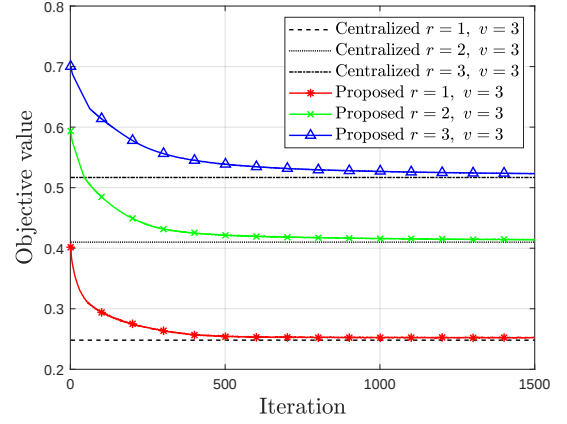


Fig. 5. Convergence performance of the proposed algorithm.

as the initial solution. When all the communication links are reliable, i.e., $\mathbf{P} = \mathbf{11}^T$, $\mathbf{W} = \frac{1}{M}\mathbf{11}^T$ is the optimal solution and we have $\rho(\bar{\mathbf{W}}) = 0$. From Fig. 5, the proposed algorithm consistently converges to the global optimal solution given by the centralized algorithm under different settings. Moreover, as the parameter r increases, the convergence value of the $\rho(\bar{\mathbf{W}})$ increases and the convergence time becomes longer. As shown in Fig. 4, when the value of r is larger, there are more communication links with low reliability and matrix $\mathbf{P}$ deviates further from the ideal case $\mathbf{P} = \mathbf{11}^T$. In this case, initial solution $\mathbf{W} = \frac{1}{M}\mathbf{11}^T$ is far from optimal and leads to a longer convergence time.

D. Performance under Different Settings

In this subsection, we investigate the proposed algorithm under different system configurations.

Fig. 6 and Fig. 7 plot the average test accuracy and the minimum test accuracy, respectively. In these figures, we set $r = 2$ and change the value of v . From these figures, the test accuracy decreases with the increase of parameter v while and the accuracy gap is larger with respect to minimum accuracy. The reason is that optimized second largest (in magnitude) eigenvalue $\rho(\bar{\mathbf{W}})$ decreases (from 0.54 when $v = 2$ to 0.15 when $v = 10$) as the value of v increases.

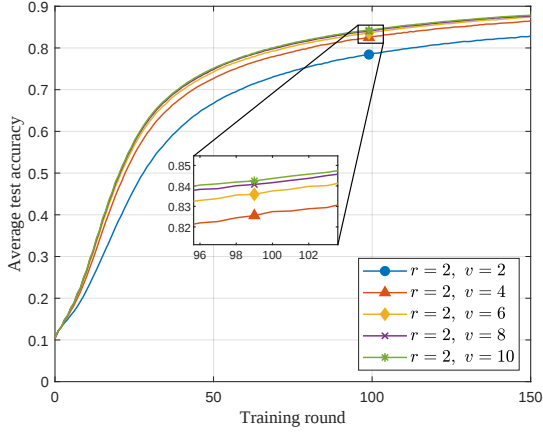


Fig. 6. Average test accuracy versus training round.

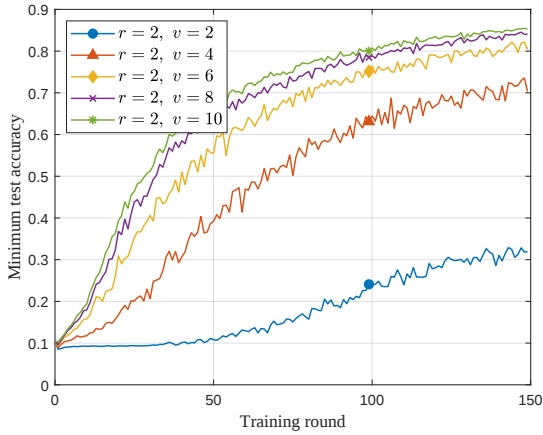


Fig. 7. Minimum test accuracy versus training round.

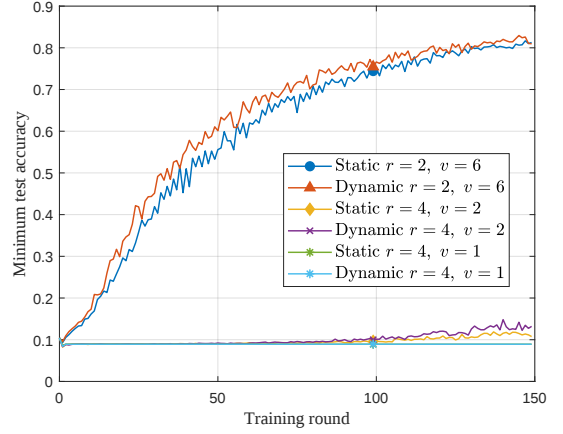


Fig. 8. Average test accuracy versus training round.

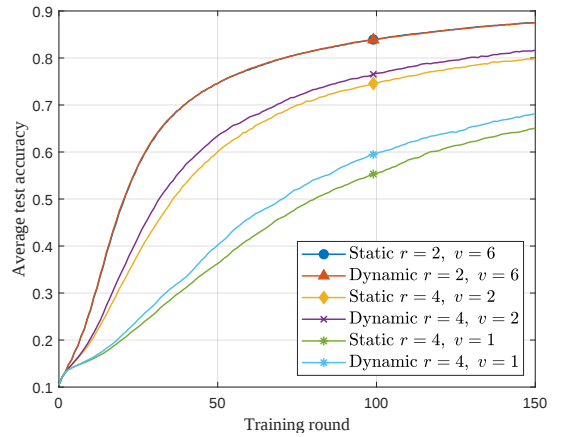


Fig. 9. Minimum test accuracy versus training round.

Fig. 8 and 9 compare the performance of the decentralized federated learning (DFL) system under static and dynamic link reliability with varying r and v . In the static scenario, user positions and link qualities remain constant throughout the training process, requiring a single optimization of aggregation weights. In contrast, the dynamic scenario involves random user position updates in each round, leading to varying link qualities and dynamic re-optimization of aggregation weights. The left figure shows minimum test accuracy, while the right panel presents average test accuracy. Dynamic link conditions consistently improve both metrics compared to static links. This is because randomizing user positions prevents any device from being persistently subjected to poor channel conditions, balancing link reliability across devices over time. As a result, the dynamic scenario achieves better model aggregation and learning performance, demonstrating the method's robustness to real-world network dynamics.

E. Performance Comparison in Complex System

In this subsection, we compare the proposed scheme with existing state-of-the-art schemes under a large-scale complex system. Specifically, we set the link reliability factor, $r = 4, v = 2$, and the device number $M = 200$. We conduct DFL training on the challenging FashionMNIST dataset [43].

Other settings are the same as that given in Section V-A. The followings are the benchmarks used in comparison.

- **Weight design via centralized optimization (WD via CO) [21]:** Assume there is a central entity in the system that can collect the link reliability information, i.e., matrix $\mathbf{P}$, from all devices. This central entity optimizes the aggregation weights by solving (10) and distributes the results to the corresponding edge devices.
- **Equal weight design without reliability consideration (EW w/o RC) [44]:** We use the average aggregation weights for each device, i.e., $\mathbf{W} = \frac{1}{M} \mathbf{1} \mathbf{1}^T$. Since this setting consider no link reliability $\mathbf{P}$, the resulting $\bar{\mathbf{W}}$ does not minimize $\rho(\bar{\mathbf{W}})$ in general.
- **Weight design with reliable communication (WD with RC):** We set all the communication links are reliable, i.e., $\mathbf{P} = \mathbf{1} \mathbf{1}^T$. In this case, the optimal aggregation weights $\mathbf{W}$ is $\frac{1}{M} \mathbf{1} \mathbf{1}^T$ since this $\mathbf{W}$ makes $\rho(\bar{\mathbf{W}}) = 0$.
- **Weight design via Metropolis-Hastings algorithm (WD via MH) [45]:** Following the idea of the Metropolis-Hastings (MH) algorithm [45], we construct a symmetric and stochastic aggregation matrix based on the link reliability, p_{ij} . Specifically, the aggregation weights

are defined as

$$w_{ij} = \begin{cases} \frac{p_{ij}}{\max\{d_i, d_j\}}, & i \neq j, \\ 1 - \sum_{k \neq i} w_{ik}, & i = j, \end{cases} \quad (27)$$

where the weighted degree of device i is given by $d_i = \sum_k p_{ik}$. This design follows the principle of the MH algorithm, ensuring both symmetry ($w_{ij} = w_{ji}$) and stochasticity ($\sum_j w_{ij} = 1$), while continuously reflecting the reliability of communication links in the aggregation process.

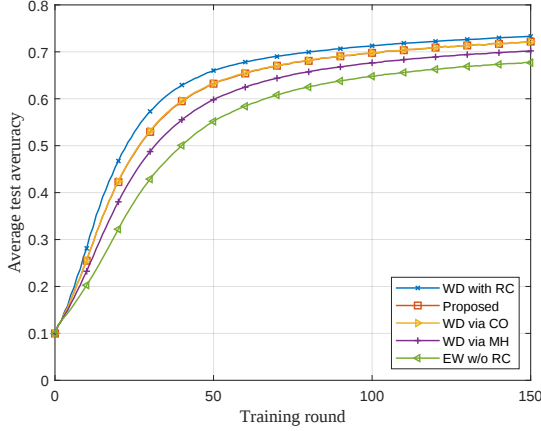


Fig. 10. Average test accuracy of different schemes.

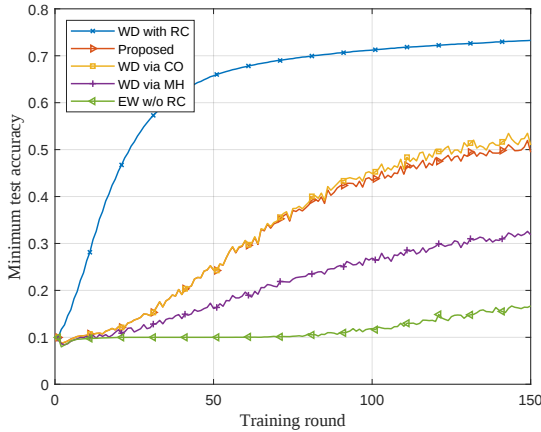


Fig. 11. Minimum test accuracy of different schemes.

The average test accuracy is plotted in Fig. 10 and minimum test accuracy is in Fig. 11. From the figures, the WD with RC scheme exhibits the best learning performance because we set all communication links reliable in the WD with RC and in this case the WD with RC scheme chooses optimal aggregation weights $\mathbf{W} = \frac{1}{M} \mathbf{1}\mathbf{1}^T$ such that $\rho(\bar{\mathbf{W}}) = 0$. Moreover, the performance of the proposed algorithm is almost the same as that of WD via CO defined before in both figures and similar to optimal performance given by the WD with RC. This demonstrates that the proposed distributed subgradient optimization algorithm can achieve the same performance as the centralized method that is under the assistance of a central entity. This observation can be explained by our

convergence analysis. Since the optimization problem (10) is convex, the centralized method attains the globally optimal aggregation weights. Meanwhile, the projected subgradient algorithm is proven to converge to a neighborhood of the optimum (Section IV-F). Therefore, the proposed distributed realization naturally achieves performance comparable to the centralized one. Furthermore, the performance of the WD via WH scheme lags far behind the proposed algorithm because WD via WH heuristically designs aggregation weights based on the topology formed by the link reliability p_{ij} , which does not achieve minimal $\rho(\bar{\mathbf{W}})$. Finally, the EW w/o RC scheme shows the worst learning performance because this design completely ignores the link reliability.

VI. CONCLUSIONS

In this paper, we investigated distributed aggregation weight optimization for DFL. We derived an ergodic convergence bound by capturing the impact of aggregation weights on the learning performance over communication networks. Based on this, we formulated a weight optimization problem to minimize the convergence bound. We proposed a distributed subgradient algorithm to solve this problem. In this way, we established a completely distributed DFL system, where optimization, communication, and learning processes are all distributed. Based on our simulation results, the proposed algorithm reached the performance of the centralized method. Future work may consider practical factors to balance performance and efficiency.

APPENDIX A PROOF OF PROPOSITION 1

We assume $\lambda \leq \frac{1}{\omega}$. Begin with $f\left(\frac{\mathbf{X}^{(t+1)}\mathbf{1}}{M}\right)$

$$\begin{aligned} & \mathbb{E} f\left(\frac{\mathbf{X}^{(t+1)}\mathbf{1}}{M}\right) \\ &= \mathbb{E} f\left(\frac{\mathbf{X}^{(t)}\widehat{\mathbf{W}}^{(t)}\mathbf{1}}{M} - \lambda \frac{\partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)})\mathbf{1}}{M}\right) \\ &\stackrel{(a)}{\leq} \mathbb{E} f\left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M}\right) - \lambda \mathbb{E} \left\langle \nabla f\left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M}\right), \frac{\partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)})\mathbf{1}}{M} \right\rangle \\ &\quad + \frac{\omega\lambda^2}{2} \mathbb{E} \left\| \frac{\partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)})\mathbf{1}}{M} \right\|^2 \\ &\stackrel{(b)}{=} \mathbb{E} f\left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M}\right) - \frac{\lambda}{2} \mathbb{E} \left\| \nabla f\left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M}\right) \right\|^2 \\ &\quad + \frac{\lambda}{2} \mathbb{E} \left\| \nabla f\left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M}\right) - \frac{\partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)})\mathbf{1}}{M} \right\|^2 \\ &\quad - \frac{\lambda}{2} \mathbb{E} \left\| \frac{\partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)})\mathbf{1}}{M} \right\|^2 + \frac{\omega\lambda^2}{2} \mathbb{E} \left\| \frac{\partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)})\mathbf{1}}{M} \right\|^2 \\ &\stackrel{(c)}{\leq} \mathbb{E} f\left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M}\right) - \frac{\lambda}{2} \mathbb{E} \left\| \nabla f\left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M}\right) \right\|^2 \\ &\quad + \frac{\lambda}{2} \mathbb{E} \left\| \nabla f\left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M}\right) - \frac{\partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)})\mathbf{1}}{M} \right\|^2, \end{aligned}$$

where (a) is due to ω -smoothness and $\bar{\mathbf{W}}\mathbf{1} = \mathbf{1}$; (b) is due to $2\langle a, b \rangle = \|a\|^2 + \|b\|^2 - \|a - b\|^2$; (c) is due to the Assumption $\lambda \leq \frac{1}{\omega}$. Then, we turn to bound the last term in the above inequality as follows.

$$\begin{aligned}
& \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M} \right) - \frac{\partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)})\mathbf{1}}{M} \right\|^2 \\
&= \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M} \right) - \frac{\partial f(\mathbf{X}^{(t)})\mathbf{1}}{M} \right. \\
&\quad \left. - \frac{\partial f(\mathbf{X}^{(t)})\mathbf{1}}{M} + \frac{\partial f(\mathbf{X}^{(t)})\mathbf{1}}{M} \right\|^2 \\
&= \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M} \right) - \frac{\partial f(\mathbf{X}^{(t)})\mathbf{1}}{M} \right\|^2 \\
&\quad + \mathbb{E} \left\| \frac{\partial f(\mathbf{X}^{(t)})\mathbf{1}}{M} - \frac{\partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)})\mathbf{1}}{M} \right\|^2 \\
&\quad + 2\mathbb{E} \left\langle \nabla f \left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M} \right) - \frac{\partial f(\mathbf{X}^{(t)})\mathbf{1}}{M}, \right. \\
&\quad \left. \frac{\partial f(\mathbf{X}^{(t)})\mathbf{1}}{M} - \frac{\partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)})\mathbf{1}}{M} \right\rangle \\
&= \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M} \right) - \frac{\partial f(\mathbf{X}^{(t)})\mathbf{1}}{M} \right\|^2 \\
&\quad + \mathbb{E} \left\| \frac{\partial f(\mathbf{X}^{(t)})\mathbf{1}}{M} - \frac{\partial F(\mathbf{X}^{(t)}, \boldsymbol{\xi}^{(t)})\mathbf{1}}{M} \right\|^2 \\
&\leq \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M} \right) - \frac{\partial f(\mathbf{X}^{(t)})\mathbf{1}}{M} \right\|^2 + \frac{\alpha^2}{M},
\end{aligned}$$

where the last inequality is obtained from Assumption 5. We then bound the second last term on the right hand side (RHS) of the above inequality as follows.

$$\begin{aligned}
& \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M} \right) - \frac{\partial f(\mathbf{X}^{(t)})\mathbf{1}}{M} \right\|^2 \\
&= \mathbb{E} \left\| \frac{1}{M} \sum_{i=1}^M \left(\nabla f_i \left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M} \right) - \nabla f_i(\mathbf{X}^{(t)}\mathbf{e}_i) \right) \right\|^2 \\
&\leq \frac{1}{M} \sum_{i=1}^M \mathbb{E} \left\| \nabla f_i \left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M} \right) - \nabla f_i(\mathbf{X}^{(t)}\mathbf{e}_i) \right\|^2 \\
&\leq \frac{\omega^2}{M} \sum_{i=1}^M \mathbb{E} \left\| \frac{\mathbf{X}^{(t)}\mathbf{1}}{M} - \mathbf{X}^{(t)}\mathbf{e}_i \right\|^2,
\end{aligned}$$

where $\mathbf{e}_i$ is the vector with appropriate dimensions and only the i -th element of $\mathbf{e}_i$ is 1 while all others are 0, and the last inequality is due to the ω -smoothness. We call $\frac{1}{M} \sum_{i=1}^M \mathbb{E} \left\| \frac{\mathbf{X}^{(t)}\mathbf{1}}{M} - \mathbf{X}^{(t)}\mathbf{e}_i \right\|^2$ as the consensus error at the t -th round.

Till now, it can be concluded that

$$\begin{aligned}
& \mathbb{E} f \left(\frac{\mathbf{X}^{(t+1)}\mathbf{1}}{M} \right) \leq \mathbb{E} f \left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M} \right) - \frac{\lambda}{2} \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(t)}\mathbf{1}}{M} \right) \right\|^2 \\
&\quad + \frac{\lambda\omega^2}{2M} \sum_{i=1}^M \mathbb{E} \left\| \frac{\mathbf{X}^{(t)}\mathbf{1}}{M} - \mathbf{X}^{(t)}\mathbf{e}_i \right\|^2 + \frac{\lambda\alpha^2}{2M}. \tag{28}
\end{aligned}$$

We proceed with bounding $\Xi_i^{(t)} = \mathbb{E} \left\| \frac{\mathbf{X}^{(t)}\mathbf{1}}{M} - \mathbf{X}^{(t)}\mathbf{e}_i \right\|^2$.

$$\begin{aligned}
\Xi_i^{(t)} &= \mathbb{E} \left\| \frac{1}{M} \left(\mathbf{X}^{(t-1)}\widehat{\mathbf{W}}^{(t-1)}\mathbf{1} - \lambda \left(\partial F(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t-1)}) \right) \mathbf{1} \right) \right. \\
&\quad \left. - \left(\mathbf{X}^{(t-1)}\widehat{\mathbf{W}}^{(t-1)}\mathbf{e}_i - \lambda \left(\partial F(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t-1)}) \right) \mathbf{e}_i \right) \right\|^2 \\
&= \mathbb{E} \left\| \frac{1}{M} \left(\mathbf{X}^{(0)}\mathbf{1} - \sum_{i=0}^{t-1} \lambda \left(\partial F(\mathbf{X}^{(i)}, \boldsymbol{\xi}^{(i)}) \right) \mathbf{1} \right) - \right. \\
&\quad \left(\mathbf{X}^{(0)} \prod_{m=0}^{(t-1)} \widehat{\mathbf{W}}^{(m)}\mathbf{e}_i - \sum_{j=0}^{t-1} \lambda \left(\partial F(\mathbf{X}^{(j)}, \boldsymbol{\xi}^{(j)}) \right) \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)}\mathbf{e}_i \right) \right\|^2 \\
&= \mathbb{E} \left\| \mathbf{X}^{(0)} \left(\frac{1}{M} - \prod_{m=0}^{(t-1)} \widehat{\mathbf{W}}^{(m)}\mathbf{e}_i \right) - \right. \\
&\quad \left. \sum_{j=0}^{t-1} \lambda \left(\partial F(\mathbf{X}^{(j)}, \boldsymbol{\xi}^{(j)}) \right) \left(\frac{1}{M} - \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)}\mathbf{e}_i \right) \right\|^2 \\
&= \mathbb{E} \left\| \sum_{j=0}^{t-1} \lambda \left(\partial F(\mathbf{X}^{(j)}, \boldsymbol{\xi}^{(j)}) \right) \left(\frac{1}{M} - \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)}\mathbf{e}_i \right) \right\|^2 \\
&= \lambda^2 \mathbb{E} \left\| \sum_{j=0}^{t-1} \left(\partial F(\mathbf{X}^{(j)}, \boldsymbol{\xi}^{(j)}) - \partial f(\mathbf{X}^{(j)}) + \partial f(\mathbf{X}^{(j)}) \right) \right. \\
&\quad \left. \left(\frac{1}{M} - \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)}\mathbf{e}_i \right) \right\|^2 \\
&\leq 2\lambda^2 \mathbb{E} \left\| \sum_{j=0}^{t-1} \partial f(\mathbf{X}^{(j)}) \left(\frac{1}{M} - \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)}\mathbf{e}_i \right) \right\|^2 + \\
&\quad 2\lambda^2 \mathbb{E} \left\| \sum_{j=0}^{t-1} \left(\partial F(\mathbf{X}^{(j)}, \boldsymbol{\xi}^{(j)}) - \partial f(\mathbf{X}^{(j)}) \right) \left(\frac{1}{M} - \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)}\mathbf{e}_i \right) \right\|^2, \tag{29}
\end{aligned}$$

where we use $\mathbf{X}^{(0)} = 0$. We now bound the first term on the RHS of inequality (29).

$$\begin{aligned}
& \mathbb{E} \left\| \sum_{j=0}^{t-1} \left(\partial F(\mathbf{X}^{(j)}, \boldsymbol{\xi}^{(j)}) - \partial f(\mathbf{X}^{(j)}) \right) \left(\frac{1}{M} - \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)}\mathbf{e}_i \right) \right\|^2 \\
&\leq \sum_{j=0}^{t-1} \mathbb{E} \left\| \left(\partial F(\mathbf{X}^{(j)}, \boldsymbol{\xi}^{(j)}) - \partial f(\mathbf{X}^{(j)}) \right) \right\|^2 \\
&\quad \left\| \left(\frac{1}{M} - \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)}\mathbf{e}_i \right) \right\|^2 \\
&\leq \sum_{j=0}^{t-1} \mathbb{E} \left\| \left(\partial F(\mathbf{X}^{(j)}, \boldsymbol{\xi}^{(j)}) - \partial f(\mathbf{X}^{(j)}) \right) \right\|_F^2 \\
&\quad \left\| \left(\frac{1}{M} - \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)}\mathbf{e}_i \right) \right\|^2
\end{aligned}$$

$$\begin{aligned}
&\stackrel{(a)}{\leq} M\alpha^2 \sum_{j=0}^{t-1} \mathbb{E} \left\| \left(\frac{\mathbf{1}\mathbf{1}^T}{M} - \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)} \mathbf{e}_i \right) \right\|^2 \\
&\stackrel{(b)}{=} M\alpha^2 \sum_{j=0}^{t-1} \mathbb{E} \left\| \prod_{m=j+1}^{(t-1)} \left(\frac{\mathbf{1}\mathbf{1}^T}{M} - \widehat{\mathbf{W}}^{(m)} \right) \mathbf{e}_i \right\|^2,
\end{aligned}$$

where (a) is from Assumption 5, and (b) is from the fact that $\widehat{\mathbf{W}}^{(m)}$ is a doubly stochastic matrix. We then bound

$$\begin{aligned}
&\mathbb{E} \left\| \prod_{m=j+1}^{(t-1)} \left(\frac{\mathbf{1}\mathbf{1}^T}{M} - \widehat{\mathbf{W}}^{(m)} \right) \mathbf{e}_i \right\|^2 \\
&= \mathbb{E} \left\{ \mathbf{e}_i^T \left(\prod_{m=t-1}^{(j+1)} \left(\frac{\mathbf{1}\mathbf{1}^T}{M} - \widehat{\mathbf{W}}^{(m)} \right)^T \right) \right. \\
&\quad \left. \left(\prod_{m=j+1}^{(t-1)} \left(\frac{\mathbf{1}\mathbf{1}^T}{M} - \widehat{\mathbf{W}}^{(m)} \right) \mathbf{e}_i \right) \right\} \\
&= \mathbb{E} \left\{ \mathbf{e}_i^T \left(\prod_{m=t-1}^{(j+2)} \left(\frac{\mathbf{1}\mathbf{1}^T}{M} - \widehat{\mathbf{W}}^{(m)} \right)^T \right) \right. \\
&\quad \left. \mathbb{E} \left[\left(\frac{\mathbf{1}\mathbf{1}^T}{M} - \widehat{\mathbf{W}}^{(j+1)} \right)^T \left(\frac{\mathbf{1}\mathbf{1}^T}{M} - \widehat{\mathbf{W}}^{(j+1)} \right) \right] \right. \\
&\quad \left. \left(\prod_{m=j+2}^{(t-1)} \left(\frac{\mathbf{1}\mathbf{1}^T}{M} - \widehat{\mathbf{W}}^{(m)} \right) \right) \mathbf{e}_i \right\} \\
&= \mathbb{E} \left\{ \mathbf{e}_i^T \left(\prod_{m=t-1}^{(j+2)} \left(\frac{\mathbf{1}\mathbf{1}^T}{M} - \widehat{\mathbf{W}}^{(m)} \right)^T \right) \right. \\
&\quad \left. \left(\overline{\mathbf{W}}^2 - \frac{\mathbf{1}\mathbf{1}^T}{M} \right) \left(\prod_{m=j+2}^{(t-1)} \left(\frac{\mathbf{1}\mathbf{1}^T}{M} - \widehat{\mathbf{W}}^{(m)} \right) \right) \mathbf{e}_i \right\} \\
&\stackrel{(a)}{\leq} \lambda_1 \left(\overline{\mathbf{W}}^2 - \frac{\mathbf{1}\mathbf{1}^T}{M} \right) \mathbb{E} \left\{ \mathbf{e}_i^T \left(\prod_{m=t-1}^{(j+2)} \left(\frac{\mathbf{1}\mathbf{1}^T}{M} - \widehat{\mathbf{W}}^{(m)} \right)^T \right) \right. \\
&\quad \left. \left(\prod_{m=j+2}^{(t-1)} \left(\frac{\mathbf{1}\mathbf{1}^T}{M} - \widehat{\mathbf{W}}^{(m)} \right) \right) \mathbf{e}_i \right\} \\
&\leq \left(\lambda_1 \left(\overline{\mathbf{W}}^2 - \frac{\mathbf{1}\mathbf{1}^T}{M} \right) \right)^{t-j-1} \\
&\stackrel{(b)}{=} \left(\lambda_1 \left(\overline{\mathbf{W}}^2 - \left(\overline{\mathbf{W}}^2 \right)^\infty \right) \right)^{t-j-1} \\
&\leq \left(\rho \left(\overline{\mathbf{W}}^2 \right) \right)^{t-j-1}, \tag{30}
\end{aligned}$$

where (a) is from the Rayleigh quotient inequality, and (b) is from the fact that $\overline{\mathbf{W}}^2$ is a doubly stochastic matrix. By applying the analysis in [46], the second term on the RHS of (29) can be eventually bounded as

$$\mathbb{E} \left\| \sum_{j=0}^{t-1} \partial f(\mathbf{X}^{(j)}) \left(\frac{\mathbf{1}}{M} - \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)} \mathbf{e}_i \right) \right\|^2$$

$$\begin{aligned}
&\leq 3 \sum_{j=0}^{t-1} \sum_{h=1}^M \mathbb{E} \omega^2 \Xi_h^{(j)} \left\| \left(\frac{\mathbf{1}}{M} - \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)} \mathbf{e}_i \right) \right\|^2 \\
&+ 6 \sum_{j=0}^{t-1} \left(\sum_{h=1}^M \mathbb{E} \omega^2 \Xi_h^{(j)} + \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(j)} \mathbf{1}}{M} \right) \mathbf{1}^\top \right\|^2 \right) \\
&\times \frac{\sqrt{\rho(\overline{\mathbf{W}}^2)}^{k-j-1}}{1 - \sqrt{\rho(\overline{\mathbf{W}}^2)}} + \frac{9n\beta^2}{(1 - \sqrt{\rho(\overline{\mathbf{W}}^2)})^2} \\
&+ 3 \sum_{j=0}^{t-1} \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(j)} \mathbf{1}}{M} \right) \mathbf{1}^\top \right\|^2 \left\| \left(\frac{\mathbf{1}}{M} - \mathbf{W}^{t-j-1} \mathbf{e}_i \right) \right\|^2.
\end{aligned}$$

Now we are able to bound $\Xi_i^{(t)}$.

$$\begin{aligned}
\Xi_i^{(t)} &\leq 12\lambda^2 \sum_{j=0}^{t-1} \left(\sum_{h=1}^M \omega^2 \mathbb{E} \Xi_h^{(j)} + \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(j)} \mathbf{1}}{M} \right) \mathbf{1}^\top \right\|^2 \right) \\
&\times \frac{\sqrt{\rho(\overline{\mathbf{W}}^2)}^{k-j-1}}{1 - \sqrt{\rho(\overline{\mathbf{W}}^2)}} + \frac{2\lambda^2 M\alpha^2}{1 - \rho(\overline{\mathbf{W}}^2)} + \frac{18\lambda^2 M\beta^2}{(1 - \sqrt{\rho(\overline{\mathbf{W}}^2)})^2} \\
&+ 6\lambda^2 \sum_{j=0}^{t-1} \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(j)} \mathbf{1}}{M} \right) \mathbf{1}^\top \right\|^2 \left\| \left(\frac{\mathbf{1}}{M} - \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)} \mathbf{e}_i \right) \right\|^2 \\
&+ 6\lambda^2 \omega^2 \sum_{j=0}^{t-1} \sum_{h=1}^M \mathbb{E} \Xi_h^{(j)} \left\| \left(\frac{\mathbf{1}}{M} - \prod_{m=j+1}^{(t-1)} \widehat{\mathbf{W}}^{(m)} \mathbf{e}_i \right) \right\|^2 \\
&\leq \frac{2\lambda^2 M\alpha^2}{1 - \rho(\overline{\mathbf{W}}^2)} + \frac{18\lambda^2 M\beta^2}{(1 - \sqrt{\rho(\overline{\mathbf{W}}^2)})^2} \\
&+ 6\lambda^2 \sum_{j=0}^{t-1} \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(j)} \mathbf{1}}{M} \right) \mathbf{1}^\top \right\|^2 \Omega \left(\rho(\overline{\mathbf{W}}^2) \right) \\
&+ 6\lambda^2 \omega^2 \sum_{j=0}^{t-1} \sum_{h=1}^M \mathbb{E} \Xi_h^{(j)} \Omega \left(\rho(\overline{\mathbf{W}}^2) \right),
\end{aligned}$$

where

$$\Omega \left(\rho(\overline{\mathbf{W}}^2) \right) = \left(\rho(\overline{\mathbf{W}}^2) \right)^{k-j-1} + \frac{2\sqrt{\rho(\overline{\mathbf{W}}^2)}^{k-j-1}}{1 - \sqrt{\rho(\overline{\mathbf{W}}^2)}}.$$

We next bound the consensus error $\frac{1}{M} \sum_{i=1}^M \Xi_i^{(t)}$ as

$$\begin{aligned}
&\frac{1}{M} \sum_{i=1}^M \mathbb{E} \left\| \frac{\mathbf{X}^{(t)} \mathbf{1}}{M} - \mathbf{X}^{(t)} \mathbf{e}_i \right\|^2 \leq \frac{2\lambda^2 M\alpha^2}{1 - \rho(\overline{\mathbf{W}}^2)} + \frac{18\lambda^2 M\beta^2}{(1 - \sqrt{\rho(\overline{\mathbf{W}}^2)})^2} \\
&+ 6\lambda^2 \sum_{j=0}^{t-1} \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(j)} \mathbf{1}}{M} \right) \mathbf{1}^\top \right\|^2 \Omega \left(\rho(\overline{\mathbf{W}}^2) \right) \\
&+ 6\lambda^2 \omega^2 \sum_{j=0}^{t-1} \sum_{i=1}^M \mathbb{E} \left\| \frac{\mathbf{X}^{(j)} \mathbf{1}}{M} - \mathbf{X}^{(j)} \mathbf{e}_i \right\|^2 \Omega \left(\rho(\overline{\mathbf{W}}^2) \right).
\end{aligned}$$

The consensus error appears on the both sides of the above inequality. By summing the above inequality from $t = 0$ to $T - 1$, rearranging the summation, and taking relaxation, the

overall bound for the consensus error is

$$\begin{aligned} & \frac{1}{M} \sum_{t=0}^{T-1} \sum_{i=1}^M \mathbb{E} \left\| \frac{\mathbf{X}^{(t)} \mathbf{1}}{M} - \mathbf{X}^{(t)} \mathbf{e}_i \right\|^2 \\ & \leq \frac{2\lambda^2}{(1 - \rho(\overline{\mathbf{W}}^2)) \left(1 - \frac{18M\lambda^2\omega^2}{(1 - \sqrt{\rho(\overline{\mathbf{W}}^2)})^2}\right)} \\ & \times \left(M\alpha^2 T + 9M\beta^2 T + 9 \sum_{t=0}^{T-1} \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(t)} \mathbf{1}}{M} \right) \mathbf{1}^\top \right\|^2 \right). \end{aligned} \quad (31)$$

Summing the inequality (28) from $t = 0$ to $t = T - 1$ while applying (31), we obtain

$$\begin{aligned} & \frac{1}{2} \sum_{t=0}^{T-1} \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(t)} \mathbf{1}}{M} \right) \right\|^2 \leq \frac{f_0 - f^*}{\lambda} + \frac{\lambda\omega\alpha^2 T}{2M} \\ & + \left(\alpha^2 T + 9\beta^2 T + 9 \sum_{t=0}^{T-1} \mathbb{E} \left\| \nabla f \left(\frac{\mathbf{X}^{(t)} \mathbf{1}}{M} \right) \right\|^2 \right) G(\overline{\mathbf{W}}^2), \end{aligned}$$

where $G(\overline{\mathbf{W}}^2) = \frac{M\lambda^2\omega^2}{(1 - \sqrt{\rho(\overline{\mathbf{W}}^2)})^2 - 18M\lambda^2\omega^2}$. Rearranging the above inequality, we obtain the final convergence bound given in Theorem 1.

APPENDIX B PROOF OF PROPOSITION 2

For the second order statistic $\overline{\mathbf{W}}^2$, we have

$$\overline{\mathbf{W}}^2 = (\overline{\mathbf{W}})^2 + \widetilde{\mathbf{W}}, \quad (32)$$

where $\widetilde{\mathbf{W}} = \overline{\mathbf{W}}^2 - (\overline{\mathbf{W}})^2$ has a variance-like form. Suppose $\rho(\overline{\mathbf{W}}^2) = \lambda_2(\overline{\mathbf{W}}^2)$, since $\overline{\mathbf{W}}^2$ is a real symmetric matrix, from the spectral stability corollary of Weyl's inequality [47], we have

$$\lambda_2(\overline{\mathbf{W}}^2) = \lambda_2((\overline{\mathbf{W}})^2 + \widetilde{\mathbf{W}}) \leq \lambda_2((\overline{\mathbf{W}})^2) + \lambda_1(\widetilde{\mathbf{W}}). \quad (33)$$

Here, we see $\lambda_2((\overline{\mathbf{W}})^2) + \lambda_1(\widetilde{\mathbf{W}})$ is an upper bound of $\rho(\overline{\mathbf{W}}^2)$. So $\lambda_2((\overline{\mathbf{W}})^2) + \lambda_1(\widetilde{\mathbf{W}})$ can be an objective function of the minimization problem of $\rho(\overline{\mathbf{W}}^2)$.

Note that $\widetilde{\mathbf{W}} = \overline{\mathbf{W}}^2 - (\overline{\mathbf{W}})^2 = \mathbb{E}\{(\widehat{\mathbf{W}}^{(t)})^\top (\widehat{\mathbf{W}}^{(t)})\} - \mathbb{E}\{(\widehat{\mathbf{W}}^{(t)})\}^2 = \mathbb{E}\{(\widehat{\mathbf{W}}^{(t)})^2\} - \mathbb{E}\{(\widehat{\mathbf{W}}^{(t)})\}^2$, which is associated with the first-order and second-order statistics of variable $\widehat{\mathbf{W}}^{(t)}$. The expression of $\widetilde{\mathbf{W}}$ is quite similar to the autocovariance of a random variable. For the (i, j) -th element of $\widetilde{\mathbf{W}}$, we have

$$\widetilde{w}_{ij} = \sum_{k=1}^M \left[\mathbb{E} \left\{ \widehat{w}_{ki}^{(t)} \widehat{w}_{kj}^{(t)} \right\} - \mathbb{E} \left\{ \widehat{w}_{ki}^{(t)} \right\} \mathbb{E} \left\{ \widehat{w}_{kj}^{(t)} \right\} \right]. \quad (34)$$

We see that the (i, j) -th element of $\widetilde{\mathbf{W}}$ is the sum of the covariance of the corresponding elements between the i -th and j -th columns of matrix $\widehat{\mathbf{W}}^{(t)}$. Since the reliability of different links is independent (from Assumption 2), we have

$$\widetilde{w}_{ij} = 2w_{ij}^2 (p_{ij}^2 - p_{ij}), \quad \forall i \neq j, \quad (35)$$

$$\widetilde{w}_{ii} = 2 \sum_{k=1}^M w_{ki}^2 (p_{ki} - p_{ki}^2), \quad \forall i. \quad (36)$$

Since $|\widetilde{w}_{ii}| \geq \sum_{j=1, j \neq i}^M |\widetilde{w}_{ij}|$, $\widetilde{\mathbf{W}}$ is a diagonally dominant matrix. The eigenvalues of $\widetilde{\mathbf{W}}$ can be estimated as its diagonal elements and the largest eigenvalue $\lambda_1(\widetilde{\mathbf{W}}) \approx \max\{\widetilde{w}_{ii} \mid \forall i\}$. Since $0 \leq p_{ij} \leq 1, \forall i, j$, we have $0 \leq \widetilde{w}_{ii} \leq \frac{1}{2} \sum_{i=1}^M w_{ki}^2, \forall i$.

In the large-scale systems where the number of devices approaches infinity, there cannot not be a model of specific device which dominates the each aggregation process. If each model shares the equal weight in aggregation, i.e., $w_{ki} = \frac{1}{M}, \forall k, i$, we have $\lim_{M \rightarrow \infty} \sum_{j=1}^M w_{ki}^2 = \frac{1}{M} = 0, \forall i$ and hence $\widetilde{w}_{ii} = 0, \forall i$. Combing the above analysis, we can prove that $\lambda_1(\widetilde{\mathbf{W}}) \approx 0$ and $\lambda_2(\overline{\mathbf{W}}^2) \leq \lambda_2((\overline{\mathbf{W}})^2)$.

Similarly, we can also prove $\rho(\overline{\mathbf{W}}^2) \leq \lambda_2((\overline{\mathbf{W}})^2)$ for the case $\rho(\overline{\mathbf{W}}^2) = -\lambda_M(\overline{\mathbf{W}}^2)$. Therefore, $\lambda_2((\overline{\mathbf{W}})^2)$ is an upper bound of $\rho(\overline{\mathbf{W}}^2)$. Furthermore, since $\lambda_2((\overline{\mathbf{W}})^2) = (\max\{\lambda_2(\overline{\mathbf{W}}), -\lambda_M(\overline{\mathbf{W}})\})^2$, we can consider $\rho(\overline{\mathbf{W}}) = \max\{\lambda_2(\overline{\mathbf{W}}), -\lambda_M(\overline{\mathbf{W}})\}$ as a simplified upper bound and turn to use it as the surrogate objective function in the optimization.

APPENDIX C PROOF OF PROPOSITION 4

We establish the convergence of Algorithm 2 following the framework of approximate subgradient methods [38]. Given the feasible set $\mathcal{C}$ defined in (10b), Algorithm 2 performs the following update:

$$\begin{aligned} \mathbf{W}(n+1) &= \Pi_{\mathcal{C}}(\mathbf{W}(n) - \gamma_n g(\mathbf{W}(n))), \\ g(\mathbf{W}(n)) &\in \partial_{\epsilon_n} f(\mathbf{W}(n)), \end{aligned} \quad (37)$$

where $\Pi_{\mathcal{C}}$ denotes the projection onto $\mathcal{C}$, and $\partial_{\epsilon_n} f(\mathbf{W}(n))$ is the ϵ_n -subdifferential defined by

$$\begin{aligned} \partial_{\epsilon_n} f(\mathbf{W}(n)) &= \left\{ g : f(\mathbf{W}) \geq f(\mathbf{W}(n)) + \langle g, \mathbf{W} - \mathbf{W}(n) \rangle - \epsilon_n, \forall \mathbf{W} \in \mathcal{C} \right\}. \end{aligned}$$

Let $\eta_n = \frac{1}{2} \|g(\mathbf{W}(n))\|_F^2 \gamma_n$ and $\delta_n = \eta_n + \epsilon_n$. From [38], if $\sum_n \gamma_n = \infty$, then

$$\liminf_n f(\mathbf{W}(n)) \leq f^* + \delta, \quad \delta = \limsup_n \delta_n. \quad (38)$$

At iteration n , the eigenvector $\mathbf{v}(n)$ computed by Algorithm 1 approximates the exact eigenvector $\mathbf{v}_r(n)$ associated with $\rho(\overline{\mathbf{W}})$ and satisfies $\|\mathbf{v}(n) - \mathbf{v}_r(n)\|_2^2 \leq \epsilon_n$. Let $g_r(\mathbf{W}(n))$ denote the exact subgradient derived from $\mathbf{v}_r(n)$ and $g(\mathbf{W}(n))$ the one used in the algorithm. For any feasible $\mathbf{W} \in \mathcal{C}$,

$$\begin{aligned} \lambda_2(\mathbf{W}) &\geq \lambda_2(\mathbf{W}(n)) + \langle g_r, \mathbf{W} - \mathbf{W}(n) \rangle \\ &= \lambda_2(\mathbf{W}(n)) + \langle g, \mathbf{W} - \mathbf{W}(n) \rangle - \langle g - g_r, \mathbf{W} - \mathbf{W}(n) \rangle. \end{aligned}$$

Hence the inexactness term satisfies

$$\epsilon_n = \sup_{\mathbf{W} \in \mathcal{C}} \langle g - g_r, \mathbf{W} - \mathbf{W}(n) \rangle = c \|g - g_r\|_F^2, \quad (39)$$

where $c > 0$ is a scaling constant.

The (i, j) -th element of $g - g_r$ is

$$\begin{aligned} (g - g_r)_{ij} &= p_{ij} [(v_i - v_j)^2 - (v_{r,i} - v_{r,j})^2] \\ &= p_{ij} [(v_i - v_{r,i} - v_j + v_{r,j})(v_i - v_j + v_{r,i} - v_{r,j})]. \end{aligned}$$

Since $|v_i - v_{r,i}| \leq \sqrt{\varepsilon_n}$ for all i and the eigenvectors are normalized ($\|v\|_2 = 1$), one has $|v_i - v_j| \leq \sqrt{2}$, which yields

$$(g - g_r)_{ij}^2 \leq 32 p_{ij}^2 \varepsilon_n. \quad (40)$$

Summing over all existing links gives

$$\|g - g_r\|_F^2 \leq 32 \varepsilon_n \sum_{(i,j)} p_{ij}^2. \quad (41)$$

Substituting (41) into (39) yields

$$\epsilon_n \leq 32c \varepsilon_n \sum_{(i,j)} p_{ij}^2 = \mathcal{O}(\varepsilon_n). \quad (42)$$

Now we choose a diminishing stepsize $\gamma_n = 1/n$. From (21) we see the subgradient norm $\|g(\mathbf{W}(n))\|_F$ remains bounded, and hence $\eta_n \rightarrow 0$. Therefore, from (38) and (42),

$$\liminf_n f(\mathbf{W}(n)) \leq f^* + \delta, \quad \delta = \limsup_n \epsilon_n,$$

which proves Proposition 4.

REFERENCES

- [1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [2] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, 2018.
- [3] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.
- [4] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.
- [5] S. Zhou and G. Y. Li, "Fedgia: An efficient hybrid algorithm for federated learning," *IEEE Trans. Signal Process.*, vol. 71, pp. 1493–1508, 2023.
- [6] —, "Federated learning via inexact admm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9699–9708, 2023.
- [7] S. Liu, C. Liu, D. Wen, and G. Yu, "Efficient collaborative learning over unreliable D2D network: Adaptive cluster head selection and resource allocation," *IEEE Trans. Commun.*, 2024.
- [8] S. Savazzi, M. Nicoli, M. Bennis, S. Kianoush, and L. Barbieri, "Opportunities of federated learning in connected, cooperative, and automated industrial systems," *IEEE Commun. Mag.*, vol. 59, no. 2, pp. 16–21, 2021.
- [9] J. N. Tsitsiklis, "Problems in Decentralized Decision Making and Computation," Ph.D. dissertation, Massachusetts Institute of Technology, 1984.
- [10] A. Agarwal, M. J. Wainwright, and J. C. Duchi, "Distributed dual averaging in networks," *Adv. Neural Inf. Process. Syst.*, vol. 23, 2010.
- [11] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Explicit convergence rate of a distributed alternating direction method of multipliers," *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 892–904, 2015.
- [12] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Trans. Autom. Control*, vol. 54, no. 11, pp. 2506–2517, 2009.
- [13] S. Sundhar Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, pp. 516–545, 2010.
- [14] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, 2020.
- [15] D. Basu, D. Data, C. Karakus, and S. Diggavi, "Qsparse-local-SGD: Distributed SGD with quantization, sparsification and local computations," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
- [16] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized sgd with changing topology and local updates," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2020, pp. 5381–5393.
- [17] L. Yuan, Z. Wang, L. Sun, S. Y. Philip, and C. G. Brinton, "Decentralized federated learning: A survey and perspective," *IEEE Internet Things J.*, 2024.
- [18] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, 2021.
- [19] H. Xing, O. Simeone, and S. Bi, "Decentralized federated learning via SGD over wireless D2D networks," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun.*, 2020, pp. 1–5.
- [20] Z. Zhai, X. Yuan, and X. Wang, "Decentralized federated learning via mimo over-the-air computation: Consensus analysis and performance optimization," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2024.
- [21] H. Ye, L. Liang, and G. Y. Li, "Decentralized federated learning with unreliable communications," *IEEE J. Sel. Top. Signal Process.*, vol. 16, no. 3, pp. 487–500, 2022.
- [22] R. Chen, L. Li, K. Xue, C. Zhang, M. Pan, and Y. Fang, "Energy efficient federated learning over heterogeneous mobile devices via joint design of weight quantization and wireless transmission," *IEEE Trans. Mobile Comput.*, vol. 22, no. 12, pp. 7451–7465, 2023.
- [23] L. Wang, Y. Xu, H. Xu, M. Chen, and L. Huang, "Accelerating decentralized federated learning in heterogeneous edge computing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 9, pp. 5001–5016, 2023.
- [24] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "Braitentor: A peer-to-peer environment for decentralized federated learning," *arXiv preprint arXiv:1905.06731*, 2019.
- [25] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Commun. Surv. Tutor.*, 2023.
- [26] Z. Jiang, D. Wen, S. Liu, G. Zhu, and G. Yu, "Partitioned edge learning over fast fading channels," *IEEE Trans. Veh. Technol.*, vol. 74, no. 6, pp. 8561–8576, 2025.
- [27] M. Zinkevich, M. Weimer, L. Li, and A. Smola, "Parallelized stochastic gradient descent," *Adv. Neural Inf. Process. Syst.*, vol. 23, 2010.
- [28] A. Koloskova, S. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3478–3487.
- [29] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms," *arXiv preprint arXiv:1808.07576*, 2018.
- [30] X. Li, Y. Xu, J. H. Wang, X. Wang, and J. Lui, "Decentralized stochastic proximal gradient descent with variance reduction over time-varying networks," *arXiv preprint arXiv:2112.10389*, 2021.
- [31] D. B. West *et al.*, *Introduction to graph theory*. Prentice Hall Upper Saddle River, 2001, vol. 2.
- [32] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.
- [33] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," in *Proc. Annu. ACM Symp. Theory Comput.*, 2004, pp. 561–568.
- [34] Y. Xu, "On the convergence of higher-order orthogonal iteration," *Linear Multilinear Algebra*, vol. 66, no. 11, pp. 2247–2265, 2018.
- [35] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.
- [36] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2013.
- [37] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.
- [38] K. C. Kiwiel, "Convergence of approximate and incremental subgradient methods for convex optimization," *SIAM J. Optim.*
- [39] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, 2012.
- [40] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Stat.*, 2017, pp. 1273–1282.
- [41] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 1–5, 2016.
- [42] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, 2020.
- [43] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
- [44] Z. Lin, Y. Gong, and K. Huang, "Distributed over-the-air computing for fast distributed optimization: Beamforming design and convergence

- analysis,” *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 274–287, 2022.
- [45] W. K. Hastings, “Monte Carlo sampling methods using Markov chains and their applications,” 1970.
 - [46] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, “Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent,” *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
 - [47] H. Weyl, “Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung),” *Math. Ann.*, vol. 71, no. 4, pp. 441–479, 1912.