Distributed Learning over Networks with Graph-Attention-Based Personalization

Zhuojun Tian, Zhaoyang Zhang, *Senior Member, IEEE,* Zhaohui Yang,
Richeng Jin, and Huaiyu Dai, *Fellow, IEEE*

***Abstract*—In conventional distributed learning over a network, multiple agents collaboratively build a common machine learning model. However, due to the underlying non-i.i.d. data distribution among agents, the unified learning model becomes inefficient for each agent to process its locally accessible data. To address this problem, we propose a graph-attention-based personalized training algorithm (GATTA) for distributed deep learning. The GATTA enables each agent to train its local personalized model while exploiting its correlation with neighboring nodes and utilizing their useful information for aggregation. In particular, the personalized model in each agent is composed of a global part and a node-specific part. By treating each agent as one node in a graph and the node-specific parameters as its features, the benefits of the graph attention mechanism can be inherited. Namely, instead of aggregation based on averaging, it learns the specific weights for different neighboring nodes without requiring prior knowledge about the graph structure or the neighboring nodes' data distribution. Furthermore, relying on the weight-learning procedure, we develop a communication-efficient GATTA by skipping the transmission of information with small aggregation weights. Additionally, we theoretically analyze the convergence properties of GATTA for non-convex loss functions. Numerical results validate the excellent performances of the proposed algorithms in terms of convergence and communication cost.**

***Index Terms*—Distributed learning, personalized learning, statistical heterogeneity, decentralized network.**

I. Introduction

With the rapid development of deep learning as well as the growing storage and computational capacity of devices, distributed deep learning has attracted great attention recently. It can be widely applied in many areas such as cooperative localization in 5G networks, distributed signal processing and recommender system. In conventional distributed learning procedures, each agent has access to its own training data and cooperates with others to obtain a common global model. However, in practical scenarios, the agents distributed in different geographical locations always have their local partial view and tend to access data with heterogeneous distributions, i.e., the data distribution is non-i.i.d. for different agents. Take the collaborative location problem as an example. The base stations located in different positions may have diverse surroundings, leading to different data distributions and projections from the input such as channel state information (CSI) to the output user location. In such non-i.i.d. conditions, the consensus model shared among all agents may have poor performance for the locally accessible data in each agent. This problem motivates us to address the challenge of statistical heterogeneity in distributed learning, through developing the personalized model for each agent.

In this work, we investigate the decentralized communication network, which does not require a central server and is thus more robust by removing the heavy communication burden concentrated on the central server. In every round of the decentralized learning, each agent executes a local update of the model and then shares the updated model with neighboring nodes for aggregation. A dedicated aggregation procedure is expected to utilize the effective information from neighboring nodes, which however is always implicit and difficult to be explicitly characterized in the non-i.i.d. scenario. Besides, the well-known decentralized stochastic gradient descent (D-SGD) [2] aggregates the model parameters through averaging or weighted averaging, which can lead to performance loss since this aggregation does not take account of the non-i.i.d. data distribution. Thus, it is necessary and appealing to conceive an aggregation procedure for non-i.i.d. conditions in distributed learning over a network. Moreover, the aggregation procedure can be utilized to further reduce the communication cost during the training process.

Recently, we have witnessed significant progress in solving non-i.i.d. challenges of Federated Learning (FL) [3–20]. FL is a centralized learning framework [21, 22] requiring a central server for model aggregation. The experiments and analysis in [23] show the significant performance degradation of FL when the local data is non-i.i.d., highlighting the necessity of personalization. To solve the non-i.i.d. challenges through personalization techniques, Meta-Learning methods are applied in FL [3–5]. Smith *et.al.* [6] applied the multi-task learning (MTL) to FL and proposed the novel optimization method called MOCHA to solve the formulated MTL problem. The authors in [7] addressed the statistical heterogeneity by

The conference version of this paper has been accepted by IEEE ICC'23 Workshop on Edge Learning over 5G Mobile Networks and Beyond [1]. The work of Zhuojun Tian and Zhaoyang Zhang was supported in part by National Natural Science Foundation of China under Grants U20A20158 and 61725104, in part by the National Key R&D Program of China under Grants 2020YFB1807101 and 2018YFB1801104, and in part by Zhejiang Provincial Key R&D Program under Grant 2023C01021. The work of Richeng Jin was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant No. LQ23F010021. The work of Huaiyu Dai was supported in part by the U.S. National Science Foundation under Grant CNS-1824518 and Grant ECCS-2203214. (Corresponding author: Zhaoyang Zhang)

Z. Tian (email: dankotian@zju.edu.cn), Z. Zhang (email: ning_ming@zju.edu.cn), Z. Yang (email: yang_zhaohui@zju.edu.cn) and R. Jin (email: richengjin@zju.edu.cn) are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China, and also with Zhejiang Provincial Key Laboratory of Info. Proc., Commun. & Netw. (IPCAN), Hangzhou 310027, China.

H. Dai (e-mail: huaiyu_dai@ncsu.edu) is with the Department of Electrical and Computer Engineering, NC State University, USA.

clustering the agents and using the graph convolution networks to share knowledge across different clusters. The works in [8] and [9] apply the structural neural network architecture which consists of common layers across agents and the agent-specific layer for personalization. In addition to the non-i.i.d. challenges, some recent progress has been made on the communication-efficient implementation of FL in wireless communication system [24–27], through resource allocation [24, 25], reducing communication cost per iteration [26] and accelerating convergence [27].

Different from FL, decentralized learning does not require a central node to collect and process all agents' information. Each agent shares the information with its neighboring nodes and aggregates the received messages locally, utilizing all agents' computational resources and alleviating the communication burden on the central server. Many recent treatises have paid attention to decentralized learning [2, 28–32], which however are all based on the i.i.d. assumption. The authors in [33–38] considered the condition of non-i.i.d. data distribution. [33] proposed the Cross-Gradient Aggregation algorithm (CGA) to solve the statistical heterogeneity problem, which however takes high communication cost and they still intend to achieve a consensus model among all agents. The authors in [35–37] proposed algorithms based on gradient tracking, where the basic idea is to replace the local gradient with a tracker of global gradient. On the other hand, the personalized model has been scarcely exploited in decentralized network. The personalization techniques for decentralized learning may have radical differences from those in FL, which need to take the network topology and local processing ability into consideration. The authors in [39] leveraged a collaboration graph to describe the relationships among the users' tasks, which is learned alternately with the models. The proposed algorithm can obtain the personalized model for each agent. However, the alternate optimization procedure involving the graph learning may lead to high computation cost. Moreover, it requires the agents to communicate beyond their current direct neighbors in the communication network, which is impractical and may lead to high communication cost.

In this work, the non-i.i.d. challenge, the personalization needs and the robustness of decentralized network motivate us to develop personalized decentralized learning algorithm. The learned model in each agent is expected to perform well w.r.t. the local data distribution, as is widely considered in practical scenarios, since the agent usually needs a personalized model to handle its local accessed data, rather than a poor-performed common global model. Inspired by the structural neural network proposed in [8], consisting of a shared data representation component and a unique head, we apply the partially-shared local model in each agent. We observe that the algorithm in [8] trains the unique heads only with local data, without requiring any information from other agents. However, the non-i.i.d. data distributed in different agents usually has certain correlation, which can be exploited and utilized based on the topological structure formed by the agents.

To achieve this goal, we treat each agent as one node and the node-specific parameters as its features. We further deal with the topological structures of the parameters by introducing the graph neural network (GNN) [40–42]. Numerous advanced models and architectures have been proposed such as federated GNN in [42] and minibatch graph convolutional networks [43]. It has also been widely applied in practical problems such as remote sensing and image processing problems [44–46]. Specifically, the recently proposed graph attention network (GAT) [47, 48] shows its effectiveness in specifying different importance for neighboring nodes, which can be utilized in the aggregation process. Besides, we observe that GAT is a parallelizable attention process without relying on any prior knowledge about the whole network, which can be used for decentralized implementations. Inspired by that, we propose to leverage the graph attention mechanism for decentralized learning, so as to pick up the effective information from other agents. Moreover, the different aggregation weights learned and assigned to various neighboring nodes can be utilized to reduce the communication cost, based on which we develop a communication-efficient training algorithm.

Our contributions can be summarized as follows:

- We propose GATTA to train the personalized model over network in non-i.i.d. condition, which fuses the graph attention mechanism into the decentralized learning. By jointly learning to specify the weights of different neighboring nodes in the training process, GATTA enables each agent to concentrate on the most relevant information received from its neighboring nodes.
- Based on the weight-learning mechanism of GATTA, we further design a communication-efficient GATTA (CE-GATTA) by skipping the transmission of less important information, which is characterized by the learned weights.
- We theoretically analyze the convergence properties of the proposed GATTA under given conditions, which provides a useful analytical approach for personalized learning. We show its convergence rate is $\mathcal{O}(\frac{1}{\sqrt{K}})$. Moreover, the range of the fusion parameter is derived, providing potential guidance on the parameter selection.
- Numerical experiments validate the superiority of GATTA and CE-GATTA compared with other methods in different datasets, including label distribution skew as well as feature distribution skew settings. Moreover, the results under different communication network topologies are evaluated and compared to show the effectiveness of GATTA more comprehensively. Different local neural network architectures are simulated to show its broad applicability. The communication cost is also investigated to show the communication efficiency of CE-GATTA.

The rest of this paper is organized as follows. Section II describes the system model and the traditional D-SGD algorithm. In Section III, the partially-shared model among agents is introduced, based on which we develop the GATTA and CE-GATTA for personalized distributed learning. Section IV illustrates the assumptions and the convergence results of the proposed algorithm, where the range of the fusion parameter is derived. The simulation results are represented in Section V, followed by the conclusion in Section VI.

Note that this article significantly extends our previous

work [1] in several ways. Firstly, we give the theoretical analysis of the convergence property, derive the convergence rate of GATTA and provide the range of the fusion parameter. Secondly, we extend GATTA to a communication-efficient variant. Last but not the least, more experiments are conducted on different local neural network architectures and on the proposed CE-GATTA. We also compare the proposed methods with more state-of-the-art approaches on non-i.i.d. data.

II. Preliminary

A. System Model

Consider a multi-agent decentralized communication network, which can be represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In $\mathcal{G}$, $\mathcal{V} = \{1, ..., N\}$ denotes the set of N distributed agents and $\mathcal{E} = \{\varepsilon_{ij}\}_{i,j\in\mathcal{V}}$ represents the set of communication links between any two adjacent agents. Let $\mathcal{N}_i$ denote the set of all neighboring agents connected with agent i and we denote the number of agents in $\mathcal{N}_i$ by $d_i = |\mathcal{N}_i|$. The adjacency matrix of $\mathcal{G}$ is denoted by $\mathbf{A}$, where $\mathbf{A}(i,j) = 1$ if $\varepsilon_{ij} \in \mathcal{E}$ and $\mathbf{A}(i,j) = 0$ otherwise.

Each agent $i \in \mathcal{V}$ has access to a local training dataset $\mathcal{D}_i = \{\boldsymbol{x}_s, \boldsymbol{y}_s\}_{s=1}^{n_i}$ with the personal data distribution over some common feature space $\mathcal{X}$ and label space $\mathcal{Y}$. n_i denotes the number of training samples in agent i. In the considered model, the data distributions in different agents are heterogeneous, known as non-i.i.d. data. In addition to the cooperative location problem introduced in Section I, another example is the distributed natural language processing (NLP) problem, where each agent has a set of local users, whose distribution over words or expressions varies from one to another. The non-i.i.d. problem also arises in a distributed sensing system, where the agents collaboratively sense some signal. Usually, the observed signal of each agent has personalized degradation, noise effects, or variabilities [49].

Let f_i denote the loss function corresponding to agent i and the global loss function of the whole network is:

$$\min F(\mathbf{V}) := \frac{1}{N}\sum\nolimits_{i=1}^{N} f_i(\boldsymbol{v}_i), \tag{1}$$

where $\boldsymbol{v}_i$ denotes the model parameters of node i. Particularly, in a supervised learning setting, $f_i(\boldsymbol{v}_i)$ stands for the expected loss over the local data distribution of agent i and is defined as $f_i(\boldsymbol{v}_i) := \mathbb{E}_{\mathcal{D}_i}[l_i(\boldsymbol{v}_i; \boldsymbol{x}_s, \boldsymbol{y}_s)]$, where $l_i(\boldsymbol{v}_i; \boldsymbol{x}_s, \boldsymbol{y}_s)$ measures the error in predicting the label $\boldsymbol{y}_s$ given the input $\boldsymbol{x}_s$ and the model parameters $\boldsymbol{v}_i$.

Conventional consensus-based methods, such as FL or decentralized stochastic gradient descent, aim at minimizing the global loss function in (1) with consensus constraints $\boldsymbol{v}_1 = \boldsymbol{v}_2 = \cdots = \boldsymbol{v}_N$. However this approach performs poorly in the heterogeneous settings and personalized learning tasks due to different distributions of $\mathcal{D}_i$. To this end, the optimization problem in (1) is proposed without consensus constraints. Observing (1), it seems that each agent can learn its own model independently, without communicating with others. However, in many typical distributed learning settings, the number of local samples is small and cannot give accurate estimation of the expectation in $f_i(\boldsymbol{v}_i)$. Thus, it cannot promise solutions with small expected risk by training completely locally. In this sense, the collaboration among agents is necessary and through exploiting the available information from other agents, the local models can be well improved.

Before developing our algorithm, we introduce the decentralized stochastic gradient descent training method applied in distributed learning to achieve consensus among the learning model of different agents.

B. Decentralized SGD Method

A widely-used method for distributed training is the decentralized stochastic gradient descent (D-SGD) [2], which averages the model parameters from neighboring agents in each iteration. D-SGD is a simple yet efficient algorithm when applied to learn a common model for agents. Specifically, in the k-th round, each agent updates the model parameters $\boldsymbol{v}_i$ in two steps: first carrying out one epoch of local stochastic gradient descent (SGD) [50] to obtain an intermediate variable and then aggregating the obtained neighboring agents' parameters to complete the update. Here one epoch refers to a few steps of stochastic gradient descent (SGD), which walks through all the local training samples. As discussed in the introduction, the simple mechanism of averaging fails to exploit the correlated information among them, which may instead lead to worse performance in the non-i.i.d. case. To this end, we propose an algorithm in the following section, which can intelligently aggregate information from neighboring nodes.

III. Distributed Learning with Graph-Attention-Based Personalization

In this section, we develop the graph attention-based personalized training algorithm for distributed learning over a network. We first present the partially-shared local model in each client, and then illustrate the graph attention-based aggregation procedure, which can learn to exploit the useful information from other agents. The attention-based distributed training algorithm GATTA for personalized learning is summarized after that. Thirdly, based on the proposed GATTA, we develop the communication-efficient GATTA.

A. The Partially-Shared Local Model

We are motivated by the work in [8], which separates the local deep neural network into the common global layers and the personalized head unique for each agent. This insight comes from the traditional machine learning which suggests that the heterogeneous data may share a global representation despite having different labels. Inspired by that, we apply the partially-shared neural network architecture as the local model in each agent. Specifically, in the local neural network, the front layers, mapping the input into lower dimensions, are shared among all agents, while the last one layer performs as the node-specific part and is unique for each agent [(a)]. The partially-shared model is shown in Fig. 1.

With the partially-shared model, after one epoch of local training, each agent transmits the model parameters to its

[(a)]In this work, we only consider the last one layer as the node-specific part for DNN. The proposed scheme can be easily generalized to multi-layer conditions with proper design for other network architectures.

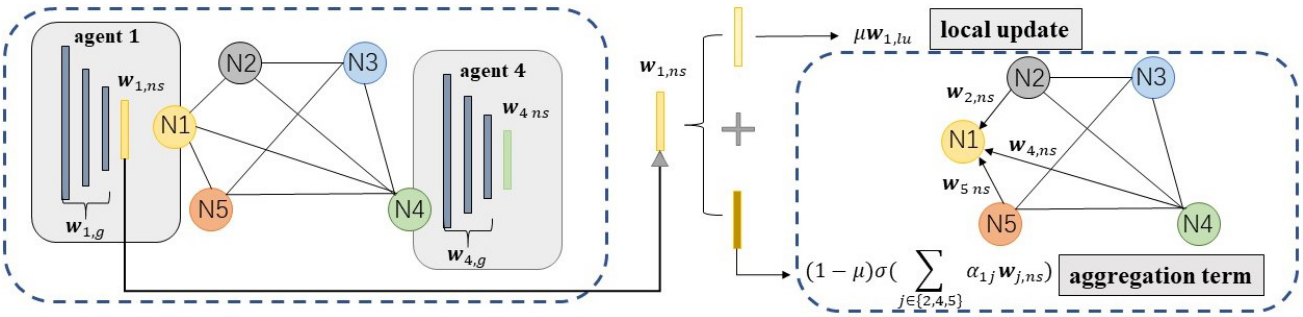

Fig. 1: The system model of GATTA: The left side represents the decentralized learning framework, where the partially-shared model is shown in the gray box. The subscript g indicates the global model part and the subscript ns stands for node-specific layer. The right side represents the two components of the node-specific parameters.

neighboring agents, including the global ones $\boldsymbol{w}_{i,g}$ as well as the node-specific ones $\boldsymbol{w}_{i,ns}$. To achieve the consensus of the global model, the efficient D-SGD method [2] is applied to update $\boldsymbol{w}_{i,g}$ in each round. [8] proposed to train the node-specific parameters only using local data set, without utilizing information from other agents. However, as mentioned before, this is inefficient especially in the condition of small number of training samples. To this end, we design an aggregation procedure for the node-specific parameters based on the graph attention mechanism.

B. The Graph Attention-Based Aggregation Method

The Graph Attention Network was first proposed in [47], which is a novel neural network architecture extending conventional neural networks to deal with graph-structured data. It can be applied to various tasks in graph domain, such as node classification and regression. Through utilizing self-attention layers, GAT can automatically configure different weights for different neighboring nodes without requiring any pre-defined weight matrix.

Inspired by this automatic weight configuration mechanism and the fact that a decentralized communication network can be viewed as a graph, we propose a graph attention based network for personalized model aggregation, aiming to dynamically exploit the effective information from other nodes to boost the local model. Each agent can be treated as one node in the graph, while its node-specific parameters are treated as local features and can be aggregated according to the graph attention mechanism. It can be implemented in a node-wise manner, which is suitable for the decentralized architecture. Following we will talk about how to utilize the graph attention mechanism in distributed personalized learning over networks.

Consider a node i, representing the agent i, with its node-specific parameters $\boldsymbol{w}_{i,ns} \in \mathbb{R}^F$. In the k-th round, after receiving the neighboring nodes' parameters in the last round, the input of the attention mechanism for each node is a set of node features $\{\boldsymbol{w}_{j,ns}^{(k-1)}\}_{j\in\mathcal{N}_i\cup\{i\}}$, consisting of its local node-specific parameters as well as those from neighboring nodes. Here the symbol $\mathcal{N}_i\cup\{i\}$ denotes the union of $\mathcal{N}_i$ and the local node. A locally shared attention mechanism $a_i(\cdot) : \mathbb{R}^F\times\mathbb{R}^F \to \mathbb{R}$ is applied in the concatenation of $\boldsymbol{w}_{i,ns}^{(k-1)}$ and $\boldsymbol{w}_{j,ns}^{(k-1)}, j \in \mathcal{N}_i$ to compute the attention coefficient:

$$e_{ij}^{(k)} = a_i(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)}), j \in \mathcal{N}_i, \tag{2}$$

where $||$ denotes the concatenation operation. The attention coefficient $e_{ij}^{(k)}$ indicates the importance of node j's parameters to node i. A softmax operation is conducted on $e_{ij}^{(k)}$ for coefficient normalization across all neighboring nodes. Thus we have:

$$\alpha_{ij}^{(k)} = \text{softmax}_j(e_{ij}^{(k)}) = \frac{\exp(e_{ij}^{(k)})}{\sum_{l\in\mathcal{N}_i}\exp(e_{il}^{(k)})}.$$

In practical implementations, to compute the attention coefficient $e_{ij}^{(k)}$ in (2), the attention mechanism $a_i(\cdot)$ is a single 1-dimensional convolution layer, parameterized by the weight parameters $\boldsymbol{\beta}_i \in \mathbb{R}^{2F}$. Applying the activation function $\sigma_G[\cdot]$, the aggregation weights computed by the attention mechanism can be expressed as:

$$\alpha_{ij}^{(k)} = \frac{\exp\left(\sigma_G\left[\boldsymbol{\beta}_i^{(k)^T}(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)})\right]\right)}{\sum_{l\in\mathcal{N}_i}\exp\left(\sigma_G\left[\boldsymbol{\beta}_i^{(k)^T}(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{l,ns}^{(k-1)})\right]\right)}. \tag{3}$$

Note that the original activation function $\sigma_G[\cdot]$ applied in [47] is LeakyReLU, while ELU is used as the activation function in our model for the sake of smoothness.

Meanwhile, considering that the aggregation of the neighboring nodes' parameters may be insufficient, we utilize another intermediate local update parameter term, denoted by $\boldsymbol{w}_{i,lu}$. The aggregation model for the node-specific parameters can thus be formulated as:

$$\boldsymbol{w}_{i,ns}^{(k)} \leftarrow \underbrace{\mu\boldsymbol{w}_{i,lu}^{(k)}}_{\text{local update}} + \underbrace{(1-\mu)\sigma\Big(\sum_{j\in\mathcal{N}_i}\alpha_{ij}^{(k)}\boldsymbol{w}_{j,ns}^{(k-1)}\Big)}_{\text{aggregation term}}, \tag{4}$$

where σ is an activation function. Here μ is the fusion parameter to balance the local update and the aggregation of parameters from neighboring nodes. We derive the range of μ in Theorem 2 in the next section, whose lower bound increases as the degree of non-i.i.d. becomes large. Note that in (4), the variables updated in the back propagation include the weight parameters $\boldsymbol{\beta}_i$ in the attention mechanism as well as the intermediate local update parameters $\boldsymbol{w}_{i,lu}$.

With the attention-based mechanism, the graph-attention-based personalized training algorithm for decentralized learning, termed as GATTA, is summarized as Algorithm 1. Here $\tilde{\mathbf{A}}$ denotes the weight matrix for aggregating the global model parameters.

For a given neural network architecture in each agent, such as AlexNet, we denote the number of its parameters by $N_v =$

Algorithm 1: Graph-Attention Based Training Algorithm (GATTA)

for *each node* $i \in \mathcal{V}$ *[in parallel]* **do**
  Initialize all the parameters in the network and set $k = 0$.
  Initialize the parameters of neighboring nodes.
while *not converged* **do**
  $k \leftarrow k + 1$.
  for *each node* $i \in \mathcal{V}$ *[in parallel]* **do**
    for *all local training samples* $\{\boldsymbol{x}_s, \boldsymbol{y}_s\} \in \mathcal{D}_i$ *randomly* **do**
      Input the training samples and **compute** the loss with the current model parameters.
      Back propagate the gradients and **Update** the model parameters.
    Obtain $\boldsymbol{w}_{i,g}^{(k-\frac{1}{2})}$, $\boldsymbol{w}_{i,lu}^{(k)}$, $\boldsymbol{\beta}_i^{(k)}$ after the local SGD.
    Calculate the node-specific parameters $\boldsymbol{w}_{i,ns}^{(k)}$ according to (4).
    Transmit the model parameters $\boldsymbol{w}_{i,g}^{(k-\frac{1}{2})}$ and $\boldsymbol{w}_{i,ns}^{(k)}$ to the neighboring nodes.
  for *each node* $i \in \mathcal{V}$ *[in parallel]* **do**
    Update the global model parameters $\boldsymbol{w}_{i,g}^{(k)} \leftarrow \sum_{j \in \mathcal{N}_i \cup \{i\}} \tilde{\mathbf{A}}(i,j) \cdot \boldsymbol{w}_{j,g}^{(k-\frac{1}{2})}$.

$N_{wg} + N_{wlu}$, where N_{wg} and N_{wlu} represent the number of parameters in the global model part and node-specific part. Then in one iteration, the number of parameters to be updated in D-SGD is N_v for one agent. The number in D-SGD with gradient tracking (GT-DSGD) [35] is $2N_v$. For GATTA, the parameters to be updated include $\boldsymbol{w}_{i,g}$, $\boldsymbol{w}_{i,lu}$ and $\boldsymbol{\beta}_i$, thus the total number is $N_v + 2N_{wlu}$. Since the node-specific layer takes a small part in the neural network, we have $N_v \leq N_v + 2N_{wlu} \leq 2N_v$.

C. Communication-Efficient GATTA for Distributed Learning

In the training process of GATTA, each agent adaptively decides how to fuse the node-specific parameters from its neighboring nodes through learning the aggregation weights. For each node, different neighboring nodes with various data distribution may have different impact on it, leading to diverse weights, especially in label distribution condition [18]. Then it naturally comes to us that in the training process, each node can stop receiving node-specific parameters from those neighboring nodes which have little positive impact on it with small weights. To better illustrate this, we plot Fig. 2 following the same setting as the first experiment in Section V. We take an arbitrary node for representation and show the learned weights of its selected 5 different neighboring nodes.

It can be observed that with the iteration going on, the weights of some neighboring nodes reduce to small values, which means those nodes have little positive impact on the local one. Motivated by this observation, we further design a communication-efficient GATTA (CE-GATTA). Specifically, we set the weight threshold τ_i. When the learned weight of j-th neighboring nodes is less than τ_i, i.e., $\alpha_{ij} < \tau_i$, the j-th neighboring node stops to transfer its node-specific parameters

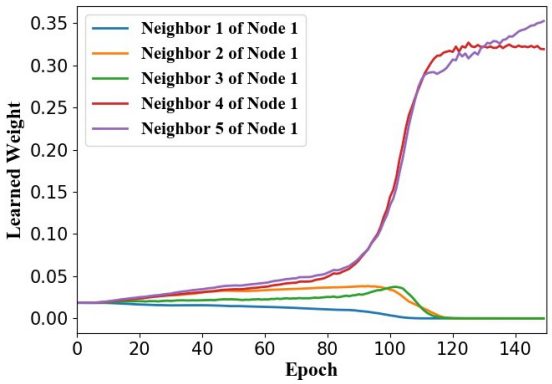

Fig. 2: Different neighboring nodes' weights

to node i. The aggregation model can thus be reformulated as:

$$\boldsymbol{w}_{i,ns}^{(k)} \leftarrow \mu \boldsymbol{w}_{i,lu}^{(k)} + (1-\mu)\sigma\Big(\sum_{j \in \mathcal{N}_{c,i}^{(k)}} \alpha_{ij}^{(k)} \boldsymbol{w}_{j,ns}^{(k-1)} \Big), \quad (5)$$

where $\mathcal{N}_{c,i}^{(k)}$ denotes the set of the selected neighboring nodes which need to transfer their node-specific parameters in the k-th iteration. Note that those neighboring nodes outside $\mathcal{N}_{c,i}$ only stop transmitting their node-specific parameters, rather than stopping transmitting the global model's parameters. This may lead to higher communication cost per epoch compared with totally stopping the communication from this node. However, it is necessary to share the global model part so as to guarantee convergence rate with the information from other nodes.

Since the aggregation weight of the removed information is small, such reduction of communication may have little impact on the convergence performance of the algorithm compared with original GATTA. Moreover, the whole information still flows over the connected communication network and CE-GATTA can adjust the aggregation weights to better fuse the information, as also indicated by experimental results. Consequently, under the proper choice of τ_i, CE-GATTA may have similar performance w.r.t. convergence rate and resultant accuracy compared with the original GATTA. Experimental results in Section V show a much faster convergence rate of GATTA compared with D-SGD. Thus, through reducing the communication per iteration as well as reducing the total communication rounds, the communication cost of the system can be highly saved. The communication-efficient GATTA can be summarized as Algorithm 2.

Note that the amount of communication cost saved by CE-GATTA is relevant to the value of the threshold τ_i. When τ_i is relatively small, the number of removed communication nodes is small and has little impact on the convergence curve. However, when τ_i becomes large enough, the convergence curve of CE-GATTA may become lossy and it takes more iterations to aggregate the information so as to achieve the same accuracy as GATTA. Consequently, the total communication cost maybe not necessarily saved. In this sense, there exists a trade-off between the number of iterations and the communication cost saved in each iteration, which is determined by τ_i and eventually affects the overall communication

cost. On the other hand, the performance of CE-GATTA maybe also influenced by the non-i.i.d. properties of the data distribution. For example, for the feature distribution skew condition in Section V, the learnt aggregation weight has small divergence among neighboring nodes, where the effect of CE-GATTA is weakened. In contrast, the divergence of the learnt weights in label distribution skew is large as shown in Fig. 2, and CE-GATTA has remarkable performance on saving the communication cost.

Algorithm 2: Communication-Efficient GATTA (CE-GATTA)

for *each node* $i \in \mathcal{V}$ *[in parallel]* **do**
  Initialize all the parameters in the network and set $k = 0$, $\mathcal{N}_{c,i} = \mathcal{N}_i$.
  Initialize the parameters of neighboring nodes.
while *not converged* **do**
  $k \leftarrow k + 1$.
  for *each node* $i \in \mathcal{V}$ *[in parallel]* **do**
    for *all local training samples* $\{\boldsymbol{x}_s, \boldsymbol{y}_s\} \in \mathcal{D}_i$ *randomly* [(a)] **do**
      Input the training samples and **compute** the loss with the current model parameters.
      Back propagate the gradients and **Update** the model parameters.
    Obtain $\boldsymbol{w}_{i,g}^{(k-\frac{1}{2})}$, $\boldsymbol{w}_{i,lu}^{(k)}$, $\boldsymbol{\beta}_i^{(k)}$ after the local SGD and **Calculate** $\{\alpha_{ij}^{(k)}\}, j \in \mathcal{N}_{c,i}$ according to (3).
    Calculate the node-specific parameters $\boldsymbol{w}_{i,ns}^{(k)}$ according to (5).
    For all $\alpha_{ij}^{(k)} < \tau_i$, remove j from $\mathcal{N}_{c,i}^{(k-1)}$ and get $\mathcal{N}_{c,i}^{(k)}$.
    If $\mathcal{N}_{c,i}^{(k)} = \emptyset$, then $\mathcal{N}_{c,i}^{(k)} \leftarrow \mathcal{N}_{c,i}^{(k-1)}$.
    Inform those neighboring nodes outside $\mathcal{N}_{c,i}^{(k)}$ to stop transmitting $\boldsymbol{w}_{j,ns}$.
    Transmit the model parameters $\boldsymbol{w}_{i,g}^{(k-\frac{1}{2})}$ to all neighboring nodes.
    Transmit $\boldsymbol{w}_{i,ns}^{(k)}$ to the needed neighboring nodes.
  for *each node* $i \in \mathcal{V}$ *[in parallel]* **do**
    Update the global model parameters $\boldsymbol{w}_{i,g}^{(k)} \leftarrow \sum_{j \in \mathcal{N}_i \cup \{i\}} \tilde{\mathbf{A}}(i,j) \boldsymbol{w}_{j,g}^{(k-\frac{1}{2})}$.

IV. Theoretical Results

We denote the updatable parameters of the local neural network in node i by $\boldsymbol{v}_i$, which is the concatenation of the global parameters $\boldsymbol{w}_{i,g}$, the local update parameters $\boldsymbol{w}_{i,lu}$, and the attention parameters $\boldsymbol{\beta}_i$. In the k-th round, the update rules of the parameters can be written as follows:

$$\boldsymbol{w}_{i,g}^{(k-\frac{1}{2})} = \boldsymbol{w}_{i,g}^{(k-1)} - \eta \sum_{t=0}^{T-1} g_{i,g,t}^{(k-1)} = \boldsymbol{w}_{i,g}^{(k-1)} - \eta \Delta_{i,g}^{(k-1)}, \quad (6)$$

$$\boldsymbol{w}_{i,g}^{(k)} = \sum\nolimits_{j \in \mathcal{N}_i \cup \{i\}} \tilde{\mathbf{A}}(i,j) \boldsymbol{w}_{j,g}^{(k-\frac{1}{2})}, \quad (7)$$

$$\boldsymbol{w}_{i,lu}^{(k)} = \boldsymbol{w}_{i,lu}^{(k-1)} - \eta \sum_{t=0}^{T-1} g_{i,lu,t}^{(k-1)} = \boldsymbol{w}_{i,lu}^{(k-1)} - \eta \Delta_{i,lu}^{(k-1)}, \quad (8)$$

$$\boldsymbol{\beta}_i^{(k)} = \boldsymbol{\beta}_i^{(k-1)} - \eta \sum_{t=0}^{T-1} g_{i,b,t}^{(k-1)} = \boldsymbol{\beta}_i^{(k-1)} - \eta \Delta_{i,b}^{(k-1)}, \quad (9)$$

where η denotes the learning rate at the k-th communication round and T is the number of stochastic gradient descent (SGD) steps in one epoch. After the k-th communication round, $g_{i,g,t}^{(k)}$ denotes the gradient w.r.t. the global parameters $\boldsymbol{w}_{i,g}$ in the t-th SGD step. Likewise, the subscript lu in (8) and b in (9) respectively represents $\boldsymbol{w}_{i,lu}$ and $\boldsymbol{\beta}_i$. Δ denotes the accumulated gradients after one epoch of SGD. Define the gradient of the local objective w.r.t. any parameter $\boldsymbol{w}$ as $\nabla f_i(\boldsymbol{w})$, then we have $g_{i,g,t}^{(k)} = \nabla f_i(\boldsymbol{w}_{i,g,t}^{(k)}, \xi_{i,t})$, where $\xi_{i,t}$ denotes the data samples in the t-th SGD step. In the k-th iteration, we define the averaged global parameters among agents as $\bar{\boldsymbol{w}}_g^{(k)}$. Meanwhile, we use the matrix $\mathbf{W}_g$ to represent the matrix form of the global parameters for all agents, i.e., $\mathbf{W}_g = [\boldsymbol{w}_{1,g}, \boldsymbol{w}_{2,g}, ..., \boldsymbol{w}_{N,g}]$, where $\boldsymbol{w}_{i,g}$ is a column vector here. Likewise, the matrix form of $\Delta_{i,g}$ for all agents can be denoted by $\Xi_g \triangleq [\Delta_{1,g}, ..., \Delta_{N,g}]$. To this end, based on the update rule in (6) and (7), we can obtain the following update rule in matrix form.

$$\mathbf{W}_g^{(k)} = (\mathbf{W}_g^{(k-1)} - \eta \Xi_g^{(k-1)}) \tilde{\mathbf{A}}. \quad (10)$$

Before presenting our theoretical findings, we make the following assumptions, where the expectations are taken over the randomness in stochastic gradients.

Assumption 1. *(Spectral Gap) The aggregation weight matrix $\tilde{\boldsymbol{A}}$ for the global model parameters is a symmetric doubly stochastic matrix. Denote its eigenvalues by $1 = |\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_N| \geq 0$. We further assume the spectral gap $1 - \rho \in (0, 1]$, where $\rho = |\lambda_2| \in (0, 1]$.*

Assumption 2. *(Smoothness) The local objective functions f_i are L-smooth for the parameters $\boldsymbol{v}_i$ of each node $i \in \mathcal{V}$, i.e.,*

$$f_i(\boldsymbol{v}_i) \leq f_i(\boldsymbol{v}_i') + \nabla f_i(\boldsymbol{v}_i')^T (\boldsymbol{v}_i - \boldsymbol{v}_i') + \frac{L}{2} \|\boldsymbol{v}_i - \boldsymbol{v}_i'\|_2^2. \quad (11)$$

Assumption 3. *(Unbiased Local Gradient Estimator) For each node $i \in \mathcal{V}$ and $\boldsymbol{w}_i \in \{\boldsymbol{w}_{i,g}, \boldsymbol{w}_{i,lu}, \boldsymbol{\beta}_i\}$, the local gradient estimator is unbiased, i.e., $\mathbb{E}[g_{i,t}] = \nabla f_i(\boldsymbol{w}_{i,t})$, in the t-th gradient descent step.*

Assumption 4. *(Bounded Local Variance) There exist scalar $\chi > 0$ such that for each node $i \in \mathcal{V}$ and $\boldsymbol{w}_i \in \{\boldsymbol{w}_{i,g}, \boldsymbol{w}_{i,lu}, \boldsymbol{\beta}_i\}$, the variance of local gradient estimator is bounded by $\mathbb{E}[\|g_{i,t} - \nabla f_i(\boldsymbol{w}_{i,t})\|] \leq \chi$.*

Assumption 5. *(Degree of Non-i.i.d.) There exists scalar $\kappa \geq 0$ for each node $i \in \mathcal{V}$ such that for the global parameters,*

$$\frac{1}{N} \sum_{i \in \mathcal{V}} \mathbb{E} \Big\| \nabla f_i(\boldsymbol{w}_g) - \frac{1}{N} \sum_{j \in \mathcal{V}} \nabla f_j(\boldsymbol{w}_g) \Big\| \leq \kappa. \quad (12)$$

Assumption 6. *(Bounded Gradients) There exists scalar $G > 0$ such that for each node $i \in \mathcal{V}$ and any $\boldsymbol{w}_i \in \{\boldsymbol{w}_{i,lu}, \boldsymbol{\beta}_i, \boldsymbol{w}_{i,ns}\}$,*

$$\|\nabla f_i(\boldsymbol{w}_i)\|_2^2 \leq G. \quad (13)$$

Further, the gradients of the activation functions satisfy

$$\|\sigma'\|_2^2 \leq 1. \quad (14)$$

Among all assumptions, Assumption 2 for the local objective function is standard, which also restricts the activation functions to be smooth. The commonly used activation functions such as ELU, sigmoid and Tanh all satisfy this assumption. Assumption 5 limits the non-i.i.d. degree through the gradients of the global parameters $\boldsymbol{w}_g$. Assumptions 1-5 are commonly used and can be widely found in [2, 5, 17, 30, 50–52]. We further make Assumption 6 to simplify the analysis in Theorem 2, where equation (14) can be easily satisfied by most common activation functions including ELU, sigmoid and Tanh.

Here one key difference of our analysis from others in standard D-SGD is that we take the node-specific parameters into consideration. Specifically, in our analysis, the performance is evaluated under the *averaged* global model parameters among agents $\bar{\boldsymbol{w}}_g$, together with the *personalized* individual parameters including local update parameters $\boldsymbol{w}_{i,lu}$ and attention parameters $\boldsymbol{\beta}_i$. This is rational because the parameters in the global model part are updated following the standard D-SGD to achieve consensus among agents, while the other parameters are updated locally with personalization. The performance of D-SGD has been analyzed in [2, 30] based on averaged parameters. Different from them, we additionally consider the node-specific model part and combine these two kinds of parameters. Given the above assumptions, we have the following lemmas, where the expectation is over the local data samples.

Lemma 1. *Denote the variable value of $\boldsymbol{w}_i$ in the t-th SGD step by $\boldsymbol{w}_{i,t}$. Under Assumption 4, we have*

$$\mathbb{E}[\|\Delta_i^{(k)}\|_2^2] \leq T\cdot\chi^2 + \mathbb{E}[\|\sum_{t=0}^{T-1}\nabla f_i(\boldsymbol{w}_{i,t}^{(k)})\|_2^2], \tag{15}$$

for all $\boldsymbol{w}_i \in \{\boldsymbol{w}_{i,g}, \boldsymbol{w}_{i,lu}, \boldsymbol{\beta}_i\}$, and $\Delta_i \in \{\Delta_{i,g}, \Delta_{i,lu}, \Delta_{i,b}\}$ respectively.

Lemma 2. *For any learning rate satisfying $\eta < \frac{1}{4TL}$, we have the following results:*

$$\mathbb{E}[\|\boldsymbol{w}_{i,t} - \boldsymbol{w}_i\|_2^2] \leq 4T\eta^2\chi^2 + 16T^2\eta^2\|\nabla f_i(\boldsymbol{w}_i)\|_2^2. \tag{16}$$

Lemma 3. *For any $i \in \mathcal{V}$, $\boldsymbol{w}_i \in \{\boldsymbol{w}_{i,lu}, \boldsymbol{\beta}_i\}$ and $\Delta_i \in \{\Delta_{i,lu}, \Delta_{i,b}\}$ respectively, we have*

$$\begin{aligned}&\mathbb{E}\Big[-\eta\nabla f_i(\boldsymbol{w}_i^{(k-1)})^T\Delta_i^{(k-1)} + \frac{1}{2}\eta^2 L\|\Delta_i^{(k-1)}\|_2^2\Big]\\ &\leq -cT\eta\|\nabla f_i(\boldsymbol{w}_i^{(k-1)})\|_2^2 + \frac{\eta^2 TL}{2}(1+4\eta TL)\chi^2,\end{aligned} \tag{17}$$

where c is a constant satisfying $0 < c < \frac{1}{2} - 8\eta^2T^2L^2$.

Lemma 4. *For the global parameters $\boldsymbol{w}_g$, we have*

$$\mathbb{E}\Big\|\nabla F(\bar{\boldsymbol{w}}_g^{(k)}) - \nabla F(\boldsymbol{w}_g^{(k)})\Big\|_2^2 \leq \frac{L^2}{N}\sum_{i\in\mathcal{V}}\mathbb{E}\|\bar{\boldsymbol{w}}_g^{(k)} - \boldsymbol{w}_{i,g}^{(k)}\|_2^2, \tag{18}$$

where $\nabla F(\bar{\boldsymbol{w}}_g^{(k)}) \triangleq \frac{1}{N}\sum_{i\in\mathcal{V}}\nabla f_i(\bar{\boldsymbol{w}}_g^{(k)})$ and $\nabla F(\boldsymbol{w}_g^{(k)}) \triangleq \frac{1}{N}\sum_{i\in\mathcal{V}}\nabla f_i(\boldsymbol{w}_{i,g}^{(k)})$

Lemma 5. *For the averaged global parameters $\bar{\boldsymbol{w}}_g$, we have*

$$\bar{\boldsymbol{w}}_g^{(k)} - \bar{\boldsymbol{w}}_g^{(k-1)} = -\frac{\eta}{N}\sum_{i\in\mathcal{V}}\Delta_{i,g}^{(k-1)} \triangleq -\eta\bar{\Delta}_g^{(k-1)}.$$

The proofs of the Lemmas can be found in Appendix A. As we talked before, the performance is measured under averaged $\bar{\boldsymbol{w}}_g$ and individual $\boldsymbol{w}_{i,lu}$, $\boldsymbol{\beta}_i$. Thus we define the partially-shared parameters in agent i as $\tilde{\boldsymbol{v}}_i$, which is the concatenation of $\bar{\boldsymbol{w}}_g$, $\boldsymbol{w}_{i,lu}$ and $\boldsymbol{\beta}_i$. Additionally, we define the concatenation of individual parameters $\boldsymbol{w}_{i,lu}$, $\boldsymbol{\beta}_i$ as $\boldsymbol{v}_{i,ns}$ and $\nabla F(\boldsymbol{v}_{ns}) \triangleq \frac{1}{N}\sum_{i\in\mathcal{V}}\nabla f_i(\boldsymbol{v}_{i,ns})$. The product of multiple weight matrices for global parameters is denoted by $\bar{\tilde{\mathbf{A}}}_{s,k-1} = \prod_{l=s}^{k-1}\tilde{\mathbf{A}}$. $\mathbf{Q} = \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$ and $\rho_{s,k-1} = \|\bar{\tilde{\mathbf{A}}}_{s,k-1} - \mathbf{Q}\|$. Additionally, we also make the following definitions.

$$A_K = \frac{1}{K}\sum_{k=1}^{K}\sum_{s=1}^{k-1}\rho_{s,k-1}^2, \quad B_K = \frac{1}{K}\sum_{k=1}^{K}\Big(\sum_{s=1}^{k-1}\rho_{s,k-1}\Big)^2,$$
$$C_K = \max_{s\in[K-1]}\sum_{k=s+1}^{K}\rho_{s,k-1}\Big(\sum_{l=1}^{k-1}\rho_{l,k-1}\Big).$$

Then based on the lemmas and definitions above, we give the convergence property of the proposed GATTA method as Theorem 1 and Corollary 1.

Theorem 1. *Provided that $\eta < \min\{\frac{1}{24TL}, \frac{1}{32TL\sqrt{C_K}}\}$, under Assumptions 1-5 made above, the iterates of GATTA algorithm satisfy the following inequality:*

$$\min_{k\in[K]}\mathbb{E}\big[\|\nabla F(\boldsymbol{v}_{ns}^{(k)})\|_2^2 + \|\nabla F(\bar{\boldsymbol{w}}_g^{(k)})\|_2^2\big] \leq \frac{F_0 - F_*}{cTK\eta} + \Phi,$$

where F_0 denotes initial value of the objective $F(\tilde{\boldsymbol{V}})$ and F_* denotes its optimal value.

$$\begin{aligned}\Phi = \frac{1}{c}\Big\{&\eta L(1+4\eta TL)\chi^2 +\\ &\frac{1}{N}\big[\eta L(4\kappa^2T + \chi^2) + 6T\eta^2\chi^2L^2\big] +\\ &64\eta^2TL^2(A_K\chi^2 + B_KT(\kappa^2 + T\eta^2\chi^2L^2))\Big\},\end{aligned}$$

c is a constant satisfying $0 < c < \frac{1}{2} - 8\eta^2T^2L^2$, and A_K, B_K, C_K are defined as above.

Its proof can be found in Appendix B. Based on Theorem 1, we have the following convergence rate for GATTA as Corollary 1.

Corollary 1. *Let the learning rate $\eta = \frac{m}{\sqrt{K}}$, where m is a constant such that $\eta < \min\{\frac{1}{24TL}, \frac{1}{32TL\sqrt{C_K}}\}$, then the convergence rate for GATTA is $\mathcal{O}(\frac{1}{\sqrt{K}})$.*

Finally, we provide the limited range of the fusion parameter μ in (4) as the following Theorem 2. We denote μ by $\mu_i^{(k)}$ to associate with a specific node i in the k-th round for better clarification.

Theorem 2. *Denote the gradient value of $\sigma_G(x_j)$ by $\sigma'_{G,j}$, where $x_j \triangleq {\boldsymbol{\beta}_i^{(k-1)}}^T(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)})$. Define*

$$\begin{aligned}&D_i^{(k)} \triangleq\\ &\Big[\sum_{j\in\mathcal{N}_i}\big\|\boldsymbol{w}_{j,ns}^{(k-1)}\|_2^2\Big]\cdot\Big[\sum_{j\in\mathcal{N}_i}\sum_{l\in\mathcal{N}_i\setminus\{j\}}\big\|\sigma'_{G,j}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)})\\ &-\sigma'_{G,l}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{l,ns}^{(k-1)})\big\|_2^2\Big].\end{aligned}$$

Then to satisfy Assumption 6, the value of the fusion parameter $\mu_i^{(k)}$ in the k-th round for node $i \in \mathcal{V}$ should be constrained in

$$1 - \frac{1}{\sqrt{d_i(d_i-1)D_i^{(k)}}} \leq \mu_i^{(k)} \leq 1. \tag{19}$$

Remark 1. *As the number of neighboring nodes d_i becomes larger, the lower bound of $\mu_i^{(k)}$ increases. Moreover, the value of $D_i^{(k)}$ reflects the degree of non-i.i.d. of the neighboring nodes to some extent. If $x_j \geq 0$ is satisfied for all $j \in \mathcal{N}_i$, $D_i^{(k)}$ can be further simplified into the following expression:*

$$D_i^{(k)} = \Big[\sum_{j \in \mathcal{N}_i} \|\boldsymbol{w}_{j,ns}^{(k-1)}\|_2^2 \Big] \times \Big[\sum_{j \in \mathcal{N}_i} \sum_{l \in \mathcal{N}_i \setminus \{j\}} \|\boldsymbol{w}_{j,ns}^{(k-1)} - \boldsymbol{w}_{l,ns}^{(k-1)}\|_2^2 \Big]. \tag{20}$$

It can be observed that when the parameters of neighboring nodes are closer, which indicates that the non-i.i.d. degree is smaller, the lower bound of $\mu_i^{(k)}$ reduces. This is rational since a small value of $\mu_i^{(k)}$ represents more impact of the aggregation term under smaller non-i.i.d. degree.

Theorem 2 provides the lower bound of the fusion parameter, below which the convergence of GATTA cannot be guaranteed. We refer the readers to Appendix C for detailed proof. In the practical implementations, we do not focus on the fusion parameter design for each single node. For the sake of simplicity, we denote the fusion parameter for all the nodes by μ as applied in (4) and choose its value through experiments.

V. Numerical Experiments

In this section, we numerically evaluate the performance of our proposed algorithms under non-i.i.d. conditions. In particular, we consider a multi-agent communication network with N nodes, whose topology is generated randomly using the *Erdos_Renyi* random graph model, with the connectivity probability equal to p. If not specified, we apply the widely-used AlexNet architecture in each agent, which is a representative DNN and CNN architecture. The node-specific layer gets its parameters according to (4) or (5), including weights and biases. Meanwhile, to make the loss function smooth, we apply ELU as the activation functions for the whole network.

The performance is evaluated on the image classification problem. To validate the algorithm more comprehensively, we simulate on two different settings of non-i.i.d.: label distribution skew [18] and feature distribution skew [16]. For the label distribution skew, we consider the 10-class classification problem over CIFAR-10 [53] and randomly choose c_i labels assigned for each agent, which reflects the non-i.i.d. data distribution. The training samples corresponding to the same label are averaged and randomly assigned to the agents. The testing samples are assigned to agents corresponding to their local label distributions. Note that the number of training samples with respect to one label is averaged among agents with that label. Meanwhile, for a general and comprehensive evaluation of the learned model, the testing samples are assigned to agents with all of them corresponding to the local label distribution.

For the feature distribution skew, we consider the 62-class classification problem over FEMNIST [54], which contains images of different characters written by different writers. We randomly assign e_i different writers with their written characters for each agent and use 75% of them for training, 25% for testing. The performance metric is the average of the testing accuracy among the agents. We compare the proposed GATTA and CE-GATTA with three baseline methods: centralized FL [21], D-SGD [2] as well as independent learning in each agent (IL). For all distributed learning methods, agents exchange messages after one epoch of local training.

A. Evaluation of convergence on the Different Datasets

Label Distribution Skew. We first show the results on the CIFAR-10 dataset under different numbers of local labels $c_i = 3, 4, 5$ and different numbers of local training samples n_i. The communication network is generated randomly with $N = 100$ and $p = 0.6$. In each agent, the local neural network is made of two 5×5 convolutional layers, each followed by a 3×3 max pooling layer with stride 2, and three fully connected layers. The last fully-connected layer is the node-specific layer. For all the algorithms, the local optimizer is RMSProp [55]. The learning rate of IL is set to $\eta = 0.01$, while for others $\eta = 0.001$. All of the learning rates are tuned from $\{0.1, 0.01, 0.001, 0.0001\}$ and we set $\mu = 0.9$ through experiments. The threshold for CE-GATTA is set to $\tau_i = \frac{1}{4d_i}$. The results are shown in Figure 3.

As shown in Fig. 3, in the label skew condition, GATTA and CE-GATTA outperform the baseline methods in both convergence rate as well as resultant accuracy. Here FL and D-SGD shows similar results due to the average and consensus procedure. Also, theorem in [2] proves the same convergence rate of the centralized and decentralized method. Secondly, comparing the results under different c_i, it can be observed that the superiority of GATTA over FL and D-SGD is more pronounced under smaller c_i. This is because a smaller c_i indicates less relativity among agents and the personalization technique is more effective. Meanwhile, the independent learning method IL performs best in $c_i = 3$.

Moreover, the communication-efficient implementation of GATTA shows almost the same convergence property as original GATTA. This is resulted from that the information transmission CE-GATTA removed is redundant or useless, and has little impact on the performance of the algorithm. In this way, the reduction of communication is effective without increasing the iteration number.

Feature Distribution Skew. We next evaluate the performance in feature distribution skew condition through assigning different writers in FEMNIST for different agents. The number of writers in each agent is set to $e_i = 2, 4, 6$ respectively. The local network is made of two 3×3 convolutional layers, each followed by a 2×2 max pooling layer with stride 2, and two fully connected layers, the last one of which serves as the node-specific layer. We set $\eta = 0.01$ for all the algorithms, as tuned from $\{0.1, 0.01, 0.001, 0.0001\}$, and $\mu = 0.7$, $\tau_i = 1/d_i$. The results are shown in Figure 4. As shown in Fig. 4, in the feature skew condition, GATTA and CE-GATTA also show

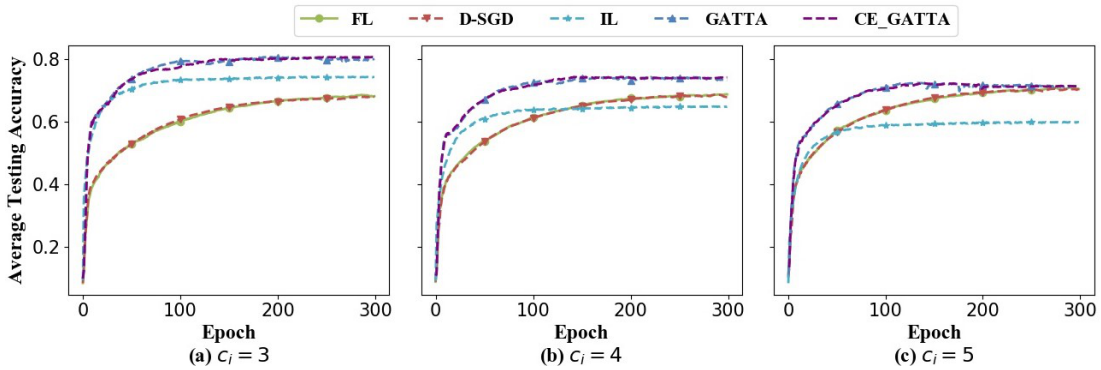

Fig. 3: Convergence behaviours for CIFAR-10 under various c_i

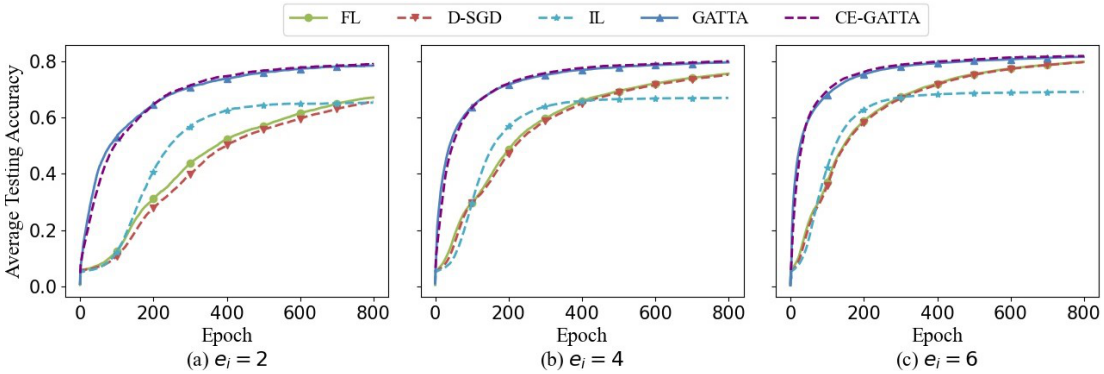

Fig. 4: Convergence behaviours for FEMNIST under various e_i

good performance compared with the other methods. It can be observed that the convergence rate of proposed algorithms is much faster than FL and D-SGD, indicating that the the proposed methods can quickly and effectively capture the useful information from other agents. Also, as e_i reduces, indicating a larger degree of non-i.i.d., the GATTA shows significant accuracy performance compared with other baseline methods, highlighting the effectiveness of proposed algorithms in the non-i.i.d. conditions.

In both Fig. 3 and 4, there exists a similar and inspiring trend that the superiority of GATTA and CE-GATTA, over FL and D-SGD, is higher under smaller number of labels or writers in each agent. To shed more light on its inherent reasons, we provide the following Remark 2.

Remark 2. *The local number of labels (c_i) or writers (e_i) affects the non-i.i.d. degree among nodes, which becomes higher when c_i or e_i decreases. When the non-i.i.d. degree becomes higher, the correlation among nodes reduces, leading to a worse performance of consensus learning methods such as FL or D-SGD and a higher superiority of personalized GATTA/CE-GATTA. On the contrary, when c_i or e_i increases, the correlation among nodes becomes larger, where a consensus model may adapt more on local data distribution and the superiority of GATTA/CE-GATTA becomes smaller.*

B. Evaluation of accuracy On Different Network Topologies

We investigate the performance of the algorithms under different network topologies. Specifically, we use FEMNIST for validation and set $e_i = 2$. The results under different numbers of agents as well as different probabilities of connectivity are evaluated. We set $\eta = 0.01$ (except for the ring topology), $\mu = 0.7$, $\tau_i = 1/d_i$ and the maximum number of rounds is 800. The results are first compared under fixed $N = 100$ and different probabilities of connectivity $p = 0.05, 0.2$. Then, we fix $p = 0.2$ and set $N = 50, 150$. We also consider a more extreme condition of a ring communication network topology with $N = 50$ nodes, where $\eta = 0.008$. We additionally compare the algorithms with four different state-of-the-art methods as follows:

- The first one is the method in [8] generalized in decentralized network, which we term as RepDL. In RepDL, each agent aggregate the parameters in the global component while updating the node-specific parameters only with local dataset. The comparison with RepDL can shows the effectiveness of the graph-based aggregation procedure. Its learning rate is $\eta = 0.01$.
- The second procedure is the traditional D-SGD following fine-tuning on different nodes for personalization, termed as DSGD-FT. Such idea has achieved good performance in federated learning. Its learning rate is $\eta = 0.01$.
- The third method is the D^2 training algorithm proposed in [34], whose learning rate is $\eta = 0.1$.
- The last method is the GT-DSGD algorithm in [35], where the decaying step-size $\eta_k = \frac{1.0}{10+\sqrt{k}}$ are adopted and the Metropolis rule is applied to define the weight matrix as suggested by [37].

$$\alpha_{ij} = \begin{cases} 1/\max\{d_i, d_j\} & \text{if} \quad j \in \mathcal{N}_i, \\ 1 - \sum_{l \in \mathcal{N}_i} \alpha_{il} & \text{if} \quad j = i, \\ 0 & \text{otherwise.} \end{cases}$$

The communication networks are generated randomly using the *Erdos_Renyi* model and the testing accuracy results are averaged over 5 trails as reported in Table I, along with the 95% confidence intervals. As shown in Table I, the proposed algorithms have the best performance among all algorithms, even under the sparse connectivity $p = 0.05$ and extreme condition of ring communication network. Moreover, the resultant accuracy of CE-GATTA is similar to that of GATTA. This is because of the mechanism of CE-GATTA, which can learn and adjust to fuse the information from the selected nodes. And such fusion may utilize the whole information flowing over the communication network. D^2 algorithm fails to converge when $N = 100, p = 0.05$, $N = 50, p = 0.2$ and in the ring topology.

C. Generalization to Other DNN Architecture

In this subsection, we simulate the proposed GATTA on other kinds of local neural network architecture. Different from the AlexNet above, we apply ResNet-18 [56] for CIFAR-10 and MLP for FEMNIST. Specifically, in ResNet-18, each convolutional layer is followed by a batch-normalization layer, whose shift and scale are trainable parameters. The MLP is a $784 - 400 - 100 - 62$ architecture with three fully connected (FC) layers, and each of the first two FC layers is followed by a batch-normalization layer. The learning rate is 0.001 for all methods on ResNet-18, and 0.1 for all approaches on MLP. In both networks, the last fully-connected layer is treated as the node-specific layer for GATTA. Moreover, we compare the results with another method proposed in [19], where the batch-normalization layers are not averaged in the training process and only trained with local data. We name it as BN-DSGD. The other settings are same as those in Section V-A and we set $c_i = 3$ for CIFAR-10, $e_i = 2$ for FEMNIST. The results in ResNet-18 and MLP are shown in Fig. 5 and 6.

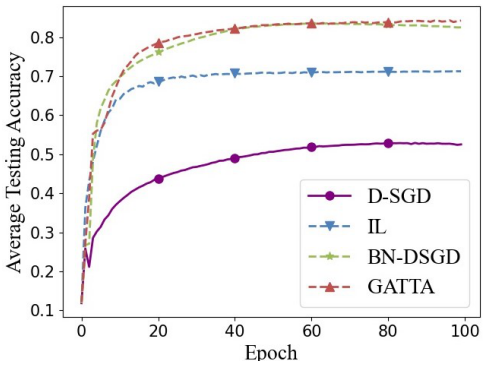

Fig. 5: Convergence behaviors for CIFAR-10 on ResNet-18

In Fig. 5, GATTA and BN-DSGD share similar performance, while in Fig. 6, GATTA outperforms BN-DSGD. Note that BN-DSGD requires the network architecture having the batch-normalization layer and its performance highly relies on the number of local data samples training the batch-normalization layers. Meanwhile, the results validate the efficiency and superiority of the proposed GATTA on different local DNN architectures.

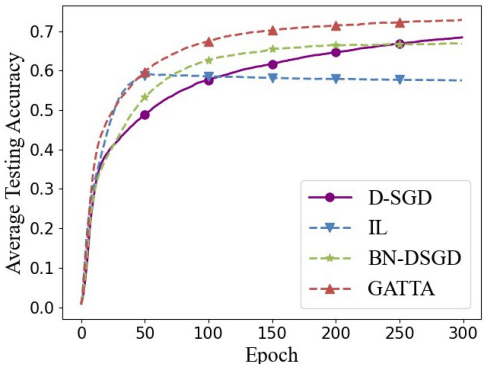

Fig. 6: Convergence behaviors for FEMNIST on MLP

D. Evaluation of Communication Cost

In this part, we evaluate the communication cost of CE-GATTA and compare with the traditional D-SGD method. Note that here we focus on the decentralized communication network topology without a fusion center, so we do not conduct FL for comparison. Specifically, we measure the communication cost by the total number of parameters transmitted. The algorithms stop when they achieve the accuracy requirements ($0.79\%, 0.75\%, 0.72\%$ for $c_i = 3, 4, 5$ respectively) or the maximum iteration number. The setting of the simulation is the same as the label skew condition in Section V-A. We first show the reduction of communication cost with epoch in Fig. 7, where the BaseLine refers to the methods of D-SGD or GATTA, which transmits all the parameters to all the neighboring nodes. From Fig. 7, it can be observed that as the iteration goes on, the communication cost of CE-GATTA per epoch reduces by stopping the transmission of less important parameters. Moreover, when the learning of the weight specification comes to converge, the condition of $c_i = 3$ takes the least communication cost. It is rational since a smaller c_i indicates less relativity among agents and there can be more ineffective information stopped to be transmitted.

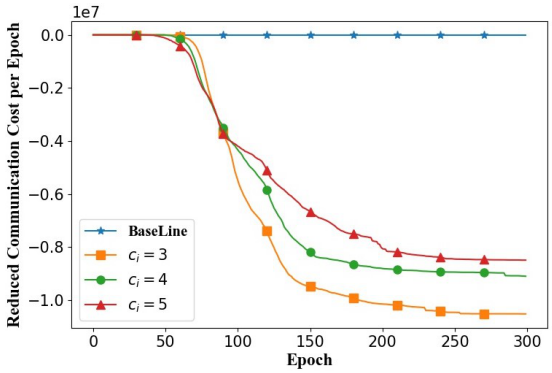

Fig. 7: The reduced communication cost with epoch

Then we show the results of total communication cost in Table. II. It can be observed that compared with traditional D-SGD, CE-GATTA largely reduces the communication cost resulted from the faster convergence rate and less information transmission per epoch.

In the following, we focus on the performance of CE-GATTA under different threshold τ_i. As we talked in Section

TABLE I: Comparison of average testing accuracy under different network topologies.

Algorithms	Network Parameters				
	$N = 100, p = 0.05$	$N = 100, p = 0.2$	$N = 50, p = 0.2$	$N = 150, p = 0.2$	$N = 50$, ring
FL	$67.39 \pm 0.13\%$	$67.44 \pm 0.15\%$	$67.20 \pm 0.16\%$	$67.40 \pm 0.14\%$	$67.15 \pm 0.14\%$
D-SGD	$65.91 \pm 0.27\%$	$66.08 \pm 0.16\%$	$66.33 \pm 0.25\%$	$66.21 \pm 0.21\%$	$66.26 \pm 0.20\%$
IL	$65.49 \pm 0.06\%$	$65.53 \pm 0.05\%$	$65.07 \pm 0.09\%$	$65.14 \pm 0.05\%$	$65.06 \pm 0.10\%$
RepDL	$66.93 \pm 0.10\%$	$66.85 \pm 0.13\%$	$66.40 \pm 0.21\%$	$66.54 \pm 0.19\%$	$66.41 \pm 0.06\%$
DSGD-FT	$71.02 \pm 0.12\%$	$72.46 \pm 0.10\%$	$70.92 \pm 0.17\%$	$70.95 \pm 0.15\%$	$72.24 \pm 0.20\%$
D^2	–	$73.92 \pm 0.30\%$	–	$72.95 \pm 0.21\%$	–
GT-DSGD	$69.79 \pm 0.11\%$	$72.57 \pm 0.16\%$	$71.27 \pm 0.39\%$	$72.34 \pm 0.35\%$	$73.28 \pm 0.21\%$
GATTA	$\mathbf{78.78 \pm 0.21\%}$	$\mathbf{78.81 \pm 0.23\%}$	$\mathbf{78.90 \pm 0.15\%}$	$\mathbf{79.03 \pm 0.20\%}$	$\mathbf{77.10 \pm 0.27\%}$
CE-GATTA	$\mathbf{78.70 \pm 0.20\%}$	$\mathbf{78.67 \pm 0.25\%}$	$\mathbf{79.04 \pm 0.17\%}$	$\mathbf{79.05 \pm 0.23\%}$	$\mathbf{76.92 \pm 0.28\%}$

TABLE II: Comparison of communication cost

	$c_i = 3$	$c_i = 4$	$c_i = 5$
D-SGD	4.0204×10^{12}	4.0204×10^{12}	4.0204×10^{12}
CE-GATTA	1.6076×10^{12}	1.6578×10^{12}	1.6580×10^{12}
Reduction	60.0%	58.8%	58.8%

III-C, when the threshold is small or the number of epochs is large, there exists little difference of the resultant accuracy over different τ_i. To better show the difference and reveal the trade-off, we choose τ_i with relatively large values, where $\tau_i = 1/d_i, 2/d_i, 3/d_i, 4/d_i$. And the total communication cost is calculated until the accuracy achieves 78%. Then we present the following Table III to show the comparison of communication cost.

TABLE III: Comparison of communication cost ($\times 10^{12}$)

$\tau_i = 1/4d_i$	$\tau_i = 1/d_i$	$\tau_i = 2/d_i$	$\tau_i = 3/d_i$	$\tau_i = 4/d_i$
1.0553	1.0347	1.2353	1.2055	1.2657

It can be observed that the communication cost does not necessarily become smaller with the increasing τ_i, due to a larger number of epochs to achieve the required accuracy. Consequently, there exists a best choice of the threshold for CE-GATTA saving the communication cost most.

VI. Conclusion

We considered the statistical heterogeneous problem in the decentralized deep learning and proposed a graph-attention-based personalization method called GATTA. The GATTA enables each agent to adaptively utilize the information from neighboring agents. This can be implemented through learning specify weights for different neighboring agents in the training process, based on which we designed a communication-efficient GATTA. We also derived the theoretical convergence properties of GATTA and provided the range of the fusion parameter. Finally, we compared the performances of the proposed algorithms with other distributed learning algorithm under different datasets, non-i.i.d. settings, and network topologies. The experiment results validated the superiority of the proposed algorithms over conventional schemes.

The algorithm with rigorous theoretical guarantees provides a broad impact on improving the local learning quality for applications that deploy decentralized learning. Although the local personalized model and the experiments are based on deep neural networks, the proposed graph-attention-based personalization technique could be generalized to other learning networks with proper design. Thus, one of our future researching topics is to generalize the personalized model into other neural networks. Another important issue is the theoretical convergence analysis of CE-GATTA, which could shed more light on its overall performance w.r.t. communication and computation costs. Additionally, it is promising to apply the proposed algorithm to practical wireless communication problems, such as collaborative location for multiple base stations.

Appendix

A. Proof of the Lemmas

The proof of Lemma 1 is as follows.

Proof:

$$\begin{aligned}\mathbb{E}[\|\Delta_i^{(k)}\|_2^2] &= \mathbb{E}[\|\sum_{t=0}^{T} g_{i,t}^{(k)}\|_2^2] \\ &\overset{(a)}{=} \mathbb{E}[\|\sum_{t=0}^{T-1}(g_{i,t}^{(k)} - \nabla f_i(\boldsymbol{w}_{i,t}^{(k)}))\|_2^2] + \mathbb{E}[\|\sum_{t=0}^{T-1}\nabla f_i(\boldsymbol{w}_{i,t}^{(k)})\|_2^2] \\ &\overset{(b)}{\leq} T\cdot\chi^2 + \mathbb{E}[\|\sum_{t=0}^{T-1}\nabla f_i(\boldsymbol{w}_{i,t}^{(k)})\|_2^2], \end{aligned} \tag{21}$$

where (a) follows from the fact that $\mathbb{E}[\|\boldsymbol{x}\|_2^2] = \mathbb{E}[\|\boldsymbol{x} - \mathbb{E}[x]\|_2^2] + \|\mathbb{E}[x]\|_2^2$ and (b) follows from the unbiased estimator. ■

The proof of Lemma 2 is as follows, which is similar to that of Lemma 2 in [51].

Proof: We have that

$$\begin{aligned}&\mathbb{E}[\|(\boldsymbol{w}_{i,t}^{(k)} - \boldsymbol{w}_i^{(k)})\|_2^2] = \mathbb{E}[\|(\boldsymbol{w}_{i,t-1}^{(k)} - \boldsymbol{w}_i^{(k)}) - \eta g_{i,t-1}^{(k)}\|_2^2] \\ &= \mathbb{E}[\|(\boldsymbol{w}_{i,t-1}^{(k)} - \boldsymbol{w}_i^{(k)}) - \eta(g_{i,t-1}^{(k)} - \nabla f_i(\boldsymbol{w}_{i,t-1}^{(k)})\end{aligned}$$

$$
\begin{aligned}
&+\nabla f_i(\boldsymbol{w}_{i,t-1}^{(k)})-\nabla f_i(\boldsymbol{w}_i^{(k)})+\nabla f_i(\boldsymbol{w}_i^{(k)}))\|_2^2]\\
&\overset{(a)}{\leq}\mathbb{E}[\|(\boldsymbol{w}_{i,t-1}^{(k)}-\boldsymbol{w}_i^{(k)})-\eta(\nabla f_i(\boldsymbol{w}_{i,t-1}^{(k)})-\nabla f_i(\boldsymbol{w}_i^{(k)})\\
&+\nabla f_i(\boldsymbol{w}_i^{(k)}))\|_2^2]+E[\|\eta(g_{i,t-1}^{(k)}-\nabla f_i(\boldsymbol{w}_{i,t-1}^{(k)})\|_2^2]\\
&\overset{(b)}{\leq}(1+\frac{1}{2T-1})\mathbb{E}[\|\boldsymbol{w}_{i,t-1}^{(k)}-\boldsymbol{w}_i^{(k)}\|_2^2]\\
&+(1+2T-1)\mathbb{E}[\|\eta(\nabla f_i(\boldsymbol{w}_{i,t-1}^{(k)})-\nabla f_i(\boldsymbol{w}_i^{(k)})\\
&+\nabla f_i(\boldsymbol{w}_i^{(k)}))\|_2^2]+E[\|\eta(g_{i,t-1}^{(k)}-\nabla f_i(\boldsymbol{w}_{i,t-1}^{(k)})\|_2^2]\\
&\leq(1+\frac{1}{2T-1})\mathbb{E}[\|\boldsymbol{w}_{i,t-1}^{(k)}-\boldsymbol{w}_i^{(k)}\|_2^2] \qquad (22)\\
&+4T\mathbb{E}[\|\eta(\nabla f_i(\boldsymbol{w}_{i,t-1}^{(k)})-\nabla f_i(\boldsymbol{w}_i^{(k)}))\|_2^2]\\
&+4T\mathbb{E}[\|\eta\nabla f_i(\boldsymbol{w}_i^{(k)}))\|_2^2]+E[\|\eta(g_{i,t-1}^{(k)}-\nabla f_i(\boldsymbol{w}_{i,t-1}^{(k)})\|_2^2]\\
&\leq(1+\frac{1}{2T-1}+4T\eta^2L^2)\mathbb{E}[\|\boldsymbol{w}_{i,t-1}^{(k)}-\boldsymbol{w}_i^{(k)}\|_2^2]\\
&+4T\mathbb{E}[\|\eta\nabla f_i(\boldsymbol{w}_i^{(k)}))\|_2^2]+\eta^2\chi^2\\
&\overset{(c)}{\leq}(1+\frac{1}{T-1})\mathbb{E}[\|\boldsymbol{w}_{i,t-1}^{(k)}-\boldsymbol{w}_i^{(k)}\|_2^2]\\
&+4T\mathbb{E}[\|\eta\nabla f_i(\boldsymbol{w}_i^{(k)}))\|_2^2]+\eta^2\chi^2,
\end{aligned}
$$

where (a) follows from Assumption 3 that $g_{i,t-1}^{(k)}$ is an unbiased estimation of $\nabla f_i(\boldsymbol{w}_{i,t-1}^{(k)})$. (b) follows from $(x+y)^2\leq(1+\frac{1}{K})x^2+(1+K)y^2$ and (c) follows from $\eta<\frac{1}{24TL}$.

Unrolling the recursion, we get

$$
\begin{aligned}
&\mathbb{E}[\|(\boldsymbol{w}_{i,t}^{(k)}-\boldsymbol{w}_i^{(k)})\|_2^2]\leq \qquad (23)\\
&\sum_{p=0}^{t-1}(1+\frac{1}{T-1})^p\Big[\eta^2\chi^2+4T\mathbb{E}[\|\eta\nabla f_i(\boldsymbol{w}_i^{(k)}))\|_2^2]\Big]\\
&\leq(T-1)\times[(1+\frac{1}{T-1})^T-1]\times\Big[\eta^2\chi^2+4T\mathbb{E}[\|\eta\nabla f_i(\boldsymbol{w}_i^{(k)}))\|_2^2]\Big]\\
&\leq 4T\eta^2\chi^2+16T^2\eta^2\mathbb{E}[\|\nabla f_i(\boldsymbol{w}_i^{(k)}))\|_2^2],
\end{aligned}
$$

where the last inequality follows from $(1+\frac{1}{T-1})^T\leq 5$ for $T>1$. ∎

The proof is Lemma 3 is as follows.

Proof: Provided that $\eta\leq\frac{1}{24TL}$, we have

$$
\begin{aligned}
&-\mathbb{E}[\eta\nabla f_i(\boldsymbol{w}_i^{(k)})^T\Delta_i^{(k)}]=-\eta\mathbb{E}<\nabla f_i(\boldsymbol{w}_i^{(k)}),\sum_{t=0}^{T-1}g_{i,t}^{(k)}>\\
&=-\eta\mathbb{E}<\nabla f_i(\boldsymbol{w}_i^{(k)}),\sum_{t=0}^{T-1}\nabla f_i(\boldsymbol{w}_{i,t}^{(k)})>\\
&\overset{(a)}{\leq}-\frac{T\eta}{2}\|\nabla f_i(\boldsymbol{w}_i^{(k)})\|_2^2-\frac{\eta}{2T}\mathbb{E}[\|\sum_{t=0}^{T-1}\nabla f_i(\boldsymbol{w}_{i,t}^{(k)})\|_2^2]\\
&+\frac{\eta}{2}\sum_{t=0}^{T-1}\mathbb{E}[\|(\nabla f_i(\boldsymbol{w}_{i,t}^{(k)})-\nabla f_i(\boldsymbol{w}_i^{(k)}))\|_2^2]\\
&\overset{(b)}{\leq}-\frac{T\eta}{2}\|\nabla f_i(\boldsymbol{w}_i^{(k)})\|_2^2-\frac{\eta}{2T}\mathbb{E}[\|\sum_{t=0}^{T-1}\nabla f_i(\boldsymbol{w}_{i,t}^{(k)})\|_2^2]\\
&+\frac{L^2\eta}{2}\sum_{t=0}^{T-1}\mathbb{E}[\|(\boldsymbol{w}_{i,t}^{(k)}-\boldsymbol{w}_i^{(k)})\|_2^2]\\
&\overset{(c)}{\leq}-\frac{T\eta}{2}\|\nabla f_i(\boldsymbol{w}_i^{(k)})\|_2^2-\frac{\eta}{2T}\mathbb{E}[\|\sum_{t=0}^{T-1}\nabla f_i(\boldsymbol{w}_{i,t}^{(k)})\|_2^2]\\
&+\frac{L^2\eta T}{2}[4T\eta^2\chi^2+16T^2\eta^2\|\nabla f_i(\boldsymbol{w}_i^{(k)})\|_2^2]\\
&=-T\eta(\frac{1}{2}-8\eta^2T^2L^2)\|\nabla f_i(\boldsymbol{w}_i^{(k)})\|_2^2+2\eta^3T^2L^2\chi^2\\
&-\frac{\eta}{2T}\mathbb{E}[\|\sum_{t=0}^{T-1}\nabla f_i(\boldsymbol{w}_{i,t}^{(k)})\|_2^2], \qquad (24)
\end{aligned}
$$

where (a) follows from $\|\sum_{i=1}^n\boldsymbol{a}_i\|_2^2\leq n\sum_{i=1}^n\|\boldsymbol{a}_i\|_2^2$, (b) follows from Assumption 2 and (c) follows from Lemma 2. Then according to Lemma 1, we have

$$
\begin{aligned}
&\mathbb{E}\Big[-\eta\nabla f_i(\boldsymbol{w}_i^{(k)})^T\Delta_i+\frac{1}{2}\eta^2L\|\Delta_i^{(k)}\|_2^2\Big] \qquad (25)\\
&\leq-T\eta(\frac{1}{2}-8\eta^2T^2L^2)\|\nabla f_i(\boldsymbol{w}_i^{(k)})\|_2^2+2\eta^3T^2L^2\chi^2\\
&+\frac{1}{2}\eta^2LT\chi^2+(\frac{\eta^2L}{2}-\frac{\eta}{2T})\mathbb{E}[\|\sum_{t=0}^{T-1}\nabla f_i(\boldsymbol{w}_{i,t}^{(k)})\|_2^2]\\
&\overset{(a)}{\leq}-T\eta(\frac{1}{2}-8\eta^2T^2L^2)\|\nabla f_i(\boldsymbol{w}_i^{(k)})\|_2^2+\frac{\eta^2TL}{2}(1+4\eta TL)\chi^2,
\end{aligned}
$$

where (a) follows from $\eta<\frac{1}{24TL}<\frac{1}{TL}$. ∎

The proof of Lemma 4 is as follows.

Proof:

$$
\begin{aligned}
&\mathbb{E}\Big\|\frac{1}{N}\sum_{i\in\mathcal{V}}\nabla f_i(\bar{\boldsymbol{w}}_g^{(k)})-\frac{1}{N}\sum_{i\in\mathcal{V}}\nabla f_i(\boldsymbol{w}_{i,g}^{(k)})\Big\|_2^2\\
&=\frac{1}{N^2}\mathbb{E}\Big\|\sum_{i\in\mathcal{V}}\big[\nabla f_i(\bar{\boldsymbol{w}}_g^{(k)})-\nabla f_i(\boldsymbol{w}_{i,g}^{(k)})\big]\Big\|_2^2\\
&\leq\frac{L^2}{N}\sum_{i\in\mathcal{V}}\mathbb{E}\|\bar{\boldsymbol{w}}_g^{(k)}-\boldsymbol{w}_{i,g}^{(k)}\|_2^2,
\end{aligned}
$$

where the last inequality follows from the L-smoothness of the local functions. ∎

The proof of Lemma 5 is as follows.

Proof:

$$
\begin{aligned}
\bar{\boldsymbol{w}}_g^{(k)}-\bar{\boldsymbol{w}}_g^{(k-1)}&=\frac{1}{N}\mathbf{W}_g^{(k)}\mathbf{1}_N-\frac{1}{N}(\mathbf{W}_g^{(k-1)})\mathbf{1}_N\\
&=\frac{1}{N}(\mathbf{W}_g^{(k-1)}-\eta\Xi_g^{(k-1)})\tilde{\mathbf{A}}\mathbf{1}_N-\frac{1}{N}(\mathbf{W}_g^{(k-1)})\mathbf{1}_N\\
&=-\frac{\eta}{N}\Xi_g\tilde{\mathbf{A}}\mathbf{1}_N=-\frac{\eta}{N}\sum_{i\in\mathcal{V}}\Delta_{i,g}^{(k-1)}. \qquad (26)
\end{aligned}
$$

where the last equality follows from Assumption 1 that $\tilde{\mathbf{A}}$ is a symmetric doubly stochastic. ∎

B. Proof of Theorem 1

Proof: The proof is inspired by the analysis in SGD method as in [50] and D-SGD method as in [30]. We treat the parameters $\boldsymbol{w}_{i,lu},\boldsymbol{\beta}_i$ and $\bar{\boldsymbol{w}}_g$ respectively so as to correspond to their update rules.

Without loss of generality, we consider one specific node i in the following proof. According to the smoothness of the local objective function, we have

$$
f_i(\tilde{\boldsymbol{v}}_i^{(k)})-f_i(\tilde{\boldsymbol{v}}_i^{(k-1)})\leq \qquad (27)
$$

$$\begin{aligned}
&\nabla f_i(\tilde{\boldsymbol{v}}_i^{(k-1)})^T(\tilde{\boldsymbol{v}}_i^{(k)}-\tilde{\boldsymbol{v}}_i^{(k-1)})+\frac{L}{2}\|\tilde{\boldsymbol{v}}_i^{(k)}-\tilde{\boldsymbol{v}}_i^{(k-1)}\|_2^2\\
&\overset{(a)}{=}\nabla f_i(\boldsymbol{w}_{i,lu}^{(k-1)})^T(\boldsymbol{w}_{i,lu}^{(k)}-\boldsymbol{w}_{i,lu}^{(k-1)})+\frac{L}{2}\|\boldsymbol{w}_{i,lu}^{(k)}-\boldsymbol{w}_{i,lu}^{(k-1)}\|_2^2\\
&\quad+\nabla f_i(\boldsymbol{\beta}_i^{(k-1)})^T(\boldsymbol{\beta}_i^{(k)}-\boldsymbol{\beta}_i^{(k-1)})+\frac{L}{2}\|\boldsymbol{\beta}_i^{(k)}-\boldsymbol{\beta}_i^{(k-1)}\|_2^2\\
&\quad+\nabla f_i(\bar{\boldsymbol{w}}_g^{(k-1)})^T(\bar{\boldsymbol{w}}_g^{(k)}-\bar{\boldsymbol{w}}_g^{(k-1)})+\frac{L}{2}\|\bar{\boldsymbol{w}}_g^{(k)}-\bar{\boldsymbol{w}}_g^{(k-1)}\|_2^2,
\end{aligned}$$

where (a) follows from the definition that $\tilde{\boldsymbol{v}}_i$ is the concatenation of the averaged $\bar{\boldsymbol{w}}_g$ and individual $\boldsymbol{w}_{i,lu}$, $\boldsymbol{\beta}_i$. According to (8) and (9), the first four terms in the right side of (27) is equal to

$$\begin{aligned}
&-\eta\nabla f_i(\boldsymbol{w}_{i,lu}^{(k-1)})^T\Delta_{i,lu}^{(k-1)}+\frac{1}{2}\eta^2L\|\Delta_{i,lu}^{(k-1)}\|_2^2\\
&-\eta\nabla f_i(\boldsymbol{\beta}_i^{(k-1)})^T\Delta_{i,b}^{(k-1)}+\frac{1}{2}\eta^2L\|\Delta_{i,b}^{(k-1)}\|_2^2.
\end{aligned}\tag{28}$$

Take the expectation of (27) on the both sides, we have

$$\begin{aligned}
&\mathbb{E}\big[f_i(\tilde{\boldsymbol{v}}_i^{(k)})-f_i(\tilde{\boldsymbol{v}}_i^{(k-1)})\big]\leq\\
&\mathbb{E}\Big[-\eta\nabla f_i(\boldsymbol{w}_{i,lu}^{(k-1)})^T\Delta_{i,lu}^{(k-1)}-\eta\nabla f_i(\boldsymbol{\beta}_i^{(k-1)})^T\Delta_{i,b}^{(k-1)}\\
&\quad+\frac{1}{2}\eta^2L\|\Delta_{i,lu}^{(k-1)}\|_2^2+\frac{1}{2}\eta^2L\|\Delta_{i,b}^{(k-1)}\|_2^2\Big]\\
&\quad+\mathbb{E}\Big[\nabla f_i(\bar{\boldsymbol{w}}_g^{(k-1)})^T(\bar{\boldsymbol{w}}_g^{(k)}-\bar{\boldsymbol{w}}_g^{(k-1)})\Big]\\
&\quad+\frac{L}{2}\mathbb{E}\Big[\|\bar{\boldsymbol{w}}_g^{(k)}-\bar{\boldsymbol{w}}_g^{(k-1)}\|_2^2\Big].
\end{aligned}\tag{29}$$

We now focus on the right side of (29). Under Lemma 3, the first expectation term can be bounded by

$$\begin{aligned}
&\mathbb{E}\Big[-\eta\nabla f_i(\boldsymbol{w}_{i,lu}^{(k-1)})^T\Delta_{i,lu}^{(k-1)}-\eta\nabla f_i(\boldsymbol{\beta}_i^{(k-1)})^T\Delta_{i,b}^{(k-1)}\\
&\quad+\frac{1}{2}\eta^2L\|\Delta_{i,lu}^{(k-1)}\|_2^2+\frac{1}{2}\eta^2L\|\Delta_{i,b}^{(k-1)}\|_2^2\Big]\\
&\leq-cT\eta(\|\nabla f_i(\boldsymbol{w}_{i,lu}^{(k-1)})\|_2^2+\|\nabla f_i(\boldsymbol{\beta}_i^{(k-1)})\|_2^2)\\
&\quad+\eta^2TL(1+4\eta TL)\chi^2.
\end{aligned}\tag{30}$$

We add both sides of (29) from $i=1$ to $i=N$, and derive the results by N, then we have

$$\begin{aligned}
&\frac{1}{N}\sum_{i\in\mathcal{V}}\mathbb{E}\big[f_i(\tilde{\boldsymbol{v}}_i^{(k)})-f_i(\tilde{\boldsymbol{v}}_i^{(k-1)})\big]\\
&\leq\frac{-cT\eta}{N}\sum_{i\in\mathcal{V}}(\|\nabla f_i(\boldsymbol{w}_{i,lu}^{(k-1)})\|_2^2+\|\nabla f_i(\boldsymbol{\beta}_i^{(k-1)})\|_2^2)\\
&\quad+\eta^2TL(1+4\eta TL)\chi^2\\
&\quad+\mathbb{E}\big[\frac{1}{N}\sum_{i\in\mathcal{V}}\nabla f_i(\bar{\boldsymbol{w}}_g^{(k-1)})^T(\bar{\boldsymbol{w}}_g^{(k)}-\bar{\boldsymbol{w}}_g^{(k-1)})\big]\\
&\quad+\frac{L}{2}\mathbb{E}\Big[\|\bar{\boldsymbol{w}}_g^{(k)}-\bar{\boldsymbol{w}}_g^{(k-1)}\|_2^2\Big].
\end{aligned}\tag{31}$$

When Assumption 2-4 and 6 hold, then the last two terms in (31) can be bounded by

$$\begin{aligned}
&\mathbb{E}\big[\frac{1}{N}\sum_{i\in\mathcal{V}}\nabla f_i(\bar{\boldsymbol{w}}_g^{(k-1)})^T(\bar{\boldsymbol{w}}_g^{(k)}-\bar{\boldsymbol{w}}_g^{(k-1)})\big]\\
&\quad+\frac{L}{2}\mathbb{E}\Big[\|\bar{\boldsymbol{w}}_g^{(k)}-\bar{\boldsymbol{w}}_g^{(k-1)}\|_2^2\Big]\\
&\overset{(a)}{=}-\eta\mathbb{E}\Big[\nabla F(\bar{\boldsymbol{w}}_g^{(k-1)})^T\bar{\Delta}_g^{(k-1)}\Big]+\frac{L\eta^2}{2}\mathbb{E}\|\bar{\Delta}_g^{(k-1)}\|_2^2\\
&=-\eta\mathbb{E}\Big[\nabla F(\bar{\boldsymbol{w}}_g^{(k-1)})^T(\bar{\Delta}_g^{(k-1)}-\sum_{t=0}^{T-1}\nabla F(\boldsymbol{w}_{g,t}^{(k-1)}))\Big]\\
&\quad-\eta\mathbb{E}\Big[\nabla F(\bar{\boldsymbol{w}}_g^{(k-1)})^T\sum_{t=0}^{T-1}\nabla F(\boldsymbol{w}_{g,t}^{(k-1)})\Big]\\
&\quad+\frac{L\eta^2}{2}\mathbb{E}\|\bar{\Delta}_g^{(k-1)}-\sum_{t=0}^{T-1}\nabla F(\boldsymbol{w}_{g,t}^{(k-1)})+\sum_{t=0}^{T-1}\nabla F(\boldsymbol{w}_{g,t}^{(k-1)})\|_2^2\\
&\overset{(b)}{\leq}-\frac{\eta T}{2}\mathbb{E}\|\nabla F(\bar{\boldsymbol{w}}_g^{(k-1)})\|_2^2-\frac{\eta T}{2}(\frac{1}{6}-LT\eta)\mathbb{E}\|\nabla F(\boldsymbol{w}_g^{(k-1)})\|_2^2\\
&\quad+\frac{4\eta TL^2}{3N}\sum_{i\in\mathcal{V}}\mathbb{E}\|\bar{\boldsymbol{w}}_g^{(k-1)}-\boldsymbol{w}_{i,g}^{(k-1)}\|_2^2\\
&\quad+\frac{1}{N}\big[T\eta^2L(4\kappa^2T+\chi^2)+6T^2\eta^3\chi^2L^2\big],
\end{aligned}\tag{32}$$

where (a) follows from the definition $\nabla F(\bar{\boldsymbol{w}}_g)\triangleq\frac{1}{N}\sum_{i\in\mathcal{V}}\nabla f_i(\bar{\boldsymbol{w}}_g)$ and Lemma 5. The derivation of (b) deals with $\sum_{t=0}^{T-1}\nabla F(\boldsymbol{w}_{g,t}^{(k-1)})$ by subtracting and then adding $T\nabla F(\boldsymbol{w}_g^{(k-1)})$. It also requires $\eta<\frac{1}{24TL}$. Its detailed derivation is omitted here.

Define $\bar{\tilde{\mathbf{A}}}_{s,k-1}=\prod_{l=s}^{k-1}\tilde{\mathbf{A}}$, $\mathbf{Q}=\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$ and $\rho_{s,k-1}=\|\bar{\tilde{\mathbf{A}}}_{s,k-1}-\mathbf{Q}\|$. Then based on Lemma 2-6 in [30], when the learning rate is sufficiently small such that $\eta<\frac{1}{24TL}$ and $\eta<\frac{1}{32TL\sqrt{C_K}}$, following some derivations under multiple steps of SGD and the adjustment of Lemma 2-6 in [30], we could also derive that

$$\begin{aligned}
&\frac{1}{N}\sum_{i\in\mathcal{V}}\sum_{k=1}^{K}\mathbb{E}\|\bar{\boldsymbol{w}}_g^{(k-1)}-\boldsymbol{w}_{i,g}^{(k-1)}\|_2^2\\
&\leq\Big[A_K\chi^2+B_KT(\kappa^2+T\eta^2\chi^2L^2)+\\
&\qquad\frac{C_KT}{K}\sum_{k=1}^{K}\mathbb{E}\big\|\nabla F(\boldsymbol{w}_g^{(k-1)})\big\|_2^2\Big]\times\frac{24\eta^2TK}{1-48\eta^2L^2T^2C_K},
\end{aligned}\tag{33}$$

where A_K, B_K, C_K are defined as follows.

$$\begin{aligned}
&A_K=\frac{1}{K}\sum_{k=1}^{K}\sum_{s=1}^{k-1}\rho_{s,k-1}^2,\quad B_K=\frac{1}{K}\sum_{k=1}^{K}\Big(\sum_{s=1}^{k-1}\rho_{s,k-1}\Big)^2,\\
&C_K=\max_{s\in[K-1]}\sum_{k=s+1}^{K}\rho_{s,k-1}\Big(\sum_{l=1}^{k-1}\rho_{l,k-1}\Big),
\end{aligned}$$

Recall the definition of the global objective function as (1), we have $F(\tilde{\boldsymbol{V}}^{(k)})=\frac{1}{N}\sum_{i\in\mathcal{V}}f_i(\tilde{\boldsymbol{v}}_i^{(k)})$ and the left side of (31) is equal to $F(\tilde{\boldsymbol{V}}^{(k)})-F(\tilde{\boldsymbol{V}}^{(0)})$. Then we add both the right side and left side of (31) from $k=1$ to K, we could derive the following expression.

$$\begin{aligned}
&\mathbb{E}[F(\tilde{\boldsymbol{V}}^{(K)})-F(\tilde{\boldsymbol{V}}^{(0)})]\\
&\leq-\frac{cT\eta}{N}\sum_{i\in\mathcal{V}}\sum_{k=0}^{K-1}\|\nabla f_i(\boldsymbol{v}_{i,ns}^{(k)})\|_2^2-cT\eta\sum_{k=0}^{K-1}\|\nabla F(\bar{\boldsymbol{w}}_g^{(k)})\|_2^2\\
&\quad-\frac{\eta T}{2}(\frac{1}{6}-LT\eta-128\eta^2T^2L^2C_K)\sum_{k=0}^{K-1}\mathbb{E}\|\nabla F(\boldsymbol{w}_g^{(k)})\|_2^2\\
&\quad+\eta^2TLK(1+4\eta TL)\chi^2\\
&\quad+\frac{1}{N}\big[T\eta^2LK(4\kappa^2T+\chi^2)+6T^2\eta^3\chi^2L^2K\big]
\end{aligned}$$

$$+ 64\eta^3T^2L^2K(A_K\chi^2 + B_KT(\kappa^2 + T\eta^2\chi^2L^2)). \quad (34)$$

requiring the learning rate satisfies $1 - 48\eta^2L^2T^2C_K \geq \frac{1}{2}$. If we constrain $\eta < \frac{1}{24TL}$ and $\eta < \frac{1}{32TL\sqrt{C_K}}$, we have

$$\frac{1}{6} - LT\eta - 128\eta^2T^2L^2C_K > 0,$$

and we can derive that

$$\begin{aligned}&\mathbb{E}\big[F(\tilde{\boldsymbol{V}}^{(K)}) - F(\tilde{\boldsymbol{V}}^{(0)})\big] \quad (35)\\ &\leq -\frac{cT\eta}{N}\sum_{i\in\mathcal{V}}\sum_{k=0}^{K-1}\big[\|\nabla f_i(\boldsymbol{v}_{i,ns}^{(k)})\|_2^2 + \|\nabla F(\bar{\boldsymbol{w}}_g^{(k)})\|_2^2\big] + E.\end{aligned}$$

Here

$$\begin{aligned}E &= \eta^2TLK(1+4\eta TL)\chi^2\\ &+ \frac{1}{N}\big[T\eta^2LK(4\kappa^2T+\chi^2) + 6T^2\eta^3\chi^2L^2K\big]\\ &+ 64\eta^3T^2L^2K(A_K\chi^2 + B_KT(\kappa^2 + T\eta^2\chi^2L^2)).\end{aligned}$$

We also have

$$\begin{aligned}-\frac{1}{N}\sum_{i\in\mathcal{V}}\|\nabla f_i(\boldsymbol{v}_{i,ns}^{(k)})\|_2^2 &\leq -\Big\|\frac{1}{N}\sum_{i\in\mathcal{V}}\nabla f_i(\boldsymbol{v}_{i,ns}^{(k)})\Big\|_2^2\\ &= -\big\|\nabla F(\boldsymbol{v}_{ns}^{(k)})\big\|_2^2.\end{aligned}$$

Rearrange the terms in (35), we can derive

$$\begin{aligned}&\frac{1}{K}\sum\nolimits_{k=0}^{K-1}\times\mathbb{E}\big[\|\nabla F(\boldsymbol{v}_{ns}^{(k)})\|_2^2 + \|\nabla F(\bar{\boldsymbol{w}}_g^{(k)})\|_2^2\big]\\ &\leq \frac{\mathbb{E}\big[F(\tilde{\boldsymbol{V}}^{(0)}) - F(\tilde{\boldsymbol{V}}^{(K)})\big]}{cTK\eta} + \frac{E}{cTK\eta},\end{aligned}$$

which implies that

$$\min_{k\in[K]}\mathbb{E}\big[\|\nabla F(\boldsymbol{v}_{ns}^{(k)})\|_2^2 + \|\nabla F(\bar{\boldsymbol{w}}_g^{(k)})\|_2^2\big] \leq \frac{F_0 - F_*}{cTK\eta} + \Phi, \quad (36)$$

where

$$\begin{aligned}\Phi = \frac{1}{c}\Big\{&\eta L(1+4\eta TL)\chi^2\\ &+ \frac{1}{N}\big[\eta L(4\kappa^2T+\chi^2) + 6T\eta^2\chi^2L^2\big]\\ &+ 64\eta^2TL^2(A_K\chi^2 + B_KT(\kappa^2 + T\eta^2\chi^2L^2))\Big\}.\end{aligned}$$

This completes the proof. ∎

C. Proof of Theorem 2

Proof: In Theorem 2, we derive the appropriate range of the fusion parameter μ so as to sufficiently satisfy Assumption 6. We first derive the expression of $\nabla f_i(\boldsymbol{w}_{i,lu})$ and the expression of $\nabla f_i(\boldsymbol{\beta}_i)$. According to the aggregation model (4) and the gradient back propagation, we can derive that $\nabla f_i(\boldsymbol{w}_{i,lu}) = \mu\nabla f_i(\boldsymbol{w}_{i,ns})$. Then we have

$$\|\nabla f_i(\boldsymbol{w}_{i,lu})\|_2^2 = \mu^2 \times \|\nabla f_i(\boldsymbol{w}_{i,ns})\|_2^2. \quad (37)$$

According to Assumption 6, we have $\|\nabla f_i(\boldsymbol{w}_{i,lu})\|_2^2 \leq G$ and $\|\nabla f_i(\boldsymbol{w}_{i,ns})\|_2^2 \leq G$. Then it can be derived that $0 \leq \mu \leq 1$.

Following we derive the lower bound of μ in the k-th round for node i, which we denote by $\mu_i^{(k)}$ for better clarification. For simplicity, we denote the gradient value of $\sigma_G(x_j)$ by $\sigma'_{G,j}$, where $x_j \triangleq {\boldsymbol{\beta}_i^{(k-1)}}^T(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)})$. And we denote $f_e(j) = \exp(\sigma_{G,j})$ for node i. Likewise, the gradient value of σ is denoted by σ'. In this paper, σ_G is the ELU activation function. Then it can be derived that the gradient's value at the $\boldsymbol{\beta}_i^{(k-1)}$ is:

$$\begin{aligned}\nabla f_i(\boldsymbol{\beta}_i^{(k-1)}) = \nabla f_i(\boldsymbol{w}_{i,ns}^{(k-1)}) \times \frac{(1-\mu_i^{(k)})\sigma'^T}{\big[\sum_{j\in\mathcal{N}_i} f_e(j)\big]^2}\times\\ \sum_{j\in\mathcal{N}_i}\sum_{l\in\mathcal{N}_i\setminus\{j\}}\Big\{\boldsymbol{w}_{j,ns}^{(k-1)} \times f_e(j) \times f_e(l)\times \quad (38)\\ \big[\sigma'^T_{G,j}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)}) - \sigma'^T_{G,l}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{l,ns}^{(k-1)})\big]\Big\}.\end{aligned}$$

The detailed derivation is omitted here for simplicity. Take the ℓ_2 norm on both sides of (38) and it can be derived that

$$\begin{aligned}\|\nabla f_i(\boldsymbol{\beta}_i^{(k-1)})\|_2^2 = (1-\mu_i^{(k)})^2\|\nabla f_i(\boldsymbol{w}_{i,ns}^{(k-1)})\|_2^2\times \quad (39)\\ \frac{\|\sigma'\|_2^2}{\big[\sum_{j\in\mathcal{N}_i} f_e(j)\big]^4}\Big\|\sum_{j\in\mathcal{N}_i}\sum_{l\in\mathcal{N}_i\setminus\{j\}}\Big\{\boldsymbol{w}_{j,ns}^{(k-1)} \times f_e(j) \times f_e(l)\times\\ \big[\sigma'_{G,j}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)}) - \sigma'_{G,l}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{l,ns}^{(k-1)})\big]\Big\}\Big\|_2^2.\end{aligned}$$

Denote

$$\begin{aligned}f_G &\triangleq \Big\|\sum_{j\in\mathcal{N}_i}\sum_{l\in\mathcal{N}_i\setminus\{j\}}\Big\{\boldsymbol{w}_{j,ns}^{(k-1)}f_e(j)f_e(l)\\ &\quad\times\big[\sigma'_{G,j}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)}) - \sigma'_{G,l}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{l,ns}^{(k-1)})\big]\Big\}\Big\|_2^2\\ &= \Big\|\sum_{j\in\mathcal{N}_i}\boldsymbol{w}_{j,ns}^{(k-1)}\sum_{l\in\mathcal{N}_i\setminus\{j\}}\Big\{f_e(j)f_e(l)\times\\ &\quad\big[\sigma'_{G,j}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)}) - \sigma'_{G,l}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{l,ns}^{(k-1)})\big]\Big\}\Big\|_2^2\\ &\overset{(a)}{\leq} d_i\sum_{j\in\mathcal{N}_i}\Big\|\boldsymbol{w}_{j,ns}^{(k-1)}\sum_{l\in\mathcal{N}_i\setminus\{j\}}f_e(j)f_e(l)\times\\ &\quad\big[\sigma'_{G,j}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)}) - \sigma'_{G,l}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{l,ns}^{(k-1)})\big]\Big\|_2^2\\ &\overset{(b)}{\leq} d_i\Big[\sum_{j\in\mathcal{N}_i}\|\boldsymbol{w}_{j,ns}^{(k-1)}\|_2^2\Big]\times\Big[\sum_{j\in\mathcal{N}_i}\Big\|\sum_{l\in\mathcal{N}_i\setminus\{j\}}f_e(j)f_e(l)\times\\ &\quad\big[\sigma'_{G,j}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)}) - \sigma'_{G,l}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{l,ns}^{(k-1)})\big]\Big\|_2^2\Big]\\ &\overset{(c)}{\leq} d_i(d_i-1)\Big[\sum_{j\in\mathcal{N}_i}\|\boldsymbol{w}_{j,ns}^{(k-1)}\|_2^2\Big]\times\Big[\sum_{j\in\mathcal{N}_i}\sum_{l\in\mathcal{N}_i\setminus\{j\}}\Big\|f_e(j)f_e(l)\\ &\quad\times\big[\sigma'_{G,j}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)}) - \sigma'_{G,l}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{l,ns}^{(k-1)})\big]\Big\|_2^2\Big]\\ &\overset{(d)}{\leq} d_i(d_i-1)\Big[\sum_{j\in\mathcal{N}_i}\|\boldsymbol{w}_{j,ns}^{(k-1)}\|_2^2\Big]\times\\ &\Big[\sum_{j\in\mathcal{N}_i}\sum_{l\in\mathcal{N}_i\setminus\{j\}}\Big\|f_e(j)f_e(l)\Big\|_2^2\Big]\\ &\times\Big[\sum_{j\in\mathcal{N}_i}\sum_{l\in\mathcal{N}_i\setminus\{j\}}\Big\|\sigma'_{G,j}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)})-\\ &\quad\sigma'_{G,l}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{l,ns}^{(k-1)})\Big\|_2^2\Big],\end{aligned}$$

where (a) and (c) follows from $\|\sum_{i=1}^n\boldsymbol{a}_i\|_2^2 \leq n\sum_{i=1}^n\|\boldsymbol{a}_i\|_2^2$, (b) and (d) follows from $\sum_i\|\boldsymbol{a}_i\|_2^2\|\boldsymbol{b}_i\|_2^2 \leq$

$\sum_i \|\boldsymbol{a}_i\|_2^2 \sum_i \|\boldsymbol{b}_i\|_2^2$.

With $\|\nabla f_i(\boldsymbol{w}_{i,ns})\|_2^2 \leq G$ according to Assumption 6, $\|\sigma'\|_2^2 \leq 1$ according to (14) in Assumption 6 and

$$0 \leq \Big[\frac{\sum_{j\in\mathcal{N}_i}\sum_{l\in\mathcal{N}_i\setminus\{j\}}[f_e(j)f_e(l)]^2}{(\sum_{m\in\mathcal{N}_i} f_e(m))^4}\Big] \leq 1, \tag{40}$$

then we have that

$$\|\nabla f_i(\boldsymbol{\beta}_i^{(k-1)})\|_2^2 \leq (1-\mu_i^{(k)})^2 G d_i(d_i-1)\Big[\sum_{j\in\mathcal{N}_i}\big\|\boldsymbol{w}_{j,ns}^{(k-1)}\|_2^2\Big]\times \Big[\sum_{j\in\mathcal{N}_i}\sum_{l\in\mathcal{N}_i\setminus\{j\}}\Big\|\sigma'_{G,j}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)})- \sigma'_{G,l}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{l,ns}^{(k-1)})\Big\|_2^2\Big]. \tag{41}$$

We denote the last two terms by

$$D_i^{(k)} \triangleq \Big[\sum_{j\in\mathcal{N}_i}\big\|\boldsymbol{w}_{j,ns}^{(k-1)}\|_2^2\Big]\times \Big[\sum_{j\in\mathcal{N}_i}\sum_{l\in\mathcal{N}_i\setminus\{j\}}\Big\|\sigma'_{G,j}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{j,ns}^{(k-1)})- \sigma'_{G,l}\cdot(\boldsymbol{w}_{i,ns}^{(k-1)}||\boldsymbol{w}_{l,ns}^{(k-1)})\Big\|_2^2\Big]. \tag{42}$$

Then it sufficiently satisfies Assumption 6 if $(1-\mu_i^{(k)})^2 G d_i(d_i-1)D_i^{(k)} \leq G$, following which we can derive the lower bound of μ for node i in the k-th round as

$$\mu_i^{(k)} \geq 1 - \frac{1}{\sqrt{d_i(d_i-1)D_i^{(k)}}}. \tag{43}$$

■

REFERENCES

[1] Z. Tian, Z. Zhang, and R. Jin, "Graph-Attention-Based Decentralized Edge Learning for Non-IID Data," accepted to appear in *IEEE ICC'23 Workshop*.

[2] X. Lian, C. Zhang, H. Zhang, C. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 5330–5340, 2017.

[3] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *arxiv: 1909.12488*, 2019.

[4] M. Khodak, M. Balcan, and A. Talwalkar, "Adaptive gradient-based meta-learning methods," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 5915–5926, 2019.

[5] A. Fallah, A. Mokhtari, and A. E. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 3557–3568, 2020.

[6] V. Smith, C. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 4424–4434, 2017.

[7] D. Caldarola, M. Mancini, F. Galasso, M. Ciccone, E. Rodola, and B. Caputo, "Cluster-driven graph federated learning over multiple domains." in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, pp. 2743–2752, 2021.

[8] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *Proc. 38th Int. Conf. Mach. Learning, Virtual Event*, pp. 2089–2099, 2021.

[9] S. Nikoloutsopoulos, I. Koutsopoulos, and M. K. Titsias, "Personalized federated learning with exact stochastic gradient descent," *arxiv: 2202.09848*, 2022.

[10] K. Wang, R. Mathews, C. Kiddon, H. Eichner, F. Beaufays, and D. Ramage, "Federated evaluation of on-device personalization," *arxiv: 1910.10252*, 2019.

[11] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of personalization techniques for federated learning," *arxiv: 2003.08673*, 2020.

[12] T. Yu, E. Bagdasaryan, and V. Shmatikov, "Salvaging federated learning by local adaptation," *arxiv: 2002.04758*, 2020.

[13] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," *arxiv: 2002.05516*, 2020.

[14] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," *arxiv: 2003.13461*, 2020.

[15] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," *arxiv: 2002.10619*, 2020.

[16] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 19586–19597, 2020.

[17] C. T. Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 21394–21405, 2020.

[18] X. Li and D. Zhan, "Fedrs: Federated learning with restricted softmax for label distribution non-iid data," in *Proc. 27th ACM Conf. Knowledge Discovery and Data Mining (KDD)*, pp. 995–1005, 2021.

[19] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated Learning on Non-IID Features via Local Batch Normalization," in *Int. Conf. Learning Representations*, 2021.

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

[21] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat.*, pp. 1273–1282, 2017.

[22] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

[23] 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.

[24] C. Shen, J. Xu, S. Zheng and X. Chen, "Resource rationing for wireless federated learning: Concept, benefits, and challenges," *IEEE Commun. Mag.*, vol. 59, no. 5, pp. 82–87, 2021.

[25] S. Luo, X. Chen, Q. Wu, et. al., "HFEL: Joint Edge Association and Resource Allocation for Cost-Efficient Hierarchical Federated Edge Learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6535–6548, 2020.

[26] G. Zhu, Y. Du, D. Gndz, and K. Huang "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *IEEE Trans. Wireless Commun.*, 2020.

[27] M. Chen, H. V.Poor, W. Saad, and S. Cui "Convergence Time Optimization for Federated Learning Over Wireless Networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, 2021.

[28] S. Scardapane, D. Wang, and M. Panella, "A decentralized training algorithm for echo state networks in distributed big data applications," *Neural Networks*, vol. 78, pp. 65–74, 2016.

[29] Z. Jiang, A. Balu, C. Hegde, and S. Sarkar, "Collaborative deep learning in fixed topology networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 5904–5914, 2017.

[30] X. Li, W. Yang, S. Wang, and Z. Zhang, "Communication-efficient local decentralized SGD methods," *arxiv: 1910.09126*, 2019.

[31] A. Balu, Z. Jiang, S. Y. Tan, C. Hegde, Y. M. Lee, and S. Sarkar, "Decentralized deep learning using momentum-accelerated consensus," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2021.

[32] W. Liu, L. Chen, and W. Zhang, "Decentralized federated learning: Balancing communication and computing costs," *IEEE Trans. Signal and Inform. Process. over Networks*, vol. 8, pp. 131–143, 2022.

[33] Y. Esfandiari, S. Y. Tan, Z. Jiang, A. Balu, E. Herron, C. Hegde, and S. Sarkar, "Cross-Gradient Aggregation for Decentralized Learning from Non-IID Data," in *Proc. 38th Int. Conf. Mach. Learning*, 2021.

[34] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "D^2: Decentralized training over decentralized data," in *arxiv: 1803.07068*, 2018.

[35] R. Xin, U. A. Khan and S. Kar, "An Improved Convergence Analysis for Decentralized Online Stochastic Non-Convex Optimization," in *IEEE Trans. Signal Process.*, vol. 69, pp. 1842–1858, 2021.

[36] J. Zhang, and K. You, "Decentralized stochastic gradient tracking for empirical risk minimization," *arxiv: 1909.02712*, 2019.

[37] S. Pu, and A. Nedi, "Distributed stochastic gradient tracking methods." in *Math. Program.*, vol. 187, pp. 409–457, 2021.

[38] X. Lian, W. Zhang, C. Zhang, and J. Liu, "Asynchronous decentralized parallel stochastic gradient descent," in *Proc. 35th Int. Conf. Mach. Learning*, 2018.

[39] V. Zantedeschi, A. Bellet, and M. Tommasi, "Fully decentralized joint learning of personalized models and collaboration graphs," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2020.

[40] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Networks Learn. Syst*, 2019.

[41] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[42] C. Meng, S. Rambhatla, and Y. Liu, "Cross-node federated graph neural network for spatio-temporal data modeling," in *Proc. 27th ACM Conf. Knowledge Discovery and Data Mining (KDD)*, 2021.

[43] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza and J. Chanussot, "Graph Convolutional Networks for Hyperspectral Image Classification," in *IEEE Trans. Geosci. and Remote Sens.*, vol. 59, no. 7, pp. 5966-5978, July 2021.

[44] D. Hong, N. Yokoya, J. Chanussot, J. Xu, and X. X. Zhu, "Learning to propagate labels on graphs: An iterative multitask regression framework for semi-supervised hyperspectral dimensionality reduction," in *ISPRS J. Photogramm. Remote Sens.*, vol. 158, pp. 35-49, 2019.

[45] D. Hong, N. Yokoya, N. Ge, J. Chanussot, and X. X. Zhu, "Learnable manifold alignment (LeMA): A semi-supervised cross-modality learning framework for land cover and land use classification," in *ISPRS J. Photogramm. Remote Sens.*, vol. 147, pp. 193-205, 2019.

[46] K. Yan, M. Zhou, L. Liu, C. Xie, and D. Hong, "When Pansharpening Meets Graph Convolution Network and Knowledge Distillation,," in *IEEE Trans. Geosci. and Remote Sens.*, vol. 60, pp. 1-15, 2022.

[47] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learning Representations*, 2018.

[48] H. Ryu, H. Shin, and J. Park, "Multi-agent actor-critic with hierarchical graph attention network," in *Proc. 34th Conf. Artificial Intell. (AAAI)*, 2020. arxiv:

[49] D. Hong, N. Yokoya, J. Chanussot and X. X. Zhu, "An Augmented Linear Mixing Model to Address Spectral Variability for Hyperspectral Unmixing," in *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1923–1938, 2019.

[50] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Rev.*, vol. 60, no. 2, pp. 223–311, 2018.

[51] H. Yang, M. Fang and J. Liu, "Achieving Linear Speedup with Partial Worker Participation in Non-IID Federated Learning," in *Proc. 9th Int. Conf. Learning Representations*, 2021.

[52] Z. Tian, Z. Zhang, J. Wang, X. Chen, W. Wang, and H. Dai, "Distributed admm with synergetic communication and computation," in *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 501–517, 2021.

[53] A. Krizhevsky, G. Hinton, et.al., "Learning Multiple Layers of Features from Tiny Images,"2009

[54] S. Caldas, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar, "LEAF: A benchmark for federated settings," *arxiv: 1812.01097*, 2018.

[55] T. Tieleman and G. Hinton, "Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning, Lecture 6.5-RMSProp*.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, 2016.