# Towards Fair Graph Federated Learning via Incentive Mechanisms

**Chenglu Pan[1,2*†], Jiarong Xu[*2‡], Yue Yu[2], Ziqi Yang [1], Qingbiao Wu[1], Chunping Wang[3], Lei Chen[3], Yang Yang[1]**

[1]Zhejiang University, [2]Fudan University, [3]FinVolution Group
{chenglupan,yangziqi,qbwu,yangya}@zju.edu.cn, jiarongxu@fudan.edu.cn, yuyue22@m.fudan.edu.cn,
{wangchunping02,chenlei04}@xinye.com

## Abstract

Graph federated learning (FL) has emerged as a pivotal paradigm enabling multiple agents to collaboratively train a graph model while preserving local data privacy. Yet, current efforts overlook a key issue: agents are self-interested and would hesitant to share data without fair and satisfactory incentives. This paper is the first endeavor to address this issue by studying the incentive mechanism for graph federated learning. We identify a unique phenomenon in graph federated learning: the presence of agents posing potential harm to the federation and agents contributing with delays. This stands in contrast to previous FL incentive mechanisms that assume all agents contribute positively and in a timely manner. In view of this, this paper presents a novel incentive mechanism tailored for fair graph federated learning, integrating incentives derived from both model gradient and payoff. To achieve this, we first introduce an agent valuation function aimed at quantifying agent contributions through the introduction of two criteria: gradient alignment and graph diversity. Moreover, due to the high heterogeneity in graph federated learning, striking a balance between accuracy and fairness becomes particularly crucial. We introduce motif prototypes to enhance accuracy, communicated between the server and agents, enhancing global model aggregation and aiding agents in local model optimization. Extensive experiments show that our model achieves the best trade-off between accuracy and the fairness of model gradient, as well as superior payoff fairness.

## 1 Introduction

Graph data is ubiquitous, exhibiting diverse and generic connectivity patterns, yet a notable portion is distributed or isolated among distinct agents (*e.g.*, companies, research institutions). Unlocking the potential of this "closed graph data" represents a hidden gold mine in the era of big data. Recent advances of graph federated learning offer opportunities for collaboration among multiple agents without compromising data privacy, with each agent conducting local model training and sharing their updates/gradients with a global model on a server (McMahan et al. 2017; Karimireddy et al. 2020; Li et al. 2020; Zhao et al. 2018; Xie et al. 2021; Zhang et al. 2021b; Xie, Xiong, and Yang 2023; Gu et al. 2023; Zhang et al. 2021a). For instance, financial companies with their own transaction networks can join forces to enhance fraud detection model by participating in a graph federation.

However, in reality, the agents are self-interested and may not be cooperative if all agents receive the same model while their contributions differ. This implies that to achieve a competitive global graph model, there is a strong need to establish a fair graph federated learning framework that incentives agents to provide high-quality information. In view of this, several studies have explored incentive mechanisms for FL, with a primary focus on image domain (Xu and Lyu 2021; Xu et al. 2021; Deng et al. 2021; Gao et al. 2021).

These previous works almost assume that all agents contribute positively and in a timely manner (as depicted in Figure 1(a)). However, our observations in Figure 1(b) uncover substantial differences in graph federated learning, even though all the participants are honest: (1) Specific agents, like agent 4 and 5, negatively impact the entire federation; (2) The contributions of certain agents exhibit delays; for instance, agent 1 initially poses a harmful impact but contributes the most after multiple rounds of training. *As the first contribution, we unveil a unique phenomenon in the context of graph federated learning: the presence of agents posing potentially harm and contributing with delays.*

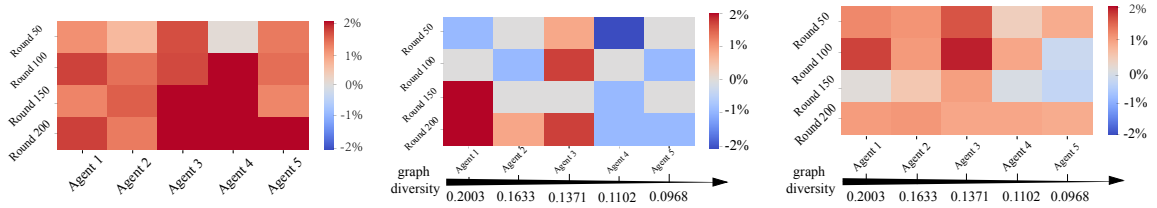
In light of the uniqueness of graph federated learning, we believe that an ideal incentive mechanism for graph federated learning should simultaneously satisfy two criteria: the capability of (1) *rewarding contributing agents while penalizing those causing harm*, and (2) *offering post-hoc compensation to agents with delayed contributions.*

However, the primary challenge lies in designing a comprehensive incentive mechanism framework that satisfies both criteria. Existing incentive mechanisms mainly fall into two lines, each with inherent limitations to the desired one. One line aims to allocate model gradients—the aggregated parameter updates/gradients that agents download from the server—to reward contributing agents (Xu and Lyu 2021; Xu et al. 2021). Yet, they cannot penalize agents that may potentially do harm and compensate those with delayed contributions. The other line focuses on the allocation of payoff (such as monetary or computational resources) (Gao et al. 2021; Yu et al. 2020), but fails to provide training-time incentives during training, potentially diminishing agent mo-

[*]These authors contributed equally.

[†]This work was done when the author was a visiting student at Fudan University.

[‡]Corresponding author.




(a) FedAvg in image dataset CIFAR 10 (b) FedAvg in graph dataset PROTEINS (c) Our model in graph dataset PROTEINS

Figure 1: The contribution of each agent in different communication rounds. Positive contributions are highlighted in red, while negative contributions are indicated in blue. The contribution of agent $i$ in $t$-th round is computed as the relative improvement in performance on the server's global test set in $t$-th round when all agents participate compared to when agent $i$ is excluded. It is observed that in Figure 1(b) of graph federated learning, certain agents make negative contributions, while in Figure 1(a) of image federated learning, all agents make positive contributions. Additionally, some agents in graph federated learning (Figure 1(b)) initially pose a harmful impact but contribute positively to the federation after multiple rounds of training.

| | *Feature 1*: reinforcement type | *Feature 2*: workflow phase |
|---|---|---|
| model gradient | reward | training-time |
| payoff | reward & punishment | post-hoc |

Table 1: Features of our proposed incentive mechanism for fair graph federated learning.

tivation.

To address this challenge, we put forth a novel incentive mechanism tailored for fair graph federated learning, outlined in Table 1. This mechanism seamlessly integrates the allocation of both *model gradients* and *payoff*, whereby model gradients can function as rewards for contributing agents and offer acknowledgment in a training-time manner, while payoff serve a dual purpose by not only imposing penalties on agents who may potentially do harm but compensating agents with delayed contributions.

Within this framework, the subsequent challenge is how to value an agent's contribution/harm to graph federation, such that the allocation can be conducted according to the agent value. Previous works typically value agents on an auxiliary validation set (Jia et al. 2019; Song, Tong, and Wei 2019; Wang et al. 2020), but selecting a validation set that is accessible and agreed upon by all agents poses challenges. To address this issue, we introduce an agent valuation function, incorporating two criteria: *gradient alignment* and *graph diversity*. Gradient alignment exploit the similarity between local gradients and the server's global gradients. However, relying solely on gradient alignment could underestimate agents with delayed contributions. This is evident in Figure 1(b), where agents with delayed contributions often exhibit high diversity. We therefore introduce an additional criterion: the graph diversity of agents' local data.

The remaining challenge lies in how to enhance the accuracy-fairness trade-off; this is critical especially in the presence of agents potentially do harm in graph federated learning. Previous endeavors have attempted to aggregate the local gradients based on the agent value (McMahan et al. 2017; Xu et al. 2021). However, this approach falls short of ensuring the model quality in graph federated learning, due to the high heterogeneity in graph data (Xie et al. 2021). This paper introduces a novel concept of *motif prototypes* as a reference coordinate between server and agents, facilitating not only the server's role in global model aggregation but also agents in optimizing their local models.

Our major contributions can be summairzed as follows:

- **Problem.** To the best of our knowledge, we are *the first* work to study the incentive mechanism for graph federated learning. We unveil a unique phenomenon in the context of graph federated learning: the presence of agents posing potentially harm and contributing with delays.
- **Method.** We propose a novel incentive mechanism tailored for fair graph federated learning that provides both model gradients and payoff for agents. Particularly, we propose to value the agents based on gradient alignment and graph diversity, and introduce motif prototypes to enhance accuracy-fairness trade-off.
- **Experiment.** In-depth experiments conducted across various settings reveal that our model not only demonstrates superiority in the allocation of model gradients and payoff, but also achieves the most favorable trade-off between fairness and accuracy among state-of-the-art baselines.

## 2 Preliminary

**Vanilla graph federated learning.** Vanilla graph federated learning involves $N$ *honest* agents, where each agent $i$ holds a local graph dataset denoted as $D_i$, comprising a set of graphs. The objective is to learn a shared global model, typically a graph neural network (GNN), across all clients, which can be formulated as

$$\min_{(\omega_i,\omega_2\cdots,\omega_N)} \sum_{i=1}^{N} \frac{|D_i|}{|D|} L(\omega_i; D_i), \tag{1}$$

where $\omega_1 = \omega_2 = \cdots = \omega_N$, $\omega_i$ represents the model parameters of agent $i$, $L(\omega_i; D_i)$ is the loss function of graph classification for local training in agent $i$, $|D_i|$ is the number of instances in client $i$, and $|D|$ is the total number of instances over all clients.

This objective is achieved through a two-step process: aggregation and distribution. In the aggregation step, the gradients uploaded by the individual agents, $\{\mathbf{u}_0^t, \cdots, \mathbf{u}_N^t\}$, are combined in the server to obtain the aggregated gradient $\mathbf{u}_{\mathcal{N}}^t$. This aggregation is performed using a weighted average, and the weights are proportional to the sizes of the local datasets:

$$\mathbf{u}_{\mathcal{N}}^t = \sum_{i=1}^{N} \frac{|D_i|}{|D|} \mathbf{u}_i^t. \tag{2}$$

In the distribution step, the aggregated gradient $\mathbf{u}_{\mathcal{N}}^t$ is sent back to each agent, and each agent $i$ receives the same gradient $\mathbf{u}_{\mathcal{N}}^t$ as a reward in $t$-th communication round.

Our current focus is solely on addressing fairness concerns related to honest agents, without considering the examination of potential cheating behaviors among agents.

**Shapely value.** In the context of FL, the Shapley value can be applied to assess the value of individual agents in the collaborative federation (Xu et al. 2021; Song, Tong, and Wei 2019). It provides a way to quantify the contribution of each agent in improving the overall performance in federation.

The Shapley value, originally introduced in cooperative game theory (Shapley 1997), is a widely used concept for evaluating the contribution of individual players in a coalition game. It measures the expected marginal value that a player brings when joining different coalitions, considering all possible permutations of players.

**Definition 1** (Shapley Value)**.** *Let $\mathcal{N}$ denote the set of all agents (*i.e.*, the grand coalition), a coalition $\mathcal{S} \subseteq \mathcal{N}$ is the subset of $\mathcal{N}$, $\Pi_{\mathcal{N}}$ is the set of all possible permutation of $\mathcal{N}$. For a given permutation $\pi \in \Pi_{\mathcal{N}}$, $\mathcal{S}_{\pi,i}$ represents the coalition of agents preceding agent $i$ in the permutation. The gradient-based Shapley value of agent $i \in \mathcal{N}$ is defined as*

$$\varphi_i := \frac{1}{N!} \sum_{\pi \in \Pi_{\mathcal{N}}} \left[ \nu\left(\mathcal{S}_{\pi,i} \cup \{i\}\right) - \nu\left(\mathcal{S}_{\pi,i}\right) \right], \tag{3}$$

*where $\nu(\mathcal{S})$ represents the value function associated with coalition $\mathcal{S}$.*

## 3 Methodology

This section introduces our incentive mechanism framework for promoting fairness in graph federated learning; see Figure 2 for an overview. We first provide an overview of our framework that encompasses the allocation of both model gradients and payoff in § 3.1. To implement this framework, we tackle two pivotal questions: how to assess an agent's contribution or potential harm within the graph federation context in § 3.2, and how to ensure the accuracy of graph federated learning in § 3.3.

### 3.1 Overview Framework

Our goal is to enhance fairness in graph federation by combining model gradients and payoff allocation mechanisms (Table 1) to reward contributing agents, penalize agents with potential harm, and provide post-hoc compensation for delayed contributions.

Before introducing the whole framework, we define the agent value, $r_i^t$, which indicates the contribution of agent $i$ in the $t$-th communication round (details about the agent value could be found in § 3.2). We proceed to elaborate on these incentive mechanisms as below.

**Model gradients allocation.** In the vanilla FL framework, all agents download the same gradients from the server (McMahan et al. 2017; Li et al. 2020), but this is unfair when dealing with agents with differing values/contributions. To address this issue, we propose that the server allocates global gradients to each agent based on their individual value, a mechanism termed model gradients allocation.

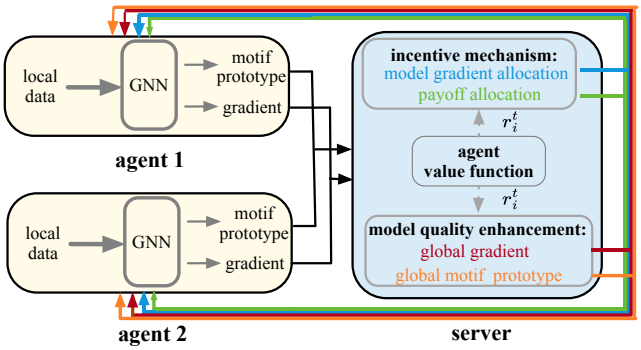



Figure 2: The proposed framework of fair graph federated learning.

To achieve this, we employ a *sparsifying gradient approach* for model gradients allocation, drawing inspiration from prior works like (Lyu et al. 2020; Xu and Lyu 2021; Xu et al. 2021). This strategy entails rewarding agents who contribute more with denser gradient rewards, while those with less contributions receive sparser gradients. Agents who potentially do harm, in turn, are assigned zero gradients.

Specifically, to differentiate the quality of the gradients allocated to different agents, we can selectively sparsify the server's aggregated global gradient. The sparsifying trick is achieved by a mask operation: when an agent's value is higher, we zero out fewer of the smaller components of the global gradient, resulting in a higher-quality gradient reward. The gradient downloaded by agent $i$ in $t$-th communication round with value $r_i^t$ is

$$\text{gradient}_i^t = \text{mask}(\mathbf{u}_{\mathcal{N}}^t, \lfloor D \tanh(\beta r_i^t) / \max_{j \in \mathcal{N}} \tanh(\beta r_j^t) \rfloor), \tag{4}$$

where the operator $\text{mask}(\mathbf{u}_{\mathcal{N}}^t, x)$ returns the largest $\max(0, x)$ components of $\mathbf{u}_{\mathcal{N}}^t$, $D$ is the total number of components in the global gradient. The hyper-parameter $\beta \geq 1$ controls the emphasis of fairness in FL: a smaller $\beta$ indicates a higher emphasis on fairness, as in this case agents with lower values will receive gradients of lower quality. In the extreme case of $\beta = \infty$, we revert to the vanilla FL.

Note that in scenarios where an agent offers no contribution or potentially do harm (*i.e.*, $r_i^t \leq 0$), no gradient is allocated to it. This policy is adopted due to the impracticality of using model gradients to penalize harmful agents, for example, by applying inverse gradients, as doing so would contradict fundamental principles of federated learning.

**Payoff allocation.** To empower the framework's ability to penalize agents and compensate those with delayed contributions, we propose a scheme to allocate payoff (*e.g.*, money or computation resources).

On one hand, agents whose actions may have a negative impact on the federation, are subjected to penalties in payoff. For agent $i$ in the $t$-th round, if $r_i^t < 0$, indicating that the agent is detrimental to the overall federation, we impose a payoff punishment $\text{payoff}_i^t = r_i^t < 0$ on the agent.

On the other hand, agents with delayed contributions are provided payoff compensation based on their performance history. This ensures that agents who have faced delays in their contributions are still acknowledged and rewarded accordingly. By examining the agent values in pre-

vious rounds, we can estimate the agent's delayed contribution. Specifically, we consider the agents, who have initially lower values but catch up in value over time, as those experiencing delayed contribution. To compensate, we allocate the payoff compensation for agent $i$ in the $t$-th round (denoted as $\mu_i^t$) as the difference between the value in the current round and the average value of the previous rounds:

$$\mu_i^t = \max(r_i^t - \frac{1}{t-1}\sum_{m=1}^{t-1} r_i^m, 0). \tag{5}$$

Given the agent value and the calculated payoff compensation, the payoff to agent $i$ in $t$-th communication round is

$$\text{payoff}_i^t = \begin{cases} r_i^t, & r_i^t < 0 \\ r_i^t + \mu_i^t, & \text{otherwise} \end{cases}, \tag{6}$$

$$\text{payoff}_i^t \leftarrow \frac{\text{payoff}_i^t}{\sum_{i=1}^N \text{payoff}_i^t} B, \tag{7}$$

where $B$ is the budget of payoff.

Given the overall framework above, two challenges remain to be solved: (1) how to define the agent value $r_i^t$; (2) while the allocation mechanism contributes to fairness, it should also guarantee the model accuracy. We will address these two problems in the following two sections.

### 3.2 Agent Valuation Function

The value of an agent is typically determined by its accuracy on an auxiliary validation set (Jia et al. 2019; Song, Tong, and Wei 2019; Wang et al. 2020). However, selecting a validation set that is accessible and agreed upon by all agents can be challenging. To decouple the agent valuation from validation, we introduce two criteria: *gradient alignment* and *graph diversity*.

**Gradient alignment.** Most approaches that utilize the Shapley value in FL typically define the value function based on an auxiliary validation set that is shared and agreed upon by all agents. To overcome the challenge and inspired from (Xu and Lyu 2021; Xu et al. 2021), we propose to utilize the gradient information as the value function for computing the Shapley value, namely *gradient-based Shapley value*, instead of using an auxiliary validation set. The gradient-based Shapley value is defined as that in Definition 1 by assigning the value function $\nu$ to be

$$\nu(\mathcal{S}) = \cos(\mathbf{u}_\mathcal{S}, \mathbf{u}_\mathcal{N}) = \langle \mathbf{u}_\mathcal{S}, \mathbf{u}_\mathcal{N} \rangle / (||\mathbf{u}_\mathcal{S}|| \cdot ||\mathbf{u}_\mathcal{N}||), \tag{8}$$

where $\nu(\mathcal{S})$ is defined as the cosine similarity between the gradient of coalition $\mathcal{S}$ (*i.e.*, local gradient from agent), denoted as $\mathbf{u}_\mathcal{S}$, and the gradient of the grand coalition $\mathcal{N}$ (*i.e.*, global gradient in server), denoted as $\mathbf{u}_\mathcal{N}$. The gradient-based Shapley value measures the gradient contribution of each agent in the federation. A positive value indicates that the agent makes a greater contribution, as its gradient is positively aligned with the global gradient. A negative value implies that the agent's gradient is in the opposite direction of the global gradient. In this case, the agent is considered as potentially detrimental to the federation.

However, calculating the exact gradient-based Shapley value of an agent costs $\mathcal{O}(2^N)$. To address this issue, we find that cosine similarity between the agent's local gradient and the server's global gradient could be used as an approximation. The error between approximated value and exact value can be bounded, as illustrated in (Xu et al. 2021). Therefore, the gradient-based Shapley value of agent $i$ in $t$-th communication round is approximated as

$$\varphi_i^t \approx \cos(\mathbf{u}_i^t, \mathbf{u}_\mathcal{N}^t), \tag{9}$$

where $\mathbf{u}_i^t$ is the local gradient that agent $i$ uploads to the server in communication round $t$, and $\mathbf{u}_\mathcal{N}^t$ is the global gradient in communication round $t$. The formal theorem and proofs can be found in Appendix A.1.

**Graph diversity.** Merely considering gradient alignment is insufficient for two reasons. First, agents with delayed contributions would face undervaluation if their assessment were solely based on gradient alignment. This occurs because agents equipped with diverse graph data hold substantial potential for delayed contributions, yet they may struggle to attain perfectly aligned gradients initially (as depicted in Figure 1(b)). Second, diverse graphs encompass a wide range of structural patterns that can be universally shared among agents, leading to better generalization (Tan et al. 2023). Consequently, to provide a more comprehensive valuation of agents, we additionally introduce the criterion of graph diversity.

However, quantifying the diversity of graph data is not straightforward due to the complex structure. Existing works use the number of subgraphs, data size or heuristic measures to quantify the diversity of graph (Yuan et al. 2021; Frasca et al. 2021; Xu et al. 2023), but overlook the inherent structural patterns within graph. To address this issue, we propose the use of motifs as a means to represent the diversity of graph data. Motifs are representative structural patterns in graphs that can reveal important information about underlying structure (Zhang et al. 2021c). Therefore, for each agent, we define graph diversity as the volume of motifs contained in agent's local graph.

**Definition 2** (Graph Diversity)**.** *The graph diversity $d_i$ of agent $i$ is defined as the ratio of the number of unique motifs categories in the agent's local graph data $k_i$ to the number of unique motifs categories in the entire data contributed by all agents K, which can be formulated as $d_i = \frac{k_i}{K}$.*

Note that the motif information necessary for computing graph diversity (*i.e.*, the motif prototypes elaborated in § 3.3) will be communicated between the agents and the server, rather than transmitting the actual graph diversity values.

Another advantage of introducing graph diversity is that it can help prevent agents from falling into the trap of converging to similar models. This is because, from the perspective of game theory, if only model gradients are allocated, agents might fall into this trap.

**Agent value updates.** Finally, *the value of agent* is defined by considering two aspects: (1) the incorporation of graph alignment and graph diversity, and (2) the consideration of both the current round's assessment and the historical assessment. First, the graph alignment $\varphi_i^t$ and graph diversity $d_i$ are combined by weight. Then, to better assess the agent

value, both the historical assessment and the value on current round should be considered. Specifically, the server updates the agent value in round $t$ via a moving average of the current value and the historical value $r_i^{t-1}$. To ensure that the values of all agents sum up to 1 in $t$-th round, the server normalizes the agent values in the last step. In summary, the update of agent value could be decomposed as:

$$r_i^t = r_i^{t-1} + \alpha_1(\varphi_i^t + \alpha_2 d_i),\ r_i^t \leftarrow r_i^t / \sum_{j\in\mathcal{N}} r_j^t, \quad (10)$$

where $\alpha_1$ can be viewed as a trade-off of the value from current round and the historical rounds, $\alpha_2$ denotes a trade-off parameter between graph alignment and graph diversity, and $r_i^0 = 1/N$ is the initial value of the agent.

### 3.3 Model Quality Enhancement

In graph federated learning, striking a balance between fairness and accuracy is of paramount importance, particularly when considering agents that have the potential to compromise the model performance due to the inherent heterogeneity in graph data. To ensure the high-accuracy federated model, there is a need for establishing a reference coordinate that can be communicated between the server and the agents. This is necessary for the server to convey its intentions and requirements to the agents, while offering agents the opportunity to refine their local models and strive for improvement.

This section first introduces a concept of motif prototype as such a reference coordinate, then presents how the server and agents utilize motif prototype for self enhancement.

**Motif prototypes.** The underlying reason graph federated learning suffers from poor accuracy lies in the challenge of transferring knowledge across different agents' graph data. This difficulty is hard to address by simply aggregating the gradients of local models on the server, as did in most FL methods. Considering these limitations, we introduce motifs, which are sub-structures with rich structural information (Milo et al. 2002). As a specific type of motif contains similar structural information across graphs, motifs can effectively operate as transferable patterns across graphs, even when encountered in agents with varied data distributions. With this transferable pattern, we introduce the concept of motif prototypes, which facilitate the transfer of knowledge across agents, ultimately resulting in enhanced accuracy. We formally define the motif prototypes as follows:

**Definition 3** (Motif Prototypes)**.** *For agent $i$, suppose that the agent's local data involve $K_i$ unique motifs. For the $k$-motif in agent $i$, we define the corresponding motif prototype as the mean of the embedding vectors of the graph instances containing $k$-th motif,* i.e.*,*

$$\mathbf{c}_{i,k}^t = \frac{1}{|D_{i,k}|} \sum_{G\in D_{i,k}} f_{\omega_i^t}(G), \quad (11)$$

*where $D_{i,k}$ is the subset of $D_i$ that is comprised of graph instances that contain $k$-th motif, $f_{\omega_i^t}(G)$ and $\omega_i^t$ are the embedding of graph instance $G$ and the parameter of embedding layers of agent $i$ in $t$-th round, respectively.*

In each communication round, except for model gradients, the motif prototypes are also communicated between the server and agents to ensure a high-accuracy federated model. Specifically, the agents upload their local motif prototypes to the server. After receiving the local motif prototypes from the agents, the server then aggregates the local motif prototypes from all agents to obtain the global motif prototypes. Once the global motif prototypes are obtained, the server distributes them back to the agents.

It is noteworthy that the communication of prototypes would not entail much privacy leakage. This is because motif prototypes are 1D-vectors derived via the computation of average statistics from the low-dimensional representations of graph instances, which is an irreversible process (Michieli and Ozay 2021; Tan et al. 2022).

**Value-based global model aggregation.** To ensure the quality of the global model, we propose a value-based global model aggregation approach. This approach aims to aggregate both the introduced motif prototypes and the gradients of agents based on their value $r_i^t$. Specifically, we assign higher weights to agents with higher agent values, while excluding agents with negative values to prevent potential harm to the global model.

We first introduce the aggregation process for motif prototypes at the server. Suppose that there are $K$ unique motifs in total. We aggregate the local motif prototypes from the agents based on their values $r_i^t$. Specifically, the global motif prototype of $k$-th motif in $t$-th round is defined as

$$\mathbf{c}_{\mathcal{N},k}^t = \frac{\sum_{i\in\mathcal{N}_k} \mathrm{ReLU}(r_i^t)\cdot \mathbf{c}_{i,k}^t}{\sum_{i\in\mathcal{N}_k} \mathrm{ReLU}(r_i^t)}, \quad (12)$$

where $\mathcal{N}_k$ denotes the set of agents that have motif $k$.

A similar strategy is employed when aggregating model gradients. The global gradient in the server for the $t$-th communication round, denoted as $\mathbf{u}_{\mathcal{N}}^t$, can be aggregated as

$$\mathbf{u}_{\mathcal{N}}^t = \frac{\sum_{i=1}^m \mathrm{ReLU}(r_i^t)\cdot \mathbf{u}_i^t}{\sum_{i=1}^m \mathrm{ReLU}(r_i^t)}, \quad (13)$$

where the ReLU function plays a crucial role in the aggregation process by excluding agents with negative effects on the federation.

**Local model training.** The global motif prototype serves as an instruction to guide the agents in updating their models in a desired direction. This also encourages non-rewarded agents to proactively identify and rectify their local issues before uploading gradients with the server. For example, agents with initially lower values $r_i^t$ have the opportunity to increase their value by following the guidance provided by the global motif prototype.

In order to facilitate this process, we introduce a regularization term that encourages the local motif prototype $\mathbf{c}_{i,k}^t$ to approach the global motif prototype $\mathbf{c}_{\mathcal{N},k}^t$ associated with motif $k$. Accordingly, the local loss on agent $i$ can be formulated as follows:

$$L(\omega_i, D_i) = L_S(F(D_i), Y) + \lambda \sum_k d(\mathbf{c}_{i,k}^t, \mathbf{c}_{\mathcal{N},k}^t), \quad (14)$$

where $L_S$ is the supervised loss that measures the discrepancy between the model predictions $F(D_i)$ and the ground truth $Y$ on the local data of agent $i$, $\lambda$ is the trade-off parameter between the supervised loss $L_S$ and the motif prototypes-based regularization, and $d$ is the L2 distance metric.

# 4 Experiments

In the experiments, we evaluate our model in graph classification task. We report the personalized accuracy and the accuracy of global model. Besides, we evaluate the model gradient fairness and payoff allocation fairness. Other results, such as ablation study of our model and parameter sensitivity, can be found in Appendix A.3.

## 4.1 Experimental Setup

**Datasets.** We use three graph classification datasets: PROTEINS(Borgwardt et al. 2005), DD (Dobson and Doig 2003), and IMDB-BINARY (Morris et al. 2020). The first two are molecule datasets, while the last one is social network, each containing a set of graphs. We retain 10% of all the graphs as the global test set for the server, and the remaining graphs are distributed to 10 agents. In each agent, we randomly split 90% for training and 10% for testing. In addition, for the payoff fairness, we introduce another setting using perturbed graph dataset. Specifically, for the graphs in each agent, we add varying ratio of perturbations to the structure. We designate 3 agents as low-quality, 3 agents as medium-quality, and 4 agents as high-quality by flipping ratios of $[0.7, 1)$, $[0.3, 0.7)$, and $[0, 0.3)$ of the total number of edges, respectively.

**Baselines.** We compare with two kinds of baselines: FL methods and payoff allocation approaches. For FL methods, we compare against (1) **Self-train**, where the agents train their models only on their local datasets; (2) two commonly used FL baselines **FedAvg** (McMahan et al. 2017) and **FedProx** (Li et al. 2020), (3) **FedSage** (Zhang et al. 2021b), a graph FL framework that adopts FedAvg with GraphSage encoder (Hamilton, Ying, and Leskovec 2017); (4) **GCFL** and **GCFL+** (Xie et al. 2021), two methods that utilize graph clustering to improve the effectiveness of graph FL; (5) **DW and EU** (Xu et al. 2021), incentive mechanisms that allocate model gradient using sparsifying gradient approach based on local data sizes and distance between local and global gradient respectively; (6) (Xu et al. 2021) and **CFFL** (Xu and Lyu 2021), incentive-based FL models in image domain (here we adapt them to our setting by substituting the encoder with the GNN used in our model).

For the payoff allocation approaches, follow (Gao et al. 2021), we set the volume-based payoff function of agent $i$ as $\Phi(i) = \log(1 + |D_i|)$, and compare against classical baselines such as (1) **equal incentive** (Yang et al. 2017), where participants get equal payoff, *i.e.*, $\Phi(i) = 1/N$; (2) **union incentive** (Gollapudi et al. 2017), where the payoff proportion of agent $i$ is $\Phi(\mathcal{N}) - \Phi(\mathcal{N} \backslash \{i\})$; (3) **individual incentive** (Yang et al. 2017), where the payoff proportion of agent $i$ is $\Phi(i)$; and (4) **Shapley incentive** (Shapley 1997), where the payoff proportion of agent $i$ is served as Eq. (3), where the value function $\nu$ is substituted with $\Phi(i)$. The actual payoff for each agent can be calculated as $\frac{\Phi(i)}{\sum_{i=1}^{N} \Phi(i)} B$, where $B$ is the budget for federated learning.

**Metrics.** We evaluate our model's performance on the classification accuracy, as well as the fairness among agents. For graph classification accuracy, we consider personalized accuracy and global accuracy. Personalized accuracy is the average accuracy of the agents on the testing sets of their local graph. Global accuracy is measured on the global test set.

To evaluate the fairness of our allocation, we consider two perspectives: model gradient and payoff. To measure the **fairness of model gradient**, we calculate the Pearson correlation coefficient $\rho(\xi, \psi)$ between the test accuracies $\xi$ achieved through self-training and that $\psi$ achieved by agents when collaborating in FL, following (Xu et al. 2021). To measure the **fairness of payoff**, the allocation approach could be assessed based on the agents with constructed low, medium, and high-quality data. If agents with high-quality data receive a larger payoff, it indicates a better payoff allocation approach.

**Implementation details.** Following the settings in (Xie et al. 2021), we evaluate on the federated graph classification task. We set the parameters $\alpha_1$ and $\alpha_2$ in Eq. (10) as 0.05 and 1, the parameter $\lambda$ in Eq. (14) as 0.1, $\beta$ in Eq. (4) as 1, and the budget $B$ of payoff as 1 (the budget $B$ in payoff allocation baselines is also set as 1). We utilized a three-layer GIN network with a hidden size of 64 and a dropout rate of 0.5 for both the server and agent models. An Adam optimizer with a learning rate of 0.001 and weight decay of $5e^{-4}$ is employed. The communication round is 200, the epoch of local training on agents is 1 and the batch size is 128. Besides, we utilized the motif extraction method in (Yu and Gao 2022). See more details in Appendix A.2.

## 4.2 Experimental Results

**Accuracy and model gradient fairness.** Table 2 present the accuracy and model gradient fairness on clean and perturbed graph datasets, respectively. Our model demonstrates the best performance in terms of model gradient fairness and global accuracy, indicating a significant advantage in fairness without compromising overall performance. Traditional FL methods like FedAvg and FedProx do not perform well in both classification accuracy and model gradient fairness in most cases. This emphasizes the need of a dedicated design encompassing both the graph federated learning model and the graph incentive mechanism. Graph FL methods like GCFL, GCFL+, FedSage and FedStar obtain either unsatisfactory accuracy and fairness, or good accuracy but with compromised fairness. This deficiency stems from the absence of an incentive mechanism within these models. The results of incentive-based federated learning methods such as DW, EU, (Xu et al. 2021) and CFFL are also non-ideal, as they are not specifically designed for graph FL. We also note the presence of a trade-off between model gradient fairness and personalized accuracy, which has also been observed and recognized in previous research (Gu et al. 2022; Ezzeldin et al. 2023); in the subsequent experiments, we proceed to further evaluate this trade-off.

**Trade-off between personalized accuracy and model gradient fairness.** We evaluate the trade-off between model gradient fairness and personalized accuracy by presenting

| Dataset | PROTEINS | | | DD | | | IMDB-BINARY | | |
|---|---|---|---|---|---|---|---|---|---|
| | model gradient fairness | global accuracy | personalized accuracy | model gradient fairness | global accuracy | personalized accuracy | model gradient fairness | global accuracy | personalized accuracy |
| Self-train | - | - | 0.712±0.011 | - | - | 0.587±0.016 | - | - | 0.779±0.018 |
| FedAvg | 0.700±0.153 | 0.748±0.006 | 0.750±0.007 | 0.259±0.209 | 0.668±0.032 | 0.657±0.029 | 0.780±0.143 | 0.779±0.007 | 0.782±0.018 |
| FedProx | 0.717±0.173 | 0.731±0.017 | 0.746±0.007 | 0.283±0.186 | 0.633±0.025 | 0.672±0.020 | 0.842±0.093 | 0.775±0.018 | 0.756±0.017 |
| FedSage | 0.770±0.048 | 0.740±0.024 | 0.741±0.025 | 0.360±0.159 | 0.671±0.021 | 0.663±0.023 | 0.870±0.086 | 0.763±0.004 | 0.768±0.003 |
| GCFL | 0.725±0.185 | - | 0.772±0.019 | 0.439±0.094 | - | **0.698**±0.013 | 0.874±0.040 | - | **0.830**±0.009 |
| GCFL+ | 0.734±0.146 | - | **0.775**±0.019 | 0.379±0.190 | - | 0.692±0.013 | 0.825±0.094 | - | 0.819±0.007 |
| FedStar | 0.763±0.043 | - | 0.717±0.138 | 0.335±0.034 | - | 0.665±0.109 | 0.795±0.071 | - | 0.776±0.086 |
| DW | 0.702±0.014 | 0.723±0.014 | 0.730±0.014 | 0.305±0.326 | 0.672±0.031 | 0.669±0.033 | 0.723±0.089 | 0.751±0.011 | 0.762±0.009 |
| EU | 0.733±0.016 | 0.725±0.048 | 0.715±0.141 | 0.381±0.081 | 0.668±0.010 | 0.658±0.382 | 0.747±0.010 | 0.755±0.023 | 0.759±0.069 |
| CFFL | 0.690±0.117 | 0.735±0.037 | 0.713±0.103 | 0.402±0.141 | 0.658±0.053 | 0.658±0.112 | 0.795±0.100 | 0.741±0.037 | 0.752±0.034 |
| (Xu et al. 2021) | 0.750±0.040 | 0.745±0.022 | 0.737±0.025 | 0.384±0.123 | 0.682±0.011 | 0.659±0.013 | 0.796±0.085 | 0.781±0.014 | 0.778±0.019 |
| ours | **0.787**±0.052 | **0.753**±0.018 | 0.751±0.017 | **0.479**±0.067 | **0.692**±0.017 | 0.680±0.017 | **0.907**±0.018 | **0.801**±0.013 | 0.794±0.005 |

Table 2: Accuracy and fairness performance on three clean graph datasets. We report the average test accuracy of all agents (denoted as personalized), the accuracy of global models on the global test dataset (denoted as global) and the performance fairness. "-" means these methods do not have a single global model on the server.

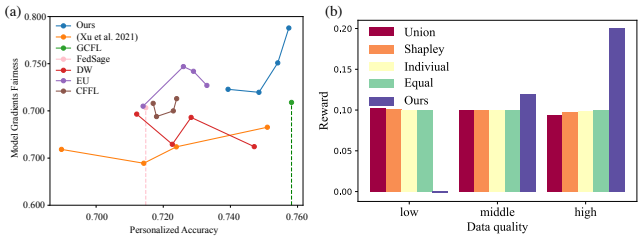



Figure 3: (a) The trade-off between model gradient fairness and personalized accuracy. (b) The average payoff per round for agents with varying data qualities.

the results of various algorithms in Figure 3(a). The ideal algorithm should be situated in the upper right corner of the figure, similar to where our algorithm is positioned, signifying both high accuracy and excellent fairness. This results effectively demonstrate the most remarkable trade-off achieved by our algorithm between personalized accuracy and model gradient fairness.

**Payoff fairness.** To evaluate the payoff fairness of our model, we compare it with different payoff allocation approaches. We conduct experiments on perturbed datasets to examine the payoff received by low, medium and high-quality agents. The results are shown in Figure 3(b), where the height of the histogram is the average amount of the payoff received by each agent per communication round. It clearly indicates that our payoff allocation mechanism exhibits the best payoff fairness, as it provides the highest payoff to agents with high-quality data, while assigning negative payoff to agents with low-quality data. In contrast, other allocation methods struggle to maintain fairness in FL.

## 5 Related Works

**Incentive mechanism.** Data value in FL is primarily assessed from two dimensions: data quantity and data quality. Concerning data quantity, existing works use data size to measure agent contribution (Zhan et al. 2020b,a). As for data quality, most works evaluate the contribution of agents with the Shapley Value (Shapley 1997), and a validation set on the server that is agreed by all agents is needed to determine the value function (Jia et al. 2019; Song, Tong, and Wei 2019; Wang et al. 2020).

Built upon the notion of data value, incentive mechanisms are established for FL to allocate model gradients or payoff. (Xu et al. 2021; Xu and Lyu 2021) allocate model gradients based on the similarity between local gradients and global gradients. Another line of works introduce payoff-sharing schemes, where (Yu et al. 2020) dynamically allocates payoff by optimizing collective utility while minimizing inequality and (Gao et al. 2021) considers the malicious agents. However, none of them have considered the uniqueness in graph federated learning.

**Graph federated learning.** Graph federated learning is divided into graph-level, subgraph-level and node-level (He et al. 2021a). In subgraph-level graph FL, each agent has a subgraph of a large entire graph (Zhang et al. 2021b; Wu et al. 2021). In node-level graph FL, each agent possesses the ego-networks of one or multiple nodes (Meng, Rambhatla, and Liu 2021; Wang et al. 2022; Zheng et al. 2021). Our work belongs to graph-level graph FL, where each agent holds a set of graphs. In this context, (Xie et al. 2021) introduces a clustered FL to deal with feature and structure heterogeneity. (Tan et al. 2023) separates the graph structure and features, sharing only the structural information across agents while retaining features for local training by each agent. (Gu et al. 2023) introduces a dynamic approach for selecting clients and model gradients to enhance efficiency and accuracy. (He et al. 2021b) proposes a multi-task learning framework that eliminates the necessity for a central server. However, none of them consider the fairness in graph federated learning.

## 6 Conclusion

In this paper, we begin by uncovering a unique phenomenon in graph federated learning: the presence of agents causing potential harm and agents contributing with delays. In light of this, we presents a novel incentive mechanism for fair graph federated learning framework, combining incentives from both model gradients and payoff. To achieve the framework, we introduce a agent valuation function considering both gradient alignment and graph diversity, and then we enhance the accuracy-fairness trade-off by introducing a novel concept of motif prototypes. Extensive experiments demonstrate our superiority in both accuracy and fairness.

## Acknowledgement

This work is supported by NSFC (62206056, 12271479), Zhejiang NSF (LR22F020005) and the Fundamental Research Funds for the Central Universities.

## References

Borgwardt, K. M.; Ong, C. S.; Schönauer, S.; Vishwanathan, S. V. N.; Smola, A.; and Kriegel, H.-P. 2005. Protein function prediction via graph kernels. *Bioinformatics*, 21 Suppl 1: i47–56.

Deng, Y.; Lyu, F.; Ren, J.; Chen, Y.-C.; Yang, P.; Zhou, Y.; and Zhang, Y. 2021. FAIR: Quality-Aware Federated Learning with Precise User Incentive and Model Aggregation. In *IEEE Conference on Computer Communications*.

Dobson, P. D.; and Doig, A. J. 2003. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4): 771–783.

Ezzeldin, Y. H.; Yan, S.; He, C.; Ferrara, E.; and Avestimehr, A. S. 2023. Fairfed: Enabling group fairness in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Frasca, F.; Bevilacqua, B.; Bronstein, M. M.; and Maron, H. 2021. Understanding and Extending Subgraph GNNs by Rethinking Their Symmetries. In *Proceedings of the International Conference on Neural Information Processing Systems*.

Gao, L.; Li, L.; Chen, Y.; Zheng, W.; Xu, C.; and Xu, M. 2021. FIFL: A Fair Incentive Mechanism for Federated Learning. In *Proceedings of the International Conference on Parallel Processing*.

Gollapudi, S.; Kollias, K.; Panigrahi, D.; and Pliatsika, V. 2017. Profit sharing and efficiency in utility games. In *Annual European Symposium on Algorithms*.

Gu, X.; Tianqing, Z.; Li, J.; Zhang, T.; Ren, W.; and Choo, K.-K. R. 2022. Privacy, accuracy, and model fairness tradeoffs in federated learning. *Computers & Security*, 122: 102907.

Gu, Z.; Zhang, K.; Bai, G.; Chen, L.; Zhao, L.; and Yang, C. 2023. Dynamic Activation of Clients and Parameters for Federated Learning over Heterogeneous Graphs. In *International Conference on Data Engineering*.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Proceedings of the International Conference on Neural Information Processing Systems*.

He, C.; Balasubramanian, K.; Ceyani, E.; Yang, C.; Xie, H.; Sun, L.; He, L.; Yang, L.; Yu, P. S.; Rong, Y.; et al. 2021a. Fedgraphnn: A federated learning system and benchmark for graph neural networks. *arXiv preprint arXiv:2104.07145*.

He, C.; Ceyani, E.; Balasubramanian, K.; Annavaram, M.; and Avestimehr, S. 2021b. Spreadgnn: Serverless multi-task federated learning for graph neural networks. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*.

Jia, R.; Dao, D.; Wang, B.; Hubis, F. A.; Hynes, N.; Gürel, N. M.; Li, B.; Zhang, C.; Song, D.; and Spanos, C. J. 2019. Towards efficient data valuation based on the shapley value. In *International Conference on Artificial Intelligence and Statistics*.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems*.

Lyu, L.; Xu, X.; Wang, Q.; and Yu, H. 2020. Collaborative fairness in federated learning. *Federated Learning: Privacy and Incentive*, 189–204.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial intelligence and statistics*.

Meng, C.; Rambhatla, S.; and Liu, Y. 2021. Cross-node federated graph neural network for spatio-temporal data modeling. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Michieli, U.; and Ozay, M. 2021. Prototype Guided Federated Learning of Visual Feature Representations. *arXiv preprint arXiv:2105.08982*.

Milo, R.; Shen-Orr, S.; Itzkovitz, S.; Kashtan, N.; Chklovskii, D.; and Alon, U. 2002. Network motifs: simple building blocks of complex networks. *Science*, 298(5594): 824–827.

Morris, C.; Kriege, N. M.; Bause, F.; Kersting, K.; Mutzel, P.; and Neumann, M. 2020. TUDataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*.

Pan, S.; Wu, J.; and Zhu, X. 2015. CogBoost: Boosting for Fast Cost-Sensitive Graph Classification. *IEEE Transactions on Knowledge and Data Engineering*, 27: 2933–2946.

Shapley, L. S. 1997. A value for n-person games. *Classics in game theory*, 69.

Song, T.; Tong, Y.; and Wei, S. 2019. Profit allocation for federated learning. In *IEEE International Conference on Big Data*.

Tan, Y.; Liu, Y.; Long, G.; Jiang, J.; Lu, Q.; and Zhang, C. 2023. Federated Learning on Non-IID Graphs via Structural Knowledge Sharing. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; and Zhang, C. 2022. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Wang, B.; Li, A.; Pang, M.; Li, H.; and Chen, Y. 2022. Graphfl: A federated learning framework for semi-supervised node classification on graphs. In *IEEE International Conference on Data Mining*.

Wang, T.; Rausch, J.; Zhang, C.; Jia, R.; and Song, D. 2020. A principled approach to data valuation for federated learning. *Federated Learning: Privacy and Incentive*, 153–167.

Wu, C.; Wu, F.; Cao, Y.; Huang, Y.; and Xie, X. 2021. Fedgnn: Federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925*.

Xie, H.; Ma, J.; Xiong, L.; and Yang, C. 2021. Federated graph classification over non-iid graphs. In *Proceedings of the International Conference on Neural Information Processing Systems*.

Xie, H.; Xiong, L.; and Yang, C. 2023. Federated node classification over graphs with latent link-type heterogeneity. In *Proceedings of the ACM Web Conference*.

Xu, J.; Huang, R.; Jiang, X.; Cao, Y.; Yang, C.; Wang, C.; and Yang, Y. 2023. Better with Less: A Data-Centric Prespective on Pre-Training Graph Neural Networks. In *Proceedings of the International Conference on Neural Information Processing Systems*.

Xu, X.; and Lyu, L. 2021. A Reputation Mechanism Is All You Need: Collaborative Fairness and Adversarial Robustness in Federated Learning. *ICML Workshop on Federated Learning for User Privacy and Data Confidentiality*.

Xu, X.; Lyu, L.; Ma, X.; Miao, C.; Foo, C. S.; and Low, B. K. H. 2021. Gradient driven rewards to guarantee fairness in collaborative machine learning. In *Proceedings of the International Conference on Neural Information Processing Systems*.

Yanardag, P.; and Vishwanathan, S. 2015. Deep graph kernels. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Yang, S.; Wu, F.; Tang, S.; Gao, X.; Yang, B.; and Chen, G. 2017. On designing data quality-aware truth estimation and surplus sharing method for mobile crowdsensing. *IEEE Journal on Selected Areas in Communications*, 35(4): 832–847.

Yu, H.; Liu, Z.; Liu, Y.; Chen, T.; Cong, M.; Weng, X.; Niyato, D. T.; and Yang, Q. 2020. A Fairness-aware Incentive Scheme for Federated Learning. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*.

Yu, Z.; and Gao, H. 2022. Molecular representation learning via heterogeneous motif graph neural networks. In *International Conference on Machine Learning*.

Yuan, H.; Yu, H.; Wang, J.; Li, K.; and Ji, S. 2021. On Explainability of Graph Neural Networks via Subgraph Explorations. In *International Conference on Machine Learning*.

Zhan, Y.; Li, P.; Qu, Z.; Zeng, D.; and Guo, S. 2020a. A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal*, 7(7): 6360–6368.

Zhan, Y.; Li, P.; Wang, K.; Guo, S.; and Xia, Y. 2020b. Big data analytics by crowdlearning: Architecture and mechanism design. *IEEE Network*, 34(3): 143–147.

Zhang, C.; Zhang, S.; Yu, J. J. Q.; and Yu, S. 2021a. FASTGNN: A Topological Information Protected Federated Learning Approach for Traffic Speed Forecasting. *IEEE Transactions on Industrial Informatics*, 17: 8464–8474.

Zhang, K.; Yang, C.; Li, X.; Sun, L.; and Yiu, S. M. 2021b. Subgraph federated learning with missing neighbor generation. In *Proceedings of the International Conference on Neural Information Processing Systems*.

Zhang, Z.; Liu, Q.; Wang, H.; Lu, C.; and Lee, C.-K. 2021c. Motif-based graph self-supervised learning for molecular property prediction. In *Proceedings of the International Conference on Neural Information Processing Systems*.

Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.

Zheng, L.; Zhou, J.; Chen, C.; Wu, B.; Wang, L.; and Zhang, B. 2021. ASFGNN: Automated separated-federated graph neural network. *Peer-to-Peer Networking and Applications*, 14(3): 1692–1704.

# A Appendix

## A.1 Additional Theorems

Let $\Pi_{\mathcal{N}}$ denote a set of all permutations of all possible permutations of $\mathcal{N}$. For a given permutation $\pi \in \Pi_{\mathcal{N}}$, $S_{\pi_i}$ represents the coalition of agents preceding agent $i$ in the permutation. Then the gradient-based Shapley value can be defined as follows:

$$\varphi_i := \frac{1}{N!} \sum_{\pi \in \Pi_{\mathcal{N}}} \left[ \nu \left( \mathcal{S}_{\pi,i} \cup \{i\} \right) - \nu \left( \mathcal{S}_{\pi,i} \right) \right], \tag{15}$$
$$\nu(\mathcal{S}) = \cos(\mathbf{u}_{\mathcal{S}}, \mathbf{u}_{\mathcal{N}}) = \langle \mathbf{u}_{\mathcal{S}}, \mathbf{u}_{\mathcal{N}} \rangle / (||\mathbf{u}_{\mathcal{S}}|| \cdot ||\mathbf{u}_{\mathcal{N}}||),$$

where $\mathbf{u}_{\mathcal{S}}$ denotes the gradient of coalition $\mathcal{S}$, and $\mathbf{u}_{\mathcal{N}}$ denotes the gradient of grand coalition $\mathcal{N}$. As the calculation process above needs $O(2^{|\mathcal{N}|})$ time complexity, we could use $\zeta_i = \cos(\mathbf{u}_i, \mathbf{u}_{\mathcal{N}})$ as an approximation , where $\mathbf{u}_i$ is the gradient that agent $i$ uploads in communication $t$. The approximation error theorem can be formulated as follows:

**Theorem 1** (Approximation Error)**.** *Let $I \in \mathbb{R}^+$. Suppose that $||\mathbf{u}_i^t|| = \tau$ and $|\langle \mathbf{u}_i, \mathbf{u}_{\mathcal{N}} \rangle| \leq 1/I$ for all $i \in \mathcal{N}$. Then $\varphi_i - L_i \zeta_i \leq I\tau^2$, where the multiplicative factor $L_i$ can be normalized.*

The detailed proof can be found in (Xu et al. 2021).

## A.2 Additional Implementation Details

**Motif extraction methods.** The motif prototype proposed above needs to define the motifs and extract the motifs from the graphs. Researchers have proposed many motif extraction methods and here we select a typical one that is proposed in (Yu and Gao 2022). In this method, the bonds and rings are selected for the motifs. We first give a formal definition of the bonds and rings that is used as the motifs.

**Definition 4** (Bonds)**.** *A bond is a tuple that contains the edge information and the nodes information at the start and the end of the edge that does not exist on a ring structure. Specifically, given the edge $E$ on the graph, the bond can be denoted as $((E[0], E[1]), e)$, where $E[0]$ and $E[1]$ are the labels of the start node and the end node of edge $E$, respectively, and $e$ is the edge label of $E$. If $e$ does not contain the label, then $e$ is labeled as 'None'.*

**Definition 5** (Rings)**.** *A ring is a tuple that contains the edge information and the node information that is contained in a cycle structure of a graph. Specifically, the ring can be formulated as the tuple $(V, E)$, where $V$ is the tuple of the labels of the nodes in the cycle structure and is ordered clockwise, $E$ is the tuple of the labels of the edges in the cycle structure and is ordered clockwise. If the edges do not contain the label, then it is labeled as 'None'.*

In molecular graphs, the bonds would be viewed as the connection types between two atoms, such as the single bond and double bond between two carbon atoms. The rings could be viewed as function groups, such as benzene rings. In other graphs, the rings represent that the nodes on the rings are in the same community, and the bonds would imply that the two nodes have connections. Therefore, the motif extraction method can be used in all kinds of graph datasets.

Note that some of the motifs may appear in most of the graphs and they would carry little information in representation learning. To reduce the influence of these motifs, we use the term frequency-inverse document frequency (TF-IDF) algorithm to select the most essential motifs in our motif vocabulary. The TF-IDF value of a motif $k$ in the graph $G$ is computed as

$$T_{k,G} = C(k)_G \left( \log \frac{1 + |D_i|}{1 + |D_{i,k}|} \right) + 1, \tag{16}$$

where $C(k)_G$ denotes the time that motif $k$ appears in graph $G$. To get the TF-IDF value $T_k$ for a motif $k$, we calculate the TF-IDF value for all graphs containing the motif $k$ as follows:

$$T_k = \frac{1}{|D_{i,k}|} \sum_{k=1}^{|D_{i,k}|} T_{i,G_{n_k}}, \tag{17}$$

where the sequence $\{n_k\}$ denotes the index of graphs that contains the motif $k$. After sorting the motif vocabulary by the TF-IDF value, we select the essential motifs based the some threshold $\beta_s \in (0, 1)$. In this work, we keep $\beta_s = 0.9$. Then we keep only the most essential motifs as the final motif vocabulary.

**Trade-off experiment settings** For our algorithm, (Xu et al. 2021), DW, EU (Xu et al. 2021) and CFFL (Xu and Lyu 2021) that are five frameworks that are customized for graph federated learning, we set the hyper-parameter $\beta$ to be 0.5, 1, 1.5, 2, respectively, and record the personalized accuracy and model gradient fairness. For GCFL and FedSage, we set the same parameter as mentioned in (Xie et al. 2021).

**Other implementation details.** Following the settings in (Xie et al. 2021), we use 10 agents in the experiment to ensure that each agent would have a total of around 100 graph instances. Our model is implemented under the following software settings:

Pytorch version 1.10.0+cu111, CUDA version 11.1, Python version 3.8.8, numpy version 1.20.3 and networkx version 2.8.7. All of the experiments are conducted on a single machine of Linux system with an Intel Xeon Gold 5118 (128G memory) and a GeForce Tesla P4 (8GB memory). The implementation can be found at https://github.com/zjunet/FairGraphFL.

**Dataset details.** We provide detailed information about the datasets that we utilized in Table 3, and give a brief introduction to these datasets.

| dataset | statistics | | | | |
|---|---|---|---|---|---|
| | #graphs | avg. #nodes | avg. #edges | #classes | node features |
| PROTEINS | 1113 | 39.06 | 72.82 | 2 | original |
| DD | 1178 | 284.32 | 715.66 | 2 | original |
| IMDB-BINARY | 1000 | 19.77 | 95.53 | 2 | degree |

Table 3: The statistics of the datasets.

- PROTEINS (Borgwardt et al. 2005) is a protein dataset consisting of enzymes and non-enzymes. In this dataset, each node represents an amino acid, and an edge connects two nodes if their distance is less than 6 Angstroms.
- DD (Dobson and Doig 2003) is an organic molecule dataset, where each node represents an atom and edges represent the chemical bonds between atoms. Nodes have different attributes, such as the type of atom, partial charge of atoms, etc.
- IMDB-BINARY (Yanardag and Vishwanathan 2015) is a movie collaboration dataset comprised of the ego-networks of 1,000 actors/actresses who have participated in movies in IMDB. In each graph, the nodes represent actors/actresses, and an edge exists between them if they have appeared in the same movie.

**Details about Figure 1.** In Figure 1(a), we conduct FedAvg in image dataset CIFAR 10 by splitting the dataset into 5 equal parts. In Figure 1(b) and (c), we conduct the experiment in graph dataset PROTEINS by splitting the dataset into 5 equal parts, using the FedAvg and our model, respectively. The contribution of each agent in different rounds is calculated as the relative improvement in performance on the server's global test set when all agents participate compared to when this agent is excluded.

## A.3 Additional Experiments

**Ablation study.** We present the result of ablation study for our model to demonstrate the importance of each component in our model. The five variants of our model remove the gradient alignment, graph diversity, model gradient, value-based aggregation and motif prototype component, respectively. The results can be found in Table 4. We can observe that: (1) Comparing our model with **ours-gradient** that removes the gradient alignment, we can see that the inclusion of gradient alignment enhances both performance and model gradient fairness. The improvement can be attributed to the fact that gradient alignment allows for a more accurate estimation of agent value, which, in turn, contributes to the enhancement of global model quality through value-based aggregation on the server. (2) Comparing our model with **ours-diversity**, which removes the graph diversity component, it is implied that the inclusion of graph diversity could bring the enhancement on both the model gradient fairness and the performance as it may lead to a more accurate estimation on the value of agents. Notably, the graph diversity component has a more significant impact on model gradient fairness compared to gradient alignment, suggesting its crucial role in assessing agent value. (3) Compared with **ours-model gradient** which gives all the agents the same gradient, we could see that the model gradient fairness would be significantly hurt, as all agents receive the same model irrespective of their contributions. However, it leads to improved personalized accuracy, as all agents receive the same global gradient. This phenomenon could be viewed as a trade-off between the accuracy and the model gradient fairness. (4) **ours-aggregation** refers to the variant that aggregates local gradients in the server as FedAvg (McMahan et al. 2017). The inclusion of our value-based aggregation may lead to an increase in accuracy and a better performance in model gradient fairness. This is because that our value-based aggregation technique reduces noise in the federation, thereby enhancing model accuracy. Additionally, it demonstrates relatively good

| | personalized accuracy | global accuracy | model gradient fairness |
|---|---|---|---|
| ours | 0.751±0.017 | 0.753±0.018 | 0.787±0.052 |
| ours-gradient | 0.743±0.017 | 0.750±0.019 | 0.736±0.145 |
| ours-diversity | 0.744±0.009 | 0.748±0.022 | 0.723±0.082 |
| ours-model gradient | 0.757±0.014 | 0.751±0.014 | 0.710±0.146 |
| ours-aggregation | 0.741±0.042 | 0.740±0.024 | 0.741±0.121 |
| ours-prototype | 0.737±0.025 | 0.741±0.022 | 0.750±0.040 |

Table 4: Ablation study on accuracy and fairness performance on clean PROTEINS dataset.

performance in terms of model gradient fairness, as it considers both gradient alignment and graph diversity to assess the value of agents and thus allocate the model gradient relatively fairly. (5) **ours-prototype** is a variant excluding the prototype-related components, and this could lead to a severe drop in accuracy due to the heterogeneity of the graph data. Moreover, the absence of the prototype-related components adversely affect the performance of model gradient fairness, as these components plays a crucial role in ensuring model quality, which in turn helps us assess the value of agents more accurately through gradient alignment.

**Hyper-parameter analysis.** We did the hyper-parameter analysis on the trade-off parameter $\lambda$ between the supervised loss and the motif prototype-based regularization. This is an important hyper-parameter in our proposed strategy. Figure 4 shows the effect of the trade-off parameter on global accuracy, personalized accuracy and model gradient fairness. We can see that a too small or too large learning rate could deteriorate the accuracy and fairness performance, and the optimal performance can be obtained when $\lambda$ is 0.1.

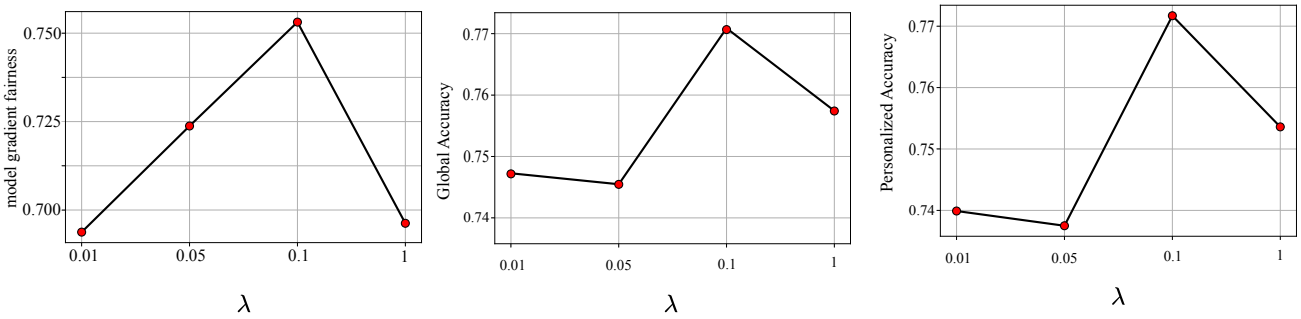



Figure 4: Personalized accuracy and model gradient fairness w.r.t. the trade-off parameter between the supervised loss and the motif prototype-based regularization.

**Experiment on large dataset** We conducted the experiment on a large dataset, TWITTER-Real-Graph-Partial dataset (Pan, Wu, and Zhu 2015), containing 144,033 graphs with an average of 4.03 nodes and 4.98 edges per graph. We compare with the most competitive Graph FL and incentive-based FL baselines. The results in Table 5 indicate our advantage even at a large scale.

| Dataset | TWITTER-Real-Graph-Partial | | |
|---|---|---|---|
| | model gradient fairness | global accuracy | personalized accuracy |
| Self-train | - | - | 0.594±0.002 |
| GCFL | 0.337±0.108 | - | 0.612±0.007 |
| (Xu et al. 2021) | 0.361±0.095 | 0.601±0.010 | 0.598±0.003 |
| ours | 0.401±0.052 | 0.620±0.014 | 0.600±0.008 |

Table 5: Results on a large scale dataset.

## A.4 Algorithm

The pseudocode of our framework is summarized in Algorithm 1. In each communication round, our framework mainly consists of three parts: (i) the server could aggregate the local gradients, update the value of agents and allocate the gradients to the agents according to their values (line 3-5). (ii) The agents then update the local models for $E$ rounds. (iii) The agents then update the local motif prototypes and the server update the global motif prototypes.

The time complexity of our model mainly consists of three parts. (1) The aggregation of local gradients and the update of the value costs $O(m)$. (2) The time complexity of local updates on each agent costs $O(E(X + k_i|D_i|))$ at most, where $X$ is the complexity of the backbone GNN. (3) The update of the global motif prototypes costs $O(Km)$ at most. Based on the analysis above, the overall time complexity of our framework on each agent in each communication round is $O((K + 1)m + EX + Ek_i|D_i|)$ at most.

## A.5 Convergence Analysis of Our Algorithm

In this subsection, we give the additional proof that our algorithm could finally convergent to a stationary point. For the convenience in the detailed proof, set $\Phi_i = (\omega_i, \phi_i)$, and decompose the parameters of the whole model $\Phi_i$ into two parts: embedding

Algorithm 1: Our federated learning framework

**Input:** A total of $m$ agents, the number of local motifs $k_i, i = 1, 2, \cdots, m$, the number of motifs across the agents $K$, communication rounds $T$, the local iterations in a communication round $E$, the initial local models $\Phi_i$, the local motif prototypes $\mathbf{c}_{i,k}^t, k = 1, 2, \cdots, K, t = 1, 2, \cdots, T$.

**Output:** The updated local models $\Phi_i^T$.

```
Initialize global prototypes for all motifs.
for each round t = 1, 2, ..., T do
    Aggregate the local gradients by Eq. (13).
    Update the value of agents by Eq. (10).
    Allocate the gradient to agents by Eq. (4).
    for each client i = 1, 2, ... m in parallel do
        Update the local model by Eq. (14) for E rounds.
    end for
    Update the global motif prototypes by Eq. (11) and (12).
end for
```

parameter $\omega_i$ and the decision layer $\phi_i$. Then the local loss function of agent $i$ can be written as:

$$L_i(\omega_i, \phi_i, D_i, Y) = L_S(F(D_i), Y) + \lambda \sum_{k=1}^{k_i} ||\mathbf{c}_{i,k}^t - \mathbf{c}_{\mathcal{N},k}^t||, \tag{18}$$

where the global motif prototype

$$\mathbf{c}_{\mathcal{N},k}^t = \frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} \frac{|D_{i,k}|}{N_k} \mathbf{c}_{i,k}^t, \tag{19}$$

and the local motif prototype

$$\mathbf{c}_{i,k}^t = \frac{1}{|D_{i,k}|} \sum_{G \in D_{i,k}} f_{\omega_i^t}(G). \tag{20}$$

As for the iteration round notification, $e \in \{1/2, 1, 2 \cdots, E\}$ to represent the local iterations. $tE$ represents the time step between prototype aggregation and gradient aggregation at the server and $tE + 1/2$ represents the time step between the prototype aggregation at the server and the first iteration on the agents, and $L_{tE+e}$ means the loss function of the $e$-th local round in the communication round $t$. Then we could have the following theorem:

**Theorem 2.** *Suppose that the local loss function $L_i$ is $\mu$-smooth, the local embedding function $f_\omega$ is L-Lipschitz continuous, the stochastic gradient is an unbiased estimator of local gradient, the variance is bounded by $\sigma^2$, and the expectation of the local gradients is bounded by G. Then the loss function in an arbitrary agent strictly monotonically decreases in every communication round when*

$$\lambda = \lambda_t = \frac{||\nabla_{tE+1/2}||}{2KLEG}, \tag{21}$$

*and*

$$\eta = \eta_{e'} = \frac{||\nabla_{tE+1/2}||^2}{\mu(E\sigma^2 + \sum_{e=1/2}^{e'} ||\nabla L_{tE+e'}||^2)}, e' = 1/2, 1, \cdots, E. \tag{22}$$

*As the loss function $L > 0$, the loss function could converge according to the monotone bounded criterion.*

To give a detailed proof, we first propose two lemmas that are important for the theorem. Let $\Phi_{t+1} = \Phi_t - \eta g_t$, we could have the following lemma:

**Lemma 1.** *From the beginning of communication round $t + 1$ to the last local update step, the loss function of an arbitrary agent can be bounded as*

$$\mathbb{E}[L_{tE+E}] \leq L_{tE+1/2} - (\eta - \frac{\mu\eta^2}{2}) \sum_{e=1/2}^{E-1} ||\nabla L_{tE+e}||^2 + \frac{\mu E \eta^2}{2} \sigma^2. \tag{23}$$

Detailed proof can be found in (Tan et al. 2022).

**Lemma 2.** *After the aggregation of local motif prototypes, the loss function of an agent can be bounded as:*

$$\mathbb{E}[L_{(t+1)E+1/2}] < L_{(t+1)E} + \lambda K L \eta E G, \tag{24}$$

*where $K$ is the number of motifs on the server.*

*Proof.*

$$L_{(t+1)E+1/2} = L_{(t+1)E} + L_{(t+1)E+1/2} - L_{(t+1)E} \tag{25}$$

$$= L_{(t+1)E} + \lambda \sum_{k=1}^{k_i} \left( ||\mathbf{c}_{i,k}^{t+1} - \mathbf{c}_{\mathcal{N},k}^{t+2}|| - ||\mathbf{c}_{i,k}^{t+1} - \mathbf{c}_{\mathcal{N},k}^{t+1}|| \right) \tag{26}$$

$$\overset{(a)}{\leq} L_{(t+1)E} + \lambda \sum_{k=1}^{k_i} \left( ||\mathbf{c}_{\mathcal{N},k}^{t+2} - \mathbf{c}_{\mathcal{N},k}^{t+1}|| \right) \tag{27}$$

$$= L_{(t+1)E} + \lambda \sum_{k=1}^{k_i} \left( \frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} \frac{|D_{i,k}|}{N_k} ||\mathbf{c}_{i,k}^{t+2} - \mathbf{c}_{i,k}^{t+1}|| \right) \tag{28}$$

$$\overset{(b)}{=} L_{(t+1)E} + \lambda \sum_{k=1}^{k_i} \left( \frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} \frac{|D_{i,k}|}{N_k} \frac{1}{|D_{i,k}|} \sum_{G \in D_{i,k}} ||f_{\omega_i^{t+2}}(G) - f_{\omega_i^{t+1}}(G)|| \right) \tag{29}$$

$$\leq L_{(t+1)E} + \lambda \sum_{k=1}^{k_i} \left( \frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} \frac{L}{N_k} \sum_{G \in D_{i,k}} ||\omega_i^{t+2} - \omega_i^{t+1}|| \right) \tag{30}$$

$$\overset{(c)}{\leq} L_{(t+1)E} + \lambda \sum_{k=1}^{k_i} \left( \frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} \frac{L}{N_k} \sum_{G \in D_{i,k}} ||\Phi_i^{t+2} - \Phi_i^{t+1}|| \right) \tag{31}$$

$$= L_{(t+1)E} + \lambda \sum_{k=1}^{k_i} \left( \frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} \frac{L|D_{i,k}|}{N_k} \eta || \sum_{e=1/2}^{E-1} g_{i,tE+e} || \right) \tag{32}$$

$$\leq L_{(t+1)E} + \lambda \sum_{k=1}^{k_i} \left( \frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} L\eta \sum_{e=1/2}^{E-1} ||g_{i,tE+e}|| \right) \tag{33}$$

After taking expectations from the random variable $\xi_t$ from both sizes of the equation, we could get:

$$\mathbb{E}[L_{(t+1)E+1/2}] \leq L_{(t+1)E} + \lambda \sum_{k=1}^{k_i} \left( \frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} L\eta \sum_{e=1/2}^{E-1} \mathbb{E}[||g_{i,tE+e}||] \right) \tag{34}$$

$$\leq L_{(t+1)E} + \lambda \sum_{k=1}^{k_i} \left( \frac{1}{\mathcal{N}_k} \sum_{i \in \mathcal{N}_k} L\eta EG \right) \tag{35}$$

$$= L_{(t+1)E} + \lambda k_i L\eta EG \leq L_{(t+1)E} + \lambda KL\eta EG, \tag{36}$$

where (a) follows from the triangle inequality, (b) follows from Eq. (11), (c) comes from the fact that $\omega_i^t$ is the subset of $\Phi_i^t$. □

**Completing proof for Theorem 2.**

*Proof.* From the above two lemmas, we could get that

$$\mathbb{E}[L_{(t+1)E+1/2}] < L_{tE+1/2} - (\eta - \frac{\mu\eta^2}{2}) \sum_{e=1/2}^{E-1} ||\nabla L_{tE+e}||^2 + \frac{\mu E\eta^2}{2}\sigma^2 + \lambda KL\eta EG. \tag{37}$$

To ensure that the loss function decreases as the communication round $t$ increases, it is easy to get

$$-(\eta - \frac{\mu\eta^2}{2}) \sum_{e=1/2}^{E-1} ||\nabla_{tE+e}||^2 + \frac{\mu E\eta^2}{2}\sigma^2 + \lambda KL\eta EG < 0. \tag{38}$$

Then the learning rate $\eta$ and the trade-off parameter $\lambda$ should satisfy:

$$\eta < \frac{2\left(\sum_{e=1/2}^{E-1} ||\nabla L_{tE+e}||^2 - \lambda KLEG\right)}{\mu(E\sigma^2 + \sum_{e=1/2}^{E-1} ||\nabla L_{tE+e}||^2)} \tag{39}$$

and

$$\lambda < \frac{\sum_{e=1/2}^{E-1} ||\nabla L_{tE+e}||^2}{KLEG} \tag{40}$$

In practice, we cannot get the gradients after the current round. Therefore, we take the $\lambda$ to keep the same during one communication round:

$$\lambda_t = \frac{||\nabla_{tE+1/2}||^2}{2KLEG}, \tag{41}$$

and keep the learning rate changing with the local update round:

$$\eta_{e'} = \frac{||\nabla_{tE+1/2}||^2}{\mu(E\sigma^2 + \sum_{e=1/2}^{e'} ||\nabla L_{tE+e'}||^2)}, e' = 1/2, 1, 2, \cdots, E. \tag{42}$$

Here we complete the proof for Theorem 2. □