Federated Graph Semantic and Structural Learning

Wenke Huang\textsuperscript{1†}, Guancheng Wan\textsuperscript{1†}, Mang Ye\textsuperscript{1,2∗} and Bo Du\textsuperscript{1,2}

\textsuperscript{1}National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence, Hubei Key Laboratory of Multimedia and Network Communication Engineering, School of Computer Science, Wuhan University, Wuhan, China.
\textsuperscript{2}Hubei Luojia Laboratory, Wuhan, China.
\{wenkehuang, guanchengwan, yemang, dubo\}@whu.edu.cn

Abstract

Federated graph learning collaboratively learns a global graph neural network with distributed graphs, where the non-independent and identically distributed property is one of the major challenge. Most relative arts focus on traditional distributed tasks like images and voices, incapable of the graph structures. This paper firstly reveals that local client distortion is brought by both node-level semantics and graph-level structure. First, for node-level semantic, we find that contrasting nodes from distinct classes is beneficial to provide a well-performing discrimination. We pull the local node towards the global node of the same class and push them away from the global node of different classes. Second, we postulate that a well-structural graph neural network possesses similarity for neighbors due to the inherent adjacency relationships. However, aligning each node with adjacent nodes hinders discrimination due to the potential class inconsistency. We transform the adjacency relationships into the similarity distribution and leverage the global model to distill the relation knowledge into the local model, which preserves the structural information and discriminability of the local model. Empirical results on three graph datasets manifest the superiority of the proposed method over counterparts.

1 Introduction

Federated learning (FL) has shown considerable potential in collaborative machine learning across distributed devices without disclosing privacy [Konečný et al., 2016; Fang and Ye, 2022]. Although FL has attracted wide research interest and witnessed remarkable progress [McMahan et al., 2017a; Zhan et al., 2020], most of them focus on the tasks like images and voices on the basis of CNN and transformer [He et al., 2016; Vaswani et al., 2017]. However, many real-world applications generate structured graphical data (e.g., knowledge graph, social network [Liu et al., 2021]), consisting of vertices and edges [Panagopoulos et al., 2021; Wang et al., 2021], while CNN and transformer can not deal with them effectively due to the inability to capture the topological structure [Kipf and Welling, 2017; Zhu et al., 2020]. For these graph applications, graph neural networks (GNN) have won praise for their impressive performance [Hamilton et al., 2017; Shchur et al., 2018] because they utilize both the independence of nodes and the unique structure of graphs to mine graph data. Therefore, for the purpose of handling the graph data across multiple participants with growing privacy concerns, Federated Graph Learning (FGL) has become a promising paradigm [Fu et al., 2022].

Notably, data heterogeneity has become an inescapable problem in Federated Learning [Kairouz et al., 2019; Huang et al., 2022a]. Specifically, the data distribution among different parties presents non-IID (identically and independently distributed) property, which results in the divergence of local direction [Huang et al., 2023; Huang et al., 2022b]. Al-
though existing methods have made efforts to restrain the local model with respect to the global model, they mainly design for typical data heterogeneity without special consideration for graph structural bias. However, previous work has demonstrated the importance of exploiting structural knowledge [Liu et al., 2023a; Liu et al., 2022b]. In this paper, we preliminarily investigate the unique characteristics of graph heterogeneity and find that there exists node semantic and graph structural bias in FGL setting. In detail, we leverage the Centered Kernel Alignment (CKA) [Kornblith et al., 2019] to measure the similarity between node representations given by different client model pairs. Especially, for structure heterogeneity, we utilize the Anonymous Walk Embedding (AWE) to generate a representation for the graph and exploit the Jensen-Shannon distance between pair of graphs to measure the discrepancy. We reveal that there exists severe divergence on both node-level semantic and graph-level structure among clients (Fig. 1). Therefore, a crucial problem for FGL is that How to calibrate both node-level semantic and graph-level structure bias in Federated Graph Learning?

For node-level semantic calibration, we argue that a well-discriminative graph neural network contrasts nodes from different classes to provide a clear decisional boundary. Inspired by the success of supervised contrastive learning [Khosla et al., 2020; Liu et al., 2023b], we naturally expect to conduct pull-push operations among different classes on local model to acquire a well-performing decisional ability. However, under federated learning, the local GNN model purely optimizes on private data and drifts towards a distinct local minimum, which means solely relying on the guidance signals provided by the local model is confusing and unreliable. Prior studies [Hu et al., 2022], attempt to reweight local model during the FL aggregation process, but this sheds little light on identity node semantics bias caused by data heterogeneity. In this work, we investigate the node semantic knowledge and calibrate it during the local training process. We propose Federated Node Semantic Contrast (FNSC), which encourages the query local node representation to be close to global node embeddings within same class and pushes it away from global node embeddings of different classes.

Besides, for graph-level structural calibration, local clients normally possess a graph that is incomplete and is biased to depict the graph structure. Existing works normally focus on reconstructing the graph structure to handle the local structural bias. For example, FedStar [Tan et al., 2023] decouples the structure information and encodes it in a personalized way, which brings extra model parameters for local updating. In this work, without more communication cost, we take a free ride to convert stiff graph structure reconstruction into structural relationship maintenance via the given global model during local training. We introduce Federated Graph Structure Distillation (FGSD). In detail, for each node, we leverage the global model to calculate the similarity of each node with its neighborhoods, based on the adjacency matrix. Then, we require the local model to generate the adjacent node similarity and mimic the global one, which leverages the global model to provide beneficial structural knowledge.

In a nutshell, we propose a novel Federated Graph Semantic and Structural Learning method (FGSSL). Our contributions are summarized as follows:
- We are the first in FGL to decouple the data heterogeneity setting to node semantic level and graph structural level bias respectively. From this perspective, we can ameliorate final degraded performance by calibrating the local training drift, which sheds good light on future research in solving the non-IID problem in FGL scenarios.
- We introduce a novel federated graph learning (FGSSL) frame for both node and graph-level calibration. The former Federated Node Semantic Contrast calibrates local node semantics with the assistance of the global model without compromising privacy. The latter Federated Graph Structure Distillation transforms the adjacency relationships from the global model to the local model, fully reinforcing the graph representation with aggregated relation.
- We conduct extensive experiments on benchmark datasets to verify that FGSSL achieves superior performance over related methods. Taking a free ride with the global model, it does not introduce additional communication rounds and shows stronger privacy since it does not require additional shared sensitive prior information.

2 Related Work

2.1 Federated Graph Learning

Federated graph learning (FGL) facilitates the distributed training of graph neural networks (GNN). Previous literature on FGL can be categorized into two types: inter-graph and intra-graph. Inter-graph FGL involves each participant possessing a set of graphs and collectively participating in federated learning (FL) to improve the modeling of local data or generate a generalizable model [Xie et al., 2021]. In contrast, intra-graph FGL involves each participant owning only a subset of the entire graph and the objective is to address missing links [Zhang et al., 2021] or discover communities [Baek et al., 2022]. However, both of them are confronted with the non-IID issue which degrades the collaboratively learned model performance. Conventional methods solving the non-IID in FL field (e.g., FedProx [Li et al., 2020] and MOON [Li et al., 2021]) meet the absence of design for FGL scenarios. Some preceding methods are dedicated to handling the non-IID problem for FGL. FedGCN [Hu et al., 2022] tries to reweight local model parameters via an attention mechanism. FILT+ [Zhu et al., 2021a] pulls the local model closer to the global model by minimizing the loss discrepancy between a local model and the global model. However, they focus on leveraging the issue from model respect and fail to effectively exploit the unique characteristics of the graph data. In this paper, we consider inter-graph FGL and deal with the non-IID via exploiting the graphic characteristics and decoupling into node-level semantic and graph-level structure calibration.

2.2 Contrastive Learning on Graphs

In recent years, contrastive learning has seen a resurgence of interest in the field of visual representation learning [He et al., 2020; Chen et al., 2020]. This success has spurred a wealth of research exploring the adaptation of contrastive learning to graph-like data for self-supervised methods [Zhu et al., 2021b; Liu et al., 2022a]. Traditional unsupervised
methods on graph representation learning approaches [Grover and Leskovec, 2016; Perozzi et al., 2014], adhere to a contrastive structure derived from the skip-gram model. The graph autoencoder (GAE) [Kipf and Welling, 2016] is a self-supervised learning technique that aims to reconstruct the graph structure while The MVGRL [Hassani and Khasabmadi, 2020] intends to do node diffusion and compare node representation to augmented graph representation in order to learn both node-level and graph-level representation. Similar to SimCLR [Chen et al., 2020], GRACE [Zhu et al., 2020] constructs two augmented views of a graph by randomly perturbing nodes and edges, and subsequently learns node representations by pushing apart representations of every other node while bringing together representations of the same node in the two different augmented graphs within the same network. Apart from self-supervised tasks, SupCon [Khosla et al., 2020] firstly extend the self-supervised batch contrastive approach to the fully-supervised setting. In this work, we examine the contrastive method in distributed systems and conduct a inter-view based contrast between the global and local models respectively. Moreover, we consider the supervised contrast that leveraging the label as a signal to choose positive samples for calibrating the node embedding to be more similar to the global node embedding.

2.3 Knowledge Distillation

Knowledge Distillation (KD) [Hinton et al., 2015] is a technique that has been extensively studied and applied in various areas of machine learning, including image classification, natural language processing, and graph representation learning. The key aspect of KD is transferring knowledge from a complex and powerful teacher model to a more limited student model. In many works, knowledge distillation is typically used to train a smaller student network under the guidance of a larger teacher network with minimal to no performance degradation [Wang and Yoon, 2021]. In practice, knowledge distillation forces the feature or logit output of the student network to be similar to that of the teacher network. Researchers have attempted to improve knowledge distillation methods by introducing new techniques such as model distillation [Mullapudi et al., 2019], feature distillation [Romero et al., 2015], and relation distillation [Park et al., 2019]. In this work, we focus on ameliorating the heterogeneity of graph structure by adapting relation-based KD techniques for the FGL domain. We first transform the adjacency relationships into similarity distribution from global view, then distill them into the local model. In this way, we leverage aggregated contextual neighborhood information from global view and calibrate the drift caused by graph structure from the locally biased data.

3 Methodology

3.1 Preliminaries

Graph Neural Network. Graph neural networks (GNN), e.g., graph convolutional networks (GCN) [Kipf and Welling, 2017] and Graph Attention Networks (GAT) ([Veličković et al., 2017]), improved the state-of-the-art in informative graph data with their elegant yet powerful designs. In general, given the structure and feature information of a graph \( \mathcal{G} = (V, A, X) \), where \( V, A, X \) denote nodes, adjacency matrix and node feature respectively, GNN targets to learn the representations of graphs, such as the node embedding \( h_i \in \mathbb{R}^d \). A GNN typically involves two steps: the processes of message propagation and neighborhood aggregation. In this process, each node in the graph iteratively collects information from its neighbors with its own information in order to update and refine its representation. Generally, an \( L \)-layer GNN can be formulated as

\[
h_i^{(l+1)} = \sigma(h_i^{(l)}, AGG([h_j^{(l)}, j \in A_i]), \forall l \in [L],
\]

where \( h_i^{(l)} \) denotes the representation of node \( v \) at the \( l \)th layer, and \( h_i^{(0)} = v_i \) represents the node feature. \( A_i \) is defined as the neighbors of node \( v_i \). \( AGG(\cdot) \) is a aggregation function that can vary for different GNN variants, and \( \sigma \) means a activation function.

After \( L \) message-passing layers, the final node embedding \( h_i \) is passed to a project head \( F \) to obtain logits:

\[
z_i = F(h_i).
\]

In this paper, we examine proposed FGSSL in node-level tasks (e.g., node classification), and \( F \) is defined as the classifier head. Specially, we utilize L-1 layers as GNN feature extractor and the L layer as \( F \).

Centralized Aggregation

In vanilla FL setting there is always a central server with \( M \) clients, the \( m \)-th client owns a private dataset \( D^m \) and |\( D | \) is the total size of samples over all clients. FedAvg [McMahan et al., 2017b] is a foundational algorithm in the field of federated learning, which serves as a starting point for the design of more advanced FL frameworks. It operates by aggregating the updated model parameters from individual clients and redistributing average of these parameters back to all clients:

\[
\theta \leftarrow \sum_{m=1}^{M} \frac{|D^m|}{|D|} \theta^m.
\]

In this study, we utilize the Federated Learning (FL) framework to enable collaborative learning on isolated graphs among multiple data owners, without the need to share raw graph data. By doing so, we aim to obtain a global node classifier. Specifically, when model parameters are set to \( \theta \) for the Graph Neural Network (GNN) encoder and classifier \( F \), we formalize the global objective:

\[
\arg \min_{\theta} \frac{1}{M} \sum_m^M \mathcal{L}^m(\theta^m; D^m).
\]

Normally, the loss function \( \mathcal{L}^m \) in Eq. (5) is cross-entropy loss as each node which is optimized with softmax operation:

\[
\mathcal{L}^C_E = -1_{c_i} \log(\text{softmax}(z_i)),
\]

where \( 1_{c_i} \) denotes the one-hot encoding of the label \( c_i \).
3.2 Motivation

Commonly, federated graph learning aims at training a shared global GNN model, where clients have their own graphs and do not expose private data. In real-world applications, heterogeneous data distribution exists among clients. Therefore, clients present divergent optimization directions, which impair the performance of the global GNN model. We also show that this client divergence manifests in node-level semantics and graph-level structure aspects. We leverage the pairwise Centered Kernel Alignment (CKA) [Kornblith et al., 2019] and calculate the similarity between arbitrary GNN models on the same input testing samples. CKA generates the similarity score ranging from 0 (not at all similar) to 1 (identical). We select 20 clients and train the local GNN model for 100 epochs, simultaneously taking the node output from different models as node representation. As shown in Fig. 1, considering both node semantics and graph structure calibration into account is beneficial to learning a better shared GNN model.

3.3 Proposed Method

Federated Node Semantic Contrast. Generally, the goal of node classification is to identify all samples. Thus, the GNN module should maintain the discernible patterns. Inspired by the success of supervised contrastive learning, we naturally expect to contrast the node features of different classes. For the local model, we pull the node feature vectors closer to the positive samples from the same semantics and push them far away from negativity with distinct classes. Specifically, for the node $v_i$, its embedding $h^m_i$ generated by local GNN encoder $G^m(\cdot)$ with its ground truth $c_i$, the positive samples are other nodes belonging to the same class $c_i$, while the negatives are the nodes from the different classes $\mathcal{C}\setminus c_i$. Our supervised, local node-wise contrastive loss is defined as:

$$
\mathcal{L}^C_{i} = - \frac{1}{|P_i|} \sum_{p \in P_i} \log \frac{\varphi(h^m_i, h^m_p, \tau)}{\sum_{k \in K_i} \varphi(h^m_i, h^m_k, \tau)},
$$

where $P_i$ and $K_i$ denote the collections of the positive and negative samples sets for the node $v_i$. We define the $\tau$ as a contrastive hyper-parameter and $\varphi$ is formulated as:

$$
\varphi(h, h', \tau) = \exp\left(\frac{h \cdot h'}{|h| |h'| / \tau}\right).
$$

However, it is widely known that private models present drift from the ideal global optima. Thus, naively leveraging the private model to provide the positive and negative sets would further skew the local optimization direction. In our work, we argue that the shared global model aggregates knowledge from multiple parties and presents less bias than the local model. In this paper, we propose Federated Node Semantic Contrast (FNSC), which leverages the global model to provide positive and negative cluster representations for each local node embedding. We further reformulate the aforementioned supervised node contrastive learning as follows:

$$
\mathcal{L}^F_{i} = - \frac{1}{|P_i|} \sum_{p \in P_i} \log \frac{\varphi(h^m_i, h^m_p, \tau)}{\sum_{k \in K_i} \varphi(h^m_i, h^m_k, \tau)},
$$

where $h^o$ denotes the node embedding generated by the GNN encoder $G^o(\cdot)$. Moreover, given the node embedding $h^m_i$ generated by local GNN encoder $G^m(\cdot)$, we pull the node $v_i$ from local view and its pairwise one $h^o_k$ in global view together, simultaneously pull it and nodes from global view with the same class $c_i$ together.

Notably, the recent success of contrastive learning in image or video processing is largely due to carefully designed image augmentations [Ye et al., 2022; Ye et al., 2019]. These augmentations allow the model to explore a wider range of underlying semantic information and obtain better performance. In this section, we adopt a similar strategy for graph data by using an augmentation module, denoted by $\text{Aug}(\cdot)$, to generate two different views of the graph. Prior research has produced various methods for graph augmentation, which can be divided into two categories: topology (structure) transformation and feature transformation (e.g., Edge Removing and Feature Masking) [Zhu et al., 2021b; Zhu et al., 2020]. In order to enforce local clients to acquire a well-discriminative ability, we leverage both augmentations in our augmentation modules. Furthermore, we propose an asymmetric design for the contrast process, which utilizes stronger augmentations for the local GNN and weaker augmentations for the global GNN, given by $G_1 = \text{Aug}_1(G)$ for strong $\text{Aug}(\cdot)$ and $G_2 = \text{Aug}_w(G)$ for weak $\text{Aug}(\cdot)$. This would give local clients great strength to optimize towards the global direction, meanwhile, the global model can provide stable contextual semantic information to local training process. We further demonstrate the effectiveness of this asymmetric augmentation strategy in Tab. 3.

Federated Graph Structure Distillation. For graph-level calibration, it is normally assumed that adjacent nodes will share similar representations. However, under federated learning, each client fails to effectively depict this relationship because local data is normally incomplete. The straightforward solution is to directly align the query local node feature with the neighborhood nodes from the global model. However, it could potentially disrupt the discriminability because neighboring nodes probably belong to different classes. Motivated by similarity knowledge distillation [Abbasi Kooohpayegani et al., 2020; Fang et al., 2021; Tejankar et al., 2021], we propose Federated Graph Structure Distillation (FGSD) to overcome semantic inconsistency of adjacent nodes, which maintains graphic structure knowledge via the support of the global model. We measure the similarity of the query node with neighboring nodes from the global model output and then optimize the local network to mimic the similarity distribution from the global view. Specifically, for the node $v_i$, $z_i$ is the logit output given by the node classifier $F$, we denote the $A_i$ as the neighborhood node set and define the $S^g(v_i, A_i)$ as the similarity of the selected node vector with adjacent nodes computed by the global model:

$$
S^g_{j} = [S^g_{j1}, \ldots, S^g_{j|A_i|}],
$$

where $\omega$ is the distillation hyper-parameter, $(\cdot)^T$ means transpose operation. Then, we measure similarity distribution
from the $m$ local model, $S^m(v_i, A_i)$, which is formed by:

$$S^m(v_i, A_i) = [S^m_1, \ldots, S^m_{|A_i|}],$$

$$S^m(v_i, A_i) = \exp \left( \sum_{j \in A_i} \exp \left( \frac{(z^m_i \cdot z^m_j)^T}{\omega} \right) \right).$$  (10)

The FGSD (Federated Graph Structure Distillation) loss is calculated as the following:

$$L_{FGSD} = S^g(v_i, A_i) \log \frac{S^g(v_i, A_i)}{S^m(v_i, A_i)}.$$  (11)

Finally, the overall objective to be maximized is then formalized as the average across all nodes of the accumulation of the losses discussed above and is defined by:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left( \mathcal{L}^{CE} + \lambda_C \mathcal{L}^{FNSC} + \lambda_D \mathcal{L}^{FGSD} \right).$$  (12)

To sum up, Federated Graph Semantic and Structural Learning (FGSSL) leverage the global model to simultaneously calibrate local model from the node-level semantics and graph-level structure, which effectively handles the heterogeneous graph data and learns a well-performing global GNN model. We further illustrate the FGSSL in the Algorithm 1.

4 Experiments

4.1 Experimental Setup

In this paper, we perform experiments on node-level tasks defined on graph data: we choose node classification to confirm the efficacy of FGSSL in various testing environments.

Datasets. For node classification, our experiments are conducted on three benchmark datasets for the citation networks:

- **Cora** [McCallum et al., 2000] dataset consists of 2708 scientific publications classified into one of seven classes. There are 5429 edges in the network of citations. 1433 distinct words make up the dictionary.
- **Citeseer** [Giles et al., 1998] dataset consists of 3312 scientific publications classified into one of six classes and 4732 edges. The dictionary contains 3703 unique words.
- **Pubmed** [Sen et al., 2008] dataset consists of 19717 scientific papers on diabetes that have been categorized into one of three categories in the PubMed database. The citation network has 44338 edges in it. A word vector from a dictionary with 500 unique terms that is TF/IDF weighted is used to describe each publication in the dataset.

Network Structure. Since the GAT [Veličković et al., 2017] is a powerful and widely used benchmark network in graph representing learning, we realize two layers GAT with parameter $\theta$, decoupling it into feature extractor $G(\cdot)$ and unified classifier $F(\cdot)$. The hidden dimensions are 128 for all datasets, and classifier $F$ maps the embedding from 128 dimensions to 7,6,3 dimensions, which is the number of classification classes for Cora, Citeseer, and Pubmed respectively.

Graph Augmentation Strategy. Generating views is a key component of contrastive learning methods. In the graph domain, different views of a graph provide different contexts for each node. We follow augmentation mentioned in [Zhu et al., 2021b], [Zhu et al., 2020] to construct a contrastive learning scheme. In FGSSL, we leverage two methods for new graph view generation, removing edges for topology and masking features for node attributes.

- **Removing edges (RE).** It randomly removes a portion of edges in the original graph.
- **Masking node features (MF).** It randomly masks a fraction of dimensions with zeros in node features.

Implement Details. We utilize the community detection algorithm: Louvain, to simulate the subgraph systems. To stimulate the non-iid scene, this algorithm partitions the graph into multiple clusters and then assigns them to clients with unbalanced node numbers. To conduct the experiments uniformly and fairly, we split the nodes into train/valid/test sets, where the ratio is 60% : 20% : 20%. As for all networks, we use SGD [Robbins and Monro, 1951] as the selected optimizer with momentum 0.9 and weight decay $5e - 4$. The communication round is 200 and the local training epoch is 4 for all datasets. The metric used in our experiments is the node classification accuracy on the testing nodes and we re-
Algorithm 1: The FGSSL Framework

Input: communication rounds $T$, local epochs $E$, participant scale $M$, $m$th client private graph data $G^m(V, A; X; Y)$, private model $\theta^m$, temperature $\tau$, distillation parameter $\omega$, loss weight $\lambda_C$ and $\lambda_D$, learning rate $\eta$.

Output: The final global model $\theta_T$

for $t = 1, 2, ..., T$
  for $m = 1, 2, ..., M$ in parallel do
    $\theta^m_t \leftarrow \text{LocalUpdating}(\theta^m_t, m)$
  end
  $\theta_{t+1} \leftarrow \sum_{m=1}^M [D^m/|D|] \theta^m_t$
end
return $\theta_T$

LocalUpdating($\theta^m_t, m$):

Initialize $G^m(\cdot), F^0(\cdot) \leftarrow \theta_T$.

Initialize $G^m(\cdot), F^m(\cdot) \leftarrow \theta_T$.

Freeze $G^0(\cdot), F^0$.

for $e = 1, 2, ..., E$ do
  $Z = \theta^m(X)$
  $L^{CE} \leftarrow C(E(Z, Y))$ in Eq. (5)
  $G_1, G_2 \leftarrow \text{Aug}_u(G), \text{Aug}_w(G)$
  $H^m, H^p \leftarrow G^m(G_1), G^p(G_2)$
  $L^{FNSC} \leftarrow (H^m, H^p)$ through Eq. (8)
  $Z^m, Z^p \leftarrow F^m(H^m), F^p(H^p)$
  $S^m(V, A) \leftarrow (Z^m)$ by Eq. (9)
  $S^p(V, A) \leftarrow (Z^p)$ by Eq. (10)
  $L^{FGSD} \leftarrow (S^m(V, A), S^p(V, A))$ through Eq. (12)
  $\theta^m \leftarrow \theta^m - \eta \nabla \mathcal{L}$
end
return $\theta^m$

port the averaged accuracy and the standard deviation over several random repetitions.

Counterparts. (1) Local each client train their model locally, (2) Global the server leverage the complete graph for training. For rigorous evaluation, we compare our FGSSL against popular federated strategies in FGL setting. (3) FedAvg [AISTATS’17 [McMahan et al., 2017b]], (4) FedProx [MLSys’21 [Li et al., 2020]], (5) FedOpt [ICLR’21 [Reddi et al., 2021]], (6) FedSage [NeurIPS’21 [Zhang et al., 2021]].

4.2 Experimental Results

Performance Comparison. The results of federated node classification for various methods under three non-IID settings are presented in Tab. 1. These results indicate that FGSSL outperforms all other baselines and demonstrates a significant and consistent improvement compared to the conventional FedAvg algorithm in the FGL setting. Additionally, personalized FL algorithms such as FedProx and FedOpt demonstrate better performance than vanilla aggregation by utilizing a universal solution to the non-IID problem. Specialized methods in the FGL field such as FedSage also perform better than common baselines, which is achieved through the simultaneous training of generative models for predicting missing links.

Convergence Analysis. Fig. 3 shows curves of the average test accuracy during the training process across five random runs conducted on the Citeseer datasets. It can be observed that FGSSL dominates the other methods in non-IID setting on the average test accuracy and achieves a stable convergence.

4.3 Ablation Study

Effects of Key Components Mechanism. To better understand the impact of specific design components on the overall performance of FGSSL, we conducted an ablation study in which we varied these components. For the variant without FNSC and FGSD, we utilize the vanilla FGL setting with 2-layer GAT. As shown in Tab. 2, by exploiting both components, the best performance is achieved in all three graph datasets. It also suggests that FNSC plays a more crucial role than FGSD, which means the calibration in node semantics is stronger than the calibration in graph structure, and feature heterogeneity is more serious than graph heterogeneity in non-IID setting. Moreover, the contribution made by FGSD is still not negligible and can benefit the learning process.

Hyper-parameter study. We compare the downstream task performance under different $\tau$ and $\omega$ values with five clients. Results are shown in Tab. 1, where Fig. 4(a) shows results when $\omega$ is fixed at 5, and Fig. 4(b) shows results under $\tau = 0.1$. It indicates that choosing $\tau$ can affect the strength of
the contrastive method, where a smaller temperature benefits training more than higher ones, but extremely low temperatures (0.01) are harder to train due to numerical instability. Across different datasets, the optimal $\tau$ is constantly around 0.1. For choosing an appropriate $\tau$ in (Eq. (9) and Eq. (10).), we find that the performance is not influenced much unless $\omega$ is set to extreme values like 0.1.

**Discussion on Augmentation Strategies.** As demonstrated in Tab. 3, different augmentation strategies were implemented within the augmentation module of proposed method. The experimental results indicate that utilizing two levels of augmentation improves performance. Specifically, on the one hand, using double-weak augmentation strategies did not result in a significant improvement when compared to baseline methods. On the other hand, double-strong augmentation strategies led to improved results as they allowed for exploration of rich semantic information through the supervised contrastive method. Additionally, the combination of strong and weak augmentation strategies at local and global levels, respectively, resulted in the highest overall performance, in accordance with our descriptions of them in Sec. 3.3.

| Methods | Cora $M = 5$ | Cora $M = 7$ | Cora $M = 10$ | Citeseer $M = 5$ | Citeseer $M = 7$ | Citeseer $M = 10$ | Pubmed $M = 5$ | Pubmed $M = 7$ | Pubmed $M = 10$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>87.78 ±1.34</td>
<td>76.91 ±1.02</td>
<td>83.81 ±0.33</td>
<td>87.51 ±0.41</td>
<td>87.75 ±0.41</td>
<td>87.60 ±0.53</td>
<td>88.34 ±0.34</td>
<td>88.01 ±0.23</td>
<td>88.25 ±0.60</td>
</tr>
<tr>
<td>Local</td>
<td>61.54 ±0.83</td>
<td>73.85 ±1.20</td>
<td>85.21 ±0.24</td>
<td>77.82 ±0.13</td>
<td>77.91 ±0.59</td>
<td>77.15 ±0.45</td>
<td>86.22 ±0.25</td>
<td>86.22 ±0.25</td>
<td>85.27 ±0.12</td>
</tr>
<tr>
<td>FedAvg</td>
<td>86.63 ±0.35</td>
<td>76.37 ±0.43</td>
<td>85.29 ±0.83</td>
<td>85.27 ±0.43</td>
<td>86.27 ±0.12</td>
<td>85.61 ±0.17</td>
<td>86.01 ±0.17</td>
<td>86.01 ±0.17</td>
<td>86.27 ±0.12</td>
</tr>
<tr>
<td>FedProx</td>
<td>86.60 ±0.59</td>
<td>77.15 ±0.45</td>
<td>85.21 ±0.24</td>
<td>84.01 ±0.59</td>
<td>86.27 ±0.12</td>
<td>86.01 ±0.17</td>
<td>86.27 ±0.12</td>
<td>86.01 ±0.17</td>
<td>86.27 ±0.12</td>
</tr>
<tr>
<td>FedOpt</td>
<td>86.31 ±0.24</td>
<td>76.96 ±0.34</td>
<td>84.39 ±0.42</td>
<td>84.10 ±0.19</td>
<td>84.39 ±0.42</td>
<td>84.10 ±0.19</td>
<td>84.39 ±0.42</td>
<td>84.10 ±0.19</td>
<td>84.39 ±0.42</td>
</tr>
<tr>
<td>FedSage</td>
<td>88.86 ±0.15</td>
<td>84.39 ±0.42</td>
<td>84.39 ±0.42</td>
<td>84.39 ±0.42</td>
<td>84.39 ±0.42</td>
<td>84.39 ±0.42</td>
<td>84.39 ±0.42</td>
<td>84.39 ±0.42</td>
<td>84.39 ±0.42</td>
</tr>
<tr>
<td>FGSSL</td>
<td>88.34 ±0.34</td>
<td>84.43 ±0.23</td>
<td>87.75 ±0.41</td>
<td>87.75 ±0.41</td>
<td>87.75 ±0.41</td>
<td>87.75 ±0.41</td>
<td>87.75 ±0.41</td>
<td>87.75 ±0.41</td>
<td>87.75 ±0.41</td>
</tr>
</tbody>
</table>

Table 1: Comparison with the state-of-the-art methods on Cora, Citeseer and Pubmed datasets. The best result is bolded. $\uparrow$ means improved accuracy compared with FedAvg. ± presents the standard deviation. Please see details in Sec. 4.2.

| FNSC FGSD | Cora $M = 5$ | Cora $M = 7$ | Cora $M = 10$ | Citeseer $M = 5$ | Citeseer $M = 7$ | Citeseer $M = 10$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>86.63 ±0.35</td>
<td>76.37 ±0.43</td>
<td>85.29 ±0.83</td>
<td>77.15 ±0.45</td>
<td>85.21 ±0.24</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>86.86 ±0.32</td>
<td>87.91 ±0.93</td>
<td>84.39 ±0.42</td>
<td>84.39 ±0.42</td>
<td>84.39 ±0.42</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>88.01 ±0.23</td>
<td>79.89 ±0.79</td>
<td>84.39 ±0.42</td>
<td>77.82 ±0.13</td>
<td>77.91 ±0.59</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>88.34 ±0.34</td>
<td>80.43 ±0.23</td>
<td>87.75 ±0.41</td>
<td>87.75 ±0.41</td>
<td>87.75 ±0.41</td>
</tr>
</tbody>
</table>

Table 2: Ablation study of key components of our method in Cora and Citeseer datasets with clients S/7/10. See Sec. 4.3 for details.

| Local Global | Cora $M = 5$ | Cora $M = 7$ | Cora $M = 10$ | Citeseer $M = 5$ | Citeseer $M = 7$ | Citeseer $M = 10$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>weak weak</td>
<td>87.24 ±0.99</td>
<td>77.72 ±0.97</td>
<td>77.72 ±0.97</td>
<td>77.72 ±0.97</td>
<td>87.24 ±0.99</td>
<td>77.72 ±0.97</td>
</tr>
<tr>
<td>weak strong</td>
<td>86.91 ±0.68</td>
<td>79.59 ±0.79</td>
<td>79.59 ±0.79</td>
<td>79.59 ±0.79</td>
<td>86.91 ±0.68</td>
<td>79.59 ±0.79</td>
</tr>
<tr>
<td>strong strong</td>
<td>88.01 ±0.23</td>
<td>79.89 ±0.79</td>
<td>79.89 ±0.79</td>
<td>79.89 ±0.79</td>
<td>88.01 ±0.23</td>
<td>79.89 ±0.79</td>
</tr>
</tbody>
</table>

Table 3: Analysis on augmentation strategies: Effect of using weak or strong augmentations for two datasets trained on the sole FNSC component with 200 epochs. See Sec. 4.3 for details.

Figure 5: Visualization of classification result. The figure number corresponds to the method on the Citeseer dataset with $m = 5$. Logits are colored based on class labels.

5 Conclusion

In this paper, we propose a novel federated graph learning framework, namely FGSSL, that mitigates the non-IID issues via appropriately calibrating the heterogeneity both on the node-level semantic and graph-level structure. We develop two key components to solve the problems respectively. On the one hand, we leverage the contrastive-based method to correct the drift node semantics from the global ones that have identical semantic information and achieve a high level of semantic discrimination at node level. On the other hand, we consider transforming adjacency relationships into a similarity distribution and utilizing a global model to distill this information into the local model, which maintains the structural information and corrects the structure heterogeneity. Experimental results illustrate that FGSSL consistently outperforms the state-of-the-art methods in federated graph scenarios.

Acknowledgement

This work is partially supported by National Natural Science Foundation of China under Grant (62176188, 62225113), the Key Research and Development Program of Hubei Province (2021BAA187), Zhejiang lab (NO.2022NF0AB01), CCF-Huawei Populus Grove Fund (CCF-HuaweiTC202203), the Special Fund of Hubei Luojia Laboratory (220100015), the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant (2019AEA170).


