Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

ADPFedGNN: Adaptive Decoupling Personalized Federated Graph Neural Network

Zeli Guan[1,2] , **Yawen Li**[3*] , **Junping Du**[1,2] , **Runqing Tang**[1,2] , **Xiaolong Meng**[1,2]
[1]School of Computer Science (National Pilot School of Software Engineering), Beijing University of Posts and Telecommunications
[2]Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia
[3]School of Economics and Management, Beijing University of Posts and Telecommunications
Beijing 100876, PR China
guanzeli@bupt.edu.cn, warmly0716@126.com, junpingdu@126.com, tangrunqing@bupt.edu.cn, mengxiaolong@bupt.edu.cn

Abstract

Personalized federated graph neural networks (PFGNN) are an emerging technology that allows multiple graph data owners to collaboratively train personalized models without sharing raw data. However, the Non-IID nature of graph data can cause the coupling of global and local knowledge parameters, which disrupts the optimization in personalized federated learning. Additionally, node neighbors may carry global and local knowledge, and their direct inclusion in training may introduce noise, degrading federated model performance. In this work, we propose the Adaptive Decoupling Personalized Federated Graph Neural Network (ADPFedGNN), which leverages multi-party collaboration to train personalized models for classifying local client graph nodes. We use two automatically updated masks and mutual information minimization to decouple global and local parameters in FGNN. We employ reinforcement learning to adaptively select appropriate neighbors for training global or local knowledge-related parameters while filtering out irrelevant nodes. We also design a personalized federated masked parameter aggregation mechanism that efficiently updates local personalized model parameters and aggregates the masked parameters. Experimental results on three public datasets demonstrate that ADPFedGNN outperforms existing methods, achieving average improvements of 5.66 percent, 5.83 percent, and 12.45 percent in ACC, F1, and Recall, respectively.

1 Introduction

Graph data finds widespread applications in various domains, such as social networks [Quan *et al.*, 2023], financial transactions [Pareja *et al.*, 2020], and recommendation systems [Yu *et al.*, 2022]. Representing data as graphs, with entities as nodes and relationships as edges, better reflects real-world scenarios [Zhang *et al.*, 2024a]. Analyzing these graphs allows leveraging node relationships to generate accurate representations, which can significantly enhance subsequent node classification tasks [Khoshraftar and An, 2024]. However, in real-world applications, high-quality data is typically owned by governments, enterprises, or other organizations. Due to privacy concerns, regulatory restrictions, and conflicting interests, data sharing is typically restricted [Wang *et al.*, 2020], thereby limiting the effectiveness of graph-based node classification models.

Federated Graph Neural Network (FGNN) enable the training of effective models using multi-party graph data while keeping data localized [Fu *et al.*, 2022; Li *et al.*, 2024b]. Due to the attribute shifts in data across clients, FGNN face the challenge of Non-IID data [Wan *et al.*, 2024], which disrupts the performance of federated learning. To address this, personalized federated learning approaches allow clients to adopt differentiated aggregation strategies [Dhillon *et al.*, 2020; Long *et al.*, 2023a; Zhang *et al.*, 2024b], primarily through techniques such as gradient weighting [He *et al.*, 2021b; Zhang *et al.*, 2023], regularization [Li *et al.*, 2020; Li *et al.*, 2021b], and client sampling [Fraboni *et al.*, 2021; Long *et al.*, 2023b]. In FGNN, each client holds knowledge. During federated training, this knowledge is encoded into the model parameters, comprising global knowledge applicable across clients and client-specific local knowledge relevant only to the individual client. Existing methods typically couple global knowledge-related parameters with local knowledge-related parameters, which can lead to interference from local knowledge on global knowledge-related parameters during the federated learning process.

For graph data, the Non-IID nature introduces structural shift issues. Some approaches mitigate these shifts by sharing graph data information [Zhang *et al.*, 2021a; Huang *et al.*, 2023]. However, graph node neighbors may carry global or local knowledge, and directly incorporating them into federated training without distinguishing their suitability for training global or local knowledge-related parameters can introduce noise, ultimately interfering with the training process and degrading the performance of FGNN [Tang *et al.*, 2021; Li *et al.*, 2021a]. Determining which neighbors should con-

*Corresponding author

tribute to training parameters related to global or local knowledge is crucial. Nevertheless, there is a lack of studies that effectively address how to select and assign neighbors to these roles in FGNN.

Based on the above, we identify two key challenges for PFGNN:

- Challenge 1: How to decouple global knowledge from client-specific local knowledge parameters during the training process, ensuring that only parameters related to global knowledge are included in federated aggregation, while effectively handling attribute shifts in feature distributions across clients.
- Challenge 2: How to effectively select neighbors that contribute to the training of either global or local knowledge-related parameters, filtering out irrelevant nodes that do not contribute to model performance, especially under structural shifts where neighbor relationships vary across clients.

To address these two challenges, we propose Adaptive Decoupling Personalized Federated Graph Neural Network (ADPFedGNN), which leverages multi-party graph data to train models capable of classifying nodes in each client's graph data. For Challenge 1, we propose federated mask-based parameter decoupling method to separate local and global knowledge-related parameters, and personalized federated masked parameter aggregation method to prevent interference between these parameters during federated aggregation. For Challenge 2, we propose reinforcement federated adaptive neighbor selection strategy, which adaptively selects node neighbors for federated training. ADPFedGNN effectively prevents interference from local knowledge-related parameters on other clients' models, while efficiently leveraging multi-party data to train personalized federated Graph Neural Network. Extensive experiments conducted on three public datasets validate the effectiveness of the proposed method.

Our main contributions include:

- We propose a federated mask based parameter decoupling method that utilizes an automatically updated mask mechanism and mutual information minimization to decouple model parameters into global and local components. The mask is updated based on the client's local training gradients and their similarity to the global model, ensuring effective decoupling.
- We propose a personalized federated masked parameter aggregation method that aggregates the masked model parameters from each client to form the global model. The global parameters are then updated using the inverted local masks, preventing the global model from disrupting the local models' adaptability to client-specific data.
- We propose a reinforcement federated adaptive neighbor selection strategy that uses reinforcement learning to select suitable node neighbors for training global or local knowledge related parameters while filtering out irrelevant nodes. This approach enhances the performance of PFGNN.

2 Related Work

2.1 Federated Learning

Federated learning is a distributed paradigm that addresses data silos [Liu *et al.*, 2024; Li *et al.*, 2024a]. For graph data, FedGraphNN [He *et al.*, 2021a] is a federated learning benchmark system for GNN, supporting multi-domain datasets. SpreadGNN [He *et al.*, 2021b] enables federated multi-task learning, and FGGP [Wan *et al.*, 2024] enhances generalization and classification by introducing cluster prototypes and global knowledge contrast.

However, data across clients often exhibits non-IID characteristics, making it difficult to train a single global model. Personalized federated learning methods, such as FedProx [Dhillon *et al.*, 2020], introduce regularization to address data heterogeneity, while FedSEM [Long *et al.*, 2023b] clusters clients based on model parameters to improve personalization. MOON [Li *et al.*, 2021b] improves performance through model-level contrastive learning. FedSage+[Zhang *et al.*, 2021a] improves federated node classification performance by generating missing neighbors. FedALA [Zhang *et al.*, 2023] adapts the aggregation process to local data. Despite these advancements, existing methods still couple global and local knowledge parameters, limiting their effectiveness in preventing interference from local knowledge.

2.2 Graph Sampling

As graph data scales, training on the entire graph becomes computationally and memory-intensive, making minibatch mechanisms essential for large datasets [Hamilton *et al.*, 2017]. Traditional methods, such as random sampling, ignore noisy or irrelevant links between nodes [Zhao *et al.*, 2023; Wang *et al.*, 2021]. Recent work has focused on optimizing neighbor sampling. BanditSampler [Liu *et al.*, 2020] and Thanos [Zhang *et al.*, 2021b] reduce sampling variance through multi-armed bandit formulations and novel reward functions. However, these approaches require per-node updates, making them difficult to adapt to federated learning due to the lack of shared strategies across devices.

Parameterized neighbor selection strategies have recently gained attention. Bayesian GNN [Hasanzadeh *et al.*, 2020], DSKReG [Wang *et al.*, 2021], and Learnable Sampling [Zhao *et al.*, 2023] dynamically adjust sampling probabilities, but they rely on complete computation flows, limiting their applicability in federated learning where global gradient computation is infeasible. Reinforcement learning offers a promising solution by learning parameterized strategies without relying on a complete computation flow [Lai *et al.*, 2020; Sun *et al.*, 2021; Yang *et al.*, 2020], making it ideal for federated settings.

3 Methodology

We propose ADPFedGNN to address challenges in adaptive neighbor selection and parameter coupling for personalized federated graph node classification. ADPFedGNN enables collaborative training on multi-party graph data to classify nodes in each client's local graph while ensuring data privacy.

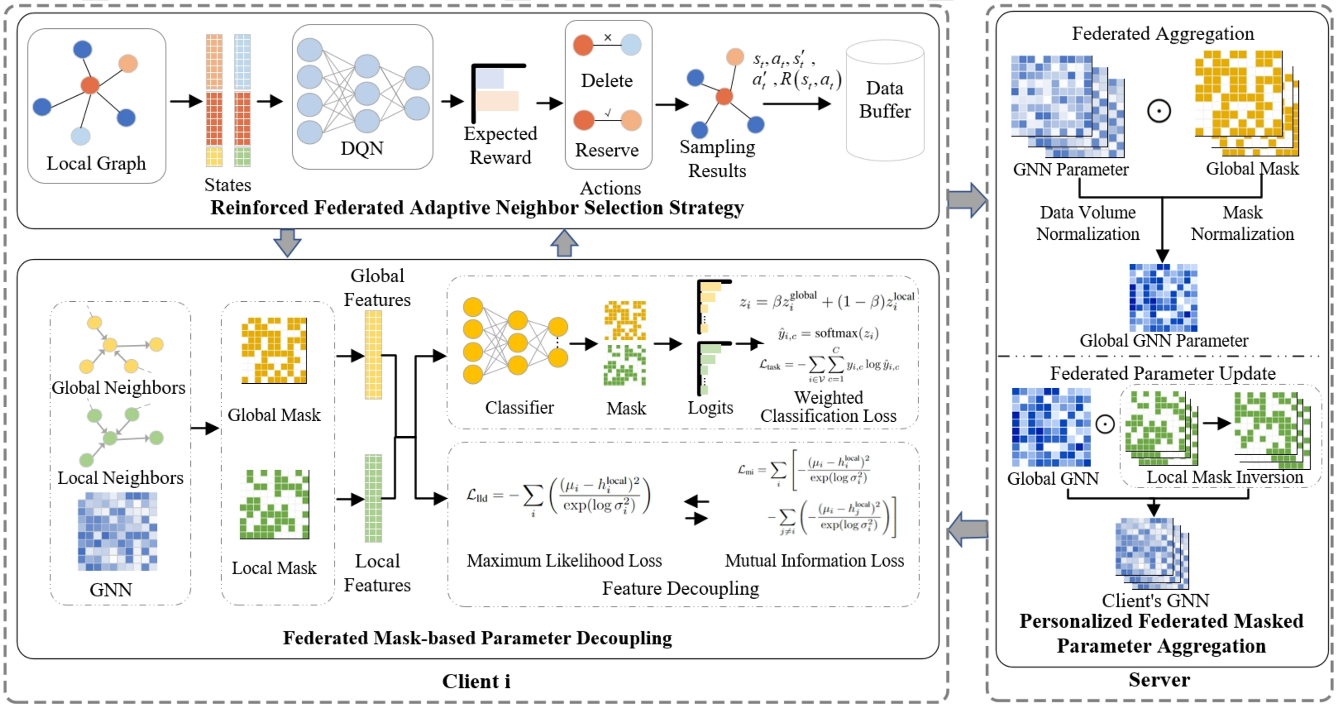


Figure 1: Overview of adaptive decoupling personalized federated graph neural network.

The method consists of three components: (1) Reinforcement federated adaptive neighbor selection strategy using reinforcement learning to select suitable neighbors for training global or local knowledge-related parameters, while filtering irrelevant nodes; (2) Federated mask-based parameter decoupling method that uses trainable masks and mutual information minimization to decouple global and local knowledge-related parameters; and (3) Personalized federated masked parameter aggregation method that shares only global parameters, preserving local model personalization. An overview of the method is shown in Figure 1.

3.1 Problem Definition

In federated graph node classification, each client C_m holds private graph data $G_m = (V_m, E_m, X_m)$ and node labels Y_m, where V_m, E_m, and X_m denote the node set, edge set, and node feature matrix. Due to privacy constraints and non-IID characteristics, clients cannot share raw data, leading to attribute and structural shifts that complicate federated learning [Wan *et al.*, 2024]. Attribute shift arises when the distribution of node features conditioned on labels varies across clients, i.e., $P_m(h \mid y) \neq P_n(h \mid y)$, even if the marginal label distribution $P(y)$ remains similar. Structural shift occurs when the relationship between edge and label distributions differs across clients, i.e., $P_m(E, Y) \neq P_n(E, Y)$, despite consistent label distributions $P_m(y) = P_n(y)$. These shifts hinder the training of a globally generalized model by introducing inconsistencies across clients, which degrade model performance. To address these challenges, we propose a personalized federated learning approach that effectively handles data heterogeneity and distribution shifts to achieve accurate node classification while preserving data privacy.

3.2 Reinforcement Federated Adaptive Neighbor Selection Strategy

We propose a reinforcement Federated Adaptive Neighbor Selection Strategy, which formulates the neighbor sampling as a Markov Decision Process (MDP) and uses a Deep Q-Network (DQN) to learn an adaptive policy.

At each training round, we adopt a minibatch mechanism to sample a two-layer block of neighbors. For each node in the batch, k neighbors are selected in each layer, forming an initial two-layer neighbor set. The DQN then estimates the expected reward for each neighbor, refining the selection by identifying suitable neighbors for updating global or local knowledge-related parameters and filtering out irrelevant or redundant ones.

State Design: The state s_{ijb} is formed by concatenating the feature vector h_i of the target node v_i, the feature vector h_j of the candidate neighbor v_j, and a binary indicator b. The binary indicator $b = 0$ denotes the selection for updating global parameters, while $b = 1$ denotes the selection for updating local parameters. The state representation enables the agent to distinguish between tasks for global and local parameter updates. The state is represented as:

$$s_{ijb} = [h_i; h_j; b] \tag{1}$$

Action Design: The action $a_{ijb} \in \{0, 1\}$ determines whether the candidate neighbor v_j is selected for the target node v_i. Actions are generated by the DQN network, which outputs the state-action value $Q(s_{ijb}, a_{ijb}; \theta_\text{q})$. Action selection is based on the Q-value distribution:

$$P(a_{ijb} = a \mid s_{ijb}) = \frac{\exp(Q(s_{ijb}, a; \theta_{\text{q}}))}{\sum_{a'} \exp(Q(s_{ijb}, a'; \theta_{\text{q}}))} \quad (2)$$

By selecting actions with higher Q-values, the DQN network identifies suitable node neighbors for the task while filtering out irrelevant nodes.

The reward $r(s_{ijb}, a_{ijb})$ is defined as the difference between the federated task loss $\mathcal{L}^r_{\text{task}}$ at round r and the average task loss from previous rounds:

$$r(s_{ijb}, a_{ijb}) = \lambda_{\text{rl}} \left(\mathcal{L}^r_{\text{task}} - \text{mean}\left(\{\mathcal{L}^{r'}_{\text{task}}\}_{r' \in \mathcal{R}} \right) \right) \quad (3)$$

Experience Replay Mechanism: To stabilize training, we use an experience replay mechanism, storing state-action-reward-next state tuples in a buffer $\mathcal{D}$. Minibatches are randomly sampled from $\mathcal{D}$ to update the DQN, helping the agent learn adaptive neighbor selection for better federated graph node classification.

3.3 Federated Mask-Based Parameter Decoupling Method

We propose a federated mask-based parameter decoupling method. This approach uses a GNN with automatic update masks to construct a global model suitable for federated aggregation and a local model retained only on the client side.

Global and local features are extracted using GNN with separate masks to ensure feature-level decoupling. The masks are updated after each federated training round, with global masks, $\mathbf{M}^{\text{feat}}_{\text{global}}$ and $\mathbf{M}^{\text{cls}}_{\text{global}}$, preserve global knowledge-related parameters, applicable across clients, while the local masks, $\mathbf{M}^{\text{feat}}_{\text{local}}$ and $\mathbf{M}^{\text{cls}}_{\text{local}}$, retain local knowledge-related parameters. The global mask assigns a value of 1 to the top $q\%$ of parameters most similar to the local training gradients and global non-zero gradients. Similarly, the local mask assigns a value of 1 to the top $q\%$ of parameters with the highest non-zero gradients from local training. This mechanism ensures that during the update, both the global and local masks focus on the most influential parameters for each respective task.

The global feature extraction is defined as:

$$h^{\text{global}}_i = \text{GNN}(x_i, N(i, a_{i,t,0}); \theta_{\text{gnn}} \odot \mathbf{M}^{\text{feat}}_{\text{global}}) \quad (4)$$

where $N(i, a_{i,t,0})$ denotes the set of neighbors selected for global parameter updates when $b = 0$. When $b = 1$, the set of selected local neighbors for updating local parameters is used for local feature extraction, defined as:

$$h^{\text{local}}_i = \text{GNN}(x_i, N(i, a_{i,t,1}); \theta_{\text{gnn}} \odot \mathbf{M}^{\text{feat}}_{\text{local}}) \quad (5)$$

where $N(i, a_{i,t,1})$ refers to the set of neighbors selected for local parameter updates.

To decouple global and local features, we minimize their mutual information using the Contrastive Log-ratio Upper Bound (CLUB) [Cheng *et al.*, 2020], which estimates the upper bound of mutual information $\mathcal{L}_{\text{mi}}$ and reduces it to achieve feature independence:

$$\mathcal{L}_{\text{mi}} = \sum_i \left[-\frac{(\mu_i - h^{\text{local}}_i)^2}{\exp(\log \sigma^2_i)} - \sum_{j \neq i} \left(-\frac{(\mu_i - h^{\text{local}}_j)^2}{\exp(\log \sigma^2_i)} \right) \right] \quad (6)$$

where μ_i and $\log \sigma^2_i$ are the mean and log-variance estimated from global features h^{global}_i using neural networks:

$$\mu_i = f_\mu(h^{\text{global}}_i; \theta_\mu), \quad \log \sigma^2_i = f_{\log \sigma^2}(h^{\text{global}}_i; \theta_{\log \sigma^2}) \quad (7)$$

where θ_μ and $\theta_{\log \sigma^2}$ represent the network parameters. These parameters are optimized through maximum likelihood estimation with the log-likelihood loss:

$$\mathcal{L}_{\text{lld}} = -\sum_i \left(\frac{(\mu_i - h^{\text{local}}_i)^2}{\exp(\log \sigma^2_i)} \right) \quad (8)$$

To classify nodes, the global and local features are passed through masked classifiers, focusing on their respective information types. The global classifier output is defined as:

$$z^{\text{global}}_i = f_{\text{cls}}(h^{\text{global}}_i; \theta_{\text{cls}} \odot \mathbf{M}^{\text{cls}}_{\text{global}}) \quad (9)$$

where z^{global}_i is the global classifier output for node i, f_{cls} represents the classifier function, θ_{cls} denotes the classifier parameters, and $\mathbf{M}^{\text{cls}}_{\text{global}}$ is the global classifier mask. Similarly, the local classifier output is defined as:

$$z^{\text{local}}_i = f_{\text{cls}}(h^{\text{local}}_i; \theta_{\text{cls}} \odot \mathbf{M}^{\text{cls}}_{\text{local}}) \quad (10)$$

where z^{local}_i is the local classifier output for node i.

The final classifier output is obtained by fusing the global and local logits:

$$z_i = \beta z^{\text{global}}_i + (1 - \beta) z^{\text{local}}_i \quad (11)$$

where z_i is the final output for node i, and $\beta \in [0, 1]$ balances the contributions of global and local classifiers. This weighted fusion ensures that the model can effectively leverage both globally shared knowledge and locally specialized patterns, enabling accurate node classification across diverse client data distributions.

The task is graph node classification, and the task loss is computed using cross-entropy:

$$\mathcal{L}_{\text{task}} = -\sum_{i \in \mathcal{V}} \sum_{c=1}^{C} y_{i,c} \log \hat{y}_{i,c}, \quad \hat{y}_{i,c} = \text{softmax}(z_i) \quad (12)$$

where $\hat{y}_{i,c}$ denotes the predicted probability of node i belonging to class c, $y_{i,c}$ is the one-hot encoded ground truth label, and $\mathcal{V}$ represents the set of nodes.

The total loss integrates the classification loss, mutual information loss, and an L2 regularization loss on the model parameters to prevent overfitting:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{mi}} \mathcal{L}_{\text{mi}} + \lambda_{\text{reg}} \left(\|\theta_{\text{gnn}}\|^2_2 + \|\theta_{\text{cls}}\|^2_2 \right) \quad (13)$$

where λ_{mi} and λ_{reg} are hyperparameters balancing the classification, mutual information, and regularization terms.

3.4 Personalized Federated Masked Parameter Aggregation Method

To achieve efficient aggregation of the global model while preserving the personalization of local models, we propose a personalized federated masked parameter aggregation method. In each federated training round, clients use two automatically updated masks, global masks $\mathbf{M}_{\text{global}}$ and local masks $\mathbf{M}_{\text{local}}$, to decouple their model parameters into global and local components. These masks are applied uniformly to both the feature extraction layers and the classifier parameters, ensuring that the global model focuses on generalizable knowledge while the local model preserves client-specific characteristics.

The aggregation goal is to compute a weighted average of global model parameters based on local data sizes and the global masks from selected clients $\mathcal{S}$. Let θ_i denote the local model parameters of client i, and $\mathbf{M}_i^{\text{global}}$ represent its global mask. The server updates the global model parameters θ^{global} using:

$$\theta^{\text{global}} = \frac{1}{\sum_{i \in \mathcal{S}} n_i} \sum_{i \in \mathcal{S}} n_i \cdot (\theta_i \odot \mathbf{M}_i^{\text{global}}) \tag{14}$$

After aggregation, the global model is distributed to clients for local updates. For client i, the local model parameters θ_i are updated using:

$$\theta_i[p] = \theta^{\text{global}}[p] \cdot (1 - \mathbf{M}_i^{\text{local}}[p]) + \theta_i[p] \cdot \mathbf{M}_i^{\text{local}}[p], \quad \forall p \tag{15}$$

where $\theta^{\text{global}}[p]$ is the global model parameter at position p, $\theta_i[p]$ is the local model parameter of client i at position p, and $\mathbf{M}_i^{\text{local}}[p]$ is the local mask value. The complementary mask $1 - \mathbf{M}_i^{\text{local}}[p]$ ensures that global parameters are applied only at positions where the local mask value is zero. The training process is detailed in Algorithm 1.

Algorithm 1 Training process of ADPFedGNN for a single epoch

Input: Local data G, batch size B.
Output: $\theta_\mu, \theta_{\log\sigma^2}, \theta_{\text{gnn}}, \theta_{\text{cls}}, \theta_q$.
Initialize parameters $\theta_{\text{gnn}}, \theta_{\text{cls}}, \theta_\mu, \theta_{\log\sigma^2}, \mathbf{M}_{\text{global}}, \mathbf{M}_{\text{local}}$.
Client-Side Training:
for batch $b \leftarrow 0$ to $B-1$ **do**
    Compute actions with Equation (2) to filter random neighbors and obtain $N(i, a_{i,t,0})$ and $N(i, a_{i,t,1})$;
    Extract global and local features using Equations (4) and (5);
    Estimate μ and $\log\sigma^2$ using Equation (7);
    Compute classification task loss using Equation (12);
    Update parameters $\theta_{\text{gnn}}, \theta_{\text{cls}}$;
    Update parameters $\theta_\mu, \theta_{\log\sigma^2}$ using the log-likelihood loss in Equation (8);
    Calculate reward using Equation (3);
    Save experience $(s_t, a_t, s'_t, a'_t, R(s_t, a_t))$ in replay buffer;
    Train Q-network θ_q on sampled mini-batches from the replay buffer;
end for
Server-Side Aggregation:
Perform federated aggregation for $\theta_\mu, \theta_{\log\sigma^2}, \theta_{\text{gnn}}$, $\theta_{\text{cls}}, \theta_q$ through Equations (14) and (15).

4 Experimental Analysis

4.1 Datasets

We conduct experiments on three public graph datasets: Cora [Sen *et al.*, 2008], CiteSeer [Sen *et al.*, 2008], and PubMed [Namata *et al.*, 2012]. To simulate federated learning with varying levels of data heterogeneity, we partition the datasets using two methods: Louvain community partitioning [Peng *et al.*, 2022], which assigns nodes to clients based on community structures, and Dirichlet label partitioning [Zhang *et al.*, 2021a], which simulates non-IID client distributions using the Dirichlet distribution, where the parameter α controls the heterogeneity—smaller values result in more imbalanced client data distributions.

4.2 Baseline Methods

We compare ADPFedGNN with the following baselines: FedAvg [McMahan *et al.*, 2017], Clustered Sampling [Fraboni *et al.*, 2021], FedProx [Dhillon *et al.*, 2020], MOON [Li *et al.*, 2021b], FedALA [Zhang *et al.*, 2023], FedSEM [Long *et al.*, 2023a], FGGP [Wan *et al.*, 2024], and FedSage+ [Zhang *et al.*, 2021a].

4.3 Experimental Setup

We use GraphSage [Hamilton *et al.*, 2017] and GAT [Veličković *et al.*, 2017] as the backbone networks. GAT experiments are limited to ADPFedGNN, FedAvg, and FedProx, as GraphSage delivers the best overall performance, while GAT's sensitivity to neighbor information makes it more suitable for evaluating neighbor selection effectiveness. The DQN component for reinforcement learning is implemented with two hidden layers, each of size 128. A fixed client selection ratio of 0.25 is applied throughout the experiments. Model performance is evaluated using accuracy (ACC), macro-F1 score (F1), and macro-recall (Recall). These metrics offer a comprehensive evaluation of both overall and per-class performance in node classification. The reported results for ACC, F1, and Recall are presented as percentages, i.e., all values are multiplied by 100 for clarity and better comparability. For detailed experimental settings, please refer to the Appendix.

4.4 Comparative Experiment Analysis

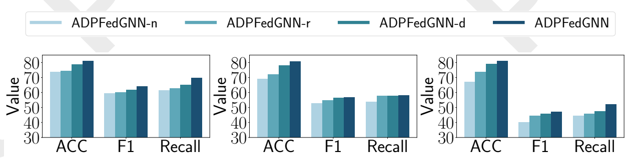
We conduct extensive experiments on the Cora, CiteSeer, and PubMed datasets, partitioning clients into 5 groups using the Louvain community method. The results, shown in Table 1, indicate that ADPFedGNN surpasses the second-best method, FedALA, with average improvements of 5.66 percent, 5.83 percent, and 12.45 percent in ACC, F1, and Recall, respectively, for the GraphSage backbone. Significant improvements are also observed with the GAT backbone.

These experimental results demonstrate the effectiveness of ADPFedGNN. The reinforcement federated adaptive neighbor selection strategy enhances robustness and stability by adaptively selecting relevant neighbors for training global or local knowledge-related parameters, addressing Challenge 1. The federated mask-based parameter decoupling method effectively separates global and local parameters, and the personalized federated masked parameter aggregation ensures the efficient use of global knowledge while minimizing local interference, addressing Challenge 2. Together, these innovations enable ADPFedGNN to consistently outperform baseline methods across diverse datasets and configurations.

4.5 Ablation Study

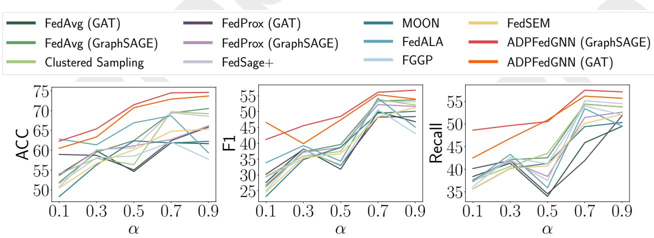
We evaluate the contributions of key components in ADPFedGNN through ablation studies on the Cora dataset with 5 clients, using GraphSage as backbone models. The results, shown in Figure 2, compare the full ADPFedGNN model with three ablated variants: ADPFedGNN-d, which integrates the federated mask-based parameter decoupling and personalized federated masked parameter aggregation methods—designed to function together as the latter depends on the former; ADPFedGNN-r, which includes the reinforcement federated adaptive neighbor selection strategy; and ADPFedGNN-n, a baseline without these components.

The results demonstrate that the full ADPFedGNN model achieves the best performance across all metrics, confirming the necessity of integrating all proposed components. Specifically, ADPFedGNN-r enhances performance by adaptively selecting neighbors for training relevant parameters, while ADPFedGNN-d effectively decouples global and local parameters through adaptive masking and mutual information minimization, ensuring the efficient utilization of global knowledge and the preservation of client-specific characteristics.



(a) n=3 (b) n=5 (c) n=10

Figure 2: Ablation study results of ADPFedGNN.



(a) ACC (b) F1 (c) Recall

Figure 3: The impact of data non-IID characteristics with varying α.

4.6 The Impact of α on Model Performance

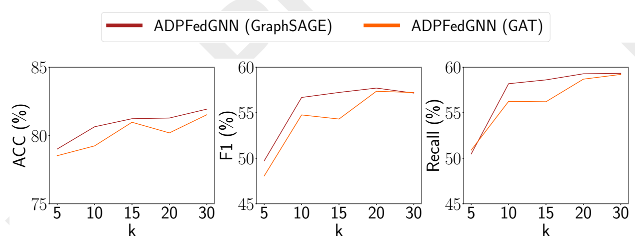
We evaluate the impact of data non-IID characteristics on the Cora dataset with 5 clients using Dirichlet label partitioning, varying the α parameter from 0.1 to 0.9 to simulate different levels of data heterogeneity. As shown in Figure 3, as α increases, the data distribution shifts from highly imbalanced to more balanced, leading to improved performance across most methods. ADPFedGNN consistently achieves superior results under all α settings and demonstrates remarkable effectiveness in scenarios with extreme imbalance at low α values, highlighting its robustness and adaptability to non-IID data.

4.7 Effect of Client Select Rate on Model Effectiveness

We evaluate the impact of client participation rates on the Cora dataset with 5 clients using Louvain community partitioning under select rates of 0.2, 0.5, 0.7, and 0.9. As shown in Table 2, ADPFedGNN consistently outperforms all baseline methods across select rates. With the GAT backbone, ADPFedGNN demonstrates significant improvements at lower select rates, highlighting the effectiveness of its reinforcement adaptive neighbor selection strategy. Similarly, with the GraphSage backbone, it achieves superior accuracy, macro-F1, and recall, demonstrating that the federated mask-based parameter decoupling method ensures effective utilization of global knowledge even with limited client participation. These results confirm the adaptability and robustness of ADPFedGNN across varying participation levels.

4.8 Impact of Neighbor Sampling Size on Model Performance

We evaluate the effect of neighbor sampling size k on model performance using the Cora dataset with 5 clients. As shown in Figure 4, increasing k enhances accuracy, macro-F1, and recall by providing richer contextual information. However, performance improvements slow when k exceeds 10, suggesting that a moderate sampling size is sufficient for effective learning. Larger k also increases memory consumption, posing challenges in resource-constrained environments. Notably, GAT benefits more from larger k values, as the expanded neighbor set allows for better attention mechanism learning, leading to faster performance gains compared to GraphSage. The reinforcement adaptive neighbor selection strategy effectively identifies suitable neighbors for updating global or local knowledge-related parameters while filtering out noisy neighbors.



(a) ACC (b) F1 (c) Recall

Figure 4: Effect of neighbor sampling size k.

4.9 Effect of β on Model Performance

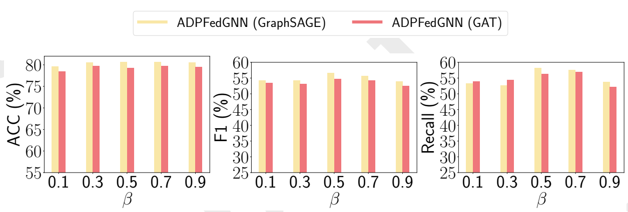
We assess the impact of the hyperparameter β on model performance using the Cora dataset with 5 clients. As shown in Figure 5, performance peaks around $\beta = 0.5$, indicating that a balanced integration of global and local features

Dataset	Cora			CiteSeer			PubMed		
Metrics	ACC	F1	Recall	ACC	F1	Recall	ACC	F1	Recall
FedAvg(GAT)	66.27±2.27	49.77±3.87	50.88±3.90	66.61±2.03	56.94±2.43	57.47±2.05	81.03±0.59	64.17±1.36	60.81±1.91
FedProx(GAT)	65.29±2.26	44.91±2.51	46.56±1.79	63.15±1.65	51.31±1.34	53.16±1.60	76.52±4.63	62.11±5.39	60.08±5.42
FedAvg(GraphSage)	68.97±1.76	52.56±4.90	53.67±5.27	68.48±1.63	57.72±1.38	59.74±1.64	87.49±0.40	82.18±0.73	82.71±1.25
Clustered Sampling(GraphSage)	73.71±0.95	50.13±1.23	52.08±1.26	67.71±3.32	57.71±3.28	60.30±2.89	87.06±2.43	82.01±3.68	82.06±3.03
FedProx(GraphSage)	71.53±1.66	51.68±3.82	53.24±3.46	69.98±1.14	59.23±2.36	59.85±2.54	87.69±0.14	81.80±0.69	81.65±1.06
FedSage+(GraphSage)	71.04±3.72	53.01±4.33	53.97±4.35	69.96±3.38	58.01±3.74	59.03±3.01	87.51±1.88	82.99±2.15	82.23±2.11
MOON(GraphSage)	72.46±1.85	52.64±2.98	52.71±2.61	67.51±2.02	59.19±3.00	60.01±2.29	86.95±2.64	81.17±4.08	81.98±2.93
FedALA(GraphSage)	73.13±2.41	53.96±4.83	55.13±4.52	70.51±1.50	59.88±2.03	61.87±1.90	86.88±0.53	82.36±0.69	68.24±1.02
FGGP(GraphSage)	64.84±2.94	48.96±4.19	49.08±3.31	69.33±1.48	60.40±1.47	60.96±1.58	87.49±1.66	81.02±4.32	81.66±3.63
FedSEM(GraphSage)	72.69±1.92	53.87±4.13	54.81±5.07	67.96±1.70	59.36±2.95	62.06±2.79	87.56±0.46	82.29±0.53	82.46±0.33
ADPFedGNN(GAT)	**79.24±4.26**	**54.76±4.89**	**56.25±5.08**	**70.38±3.24**	**61.04±2.41**	**61.21±2.55**	**86.97±2.13**	**71.65±2.44**	**71.93±2.52**
ADPFedGNN(GraphSage)	**80.64±4.24**	**56.68±4.92**	**58.19±5.01**	**73.54±3.95**	**64.92±3.18**	**64.99±3.09**	**89.83±2.12**	**85.67±2.46**	**86.46±2.51**

Table 1: Performance comparison on three datasets.

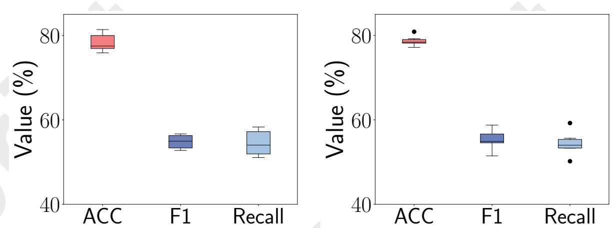
Select Rate	0.5			0.7			0.9		
Metrics	ACC	F1	Recall	ACC	F1	Recall	ACC	F1	Recall
FedAvg(GAT)	66.26±1.83	51.08±2.38	52.77±2.74	59.63±2.81	43.63±3.31	47.47±3.54	60.72±1.36	46.77±4.28	52.66±4.63
FedProx(GAT)	65.21±0.95	46.15±2.55	48.22±2.74	64.31±2.16	47.88±4.44	50.77±3.68	66.20±2.52	52.31±3.42	52.08±3.67
FedAvg(GraphSage)	68.14±2.45	53.33±4.02	53.97±3.46	68.69±1.72	45.87±2.23	49.92±1.04	69.42±1.34	53.65±2.55	53.74±2.66
Clustered Sampling(GraphSage)	74.33±2.20	50.91±2.29	52.76±2.22	75.36±1.18	49.48±4.52	51.43±5.03	74.09±3.09	53.33±3.99	54.09±4.25
FedProx(GraphSage)	72.75±1.95	50.16±3.39	51.19±3.71	71.20±1.67	59.72±2.40	53.83±2.64	71.98±1.70	54.99±3.09	53.89±3.57
FedSage+(GraphSage)	68.82±3.36	53.86±4.99	54.11±5.09	70.53±3.07	48.14±4.71	50.84±4.92	70.83±3.55	54.01±3.62	52.74±3.53
MOON(GraphSage)	73.45±1.16	53.41±3.69	54.46±3.93	72.75±2.51	51.16±2.60	52.45±2.19	72.28±1.23	55.01±3.15	51.52±1.99
FedALA(GraphSage)	73.94±1.54	52.52±2.11	53.49±2.53	71.38±1.15	48.91±2.45	51.42±2.26	72.01±1.63	53.29±5.33	52.83±5.43
FGGP(GraphSage)	65.95±0.72	51.56±2.35	52.51±2.06	66.32±0.66	47.26±3.01	50.61±2.38	68.31±2.63	50.23±3.20	52.15±2.93
FedSEM(GraphSage)	73.71±0.85	54.22±4.79	54.78±4.92	68.95±1.97	53.11±4.88	52.14±4.10	70.52±1.34	54.99±4.59	52.31±4.45
ADPFedGNN(GAT)	**79.72±4.31**	**56.26±4.02**	**56.26±4.19**	**79.19±4.78**	**53.39±4.65**	**54.77±5.82**	**80.83±4.02**	**56.84±4.63**	**54.47±4.46**
ADPFedGNN(GraphSage)	**81.39±4.53**	**58.41±4.17**	**59.31±3.94**	**80.71±4.94**	**56.32±3.65**	**57.04±4.74**	**81.06±3.26**	**60.21±4.59**	**59.53±3.97**

Table 2: Impact of client selection rate on model performance for the Cora dataset.



(a) ACC (b) F1 (c) Recall

Figure 5: Effect of the parameter β.



(a) λ_{mi} (b) λ_{reg}

Figure 6: Sensitivity analysis of λ_{mi} and λ_{reg}.

effectively leverages both knowledge sources. Performance declines when β approaches extremes; an overemphasis on global features ($\beta \approx 0.9$) limits local knowledge utilization, while excessive focus on local features ($\beta \approx 0.1$) hinders federated knowledge sharing. However, within a reasonable range, performance remains stable, demonstrating the robustness of ADPFedGNN across various settings.

4.10 Sensitivity Analysis of λ_{mi} and λ_{reg}

We conduct sensitivity analysis experiments on the Cora dataset with 5 clients to evaluate the impact of the mutual information loss weight λ_{mi} and the regularization loss weight λ_{reg}. The experiments are performed with λ_{mi} values ranging from 0.1 to 0.9 and λ_{reg} values from 0.001 to 0.009. The results indicate that the optimal performance is achieved when $\lambda_{\text{mi}} = 0.3$ and $\lambda_{\text{reg}} = 0.003$. Figure 6 presents the results of the sensitivity analysis. The analysis shows that λ_{mi} results in some fluctuations across the three evaluation metrics; however, the median results remain competitive and within an acceptable range, demonstrating the effectiveness of the model. In contrast, λ_{reg} exhibits relatively smaller fluctuations across the three metrics, indicating that the model remains stable under different values of this parameter.

5 Conclusion

In this paper, we propose Adaptive Decoupling Personalized Federated Graph Neural Networks (ADPFedGNN) to enhance personalized federated graph node classification. ADPFedGNN effectively addresses the challenges of the neighbor selection and parameter coupling in non-IID federated environments. By employing a reinforcement adaptive neighbor selection strategy, it selects suitable node neighbors for training global or local knowledge-related parameters while filtering out irrelevant nodes, thereby enhancing model performance. Additionally, the federated mask-based parameter decoupling method separates global and local parameters, while the personalized federated masked parameter aggregation method enables effective parameter sharing without interfering with local feature learning. Experimental results on public datasets demonstrate that ADPFedGNN surpasses existing methods, achieving average improvements of 5.66 percent, 5.83 percent, and 12.45 percent in ACC, F1, and Recall, respectively.

Acknowledgements

This work was supported by the 8th Young Elite Scientists Sponsorship Program by CAST (2022QNRC001), National Natural Science Foundation of China (62172056, 62192784, U22B2038, U23A20319).

References

[Cheng *et al.*, 2020] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning*, pages 1779–1788. PMLR, 2020.

[Dhillon *et al.*, 2020] Inderjit S Dhillon, Dimitris S Papailiopoulos, and Vivienne Sze. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA*. 2020.

[Fraboni *et al.*, 2021] Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In *International Conference on Machine Learning*, pages 3407–3416. PMLR, 2021.

[Fu *et al.*, 2022] Xingbo Fu, Binchi Zhang, Yushun Dong, Chen Chen, and Jundong Li. Federated graph machine learning: A survey of concepts, techniques, and applications. *ACM SIGKDD Explorations Newsletter*, 24(2):32–47, 2022.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[Hasanzadeh *et al.*, 2020] Arman Hasanzadeh, Ehsan Hajiramezanali, Shahin Boluki, Mingyuan Zhou, Nick Duffield, Krishna Narayanan, and Xiaoning Qian. Bayesian graph neural networks with adaptive connection sampling. In *International conference on machine learning*, pages 4094–4104. PMLR, 2020.

[He *et al.*, 2021a] Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Carl Yang, Han Xie, Lichao Sun, Lifang He, Liangwei Yang, Philip S Yu, Yu Rong, et al. Fedgraphnn: A federated learning system and benchmark for graph neural networks. *arXiv preprint arXiv:2104.07145*, 2021.

[He *et al.*, 2021b] Chaoyang He, Emir Ceyani, Keshav Balasubramanian, Murali Annavaram, and Salman Avestimehr. Spreadgnn: Serverless multi-task federated learning for graph neural networks. *arXiv preprint arXiv:2106.02743*, 2021.

[Huang *et al.*, 2023] Wenke Huang, Guancheng Wan, Mang Ye, and Bo Du. Federated graph semantic and structural learning. pages 3830–3838, 2023.

[Khoshraftar and An, 2024] Shima Khoshraftar and Aijun An. A survey on graph representation learning methods. *ACM Transactions on Intelligent Systems and Technology*, 15(1):1–55, 2024.

[Lai *et al.*, 2020] Kwei-Herng Lai, Daochen Zha, Kaixiong Zhou, and Xia Hu. Policy-gnn: Aggregation optimization for graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 461–471, 2020.

[Li *et al.*, 2020] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[Li *et al.*, 2021a] Anran Li, Lan Zhang, Juntao Tan, Yaxuan Qin, Junhao Wang, and Xiang-Yang Li. Sample-level data selection for federated learning. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021.

[Li *et al.*, 2021b] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021.

[Li *et al.*, 2024a] Kai Li, Xin Yuan, Jingjing Zheng, Wei Ni, Falko Dressler, and Abbas Jamalipour. Leverage variational graph representation for model poisoning on federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[Li *et al.*, 2024b] Xunkai Li, Zhengyu Wu, Wentao Zhang, Henan Sun, Rong-Hua Li, and Guoren Wang. Adafgl: A new paradigm for federated node classification with topology heterogeneity. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 2517–2530, 2024.

[Liu *et al.*, 2020] Ziqi Liu, Zhengwei Wu, Zhiqiang Zhang, Jun Zhou, Shuang Yang, Le Song, and Yuan Qi. Bandit samplers for training graph neural networks. *Advances in Neural Information Processing Systems*, 33:6878–6888, 2020.

[Liu *et al.*, 2024] Rui Liu, Pengwei Xing, Zichao Deng, Anran Li, Cuntai Guan, and Han Yu. Federated graph neural networks: Overview, techniques, and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[Long *et al.*, 2023a] Guodong Long, Ming Xie, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. Multi-center federated learning: clients clustering for better personalization. *World Wide Web*, 26(1):481–500, 2023.

[Long *et al.*, 2023b] Guodong Long, Ming Xie, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. Multi-center federated learning: clients clustering for better personalization. *World Wide Web*, 26(1):481–500, 2023.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[Namata *et al.*, 2012] Galileo Namata, Ben London, Lise Getoor, Bert Huang, and U Edu. Query-driven active surveying for collective classification. In *10th international workshop on mining and learning with graphs*, volume 8, page 1, 2012.

[Pareja *et al.*, 2020] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5363–5370, 2020.

[Peng *et al.*, 2022] Hao Peng, Ruitong Zhang, Shaoning Li, Yuwei Cao, Shirui Pan, and S Yu Philip. Reinforced, incremental and cross-lingual event detection from social messages. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):980–998, 2022.

[Quan *et al.*, 2023] Yuhan Quan, Jingtao Ding, Chen Gao, Lingling Yi, Depeng Jin, and Yong Li. Robust preference-guided denoising for graph based social recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 1097–1108, 2023.

[Sen *et al.*, 2008] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[Sun *et al.*, 2021] Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Yuanxing Ning, Philip S Yu, and Lifang He. Sugar: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism. In *Proceedings of the web conference 2021*, pages 2081–2091, 2021.

[Tang *et al.*, 2021] Zhenheng Tang, Zhikai Hu, Shaohuai Shi, Yiu-ming Cheung, Yilun Jin, Zhenghang Ren, and Xiaowen Chu. Data resampling for federated learning with non-iid labels. In *Proceedings of the International Workshop on Federated and Transfer Learning for Data Sparsity and Confidentiality in Conjunction with IJCAI, Montreal, Canada*, pages 21–22. FTLIJCAI, 2021.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Wan *et al.*, 2024] Guancheng Wan, Wenke Huang, and Mang Ye. Federated graph learning under domain shift with generalizable prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15429–15437, 2024.

[Wang *et al.*, 2020] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM 2020-IEEE conference on computer communications*, pages 1698–1707. IEEE, 2020.

[Wang *et al.*, 2021] Yu Wang, Zhiwei Liu, Ziwei Fan, Lichao Sun, and Philip S Yu. Dskreg: Differentiable sampling on knowledge graph for recommendation with relational gnn. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3513–3517, 2021.

[Yang *et al.*, 2020] Min Yang, Chengming Li, Fei Sun, Zhou Zhao, Ying Shen, and Chenglin Wu. Be relevant, non-redundant, and timely: Deep reinforcement learning for real-time event summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9410–9417, 2020.

[Yu *et al.*, 2022] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1294–1303, 2022.

[Zhang *et al.*, 2021a] Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. Subgraph federated learning with missing neighbor generation. *Advances in Neural Information Processing Systems*, 34:6671–6682, 2021.

[Zhang *et al.*, 2021b] Qingru Zhang, David Wipf, Quan Gan, and Le Song. A biased graph neural network sampler with near-optimal regret. *Advances in Neural Information Processing Systems*, 34:8833–8844, 2021.

[Zhang *et al.*, 2023] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedala: Adaptive local aggregation for personalized federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11237–11244, 2023.

[Zhang *et al.*, 2024a] Qianru Zhang, Lianghao Xia, Xuheng Cai, Siu-Ming Yiu, Chao Huang, and Christian S Jensen. Graph augmentation for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 557–569. IEEE, 2024.

[Zhang *et al.*, 2024b] Yu Zhang, Hua Lu, Ning Liu, Yonghui Xu, Qingzhong Li, and Lizhen Cui. Personalized federated learning for cross-city traffic prediction. In *33rd International Joint Conference on Artificial Intelligence, IJCAI*, pages 5526–5534, 2024.

[Zhao *et al.*, 2023] Weichen Zhao, Tiande Guo, Xiaoxi Yu, and Congying Han. A learnable sampling method for scalable graph neural networks. *Neural Networks*, 162:412–424, 2023.