Large Language Models Enhanced Personalized Graph Neural Architecture Search in Federated Learning

Hui Fang[1], Yang Gao[1], Peng Zhang[2], Jiangchao Yao[3], Hongyang Chen[4], Haishuai Wang[1]*

[1]Zhejiang Key Laboratory of Accessible Perception and Intelligent Systems,
College of Computer Science, Zhejiang University, China
[2]Cyberspace Institute of Advanced Technology, Guangzhou University, China
[3]Cooperative Medianet Innovation Center, Shanghai Jiaotong University, China
[4]Research Center for Data Hub and Security, Zhejiang Lab, China
{huif, gaoyang957, haishuai.wang}@zju.edu.cn, p.zhang@gzhu.edu.cn, Sunarker@sjtu.edu.cn, dr.h.chen@ieee.org

Abstract

Personalized federated learning (PFL) on graphs is an emerging field focusing on the collaborative development of architectures across multiple clients, each with distinct graph data distributions while adhering to strict privacy standards. This area often requires extensive expert intervention in model design, which is a significant limitation. Recent advancements have aimed to automate the search for graph neural network architectures, incorporating large language models (LLMs) for their advanced reasoning and self-reflection capabilities. However, two technical challenges persist. First, although LLMs are effective in natural language processing, their ability to meet the complex demands of graph neural architecture search (GNAS) is still being explored. Second, while LLMs can guide the architecture search process, they do not directly solve the issue of client drift due to heterogeneous data distributions. To address these challenges, we introduce a novel method, **P**ersonalized **F**ederated **G**raph **N**eural **A**rchitecture **S**earch (PFGNAS). This approach employs a task-specific prompt to identify and integrate optimal GNN architectures continuously. To counteract client drift, PFGNAS utilizes a weight-sharing strategy of supernet, which optimizes the local architectures while ensuring client-specific personalization. Extensive evaluations show that PFGNAS significantly outperforms traditional PFL methods, highlighting the advantages of integrating LLMs into personalized federated learning environments.

Code — https://github.com/HuiFang-hub/PFGNAS.

Introduction

Graph federated learning (GFL) has emerged as a prominent distributed learning paradigm, gaining widespread adoption across various fields. Especially in some critical applications such as healthcare (Gao et al. 2024; Chen et al. 2024), e-commerce (Tian et al. 2024; Fang et al. 2024), social media (Miao et al. 2024; Li et al. 2021b).

Conventional GFL models assume independent and identically distributed graph data, training a unified GNN architecture. However, real-world graph data often exhibit heterogeneous distributions (Liu et al. 2023), leading to slower convergence and reduced accuracy (Long et al. 2023; Fu et al. 2024). In contrast, personalized GFL methods learn client-specific GNN architectures within the FL framework. Recent approaches fall into two categories: global and individual personalization. The global strategy trains a single GFL model (Liang et al. 2020; Zhang et al. 2020), later adapted locally by each client. The individual strategy (Huang et al. 2021; Li et al. 2021a) develops distinct GNNs for each client, recognizing unique data distributions across clients. Although personalized graph federated learning models have succeeded, they typically present significant complexity and diversity. This complexity may cause considerable effort from human experts to discover state-of-the-art neural network architectures. Additionally, due to privacy concerns, the inability to access authentic data distributions further emphasizes the need for models that can autonomously adapt to underlying data distributions.

Graph neural architecture search (GNAS) methods have been recognized for their ability to automatically design GNNs, as highlighted in studies like (Gao et al. 2022, 2023). However, these methods remain underexplored in personalized graph federated learning, primarily due to the high communication costs associated with bilevel optimization and the insufficient utilization of client-specific data for personalization. Integrating pre-trained Large Language Models (LLMs) with extensive knowledge presents a promising solution to these challenges. Recent works like GENIUS (Zheng et al. 2023) and GPT4GNAS (Wang et al. 2023b) have leveraged LLMs to explore potential architectures within a defined search space through tailored prompts. However, incorporating LLMs into personalized federated GNAS introduces two key challenges.

First, guiding LLMs to explore graph architectures effectively is complex. Unlike traditional architecture search, which involves simpler operations, graph federated learning requires irregular message aggregation, complicating the GNN search process, especially given the diverse characteristics of training data across different graphs. While LLMs excel in NLP tasks, their ability to navigate the complexities of graph architecture search remains under investigation, making it essential to develop robust methods for guiding LLMs in this novel context.

*Corresponding author: Haishuai Wang

Second, while LLMs can assist in guiding architecture search, they cannot directly address client drift caused by data heterogeneity. In federated environments, data heterogeneity leads to client drift, distorting overall model performance and resulting in suboptimal solutions. Traditional architecture search methods follow a fixed procedure, which may struggle with the structural complexities and diversity of graph data, potentially trapping the search in local optima. Even when LLMs generate various client model recommendations, transforming these outputs into personalized models while addressing heterogeneous training challenges remains a critical issue.

To tackle the challenges previously delineated, we introduce an innovative methodology named Personalized Federated Graph Neural Architecture Search (PFGNAS). This method leverages LLMs to optimize the exploration of personalized graph neural architectures within a privacy protection environment. First, to make full use of the capability of LLMs to explore graph neural network architectures, we infuse domain-specific knowledge concerning graph datasets and pertinent GNAS tasks into our framework. The knowledge integration streamlines the architecture search process, making it more efficient. Second, to tackle client drift, we employ supernet pruning and weight sharing. We deploy a supernet on client devices to mitigate training collapse caused by model heterogeneity. Sharing relevant supernet parameters accelerates the evaluation of different GNN models, reducing training complexity and computational demands. Extensive evaluations show that PFGNAS significantly outperforms current benchmarks, highlighting the effectiveness of LLMs in graph architecture search.

Related Works

Personalized Federated Learning

In federated learning with heterogeneous data, personalized federated learning (PFL) aims to enhance each client's model through collaborative training, categorized into global and individual personalization (Tan et al. 2022). Global PFL trains a unified model that is later personalized for each client via local adaptation, as seen in LG-FedAvg (Liang et al. 2020), which mixes network layers to use local encoders with a global classifier, and Per-FedAvg (Fallah, Mokhtari, and Ozdaglar 2020), which uses meta-learning for fast personalization. FedFomo (Zhang et al. 2020) optimizes the combination of uploaded models, while pFedMe (T Dinh, Tran, and Nguyen 2020) regularizes the L2 distance between local and global models.

In contrast, individual personalization trains separate FL models for each client. Astraea (Duan et al. 2020) and FedHome (Wu et al. 2020) use data augmentation for balanced training, though this may threaten privacy. Ditto (Li et al. 2021a) improves fairness and robustness by training local and global models simultaneously, while SFL (Chen et al. 2022), FedPUB (Baek et al. 2023), and pFedGraph (Ye et al. 2023) use structural information as a regularization term. However, these models rely heavily on expert knowledge and are limited by the constraints of a single model. Consequently, we propose a personalized federated model designed to autonomously select an appropriate architecture, mitigating the need for extensive prior knowledge and enhancing adaptability across diverse data distributions.

Graph Neural Architecture Search

Graph Neural Architecture Search (GNAS) has been extensively studied to automate the development and deployment of Graph Neural Networks (GNNs). This exploration can be broadly categorized into three groups: reinforcement learning (RL) based architecture controllers like GraphNAS (Gao et al. 2021), differentiable methods like DARTS (Liu, Simonyan, and Yang 2018) and GDAS (Dong and Yang 2019), and evolutionary algorithms (Wang, Zhang, and Zhu 2022). However, these methods overlook data privacy. Some efforts, like FedNAS (He, Annavaram, and Avestimehr 2020) and DFNAS (Garg, Saha, and Dutta 2020), combine federated learning with differentiable NAS methods by incorporating federated training for a global model. FLAGNNS (Wang et al. 2023a) introduces an evolutionary optimization strategy for federated settings, but it relies on hyperparameters like population size, crossover probability, and mutation probability. Inspired by the reasoning capabilities and adaptive features of LLMs, GENIUS (Zheng et al. 2023) explores LLMs for CNN architecture search, while GPT4GNAS (Wang et al. 2023b) uses GPT-4 to guide graph neural architecture generation. However, these approaches are not well-suited for federated learning scenarios involving distributed graph datasets.

In this work, we design the PFGNAS algorithm, which uses an LLM as a controller to create a high-performance GNN architecture within a federated scenario. PFGNAS concentrates on comprehending and utilizing contextual information, facilitating individual architecture searches, and global personalized federated model optimization.

Problem Formulation

In this section, we first introduce the optimization of conventional and personalized federated learning and then define our problem of personalized federated graph neural architecture search.

LLMs for GNAS. Given a pre-trained LLM, a graph dataset G, a GNN search space Ω and evaluation metric $\mathcal{M}$, we aims to find the best architecture Λ^*:

$$\Lambda^* = \underset{\Lambda \in \mathbf{LLM}(\Omega))}{\operatorname{argmax}} \mathcal{M}(\Lambda(G)), \tag{1}$$

where $\mathbf{LLM}(\Omega)$ denotes searching the architecture in search space by an LLM.

Federated Learning. In conventional federated learning, a set of subgraphs $\{G_i\}_{i \in N}$ that are distributed and non-shared across N clients, the goal is to find the optimal global parameter w,

$$\min_{w \in \mathbb{R}^d} F(w) := \sum_{n=1}^{N} \frac{|G_n|}{|G|} f_n(w), \tag{2}$$

where $F(\cdot)$ is a function that aggregates the local objectives, it is typically set to be a weighted average of local losses,

e.g., FedAvg (McMahan et al. 2017), and the function $f_i(\cdot)$ denotes the expected loss over the data distribution of client i, $w^* = \arg\min \mathbb{E}_{(G_i,\mathbf{Y}_i)\sim\xi_i}[\mathcal{L}(w;(G_i,\mathbf{Y}_i))]$, where each client may generate data G_i via a distinct distribution ξ_i, and $\mathcal{L}(w;G_i,\mathbf{Y}_i)$ is a loss function with groudtruth label $\mathbf{Y}_i$.

Personalized Federated Learning. Personalized FL aims to perform well on N clients. In general, the existing works explore personalization through a regularization term, $\Lambda^*(w) = \min_{\{\mathbf{O},\boldsymbol{w}_1,\cdots,\boldsymbol{w}_N\}} \sum_{n=1}^{N} \frac{|G_n|}{|G|} f_n(w) + \mathcal{R}(\mathbf{O},\boldsymbol{w}_1,\cdots,\boldsymbol{w}_N)$, where O is introduced to relate clients, and $\mathcal{R}$ is a specific regularizer which is imposed to prevent w_n from over-fitting client n's limited data. However, FED-ROD (Chen and Chao 2021) has observed that robust personalized models can emerge from the local training of generic Federated Learning (FL) algorithms. Therefore, in this paper, instead of designing $\mathcal{R}$, we focus on providing distinct initial models for each client to maintain the level of personalization. Especially, a PFL *architecture* is defined as the architecture which consists of N local *models*.

Personalized Federated Graph Neural Architecture Search. Personalized federated GNAS is a bilevel optimization problem with model architecture Λ as the upper-level variable, while the global model weights w associated with the architecture as the lower-level variable. Given a GNN search space Ω, the Federated GNAS problem aims to find the optimal GNN architecture $\Lambda^* \in \Omega$ that maximizes the federated validation performance $\mathcal{P}$,

$$\begin{aligned} \Lambda^*(w^*) :&= \underset{\Lambda\in S}{\operatorname{argmax}}\, \mathcal{P}(\Lambda'(w^*),\Lambda(w^*)), \\ s.t.\ w^* &= \sum_{n=1}^{N} \frac{|G_n|}{|G|} E_{(G_i,y_i)\sim\xi_i}[\mathcal{L}(\theta_i;(G_i,\mathbf{Y}_i))], \end{aligned} \tag{3}$$

where w^* and θ_i^* denote the optimal parameters of global and local model i respectively, and Λ' refer to the historical optimal architecture.

Methodology

In this section, we introduce the PFGNAS algorithm, which optimizes personalized GNN architectures using language prompts and addresses client drift and model heterogeneity through a supernet strategy. The PFGNAS framework is illustrated in Figure 1. We design prompts that combine the search task, search space, strategy, and historical performance, allowing the LLM to explore new architectures and assign specific models to clients. To mitigate training collapse from model heterogeneity, we deploy a supernet structure on client devices, using pruning techniques to retain core modules for local training. A weight-sharing strategy accelerates the evaluation of GNN models, enabling clients to update supernet weights with the received architecture information and optimize local model weights.

Graph Architecture Search Optimization

As discussed before, to comprehensively address the task of graph architecture search and ensure the adaptability of the generated instruction prompts for each federated scenario, we design a performance-driven optimization strategy. First, we provide an introduction to the PFGNAS prompts.

Search Task. Our objective is to construct a personalized architecture for highly heterogeneous data. Specifically, we leverage LLMs to rapidly search for the optimal initial GNN architectures from the search space for each client. To tailor the LLM to different personalized federated scenarios, the search task prompt, denoted as $\mathcal{K}$, primarily focuses on three parameters: the number of clients (N), the domain knowledge (π) and specif dataset (D). To bridge the gap between NLP and graph domains, we have selected several domain-specific knowledge (π) of graph datasets (such as average shortest path length, network density, etc.) to describe the graph. For example, the shortest path length can assist in designing the layers of GNNs. Although we set the number of layers to 4, LLMs can choose layers such as Identity or ZeroConv layers to reduce actual model complexity and avoid over-smoothing of GNN models. When the network density is too high, it can guide the LLM in selecting GNN models with sampling techniques, such as GraphSage. Mathematically, this is expressed as $\mathcal{K}(N, D, \pi)$.

Search Space. The search space Ω in GNAS encompasses a variety of candidate operations and interconnections between these operations. Specifically, the architecture of personalized federated GNNs for each client is structured into three distinct blocks.

- Input Block. It projects node features into a unified embedding space and typically consists of a single layer, such as a fully connected layer.
- Middle Block. Aimed at extracting high-level node embeddings, it integrates multiple GNN operations. The repertoire of candidate operations includes nine widely utilized methods: GCN, GraphSage, GPR, GAT, GIN, SGC, ARMA, and APPNP, alongside specialized layers such as the fully-connected layer, ZeroConv layer (which outputs an all-zero tensor during forward propagation), and the Identity layer (which outputs itself).
- Output Block. It transforms the feature embeddings derived from the middle block into the final predictions.

Furthermore, the network incorporates non-linear activation functions to enable learning from complex data patterns and relationships. Each block offers five candidate activation functions: Sigmoid, Tanh, ReLU, Linear, and ELU, enhancing the model's flexibility and capability to capture non-linear dynamics.

Search strategy. The search strategy $\psi(T)$ for optimizing GNN architectures via GNAS consists of two distinct phases. During the *Exploration Stage* ($T < 2$), the LLM broadly explores the entire search space, randomly sampling operations and configurations to avoid premature convergence on locally optimal solutions. As the process transitions to the *Exploitation Stage* ($T \geq 2$), the strategy shifts to a more targeted exploration. Here, the LLM refines search parameters, focusing on promising combinations based on empirical data from earlier trials. This phase employs a self-reflective optimization approach, where the LLM iteratively queries new operation lists informed by historical performance, moving from random selection to a more structured

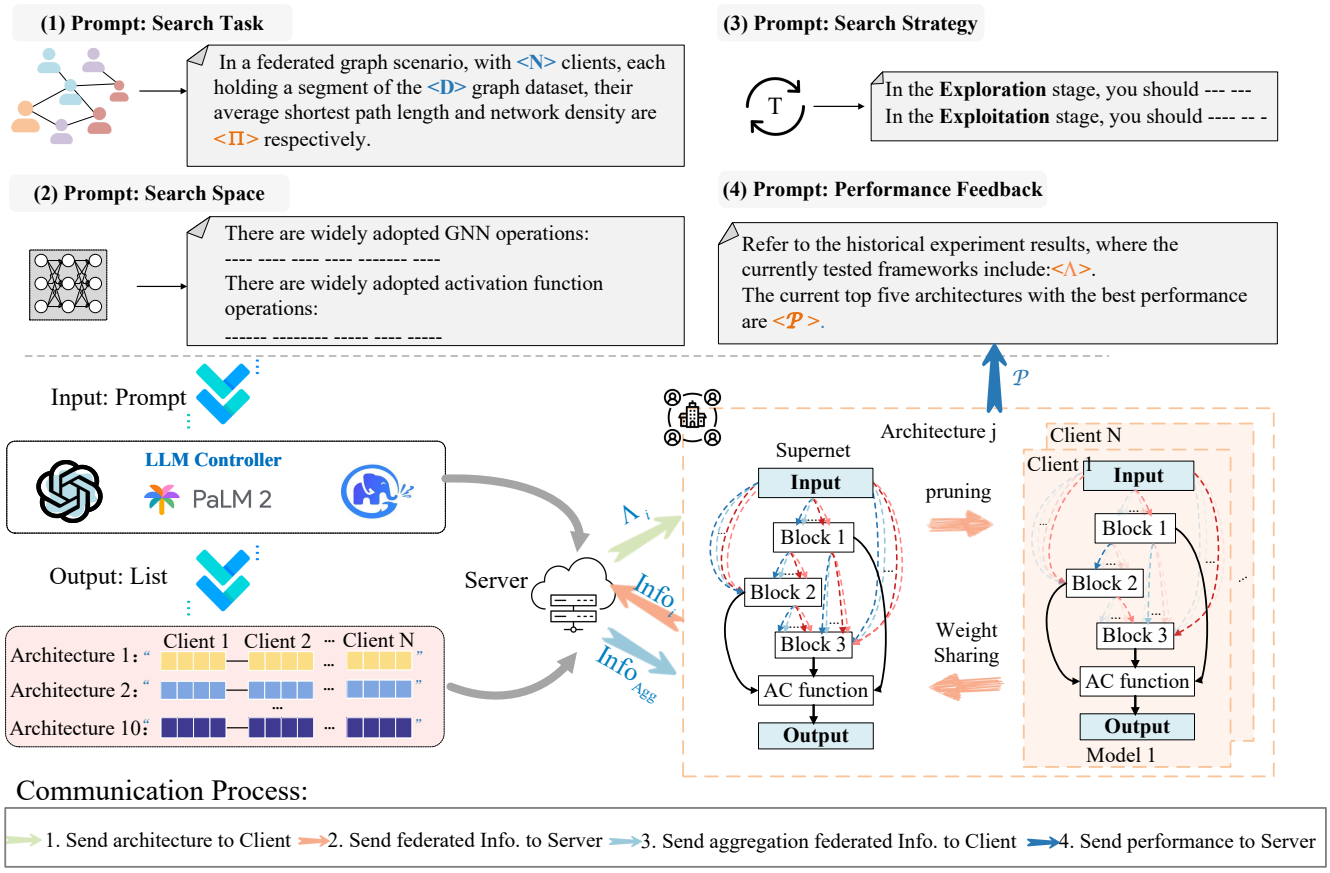


Figure 1: The overall framework of PFGNAS, <> and <> are task-specifc configurations and calculated input statistics. The framework optimal the parameters of GNAS and supernet through four communication processes. (1) These generated architectures are sent to clients. (2) Each client acquires a personalized model by pruning the supernet and begins to train the model in a federated way. Therefore, they send the Info_i (e.g., gradient and weight) to the server. (3) The server aggregates client information and sends the updated Info_{Agg} to each client. (4) After several epochs, the architecture reaches the convergence state and sends the performance of the final test to the server.

and effective exploration.

Historical performance. LLMs should consider the historical valid performance $\mathcal{P}$ of all clients to swiftly search for the optimal architecture. We aim to recommend a global GNN architecture that performs well on multiple subgraphs, so our optimization goal is to maximize the local and global federated validation performance :

$$\begin{aligned} \mathcal{P}(\Lambda(w^*)) &= \sum_{f_m} f_m\left(\Lambda(w^*); D\right), \\ \mathcal{P}(\Lambda_i(w_i^*)) &= \sum_{f_m} f_m(\Lambda_i(w_i^*); D_i), \end{aligned} \tag{4}$$

where $\mathcal{P}(\Lambda(w^*))$ and $\mathcal{P}(\Lambda_i(w_i^*))$ denote the local and global federated validation performance respectively, D and D_i denote the valid dataset of the sever and client i respectively. Here, we provide metrics f_m for validation, namely the accuracy and Micro-average Area Under the Curve (AUC).

In summary, PFGNAS utilizes those components to enable the self-reflection ability of LLM, which accelerates the efficiency of graph architectures' search for the initial local model. Mathematically, we define the best GNN architecture of clients at T-th iteration as:

$$\begin{aligned} \Lambda_T^* &= LLM\left(\mathcal{K}(N, D, \pi); \Omega; \psi(T); \mathcal{P}\right), \\ P_T &= \sum_{i=0}^{N} P_{T-1}(\Lambda_i(w_i^*)) + P_{T-1}(\Lambda(w^*)) \end{aligned} \tag{5}$$

where $P_{T-1}(\Lambda_i(w_i^*))$ denotes historical performance of local model, and $P_{T-1}(\Lambda(w^*))$ denote historical performance of global model. The optimal global architecture, denoted as Λ_T^*, is determined by a language model (LLM) via specific prompts and is dynamically updated based on performance metrics $\mathcal{P}$.

Personalized Supernet Optimization Strategy

In this section, we employ a GNN supernet optimization strategy to address client drift caused by data heterogeneity. This strategy enables all clients to train the supernet in FL scenarios collaboratively.

Here, we optimize the architecture Λ and the parameters w of super-network in an interleaved manner. First, the server undertakes a random exploration of a set of personalized federated architectures and recommends different combinations of different operations. For each architecture $\Lambda = \{\Lambda_i\}_{i=0}^{N}$, the server splits it into N models and distributes them to the corresponding clients. Subsequently, to facilitate the acquisition of a personalized initial architecture Λ_i while simultaneously preserving the extensibility of the supernet S, we implement a pruning method on supernet S. The pruning strategy is achieved by utilizing the parameter α, which represents the weights of the pruned supernet S. In this configuration, the weights of selected operators are assigned a value of 1, while those of unselected operators are set to infinitesimally small values. Consequently, the initial parameters of a GNN model Λ_i for any client can be expressed mathematically as:

$$\Lambda_i(\theta_i) = S(w) \odot \alpha, \tag{6}$$

where $\odot$ denotes the element-wise multiplication of the supernet weights with the pruning mask α. In particular, to reduce the complexity of the model, in our supernet, we save the parameters of all operations in architectures and collect them in θ_i, which is also known as the local model weight. This strategy ensures that each client starts with a model that is not only tailored to its specific needs but is also scalable within the broader architectural framework of S.

Next, we aim to update the model's parameters by minimizing the federated loss of the initial model Λ_i at round T. Let $\Lambda_i(\theta_i)$ denotes the model Λ_i of client i parametrized by θ_i. The optimal θ_i^* can be calculated as:

$$\theta_i^* \leftarrow \min_{\theta_i} \mathcal{L}(\Lambda_i(\theta_i; G_i); \mathbf{Y}_i), \tag{7}$$

where $\mathcal{L}$ denotes the cross-entropy loss function between prediction labels $\Lambda_i(\theta_i; G_i)$ and true labels $\mathbf{Y}_i$.

Then, the object is to learn the global parameter w according to the θ_i from all clients. Specifically, let $S(w)$ denote the supernet S parametrized by w. Suppose we apply FedAvg to jointly guide clients in training the supernet under FL scenarios. According to the Eq. (2), the optimal global weight w^* can be denoted as:

$$w^* := \sum_{i=1}^{N} \frac{|G_i|}{|G|} \theta_i^*. \tag{8}$$

Applying chain rule to the approximate local loss function $\mathcal{L}(\Lambda_i(\theta_i; G_i); \mathbf{Y}_i)$ the gradient of θ_i yields:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta_i} &= \frac{\partial \mathcal{L}}{\partial w} \cdot \frac{\partial w}{\partial \theta_i}, \\ &= \frac{\partial \mathcal{L}(\Lambda_i(\theta_i; G_i); \mathbf{Y}_i)}{\partial \theta_i} \cdot \frac{|G_i|}{|G|}, \\ &= \frac{\partial \mathcal{L}(\Lambda_i(S(w) \odot \alpha; G_i); \mathbf{Y}_i)}{\partial \theta_i} \cdot \frac{|G_i|}{|G|}, \end{aligned} \tag{9}$$

where $|G_i|$ denotes the number of nodes of graph G_i.

Then, we perform several local optimization steps on the personal data. According to the gradient descent algorithm, the gradient of the local model is

$$\theta_i \leftarrow \theta_i - \eta \left(\nabla_{\theta_i} \mathcal{L}((S(w) \odot \alpha; G_i); \mathbf{Y}_i) \cdot \frac{|G_i|}{|G|} \right), \tag{10}$$

where η denotes the learning rate. Therefore, the gradient of global personalized supernet $\Lambda_i(w)$ can be updated as:

$$w^* \leftarrow w - \eta \left(\nabla_w \mathcal{L}(S(w; G); y) \right). \tag{11}$$

After several epochs, the server sends the current optimal global parameters w^* to all clients for iterative updates. Until the model approaches convergence, we compute the final performance (i.e., Accuracy and ROC-AUC) of the current PFL architecture.

$$\mathcal{P}(\Lambda_i(w^*)) = \{ACC_{\Lambda_i(w^*)}, ROC_{\Lambda_i(w^*)}\}. \tag{12}$$

In particular, PFGNAS is compatible with currently prevalent federated learning methods, such as FedAvg (McMahan et al. 2017), FedSage+ (Zhang et al. 2021). In the architecture optimization stage, to generate new architectures with improved performance, the server provides the performance feedback $\mathcal{P}$ to the LLM by incorporating the performance of both historical and current federated architectures as part of the prompt.

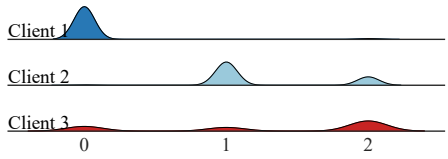
Experiments

In this section, we examine the performance of PFGNAS. First, we compare the optimal federated GNN architecture discovered by PFGNAS with the state-of-the-art baseline. Second, we compare the performance in different heterogeneous partitions to valid PFGNAS.

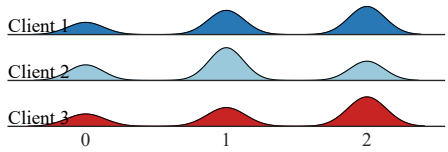
Experiment Setttings

Datasets. To validate the efficacy of our proposed methodology, we conducted simulations of federated learning scenarios utilizing three widely-recognized datasets, namely, *Cora*, *Citeseer*, and *Pubmed*. The partitioning of each dataset into N clients employs two distinct partitioning strategies (Yurochkin et al. 2019). First, We implemented a homogeneous partitioning scheme, ensuring that each client possesses an approximately equal distribution across the K classes, achieved through Dirichlet distribution sampling with $p_k \sim \text{Dir}N$ (β=10). In contrast, a heterogeneous partitioning approach was employed by simulating $p_k \sim \text{Dir}N$ (β=0.2) and allocating a proportion $p_{k,N}$ of class k instances to N clients. The depicted difference in data distribution is illustrated in Figure 2, where a lower β value corresponds to a more pronounced imbalance in the distribution.

Baselines & Metrics. We compare PFGNAS with the 8 traditional GNN methods (i.e., *GCN* (Kipf and Welling 2016), *GraphSage* (Hamilton, Ying, and Leskovec 2017), *GAT* (Velickovic et al. 2018), *GIN* (Xu et al. 2018), *SGC* (Wu et al. 2019), *ARIMA* (Yang et al. 2023) and *APPNP* (Bojchevski et al. 2020)), and take two the outstanding GNAS methods, which including *Darts* (Liu, Simonyan, and Yang 2018), *GraphNAS* (Gao et al. 2021). These methods are compared within the FedAvg federated framework. Moreover, we compare PFGNAS with *FLAGNNs* (Wang



(a) Pubmed (N=3,β=0.2)



(b) Pubmed (N=3,β=10)

Figure 2: Unbalanced sample distribution (non-I.I.D.) for Pubmed and datasets. The horizontal axis represents labels, and the vertical axis represents their corresponding numbers. A lower β value corresponds to a more pronounced imbalance in the distribution.

et al. 2023a), which is a federated GNAS baseline. Additionally, we compare with the state-of-the-art PFL method (i.e., *FedPUB* (Baek et al. 2023)). To further validate the effectiveness of PFGNAS, we designed several variants of PFGNAS and comparative experiments to understand LLM's decision-making. We constructed three PFGNAS variants to conduct ablation experiments. First, to assess the necessity of personalized model recommendations for each client, we designed the *PFGNAS-O* method, which recommends a uniform client model using LLMs. Second, to examine the logic behind PFGNAS's model recommendations, we created *PFGNAS-R*, which randomly assigns a model to each client in every round. Lastly, we explored how PFGNAS selects superior models from the population, focusing on whether it leverages LLMs' reasoning abilities. For this, we developed an evolution-based method, *PFGNAS-E*, which selects the top-5 historically outstanding architectures for crossover in each iteration. We choose accuracy and AUC to measure the performance of all methods.

Implementation Details. In our federated learning setting, we set the number of clients N to be $[3, 5, 10, 20]$ respectively, and the total round number to be 100. In particular, we use 'GLM4' as the default LLM, and we also compared it to 'GPT' with a temperature $\tau = 0.5$. Besides, we choose the Adam optimizer with a learning rate of 1e-3. The number of GNN layers to 2. We run all experiments for three random repetitions on an NVIDIA RTX 3090 GPU.

Performance under uniform data distribution

Table 1 shows the results of accuracy (i.e., ACC) of our methods and baselines on two datasets with two different federated settings. While our proposed model didn't consistently achieve the highest performance across all metrics, it outperformed other methods in most aspects. Specifically, the results demonstrated a significant accuracy improvement ranging from 0.48% to 115.41% on the Citeseer dataset. Besides, GNAS-based federated methods performed comparably to well-designed federated models like FLAGNNS and

Model	Cora (3)	Cora (5)	Citeseer (3)	Citeseer(5)
GCN	78.01±0.42	76.90±0.85	78.96±0.54	78.53±1.20
GraphSage	75.70±0.42	77.66±1.19	77.56±0.95	77.88±0.27
GAT	76.22±0.91	76.13±1.51	79.92±0.80	78.63±1.24
GIN	68.97±1.28	66.84±0.87	59.94±2.76	64.75±2.97
SGC	76.30±0.53	76.05±1.39	80.20±0.40	78.10±0.31
ARMA	53.88±8.81	64.11±9.96	78.53±1.05	77.67±1.06
APPNP	74.08±2.52	70.25±5.61	80.31±0.75	77.67±2.12
FL+Darts	44.44±18.88	61.73±1.74	50.00±15.71	37.04±9.44
FL+Graphnas	27.16±1.74	37.04±4.00	51.11±1.81	56.29±4.19
FLAGNNS	76.19±1.63	**82.20±1.27**	74.29±1.54	77.15±2.10
FedPUB	82.32±0.62	80.65±0.08	73.17±0.33	73.71±0.59
PFGNAS	**83.80±2.22**	80.86±1.73	**80.70±0.95**	**79.79±1.63**

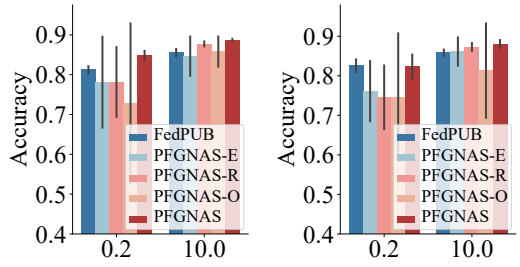
Table 1: Accuracy of node classification task on two datasets with N=3 and 5. The **bold** font highlights the best performance, and the underlined font indicates a suboptimal result.

FedPUB. Especially, our PFGNAS outperformed FedPUB on all metrics. In conclusion, the PFGNAS model exhibits superior performance across multiple datasets and metrics, outperforming traditional GNN models and several federated learning methods.

Performance under different data distribution

Figure 3 reveals the ACC performance of the global model of PFGNAS and its variant models on two different data distributions. From the results, we can observe that (1) data distribution heterogeneity reduces the model's generalization ability, resulting in decreased performance. Specifically, the overall performance under a distribution of β=0.2 is significantly lower than under β=10. This is due to the increased variability in data characteristics across clients, making it difficult to obtain a unified model that performs well for all, thus affecting convergence and outcomes. (2) The PFGNAS-O model performs the worst because a single model is less adaptable to clients with heterogeneous data, reducing generalization and overall performance. (3) PFGNAS outperforms both distributions' top personalized federated learning method, FedPUB. This is because our method leverages LLMs to provide a more flexible architecture that better adapts to varying data distributions. And (4) PFGNAS also surpasses the random and evolutionary variants, as it effectively uses historical performance feedback and domain knowledge from graph data to more accurately select and update architectures, further boosting performance.

Additionally, unlike traditional GNAS, the LLM simultaneously drives the design of multiple models and fully leverages historical performance for updates. Table 2 compares client model performance across different distributions, using GCN with FedAvg as the baseline on the PubMed dataset. The results show that FL+GraphNAS performed the worst, even falling below the baseline, with Client 1's test accuracy at 0. This suggests that FL+GraphNAS struggles to handle data distribution bias effectively, as evidenced by Client 1's data distribution in Figure 2(a). In contrast, our method demonstrated the most significant improvement,



(a) N=3, $\beta \in \{0.2, 10\}$ (b) N=5, $\beta \in \{0.2, 10\}$

Figure 3: The global accuracy in different heterogeneous Pubmed data distribution. The black line located in the center of the bar chart signifies the error range.

with an average accuracy 23.75% higher than the baseline. The results also highlight the advantage of multiple initial models in personalized federated learning. PFGNAS and PFGNAS-E adapted better to heterogeneous data distributions than a single model like PFGNAS-O. In summary, PFGNAS searched for three distinct initial models for each client, and after supernet optimization, the final accuracy reached 86.13%, this is a 19.42% improvement over PFGNAS-O and a 6.71% increase over PFGNAS-E.

Model	Client 1	Client 2	Client 3	Global	Avg ↑
FL+GCN	0.67	0.72	0.70	0.75	-
FedPUB	0.84	0.83	0.72	0.81	12.88%
FL+GraphNAS	0.00	0.48	0.67	0.77	-33.74%
PFGNAS-O	**1.00**	0.85	0.80	0.73	19.73%
PFGNAS-E	**1.00**	0.87	0.81	0.81	23.45%
PFGNAS	**1.00**	0.80	**0.84**	**0.86**	**23.75%**

Table 2: The accuracy of the PubMed dataset (N=3, β=0.2) on different clients, where Avg ↑ indicates the average improvement in performance compared to the baseline across all devices. The **bold** font highlights the best performance.

	Model	Accuracy	AUC
Citeseer	PFGNAS-GPT4	80.39±2.05	**79.46±0.17**
	PFGNAS-PALM	79.33±0.73	78.36±0.49
	PFGNAS-GLM4	**80.70±0.95**	79.32±0.20
Pubmed	PFGNAS-GPT4	79.73±4.18	93.79±0.70
	PFGNAS-PALM	85.42±3.62	94.75±0.82
	PFGNAS-GLM4	**86.52±0.79**	**95.20±0.84**

Table 3: The performance of PFGAS with different LLMs on two datasets (N=3). The **bold** font highlights the best performance, and the underlined font indicates a suboptimal result.

Case study of different LLMs

We evaluate the performance of different LLM backends: 'GPT4', 'GLM-4', and 'PALM'. Table 3 compares their accuracy, AUC, and the best architecture identified on the Citeseer and PubMed datasets. While PFGNAS-PALM generally performed well, it did not outperform PFGNAS-GLM4 in any metric. PFGNAS-GLM4 consistently achieved the highest accuracy and AUC, making it the top-performing model. Although PFGNAS-GPT4 excelled in AUC on the Citeseer dataset, its lower accuracy placed it behind PFGNAS-GLM4 overall. According to CriticBench (Lan et al. 2024), GLM-4 may outperform GPT-4 on tasks like chat and question answer, which could explain its superior performance here. These findings underscore the importance of selecting the appropriate LLM backend based on the specific metrics prioritized in different applications.

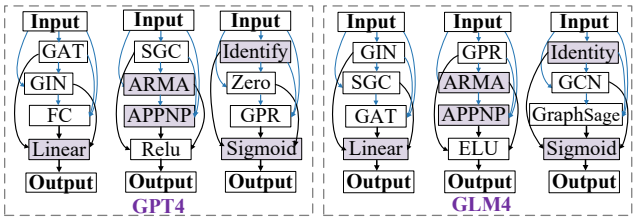


Figure 4: The comparison of the optimal architecture between GPT4 and GLM4 on Citeseer(N=3) dataset. Operators with the same insertion position and structure are highlighted with a purple background.

Additionally, we visualized the best architectures identified by GPT-4 and GLM-4 on the Citeseer dataset, as shown in Figure 4. The results reveal that similar substructures frequently appear in the optimal architectures identified by different LLMs. Notably, the optimal architectures for Client 3 consistently feature the Identity layer and the Sigmoid activation function. This consistency suggests that these structures may be crucial to the optimal architectures, reflecting specific domain characteristics of the data. The 'Identity' layer, which returns its input unchanged, might indicate that nodes in Client 3's graph dataset are closely related to their neighbors, allowing shallow GNNs to capture sufficient structural information and node features for strong predictive performance. These findings underscore the interpretability of our framework and offer valuable insights for further optimizing network architectures.

Conclusion

In this paper, we present Personalized Federated Graph Neural Architecture Search (PFGNAS), a novel approach for personalized federated graph learning. PFGNAS uses LLMs as controllers to design high-performance GNNs collaboratively. We introduce prompt learning to optimize federated graph architecture search using historical performance and strategies. A weight-sharing supernet ensures consistency across personalized models for different clients. Extensive experiments on three datasets show that PFGNAS outperforms baseline methods. In future work, we will emphasize the interpretability of LLMs in the context of federated architecture search and leveraging gradient information to enhance their search capabilities.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 62202422), the National Key R&D Program of China (Grant No. 2022ZD0160703), the National Natural Science Foundation of China (Grant Nos. 62406279, U2336202 and 62376064), and Shanghai Artificial Intelligence Laboratory.

References

Baek, J.; Jeong, W.; Jin, J.; Yoon, J.; and Hwang, S. J. 2023. Personalized subgraph federated learning. In *International Conference on Machine Learning*, 1396–1415. PMLR.

Bojchevski, A.; Gasteiger, J.; Perozzi, B.; Kapoor, A.; Blais, M.; Rózemberczki, B.; Lukasik, M.; and Günnemann, S. 2020. Scaling graph neural networks with approximate pagerank. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2464–2473.

Chen, F.; Long, G.; Wu, Z.; Zhou, T.; and Jiang, J. 2022. Personalized Federated Learning With a Graph. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence.

Chen, H.-Y.; and Chao, W.-L. 2021. On Bridging Generic and Personalized Federated Learning for Image Classification. In *International Conference on Learning Representations*.

Chen, W.; Yang, J.; Sun, Z.; Zhang, X.; Tao, G.; Ding, Y.; Gu, J.; Bu, J.; and Wang, H. 2024. DeepASD: a deep adversarial-regularized graph learning method for ASD diagnosis with multimodal data. *Translational Psychiatry*, 14(1): 375.

Dong, X.; and Yang, Y. 2019. Searching for a robust neural architecture in four gpu hours. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1761–1770.

Duan, M.; Liu, D.; Chen, X.; Liu, R.; Tan, Y.; and Liang, L. 2020. Self-balancing federated learning with global imbalanced data in mobile systems. *IEEE Transactions on Parallel and Distributed Systems*, 32(1): 59–71.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*.

Fang, H.; Xiao, Z.; Zheng, P.; Chen, H.; Li, Z.; Bu, J.; and Wang, H. 2024. Learning Co-occurrence Patterns for Next Destination Recommendation. *IEEE Transactions on Mobile Computing*, 23(6): 7225–7237.

Fu, W.; Wang, H.; Gao, C.; Liu, G.; Li, Y.; and Jiang, T. 2024. Privacy-preserving individual-level covid-19 infection prediction via federated graph learning. *ACM Transactions on Information Systems*, 42(3): 1–29.

Gao, Y.; Yang, H.; Zhang, P.; Zhou, C.; and Hu, Y. 2021. Graph neural architecture search. In *International joint conference on artificial intelligence*. International Joint Conference on Artificial Intelligence.

Gao, Y.; Zhang, P.; Yang, H.; Zhou, C.; Tian, Z.; Hu, Y.; Li, Z.; and Zhou, J. 2022. Graphnas++: Distributed architecture search for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*.

Gao, Y.; Zhang, P.; Zhou, C.; Yang, H.; Li, Z.; Hu, Y.; and Philip, S. Y. 2023. HGNAS++: efficient architecture search for heterogeneous graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(9): 9448–9461.

Gao, Y.; Zhang, X.; Sun, Z.; Chandak, P.; Bu, J.; and Wang, H. 2024. Precision Adverse Drug Reactions Prediction with Heterogeneous Graph Neural Network. *Advanced Science*, 2404671.

Garg, A.; Saha, A. K.; and Dutta, D. 2020. Direct federated neural architecture search. *arXiv preprint arXiv:2010.06223*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

He, C.; Annavaram, M.; and Avestimehr, S. 2020. Towards non-iid and invisible data with fednas: federated deep learning via neural architecture search. *arXiv preprint arXiv:2004.08546*.

Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; and Zhang, Y. 2021. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 7865–7873.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Lan, T.; Zhang, W.; Xu, C.; Huang, H.; Lin, D.; Chen, K.; and Mao, X.-l. 2024. CriticBench: Evaluating Large Language Models as Critic. *arXiv preprint arXiv:2402.13764*.

Li, T.; Hu, S.; Beirami, A.; and Smith, V. 2021a. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, 6357–6368. PMLR.

Li, Z.; Wang, H.; Zhang, P.; Hui, P.; Huang, J.; Liao, J.; Zhang, J.; and Bu, J. 2021b. Live-streaming fraud detection: A heterogeneous graph neural network approach. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 3670–3678.

Liang, P. P.; Liu, T.; Ziyin, L.; Allen, N. B.; Auerbach, R. P.; Brent, D.; Salakhutdinov, R.; and Morency, L.-P. 2020. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*.

Liu, H.; Simonyan, K.; and Yang, Y. 2018. DARTS: Differentiable Architecture Search. In *International Conference on Learning Representations*.

Liu, Q.; Wu, J.; Huang, Z.; Wang, H.; Ning, Y.; Chen, M.; Chen, E.; Yi, J.; and Zhou, B. 2023. Federated User Modeling from Hierarchical Information. *ACM Transactions on Information Systems*, 41(2): 1–33.

Long, J.; Chen, T.; Nguyen, Q. V. H.; and Yin, H. 2023. Decentralized collaborative learning framework for next POI recommendation. *ACM Transactions on Information Systems*, 41(3): 1–25.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Miao, H.; Zhong, X.; Liu, J.; Zhao, Y.; Zhao, X.; Qian, W.; Zheng, K.; and Jensen, C. S. 2024. Task Assignment With Efficient Federated Preference Learning in Spatial Crowdsourcing. *TKDE*, 36(4): 1800–1814.

T Dinh, C.; Tran, N.; and Nguyen, J. 2020. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33: 21394–21405.

Tan, A. Z.; Yu, H.; Cui, L.; and Yang, Q. 2022. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Tian, C.; Xie, Y.; Chen, X.; Li, Y.; and Zhao, X. 2024. Privacy-Preserving Cross-Domain Recommendation with Federated Graph Learning. *ACM Transactions on Information Systems*, 42(5): 1–29.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Wang, C.; Chen, B.; Li, G.; and Wang, H. 2023a. Automated Graph Neural Network Search Under Federated Learning Framework. *IEEE Transactions on Knowledge and Data Engineering*.

Wang, H.; Gao, Y.; Zheng, X.; Zhang, P.; Chen, H.; and Bu, J. 2023b. Graph neural architecture search with gpt-4. *arXiv preprint arXiv:2310.01436*.

Wang, X.; Zhang, Z.; and Zhu, W. 2022. Automated graph machine learning: Approaches, libraries and directions. *arXiv preprint arXiv:2201.01288*.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, 6861–6871. PMLR.

Wu, Q.; Chen, X.; Zhou, Z.; and Zhang, J. 2020. Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring. *IEEE Transactions on Mobile Computing*, 21(8): 2818–2832.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Yang, J.; Sun, J.; Ren, Y.; Li, S.; Ding, S.; and Hu, J. 2023. GACP: graph neural networks with ARMA filters and a parallel CNN for hyperspectral image classification. *International Journal of Digital Earth*, 16(1): 1770–1800.

Ye, R.; Ni, Z.; Wu, F.; Chen, S.; and Wang, Y. 2023. Personalized federated learning with inferred collaboration graphs. In *International Conference on Machine Learning*, 39801–39817. PMLR.

Yurochkin, M.; Agarwal, M.; Ghosh, S.; Greenewald, K.; Hoang, N.; and Khazaeni, Y. 2019. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, 7252–7261. PMLR.

Zhang, K.; Yang, C.; Li, X.; Sun, L.; and Yiu, S. M. 2021. Subgraph federated learning with missing neighbor generation. *Advances in Neural Information Processing Systems*, 34: 6671–6682.

Zhang, M.; Sapra, K.; Fidler, S.; Yeung, S.; and Alvarez, J. M. 2020. Personalized Federated Learning with First Order Model Optimization. In *International Conference on Learning Representations*.

Zheng, M.; Su, X.; You, S.; Wang, F.; Qian, C.; Xu, C.; and Albanie, S. 2023. Can GPT-4 Perform Neural Architecture Search? *arXiv preprint arXiv:2304.10970*.