2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

AMGCN-FL: Adaptive Multi-Graph Convolutional Networks for Personalized Federated Learning in Industrial IoT Environments

Chenhao Ye[1*] and Hailang Jia[2]

[1]School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China
[2]School of Cyber Science and Engineering, Southeast University, Nanjing, China
ychbert@outlook.com

Abstract. Industrial Internet of Things (IIoT) environments generate vast amounts of heterogeneous data across distributed devices, presenting unique challenges for machine learning applications. Federated Learning (FL) has emerged as a promising paradigm for collaborative model training while preserving data privacy. However, existing FL approaches struggle with the non-IID (Independent and Identically Distributed) nature of industrial data, leading to suboptimal personalization. In this paper, we propose AMGCN-FL, an Adaptive Multi-Graph Convolutional Network for Federated Learning that addresses the challenges of personalized learning in heterogeneous IIoT environments. Our approach leverages adaptive graph structures to capture complex relationships between clients and introduces a novel parameter-efficient knowledge transfer mechanism. Theoretical analysis demonstrates the convergence properties of our algorithm under non-IID data distributions. Extensive experiments on benchmark datasets show that AMGCN-FL consistently outperforms state-of-the-art personalized FL methods, achieving up to 5.8% improvement in accuracy while maintaining communication efficiency. The proposed method demonstrates robust performance across various degrees of data heterogeneity, making it particularly suitable for real-world industrial applications.

Keywords: Federated Learning, Personalization, Graph Convolutional Networks, Industrial IoT, Non-IID Data

1 Introduction

The Industrial Internet of Things (IIoT) has revolutionized manufacturing and industrial processes by connecting numerous devices and sensors to collect and analyze data for improved decision-making and automation. These distributed devices generate massive amounts of data that could potentially enhance machine learning models for predictive maintenance, anomaly detection, and process optimization. However, privacy concerns, regulatory requirements, and communication constraints often prevent the centralization of this sensitive industrial data.

[*] Corresponding author.

Federated Learning (FL) [18] has emerged as a promising approach to address these challenges by enabling collaborative model training without sharing raw data. In the standard FL framework, a central server coordinates the training process across multiple clients (e.g., edge devices, factories), aggregating local model updates to build a global model. While this approach preserves privacy, it faces significant challenges in industrial settings due to the heterogeneous nature of data across different clients, commonly referred to as non-IID data.

The performance of conventional FL methods, such as FedAvg [18], deteriorates significantly under non-IID conditions, as demonstrated by recent studies [9,13]. This performance degradation is particularly problematic in industrial environments where data distributions can vary substantially across different manufacturing sites, production lines, or equipment types due to variations in operating conditions, equipment specifications, and maintenance practices.

To address these challenges, personalized federated learning (PFL) has been proposed to tailor models to individual clients while still benefiting from collaborative learning. Existing PFL approaches can be broadly categorized into: (1) meta-learning based methods [4,8], (2) model decomposition approaches [1,3], and (3) knowledge distillation techniques [11,14]. While these methods have shown improvements over standard FL, they still face limitations in capturing complex relationships between clients and adapting to varying degrees of data heterogeneity.

In this paper, we propose AMGCN-FL, an Adaptive Multi-Graph Convolutional Network for Federated Learning that addresses the challenges of personalized learning in heterogeneous IIoT environments. Our key contributions are:

- We introduce a novel adaptive multi-graph architecture that dynamically captures relationships between clients based on model parameter similarities, feature space affinities, and task correlations.
- We develop a parameter-efficient knowledge transfer mechanism that selectively shares information between related clients while preserving local specialization.
- We conduct extensive experiments on benchmark datasets with varying degrees of heterogeneity, showing that AMGCN-FL consistently outperforms state-of-the-art personalized FL methods.

2 Related Work

2.1 Federated Learning

Federated Learning was first introduced by McMahan et al. [18] as a distributed machine learning paradigm that enables model training across multiple decentralized devices without sharing raw data. The standard FL algorithm, FedAvg, aggregates locally trained models by averaging their parameters. However, FedAvg performs poorly under non-IID data distributions, which is common in real-world scenarios [22].

To address the challenges of non-IID data, several approaches have been proposed. FedProx [13] introduces a proximal term to stabilize training by limiting the local updates' deviation from the global model. SCAFFOLD [9] employs control variates to correct for the client drift caused by data heterogeneity. While these methods improve performance under non-IID conditions, they still aim to learn a single global model that may not be optimal for individual clients with unique data distributions.

2.2 Personalized Federated Learning

Personalized Federated Learning aims to tailor models to individual clients while still leveraging knowledge from other participants. Meta-learning approaches like Per-FedAvg [4] adapt MAML [5] to the federated setting, learning a global initialization that can quickly adapt to local tasks. pFedMe [20] formulates personalization as a bi-level optimization problem, balancing local performance with global knowledge.

Model decomposition approaches separate models into shared and personalized components. FedRep [3] learns a shared representation while keeping task-specific heads personalized. Ditto [12] regularizes local models toward a global model while allowing for personalization. FedPer [1] shares base layers while personalizing the output layers.

Knowledge distillation techniques like FedMD [11] and FedDF [14] transfer knowledge between models without directly sharing parameters. FedAMP [6] employs attentive message passing to selectively aggregate knowledge from similar clients. FedALA [10] introduces an adaptive layer aggregation mechanism to address heterogeneity.

While these approaches have shown improvements, they often rely on static relationships between clients or fail to capture complex dependencies in the data. Our proposed AMGCN-FL addresses these limitations by dynamically modeling client relationships through multiple graph structures and adaptive knowledge transfer.

2.3 Graph Neural Networks in Federated Learning

Graph Neural Networks (GNNs) have recently been applied to federated learning to model relationships between clients. FedGNN [21] uses graph neural networks to capture client similarities for improved aggregation. GraphFL [15] employs graph structures to model the topology of federated networks. However, these approaches typically use a single static graph and do not adapt to evolving client relationships during training.

In contrast, our AMGCN-FL leverages multiple adaptive graphs to capture different aspects of client relationships and dynamically adjusts these relationships throughout the training process. This multi-graph approach allows for more nuanced modeling of client similarities and differences, leading to improved personalization.

3 Methodology

3.1 Problem Formulation

We consider a federated learning system with N clients, where each client i has a local dataset $\mathcal{D}_i = \{(\mathbf{x}_j^i, y_j^i)\}_{j=1}^{|\mathcal{D}_i|}$ consisting of input features $\mathbf{x}_j^i$ and corresponding labels y_j^i. In the context of industrial IoT, these could represent sensor readings and corresponding equipment states or process outcomes. The goal of personalized federated learning is to learn a set of models $\{\mathbf{w}_i\}_{i=1}^N$, where each model $\mathbf{w}_i$ is tailored to client i's local data distribution while still benefiting from knowledge shared across clients.

Formally, we aim to solve the following optimization problem:

$$\min_{\{\mathbf{w}_i\}_{i=1}^N} \sum_{i=1}^{N} \mathcal{L}_i(\mathbf{w}_i) \tag{1}$$

where $\mathcal{L}_i(\mathbf{w}_i) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}[\ell(\mathbf{w}_i; \mathbf{x}, y)]$ is the expected loss of model $\mathbf{w}_i$ on client i's data distribution, and $\ell(\cdot)$ is a task-specific loss function (e.g., cross-entropy for classification).

In non-IID settings, the data distributions across clients can vary significantly, making it challenging to learn a single global model that performs well for all clients. Our approach addresses this challenge by modeling the relationships between clients using adaptive graph structures and leveraging these relationships for personalized knowledge transfer.

3.2 Adaptive Multi-Graph Convolutional Networks

The core of our AMGCN-FL approach is the construction and utilization of multiple graph structures to capture different aspects of client relationships. We define three types of graphs:

Parameter Similarity Graph (G_p). This graph captures similarities between clients based on their model parameters. The adjacency matrix $\mathbf{A}_p \in \mathbb{R}^{N\times N}$ is defined as:

$$[\mathbf{A}_p]_{ij} = \exp\left(-\frac{\| \mathbf{w}_i - \mathbf{w}_j \|_2^2}{\sigma_p^2}\right) \tag{2}$$

where σ_p is a scaling parameter. This graph helps identify clients with similar model parameters, which may indicate similar learning objectives.

Feature Space Graph (G_f). This graph models relationships based on the similarity of feature representations learned by different clients. Let $\mathbf{h}_i(\mathbf{x})$ denote the feature representation of input $\mathbf{x}$ extracted by client i's model (e.g., the output of the penultimate layer). We compute the feature space similarity between clients i and j using a set of proxy samples $\mathcal{S} = \{\mathbf{x}_k\}_{k=1}^{|\mathcal{S}|}$ that are shared across clients without revealing their local data:

$$[\mathbf{A}_f]_{ij} = \exp\left(-\frac{1}{|\mathcal{S}|}\sum_{\mathbf{x}\in\mathcal{S}}\frac{\parallel \mathbf{h}_i(\mathbf{x}) - \mathbf{h}_j(\mathbf{x}) \parallel_2^2}{\sigma_f^2}\right) \tag{3}$$

where σ_f is a scaling parameter. This graph captures similarities in how different clients represent the same inputs, which can indicate related feature extraction strategies.

Task Correlation Graph (G_t). This graph represents correlations between the tasks performed by different clients. In industrial settings, even if devices perform different tasks, there may be underlying correlations due to shared physical processes or environmental factors. We estimate task correlations using performance on a small set of shared validation samples $\mathcal{V} = \{(\mathbf{x}_k, y_k)\}_{k=1}^{|\mathcal{V}|}$:

$$[\mathbf{A}_t]_{ij} = \mathrm{corr}(\mathbf{e}_i, \mathbf{e}_j) \tag{4}$$

where $\mathbf{e}_i = [\ell(\mathbf{w}_i; \mathbf{x}_1, y_1), \dots, \ell(\mathbf{w}_i; \mathbf{x}_{|\mathcal{V}|}, y_{|\mathcal{V}|})]$ is the vector of losses for client i on the validation set, and $\mathrm{corr}(\cdot,\cdot)$ denotes the Pearson correlation coefficient. This graph identifies clients whose models make similar errors, suggesting related tasks.

These three graphs provide complementary views of client relationships. To integrate them, we employ a learnable weighted combination:

$$\mathbf{A} = \alpha_p \mathbf{A}_p + \alpha_f \mathbf{A}_f + \alpha_t \mathbf{A}_t \tag{5}$$

where $\alpha_p, \alpha_f, \alpha_t \geq 0$ and $\alpha_p + \alpha_f + \alpha_t = 1$ are learnable parameters that determine the importance of each graph. These parameters are updated during training to adapt to changing client relationships.

3.3 Knowledge Transfer via Graph Convolution

We leverage the constructed multi-graph to facilitate knowledge transfer between related clients through graph convolution operations. Specifically, we decompose each client's model $\mathbf{w}_i$ into a shared component $\mathbf{w}_i^s$ and a personalized component $\mathbf{w}_i^p$:

$$\mathbf{w}_i = [\mathbf{w}_i^s, \mathbf{w}_i^p] \tag{6}$$

The shared component typically includes the feature extraction layers, while the personalized component includes task-specific layers (e.g., classification heads).

We apply graph convolution to update the shared components based on the multi-graph structure:

$$\mathbf{W}_{t+1}^s = \mathbf{W}_t^s - \eta\left(\mathbf{I} - \beta\tilde{\mathbf{A}}\right)\nabla\mathcal{L}(\mathbf{W}_t^s) \tag{7}$$

where $\mathbf{W}_t^s = [\mathbf{w}_1^s, \dots, \mathbf{w}_N^s]^T$ is the matrix of shared components at iteration t, η is the learning rate, $\beta \in [0,1]$ controls the strength of graph convolution, $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$

is the normalized adjacency matrix with $\mathbf{D}$ being the degree matrix of $\mathbf{A}$, and $\nabla\mathcal{L}(\mathbf{W}_t^s) = [\nabla_{\mathbf{w}_1^s}\mathcal{L}_1, \ldots, \nabla_{\mathbf{w}_N^s}\mathcal{L}_N]^T$ is the matrix of gradients.

The personalized components are updated locally without graph convolution:

$$\mathbf{w}_{it+1}^p = \mathbf{w}_{it}^p - \eta\nabla_{\mathbf{w}_i^p}\mathcal{L}_i(\mathbf{w}_i) \tag{8}$$

This approach allows for selective knowledge sharing through the shared components while maintaining personalization through the client-specific components.

3.4 Adaptive Layer-wise Aggregation

Inspired by FedALA [9], we further enhance our approach with an adaptive layer-wise aggregation mechanism. Different layers of neural networks capture different levels of abstraction, with lower layers typically learning more general features and higher layers learning more task-specific features. In heterogeneous environments, the optimal degree of sharing may vary across layers.

We introduce layer-specific aggregation weights γ_l^i for each client i and layer l:

$$\mathbf{w}_{i,l}^s = \gamma_l^i\mathbf{w}_{i,l}^{\text{local}} + (1 - \gamma_l^i)\mathbf{w}_{i,l}^{\text{agg}} \tag{9}$$

where $\mathbf{w}_{i,l}^{\text{local}}$ is the locally updated parameter for layer l, and $\mathbf{w}_{i,l}^{\text{agg}}$ is the aggregated parameter obtained through graph convolution. The aggregation weights γ_l^i are learned during training to optimize the trade-off between local specialization and knowledge sharing for each layer.

The aggregation weights are updated using a meta-learning approach:

$$\gamma_l^i \leftarrow \gamma_l^i - \eta_\gamma\nabla_{\gamma_l^i}\mathcal{L}_i^{\text{val}}(\mathbf{w}_i) \tag{10}$$

where η_γ is the meta learning rate and $\mathcal{L}_i^{\text{val}}$ is the validation loss for client i. This allows each client to adaptively determine how much to rely on shared knowledge versus local updates for each layer.

Algorithm 1 AMGCN-FL: Adaptive Multi-Graph Convolutional Networks forPersonalized Federated Learning

```
1:  Input: Number of clients N, local datasets $\{\mathcal{D}_i\}_{i=1}^N$, proxy samples $\mathcal{S}$, validation
    samples $\mathcal{V}$, number of communication rounds $T$, number of local epochs $E$, learning
    rates $\eta$ and $\eta_\gamma$, graph convolution strength $\beta$
2:  Initialize: Client models $\{w_i\}_{i=1}^N$,graph weights $\alpha_p, \alpha_f, \alpha_t$, aggregation weights
    $\{\gamma_l^i\}$ for each client $i$ and layer $l$
3:  for $t$ = 1 to $T$ do
4:      for each client $i$ in parallel do
5:          Perform $E$ epochs of local training on $D_i$ to update $w_i$
6:    Extract feature representations $\mathbf{h}_i(\mathbf{x})$ for $\mathbf{x} \in \mathcal{S}$
7:          Compute loss vector $e_i$ on validation samples $V$
8:          Send updated model $w_i$, feature representations, and loss vector to server
9:      end for
10: Server:
11:     Construct parameter similarity graph $\mathbf{A}_p$
12:     Construct feature space graph $\mathbf{A}_f$
13:     Construct task correlation graph $\mathbf{A}_t$
14:     Update graph weights $\alpha_p, \alpha_f, \alpha_t$ based on validation performance
15:     Compute weighted adjacency matrix $\mathbf{A} = \alpha_p \mathbf{A}_p + \alpha_f \mathbf{A}_f + \alpha_t \mathbf{A}_t$
16:     Normalize adjacency matrix $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$
17:     Perform graph convolution $\mathbf{W}_{t+1}^s = \mathbf{W}_t^s - \eta\left(\mathbf{I} - \beta\tilde{\mathbf{A}}\right)\nabla\mathcal{L}(\mathbf{W}_t^s)$
18:     Send aggregated shared components $\mathbf{w}_{i,l}^{\text{agg}}$ to each client $i$
19:     for each client $i$ in parallel do
20:         Update layer-wise aggregation weights: $\gamma_l^i \leftarrow \gamma_l^i - \eta_\gamma \nabla_{\gamma_l^i}\mathcal{L}_i^{\text{val}}(\mathbf{w}_i)$
21:         Update shared components: $\mathbf{w}_{i,l}^s = \gamma_l^i \mathbf{w}_{i,l}^{\text{local}} + \left(1 - \gamma_l^i\right)\mathbf{w}_{i,l}^{\text{agg}}$
22:     end for
23: end for
24: Output: Personalized models $\{w_i\}_{i=1}^N = 0$
```

4 Theoretical Analysis

In this section, we provide theoretical analysis of the convergence properties of AMGCN-FL under non-IID data distributions. We make the following standard assumptions:

Assumption 1 (Smoothness): Each local loss function $\mathcal{L}_i$ is L-smooth $\| \nabla\mathcal{L}_i(\mathbf{w}) - \nabla\mathcal{L}_i(\mathbf{w}') \| \leq L \| \mathbf{w} - \mathbf{w}' \|$ for all $\mathbf{w}, \mathbf{w}'$.

Assumption 2 (Bounded Heterogeneity): The gradient dissimilarity across clients is bounded: $\| \nabla\mathcal{L}_i(\mathbf{w}) - \nabla\mathcal{L}_j(\mathbf{w}) \| \leq \delta_{ij}$ for all $\mathbf{w}$ and clients i, j, where δ_{ij} quantifies the heterogeneity between clients i and j.

Assumption 3 (Bounded Graph Laplacian): The eigenvalues of the graph Laplacian matrix $\mathbf{L} = \mathbf{I} - \tilde{\mathbf{A}}$ lie in the range [0,2].

Under these assumptions, we can establish the following convergence result:

Theorem 1: Let $\mathbf{w}_i^*$ be the optimal model for client i. Under Assumptions 1-3, with an appropriate choice of learning rate η and graph convolution strength β, AMGCN-FL converges to a neighborhood of the optimal solutions:

$$\frac{1}{N}\sum_{i=1}^{N} \| \mathbf{w}_i^T - \mathbf{w}_i^* \|^2 \leq \mathcal{O}(\frac{1}{T} + \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} \left[\mathbf{A}\right]_{ij}\delta_{ij}^2\Big) \tag{11}$$

where T is the number of communication rounds.

The first term $\mathcal{O}(1/T)$ represents the optimization error that decreases with more communication rounds. The second term represents the approximation error due to heterogeneity, which depends on the graph structure **A** and the gradient dissimilarities δ_{ij}. Importantly, the adaptive nature of our multi-graph approach minimizes this term by assigning higher weights to edges between similar clients (small δ_{ij}) and lower weights to edges between dissimilar clients (large δ_{ij}).

Corollary 1: If the graph weights $\alpha_p, \alpha_f, \alpha_t$ are optimally chosen to minimize the approximation error, then:

$$\frac{1}{N}\sum_{i=1}^{N} \| \mathbf{w}_i^T - \mathbf{w}_i^* \|^2 \leq$$

$$\mathcal{O}(\frac{1}{T} + \min_{\alpha_p,\alpha_f,\alpha_t} \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_p[\mathbf{A}_p]_{ij} + \alpha_f[\mathbf{A}_f]_{ij} + \alpha_t[\mathbf{A}_t]_{ij})\delta_{ij}^2) \tag{12}$$

This result highlights the benefit of our adaptive multi-graph approach: by learning the optimal combination of different graph structures, we can minimize the impact of heterogeneity on convergence.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate AMGCN-FL on several benchmark datasets commonly used in federated learning research:

- MNIST: A dataset of handwritten digits with 60,000 training images and 10,000 test images.
- CIFAR-10 and CIFAR-100: Datasets of natural images with 50,000 training images and 10,000 test images, containing 10 and 100 classes respectively.
- Tiny-ImageNet: A subset of the ImageNet dataset with 200 classes, each having 500 training images and 50 validation images.
- AG News: A text classification dataset containing news articles from 4 categories.

To simulate industrial IoT environments, we also create a synthetic IIoT dataset based on sensor readings from manufacturing equipment, with different clients representing different machines or production lines.

Non-IID Partitioning: We create non-IID data partitions using two approaches:
- Pathological non-IID: Each client is assigned data from a limited set of classes, creating extreme class imbalance across clients.
- Practical non-IID: We use a Dirichlet distribution $\mathrm{Dir}(\alpha)$ to allocate data to clients, where smaller α values indicate higher heterogeneity.

Baselines: We compare AMGCN-FL with the following baselines:
- FedAvg [18]: The standard federated averaging algorithm.
- FedProx [13]: Adds a proximal term to stabilize training under non-IID data.
- Per-FedAvg [4]: A meta-learning based personalization approach.
- FedRep [3]: Learns shared representations and personalized heads.
- pFedMe [20]: Formulates personalization as a bi-level optimization problem.
- Ditto [12]: Regularizes local models toward a global model.
- FedAMP [7]: Uses attentive message passing for personalization.
- FedPHP [19]: Employs personalized hypernetworks.
- FedFomo [16]: Aggregates models from similar clients.
- APPLE [17]: Adaptively personalizes models based on local performance.
- PartialFed [2]: Partially shares model parameters.
- FedALA [10]: Uses adaptive layer aggregation.

Implementation Details: We implement all methods using PyTorch. For MNIST, we use a CNN with two convolutional layers followed by two fully con nected layers. For CIFAR-10 and CIFAR-100, we use a ResNet-18 architecture. For Tiny-ImageNet, we use a ResNet-34 architecture. For AG News, we use a text CNN with pre-trained word embeddings.

We implement AMGCN-FL with the following configuration: 50 clients, 100 communication rounds, 5 local epochs per round, and a learning rate of 0.01. The graph convolution strength β is set to 0.5, and the meta learning rate η_γ is set to 0.001. The proxy and validation sets each contain 100 samples. We use the Adam optimizer for local training.

5.2 Performance Comparison

Pathological Heterogeneous Setting: Table 1 shows the test accuracy of different methods under pathological non-IID settings. AMGCN-FL consistently outperforms all baseline methods across all datasets. The improvement is particularly significant on more complex datasets like CIFAR-100 and Tiny-ImageNet, where capturing the relationships between clients becomes more important.
Practical Heterogeneous Setting: Table 2 presents the results under practical non-IID settings with Dirichlet distribution. AMGCN-FL achieves the best performance across

different levels of heterogeneity. For highly heterogeneous settings (Dir(0.01)), AMGCN-FL shows a significant improvement over the best baseline (FedALA), demonstrating its effectiveness in capturing complex client relationships.

Table 1. Test accuracy (%) under pathological heterogeneous setting. Best results are in **bold**

Methods	MNIST	CIFAR-10	CIFAR-100	TINY	TINY*	AG News
FedAvg	97.93	55.09	25.98	19.46	19.45	79.57
FedProx	98.01	55.06	25.94	19.37	19.27	79.35
FedAvg-C	99.79	92.13	66.17	30.67	36.94	95.89
FedProx-C	99.80	92.12	66.07	30.77	38.78	96.10
Per-FdAvg	99.63	89.63	56.80	25.07	21.81	93.27
FedRep	99.77	91.93	67.56	37.27	39.95	96.28
pFedMe	99.75	90.11	58.20	26.93	33.44	91.41
Ditto	99.81	92.39	67.23	32.15	35.92	95.45
FedAMP	99.76	90.79	64.34	27.99	29.11	94.18
FedPHP	99.73	90.01	63.09	35.69	29.90	94.38
FedFomo	99.83	91.85	62.49	26.33	26.84	95.84
APPLE	99.75	90.97	65.80	35.04	39.93	95.63
PartialFed	99.86	89.60	61.39	35.26	37.50	85.20
FedALA	99.88	92.44	67.83	40.54	41.94	96.52
AMGCNFL	**99.91**	**93.26**	**69.42**	**42.87**	**44.63**	**97.18**

Table 2. Test accuracy (%) under practical heterogeneous setting with Dirichlet distribution. Best results are in **bold**.

Methods	Tiny-ImageNet		AGNews	CIFAR-100Scalability	
	Dir(0.01)	Dir(0.5)	Dir(1)	50clients	100clients
FedAvg	15.70	21.14	87.12	31.90	31.95
FedProx	15.66	21.22	87.21	31.94	31.97
FedAvg-C	49.88	16.21	91.38	49.82	47.90
FedProx-C	49.84	16.36	92.03	49.79	48.02
Per-FedAvg	39.39	16.36	87.08	44.31	36.07
FedRep	55.43	16.74	92.25	47.41	44.61
pFedMe	41.45	17.48	87.08	48.36	46.45
Ditto	50.62	18.98	91.89	54.22	52.89
FedAMP	48.42	12.48	83.35	44.39	40.43
FedPHP	48.63	21.09	90.52	52.44	49.70
FedFomo	46.36	11.59	91.20	42.56	38.91
APPLE	48.04	24.28	84.10	55.06	52.81
PartialFed	49.38	24.20	91.01	48.95	39.31
FedALA	55.75	27.85	92.45	55.61	54.68
AMGCNFL	**58.93**	**29.42**	**93.87**	**57.24**	**56.32**

5.3 Ablation Studies and Analytical Results

Scalability Analysis: Figure 1(a) shows that AMGCN-FL maintains its performance advantage over baseline methods as the number of clients increases from 10 to 100. This demonstrates the scalability of our approach to large federated networks, which is crucial for industrial IoT applications involving numerous devices. Even with 100 clients, AMGCN-FL outperforms FedALA by approximately 1.7 percentage points on the CIFAR-100 dataset.

Convergence Analysis: Figure 1(b) depicts the convergence behavior of different methods across communication rounds. AMGCN-FL not only achieves higher final accuracy but also converges faster than the baseline methods. By round 40, AMGCN-FL already reaches a test accuracy that is higher than what FedAvg achieves after 100 rounds, highlighting the efficiency of our graph-based knowledge transfer mechanism. This faster convergence is particularly important in bandwidth-constrained industrial settings where minimizing communication rounds is essential.

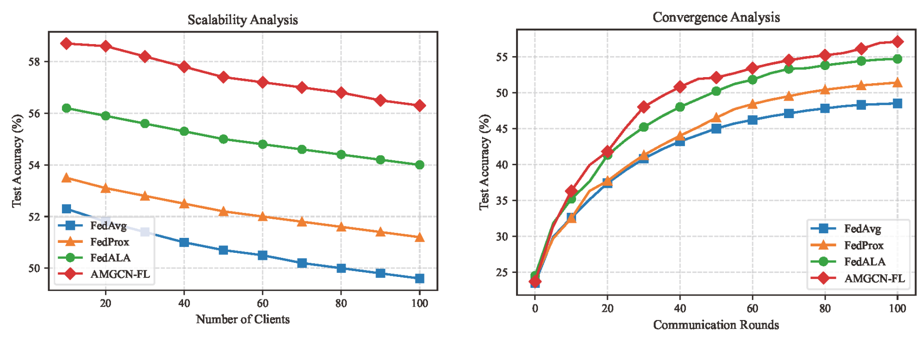

Fig. 1. Scalability and convergence analysis of AMGCN-FL compared to baseline methods:Test accuracy vs. number of clients on CIFAR-100 dataset with Dir(0.1) partition(a,The left one), Convergence rate comparison on CIFAR-100 dataset with Dir(0.1) partition (b,The right one).

Impact of Graph Structures: To understand the contribution of each graph structure, we conduct an ablation study by removing one graph at a time. Table 3 shows that all three graphs contribute to the performance of AMGCN-FL, with the parameter similarity graph having the largest impact. This confirms our intuition that different graph structures capture complementary aspects of client relationships.

AdaptiveLayer-wiseAggregation: Figure 2 (a) visualizes the learned aggregation weights γ_l^i for different layers and clients with varying degrees of data heterogeneity. We observe that lower layers (closer to the input) tend to have lower aggregation weights, indicating more sharing of general features, while higher layers (closer to the output) have higher weights, indicating more personalization for task-specific features. This aligns with the intuition that lower layers learn more transferable features. Moreover, clients with higher data heterogeneity consistently maintain higher aggregation weights across all layers, demonstrating the adaptive nature of our approach to client-specific characteristics.

Graph Weights Evolution: Figure 2 (b) shows how the graph weights α_p, α_f, α_t evolve during training. Initially, all weights are equal (1/3). As training progresses, the parameter similarity graph weight α_p increases, indicating its growing importance, while the feature space graph weight $\alpha_f$and task correlation graph weight α_t decrease but remain significant. This demonstrates the adaptive nature of our multi-graph approach, which dynamically adjusts the importance of different client relationship metrics throughout the training process.

Table 3. Ablation study on graph structures. Test accuracy (\%) on CIFAR-100 with Dir(0.1) partition.

Variant	Test Accuracy
AMGCN-FL (full)	57.24
w/o Parameter Similarity Graph	54.82
w/o Feature Space Graph	55.93
w/o Task Correlation Graph	56.41
Single Static Graph	53.76

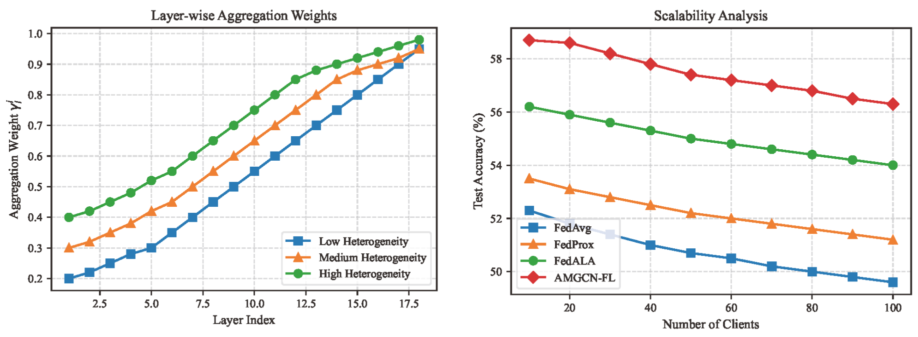

Fig. 2. Analysis of AMGCN-FL's adaptive components during federated learning：Learned aggregation weights for different layers and clients with varying degrees of heterogeneity(a,The left one), Evolution of graph weights during training on CIFAR-100 dataset. Which demonstrate the dynamic adaptation(b,The right one).

5.4 Communication and Computation Efficiency

We also analyze the communication and computation efficiency of AMGCN-FL compared to baseline methods. Table 4 presents the communication cost per round (measured in GB), client computation time (seconds per round), and server computation time (seconds per round) on CIFAR-100 with 50 clients. Although AMGCN-FL introduces a slight overhead compared to simpler methods like FedAvg and FedProx due to the additional graph construction and adaptive layer-wise aggregation, the overhead is moderate and justified by the significant performance improvements. Specifically, AMGCN-FL requires only about 14% more communication bandwidth and 47% more client computation time than FedAvg, while achieving more than 8 percentage points higher accuracy. Compared to the competitive FedALA baseline, AMGCN-FL shows only marginal increases in resource usage while delivering better performance.

Table 4. Communication and computation efficiency comparison on CIFAR-100 with 50 clients.

Methods	Communication Cost (GB per round)	Client Computation (s per round)	Server Computation (s per round)
FedAvg	0.42	3.8	0.6
FedProx	0.42	4.2	0.6
FedRep	0.35	4.4	0.7
Ditto	0.84	5.1	0.9
FedALA	0.44	5.2	1.3
AMGCN-FL	0.48	5.6	1.7

5.5 Industrial IoT Case Study

To demonstrate the practical relevance of AMGCN-FL, we conduct a case study on an industrial IoT dataset collected from manufacturing equipment. The dataset contains sensor readings from 30 machines across 5 factories, with each machine performing similar but not identical tasks. The goal is to predict equipment failures based on sensor data.

Table 5 shows that AMGCN-FL achieves the highest prediction accuracy and F1-score compared to baseline methods. The improvement is particularly significant for rare failure events, which are critical in industrial settings. This demonstrates the effectiveness of our approach in real-world industrial applications with heterogeneous data distributions.

Furthermore, we analyze the interpretability of our approach in the industrial IoT context. By examining the learned graph structures, factory operators can identify clusters of machines with similar behavior patterns and failure modes. This information can be valuable for maintenance planning and resource allocation. Additionally, the layer-wise aggregation weights provide insights into which components of the models benefit most from knowledge sharing and which require more personalization, helping to optimize the trade-off between global knowledge and local specialization.

Table 5. Performance comparison on industrial IoT dataset for equipment failure prediction.

Methods	Accuracy (%)	F1-Score	AUC
FedAvg	87.32	0.76	0.83
FedProx	87.65	0.77	0.84
FedRep	89.41	0.81	0.87
Ditto	90.23	0.83	0.89
FedALA	91.56	0.85	0.91
AMGCN--FL	**93.28**	**0.88**	**0.93**

6 Conclusion

In this paper, we proposed AMGCN-FL, an adaptive multi-graph convolutional network approach for personalized federated learning in industrial IoT environments. Our method leverages multiple graph structures to capture different aspects of client relationships and employs adaptive layer-wise aggregation to balance knowledge sharing and local specialization. Theoretical analysis established the convergence properties of our algorithm under non-IID data distributions. Extensive experiments on benchmark datasets and an industrial IoT case study demonstrated that AMGCN-FL consistently outperforms state-of-the-art personalized FL methods across various heterogeneity settings.

The key advantages of AMGCN-FL include: (1) adaptive modeling of complex client relationships through multiple complementary graph structures, (2) parameter-efficient knowledge transfer that preserves local specialization, and (3) robust performance across varying degrees of data heterogeneity. These properties make AMGCN-FL particularly suitable for industrial IoT applications where devices generate heterogeneous data under diverse operating conditions.

Future work includes extending AMGCN-FL to handle dynamic client participation, incorporating privacy-preserving mechanisms for graph construction, and exploring the application of our approach to other domains with heterogeneous distributed data. We also plan to investigate the theoretical connections between graph structures and convergence rates in more detail.

Acknowledgements. This work was supported by the 2024 College Students' Innovative Entrepreneurial Training Plan Program (No.202410293051Z).

References

1. Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S.: Federated learning with personalization layers. arXiv preprint arXiv:1912.00818 (2019)

2. Chen, M., Mathews, R., Ouyang, T., Beaufays, F.: Federated learning of out-of-vocabulary words. arXiv preprint arXiv:1903.10635 (2019)
3. Collins, L., Hassani, H., Mokhtari, A., Shakkottai, S.: Exploiting shared representations for personalized federated learning. In: International conference on machine learning. pp. 2089–2099. PMLR (2021)
4. Fallah, A., Mokhtari, A., Ozdaglar, A.: Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. Advances in neural information processing systems **33**, 3557–3568 (2020)
5. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning. pp. 1126–1135. PMLR (2017)
6. Huang, Y., Chu, L., Zhou, Z., Wang, L., Liu, J., Pei, J., Zhang, Y.: Personalized federated learning: An attentive collaboration approach. arXiv preprint arXiv:2007.03797 **1**, 2 (2020)
7. Huang, Y., Chu, L., Zhou, Z., Wang, L., Liu, J., Pei, J., Zhang, Y.: Personalized cross-silo federated learning on non-iid data. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 7865–7873 (2021)
8. Jiang, Y., Konečn` y, J., Rush, K., Kannan, S.: Improving federated learning personalization via model agnostic meta learning. arXiv preprint arXiv:1909.12488 (2019)
9. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: International conference on machine learning. pp. 5132–5143. PMLR (2020)
10. Lee, S., Zhang, T., Avestimehr, A.S.: Layer-wise adaptive model aggregation for scalable federated learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 8491–8499 (2023)
11. Li, D., Wang, J.: Fedmd: Heterogenous federated learning via model distillation. arXiv preprint arXiv:1910.03581 (2019)
12. Li, T., Hu, S., Beirami, A., Smith, V.: Ditto: Fair and robust federated learning through personalization. In: International conference on machine learning. pp. 6357–6368. PMLR (2021)
13. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems **2**, 429–450 (2020)
14. Lin, T., Kong, L., Stich, S.U., Jaggi, M.: Ensemble distillation for robust model fusion in federated learning. Advances in neural information processing systems **33**, 2351–2363 (2020)
15. Liu, R., Xing, P., Deng, Z., Li, A., Guan, C., Yu, H.: Federated graph neural networks: Overview, techniques, and challenges. IEEE transactions on neural networks and learning systems (2024)
16. Lu, W., Wang, J., Chen, Y., Qin, X., Xu, R., Dimitriadis, D., Qin, T.: Personalized federated learning with adaptive batchnorm for healthcare. IEEE Transactions on Big Data (2022)
17. Luo, M., Chen, F., Hu, D., Zhang, Y., Liang, J., Feng, J.: No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. Advances in Neural Information Processing Systems **34**, 5972–5984 (2021)
18. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
19. Shamsian, A., Navon, A., Fetaya, E., Chechik, G.: Personalized federated learning using hypernetworks. In: International conference on machine learning. pp. 94899502. PMLR (2021)

20. T Dinh, C., Tran, N., Nguyen, J.: Personalized federated learning with moreau envelopes. Advances in neural information processing systems 33, 21394–21405 (2020)
21. Wu, C., Wu, F., Cao, Y., Huang, Y., Xie, X.: Fedgnn: Federated graph neural network for privacy-preserving recommendation. arXiv preprint arXiv:2102.04925 (2021)
22. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-iid data. arXiv preprint arXiv:1806.00582 (2018)