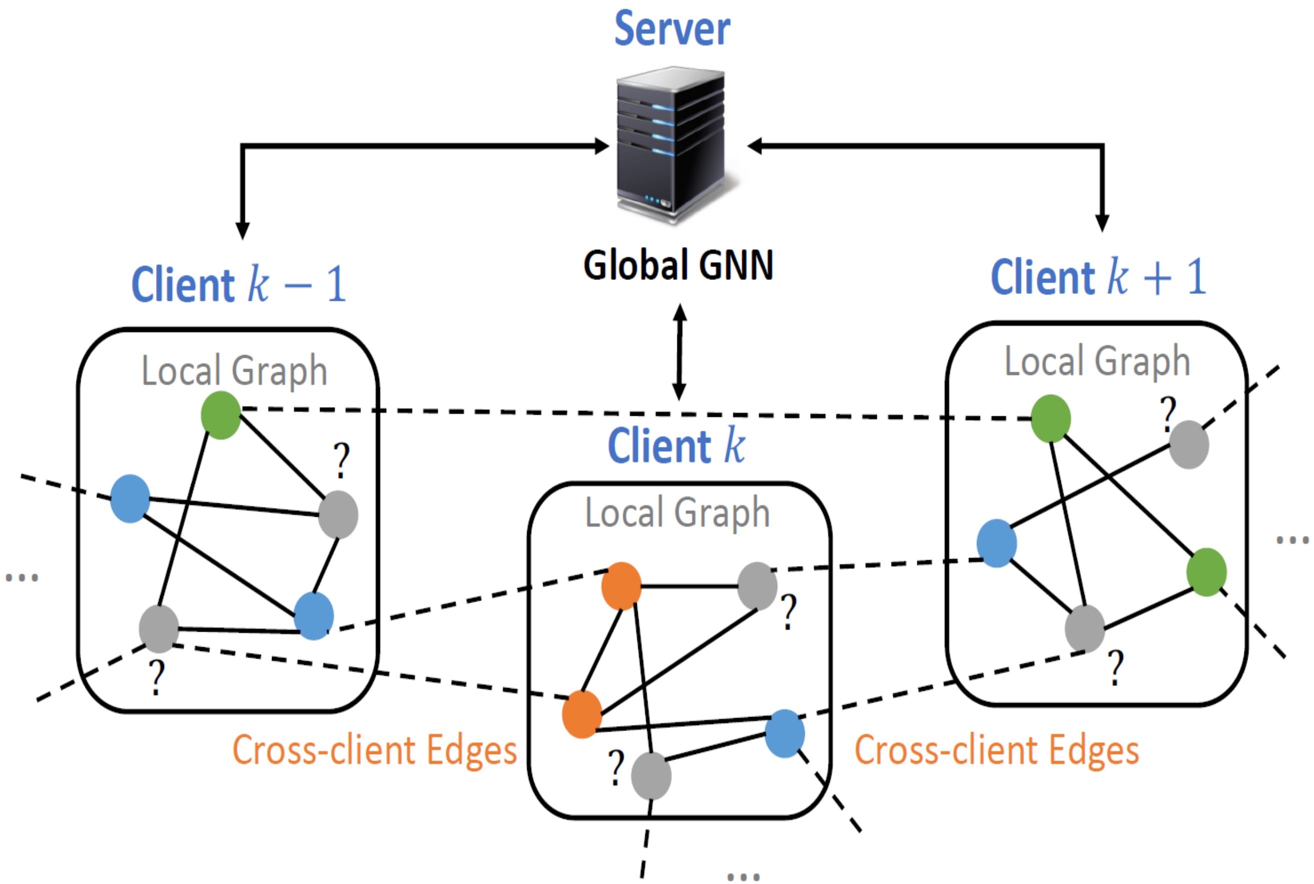
Research on Node Classification in Federated Subgraph Learning

汇报人：闫凡宇　　　时间：2025.03.05

- **Background**
- **Related Works**
- **Model: FedGRT**
- **Experiment**
- **Future Work**

Background：Federated Graph Node Classification

- Node in a graph are partitioned across clients(e.g. private data across countries)
- Cross-client edges between nodes none-exist at any client



Each client **knows**

- local graph structure
- local node feature

Each client **unknows**

- the cross-subgraph edges
- other clients feature

Target

- high accuracy of node classification
- privacy protection
- low communication overhead

Challenges

- missing cross-client edges result in key information loss
- raw feature exchange leads to privacy leakage

Node classification required node feature stored in other clients

Related Works

(2017) **FedAvg**

- FL framework.
- no graph data, no Non-IID data.

(2018) **FedGCN**

- FL with graph convolutional network;
- global graph structure

(2021) **FedSage**

- Missing neighbor generation.
- unreal node, without considering global information.

(2021) **FedGAT**

- graph attention mechanism with GFL.
- high computational complexity, no Non-IID data.

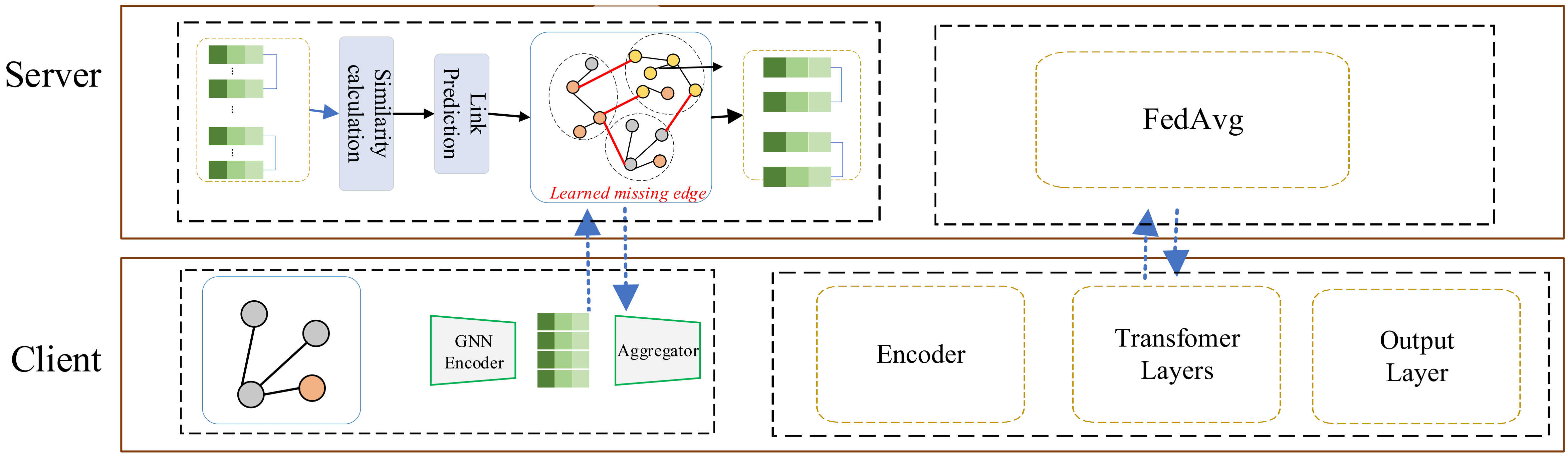
(2021) **Fed-PUB**

- personalized GFL, some non-iid data issues.
- communication overhead, model complexity.

(2024) **FedGT**

- hybrid attention mechanism, global node alignment, some data heterogeneity.
- relies on the quality of the selected global node.

Architecture of FedGRT(Federated Graph with Global Reconstruction and Transfomer)



- **Global Graph Reconstruction**: FedSage： repairs missing neighbors on subgraphs； FedGT： constructs global nodes FedGRT： reconstruct the original global graph, providing more comprehensive repaired graph information.
- **High efficiency：** Node embedding satisfying the KN constraint **enables，** If there are many neighbors, the parameter N ensures that not too many neighboring nodes are aggregated, leading to high efficiency;
  if there are few neighbors, the parameter K ensures that the model does not overly explore distant neighbors, reducing time complexity and minimizing the impact of distant nodes on the aggregation effect.
- **Security Assurance：** Only parameter exchange, without the transmission of raw information of graph.

Stage 1: Global Graph Reconstruction

Step 1: Node embedding

using random walk model to construct graph embedding representation of node

$$G_i = (V_i, E_i, X_i)$$

$$E_i \in R^{|E_i| \times 2}, X_i \in R^{|V_i| \times d}$$

$$Z_i = rw(G_i, D, M) \in R^{|V_i| \times \mathrm{D}}$$

Step 2: Feature Aggregation

aggregate the embedding vector obtained from random walks with the original feature vector

$$Z_i \in R^{|V| \times \mathrm{D}}, X \in R^{|V| \times \mathrm{d}}$$

$$H_i = [Z_i; X_i] \in R^{|V_i| \times (D+d)}$$

Step 3: Similarity Calculation

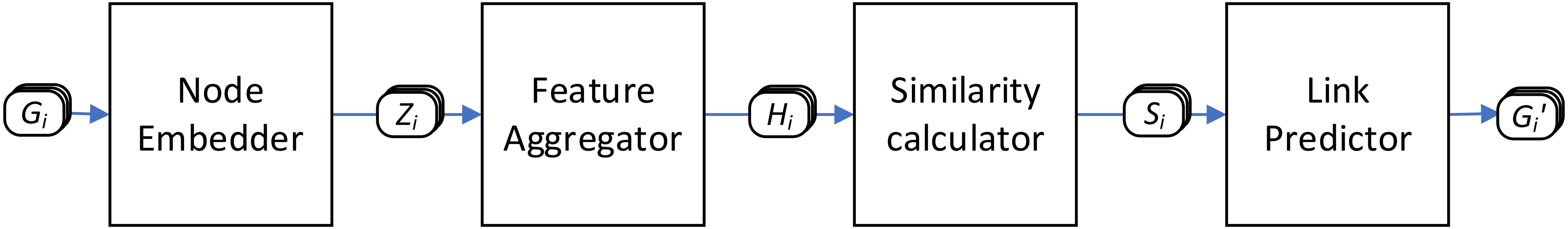
Capture the similarity between nodes by Calculate cosine similarity between nodes

$$S_{\mathrm{i}} = cs(u, v) = \frac{h_u \cdot h_v}{||h_u|| \, ||h_v||}$$

Step 4: Link Prediction

Select the K pairs of nodes with the highest similarity and add edge connections between them

$$E' = \{(u_1, v_1), \ldots, (u_K, v_K)\}$$



Stage 2：Graph Transformer Encoder

- **Neighbor aggregation satisfying KN constraint**

The sampled neighboring nodes satisfy that the number of neighbors is not greater than N and the maximum distance between sampled neighbors is not greater than K.

$$d(i,j) \le k, \forall j \in \mathcal{N}(i)$$

$$|\mathcal{N}(i)| \le N$$

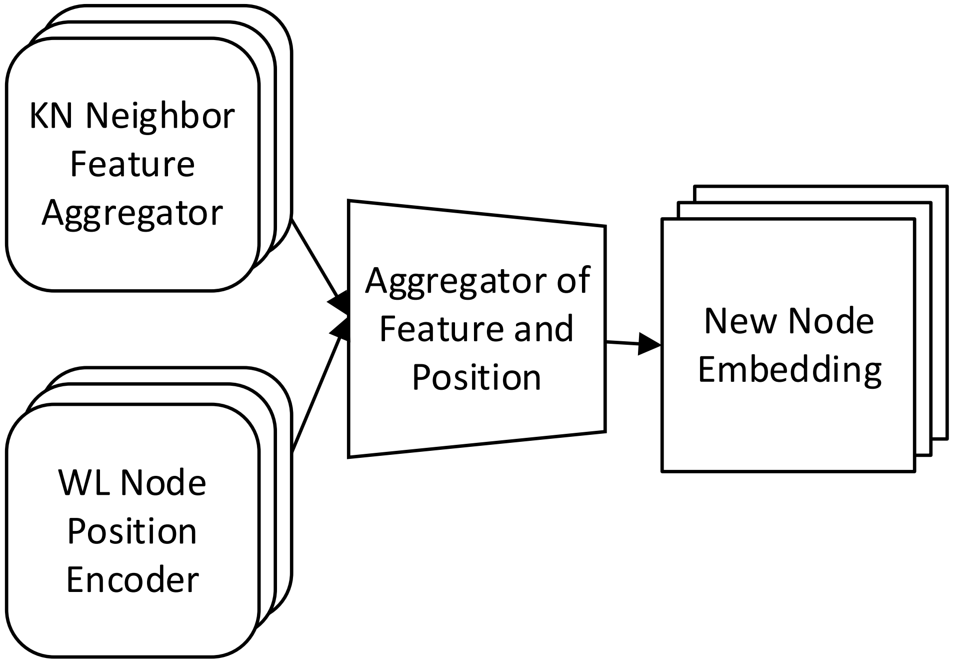
Obtain the feature embedding of node i that satisfies the KN constraint node feature embedding algorithm.

$$e_i^{(x)} = AggX\big(i, \mathcal{N}(i)\big)$$

$$AggX\big(i, \mathcal{N}(i)\big) = AggX\left(h_i, \{h_j | \forall j \in \mathcal{N}(i)\}\right) = h_i + \frac{1}{|\mathcal{N}(i)|}\sum_{j \in \mathcal{N}(i)} h_j$$

- **Position embedding based on WL algorithm**

$$\mathrm{e}_i^{(a)} = \mathrm{PE}(\mathrm{v}_i) = [\sin(\frac{WL(v_i)}{10000^{\frac{2l}{d_h}}}), \cos(\frac{WL(v_i)}{10000^{\frac{2l+1}{d_h}}})]_{l=0}^{\frac{d_h}{2}}$$



- **Aggregate node feature embedding and position encoding**

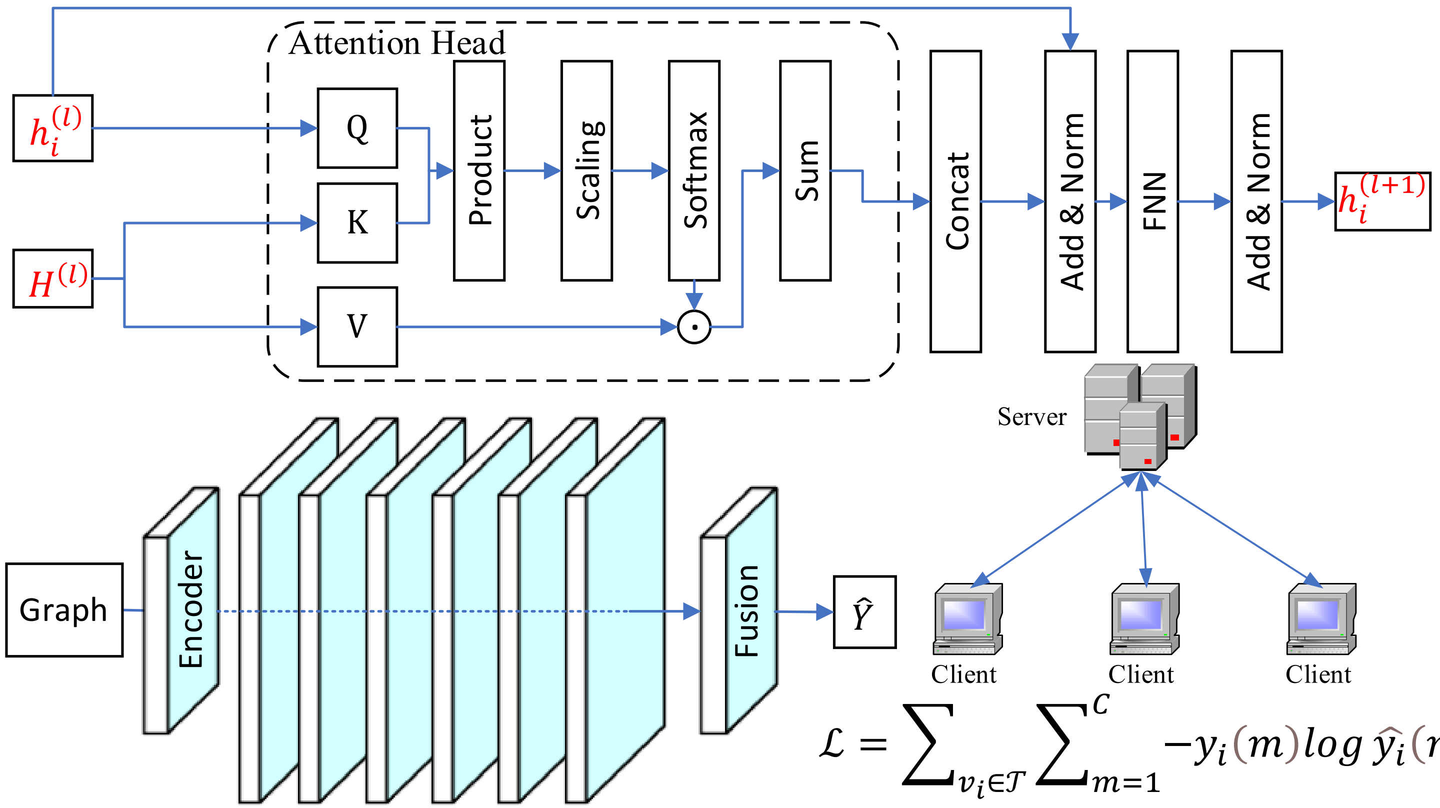
$$h_i^{(0)} = Aggregate\left(e_i^{(x)}, e_i^{(a)}\right) = \left[e_i^{(x)}\ e_i^{(a)}\right] \in R^{1\times(D+d_h)}$$

$$\mathrm{H}^{(0)} = \left[\mathrm{h}_1^{(0)}; \mathrm{h}_2^{(0)}; \ldots; \mathrm{h}_\mathrm{N}^{(0)}\right] \in R^{N\times(D+d_h)}$$

Stage 2: FedGRT Network Architecture

$$\begin{cases} H^{(0)} = \left[h_1^{(0)}; h_2^{(0)}; \dots; h_N^{(0)}\right] \\ H^{(l)} = GraphTransformer\left(H^{(l-1)}\right), \forall l \in \{1,2,\dots L\} \\ \hat{Y} = Fusion\left(H^{(L)}\right) \end{cases} \qquad \begin{cases} Q = H^{(l-1)} W_Q^{(l)} \\ K = H^{(l-1)} W_K^{(l)} \\ V = H^{(l-1)} W_V^{(l)} \end{cases}$$

$$H^{(l)} = SoftMax\left(\frac{QK^T}{\sqrt{D + d_h}}\right) V + Res\left(H^{(l-1)}, X_i\right)$$



$$\mathcal{L} = \sum_{v_i \in \mathcal{T}} \sum_{m=1}^{C} -y_i(m) \log \hat{y_i}(m)$$

Algorithm 1: Server side of FedGRT's second stage

Server

$W_S^{(0)} \leftarrow 0, W_C^{(0)} \leftarrow 0$

for *each round* $t = 1,2,\dots$ **do**

  $Z_{glob} \leftarrow Model(h_1, h_2)$

  for *each client c* **do**

    $W_{C,i}^{(t)}, \nabla Z_{glob}^i \leftarrow$ ClientTraining$(i, W_C^{(t)}, Z_{glob}^i)$

  end for

  $\nabla Z_{glob} \leftarrow$ CONCAT$(\nabla Z_{glob}^1, \dots, \nabla Z_{glob}^m)$

  $\nabla W_S^{(t)} \leftarrow Z_{glob}.backward\left(\nabla Z_{glob}\right)$

  $W_C^{(t+1)} \leftarrow \sum_i |V_i|/|V|\, W_{C,i}^{(t)}$

  $W_S^{(t+1)} \leftarrow W_S^{(t)} - \eta \nabla W_S^{(t)}$

end for

Algorithm 2: Client side of FedGRT's second stage

Client

$W_{C,i}^{(t)} \leftarrow W_C^{(t)}$

for *each round* $t = 1,2,\dots$ **do**

  $\mathcal{L}_i \leftarrow Loss\left(\tilde{Y}, Y\right)$ **and** $\boldsymbol{Z_{glob}^i}$

  $\nabla W_{C,i}^{(t)}, \nabla Z_{glob}^i \leftarrow \mathcal{L}_i.backward()$

  $W_{C,i}^{(t+1)} \leftarrow W_{C,i}^{(t)} - \eta \nabla W_{C,i}^{(t)}$

end for

return $W_{C,i}^{(t)}$ and ∇Z_{glob}^i to Server

Experiment Settings

Dataset: Cora, Citeseer, PubMed

Baselines for comparison: GCN(Kip et al., 2017), GAT(Velickovic et al., 2018), FedSage(Zhang et al., 2021), Fed-PUB(Baek et al., 2023), FedGT(Zhang et al., 2024)

Evaluation Metrics: classification accuracy

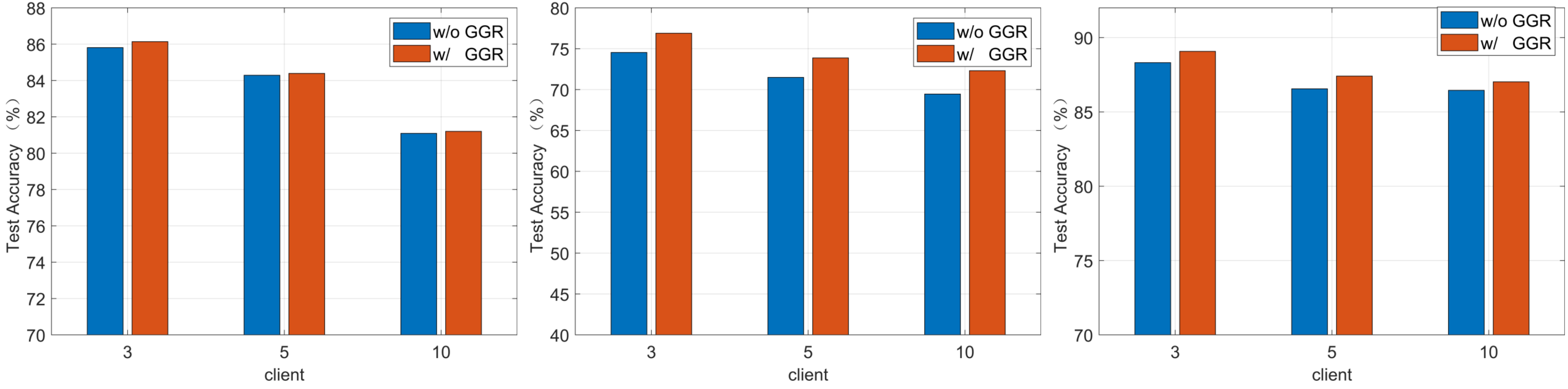
transformer layer	6
attention header	4
learning rate	0.001
weight decay	0.0005
local epoch	5
communication round	50

Dataset	Cora			Citeseer			PubMed		
$\#C$	7			6			3		
$\|V\|$	2708			3312			19717		
$\|E\|$	5429			4715			44338		
M	3	5	10	3	5	10	3	5	10
$\|V_i\|$	903	542	271	1104	662	331	6572	3943	1972
$\|E_i\|$	1675	968	450	1518	902	442	12932	7630	3789
$\|E\|'$	403	589	929	161	206	300	5543	6189	6445

Experiment Result

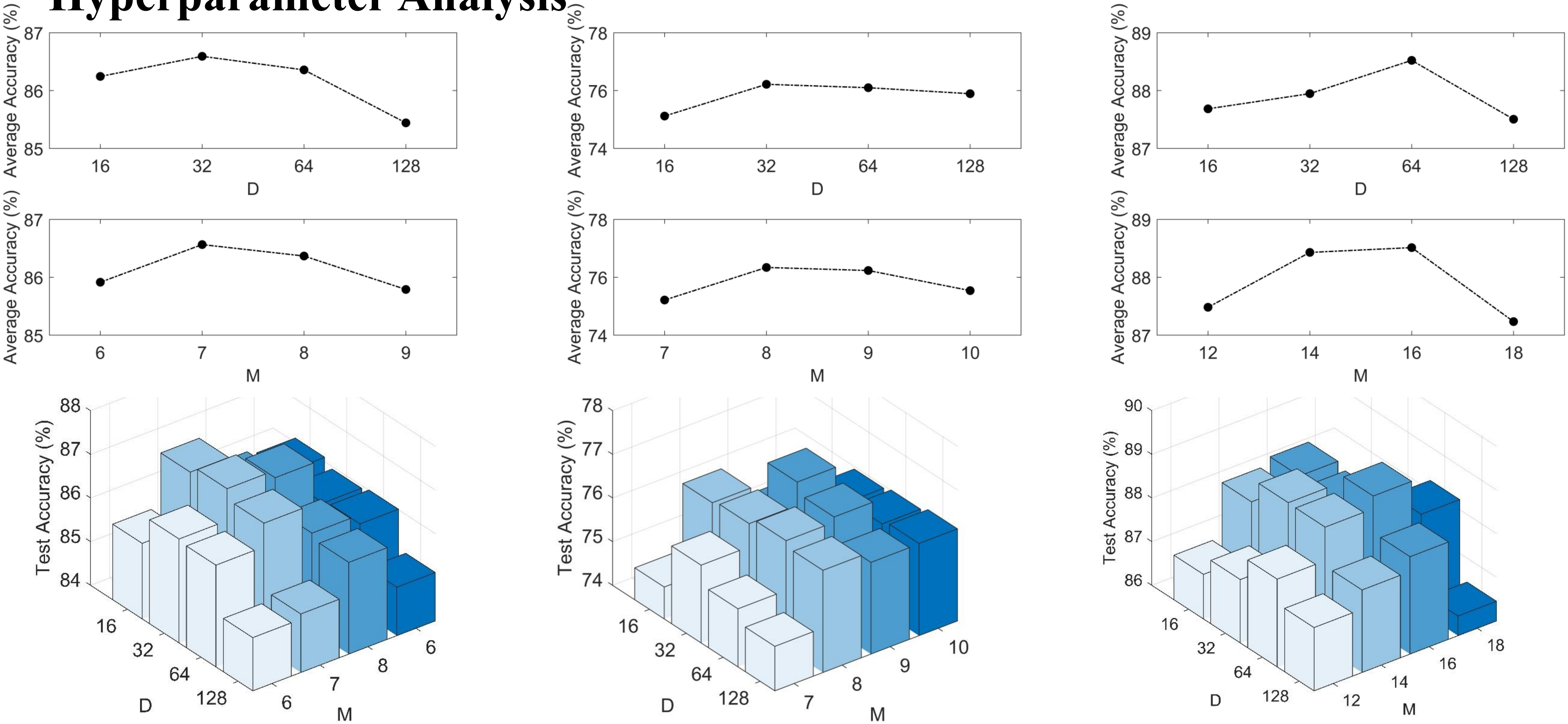
Method	Cora			CiteSeer			PubMeb		
	3 Clients	5 Clients	10 Clients	3 Clients	5 Clients	10 Clients	3 Clients	5 Clients	10 Clients
GCN	82.31	80.54	78.71	70.66	68.34	65.34	83.54	81.80	79.40
GAT	82.64	80.21	78.72	72.82	71.71	68.04	84.60	83.93	82.19
FedSage	83.71	81.11	77.21	75.36	73.67	70.34	86.59	86.34	84.23
Fed-PUB	83.72	81.45	81.11	75.38	73.89	69.03	87.83	86.81	84.66
FedGT	84.41	82.49	**81.25**	75.69	**74.01**	71.98	88.78	87.21	86.65
FedGRT	**86.14**	**84.39**	81.20	**76.89**	73.90	**72.31**	**89.07**	**87.49**	**87.02**

Ablation Study



Method	Cora			CiteSeer			PubMeb		
	3 Clients	5 Clients	10 Clients	3 Clients	5 Clients	10 Clients	3 Clients	5 Clients	10 Clients
w/o GGR	85.81	84.29	81.09	74.54	71.49	69.45	88.31	86.55	86.45
w/ GGR	86.14	84.39	81.20	76.89	73.87	72.31	89.07	87.41	87.02
↑ (%)	**0.33 ↑**	**0.10 ↑**	**0.11 ↑**	**2.35 ↑**	**2.38 ↑**	**2.86 ↑**	**0.76 ↑**	**0.86 ↑**	**0.57 ↑**

Hyperparameter Analysis



	Cora	CiteSeer	PubMeb
D	32	**32**	64
M	7	**9(8)**	16

Future Work

1 Improve the experiments

2 Continue writing the paper

Nanjing University of Posts and Telecommunications

Thanks