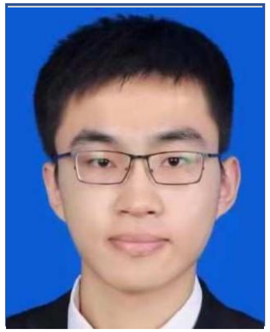
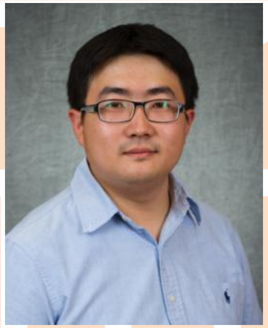
Federated Graph Learning: Recent Advances and Future Directions

Xingbo Fu   Zihan Chen   Binchi Zhang   Jundong Li

University of Virginia

Outline

- ✓ Introduction
- ✓ Subgraph Federated Learning
- ✓ Federated Graph Learning with Non-IID Graphs
- ✓ Privacy-Preserving Federated Graph Learning
- ✓ Summary and Future Directions

Outline

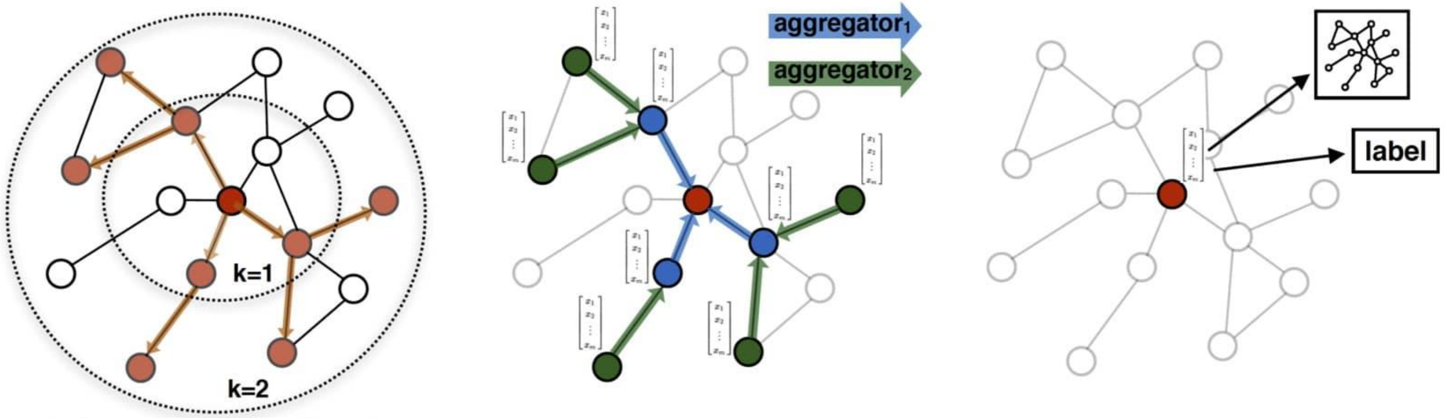
- ✓ Introduction
- ✓ Subgraph Federated Learning
- ✓ Federated Graph Learning with Non-IID Graphs
- ✓ Privacy-Preserving Federated Graph Learning
- ✓ Summary and Future Directions

Introduction

➢ What is Federated Graph Learning?

❑ Traditional Graph Learning

- Train graph learning models on graph data collected in a single machine
- Inapplicable in practice due to privacy concerns and regulations[1]



1. Sample neighborhood

2. Aggregate feature information from neighbors

3. Predict labels using aggregated information

Graph neural networks (GNNs) aggregate information from neighbors to learn node embeddings

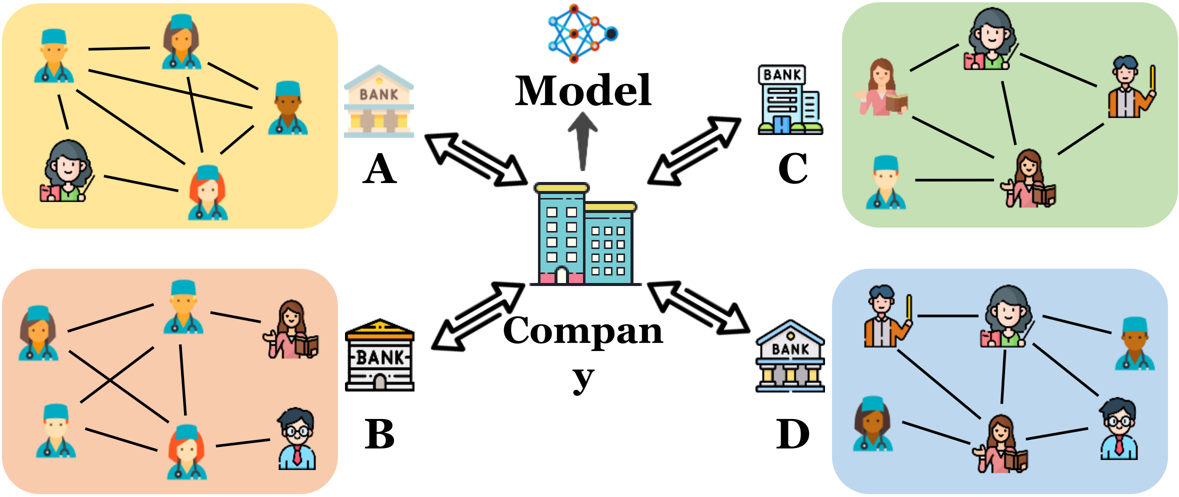
[1] Voigt, Paul, and Axel von dem Bussche. "The EU General Data Protection Regulation (GDPR) A Practical Guide." (2017).

Introduction

➢ What is Federated Graph Learning?

❑ Federated Graph Learning (FGL)

- Collaborative learning on graph data distributed in multiple clients
- Applications: financial systems, healthcare systems, medical institutes, E-commerce companies……



An example of a financial system including four banks

Introduction

➢ What is Federated Graph Learning?

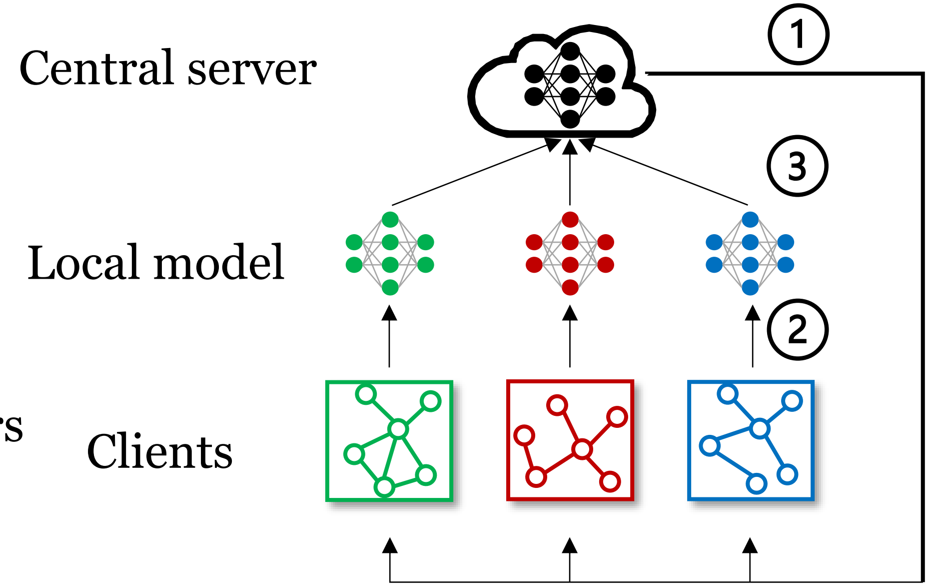
❑ Federated Graph Learning (FGL)

- Collaborative learning on graph data distributed in multiple clients
- Applications: financial systems, healthcare systems, medical institutes, E-commerce companies……
- Framework: FedAvg[1], FedProx[2], ……

① The server sends current model parameters to clients

② Each client performs local updates on its local graph data

③ The server takes a weighted average of local model parameters



[1] McMahan, Brendan, et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data." AISTATS 2017.
[2] Li, Tian, et al. "Federated Optimization in Heterogeneous Networks." MLSys 2020.

Introduction

- Research Topics in FGL

- Subgraph Federated Learning
  - Missing cross-client links
  - Community heterogeneity

- Federated Graph Learning with Non-IID Graphs
  - Cross-dataset structural knowledge sharing
  - Distribution shifts

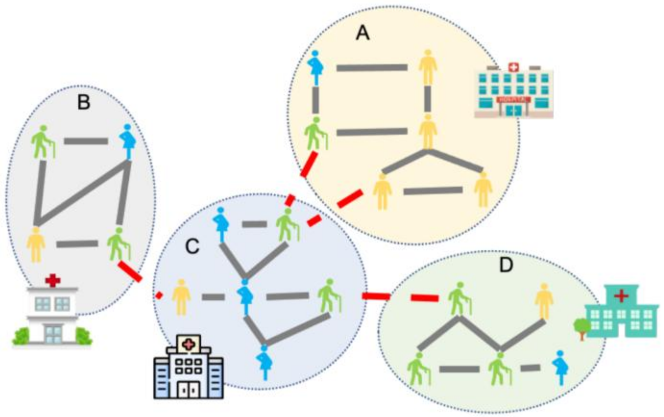
- Privacy-Preserving Federated Graph Learning
  - Entity-level privacy protection
  - Structure-level privacy protection

Introduction

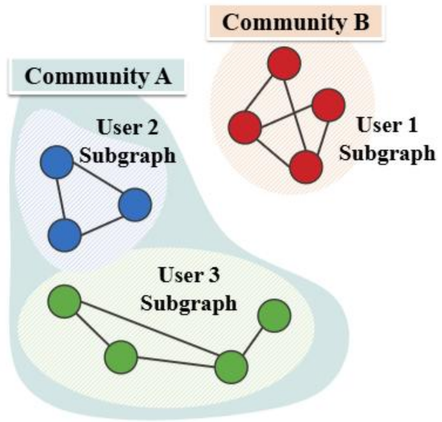
➢ Research Topics in FGL

❑ Subgraph Federated Learning

- Each client only holds a subgraph (a local view) of the global graph and cannot share raw data due to privacy or communication constraints
- Challenges: missing cross-client links & community heterogeneity



Missing cross-client links



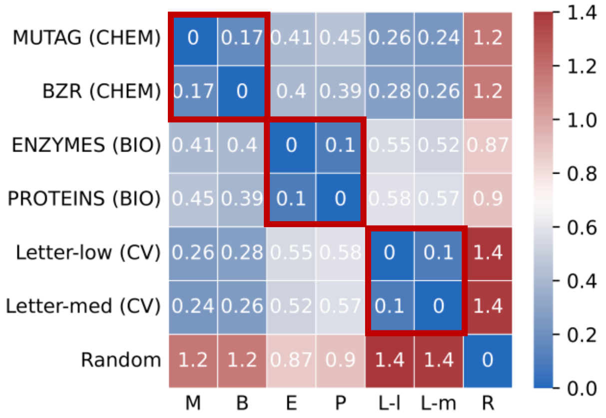
Community heterogeneity

Introduction

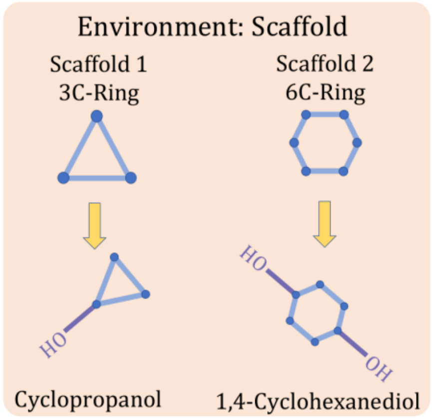
➢ Research Topics in FGL

❑ Federated Graph Learning with Non-IID Graphs

- Each client has multiple graphs and focuses on graph-level tasks (e.g., graph classification/regression)
- Graphs across clients are usually non-IID
- Challenges: cross-dataset structural knowledge sharing & distribution shifts



Cross-dataset structural knowledge sharing



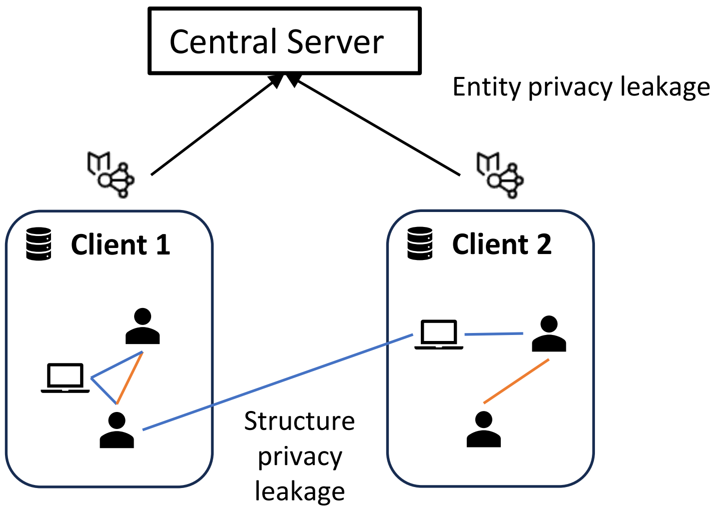
Distribution shifts

Introduction

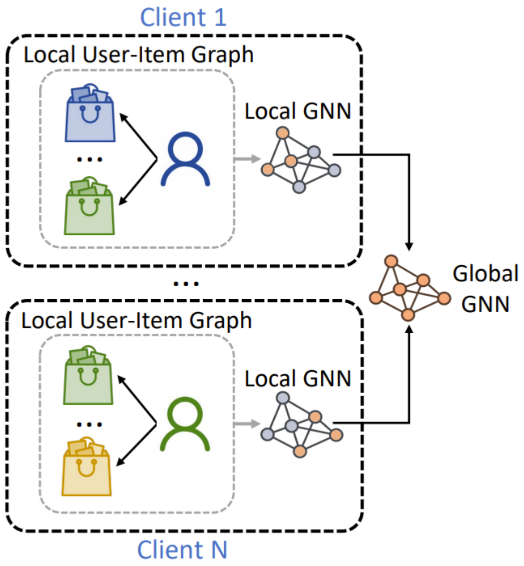
➢ Research Topics in FGL

❑ Privacy-Preserving Federated Graph Learning

- Graph data may be leaked/inferred in the central server
- Challenges: entity-level privacy protection & structure-level privacy protection



Entity-level privacy protection



Structure-level privacy protection

Introduction

➢ Tutorial Outline

Research Topics	Challenges	Techniques	Representative Works
Subgraph Federated Learning (25 mins)	Missing cross-client links	Missing neighbor generator	FedSage+
	Community heterogeneity	Functional similarity matching + personalized parameter masking	Fed-PUB
FGL with Non-IID Graphs (25 mins)	Cross-dataset structural knowledge sharing	Structure knowledge sharing	FedStar
	Distribution shifts	Virtual node optimization	FedVN
Privacy-Preserving FGL (25 mins)	Entity-level privacy protection	(Local) differential privacy	FedSoG
	Structure-level privacy protection	Local information mixup	FedGNN, FedEgo

Outline

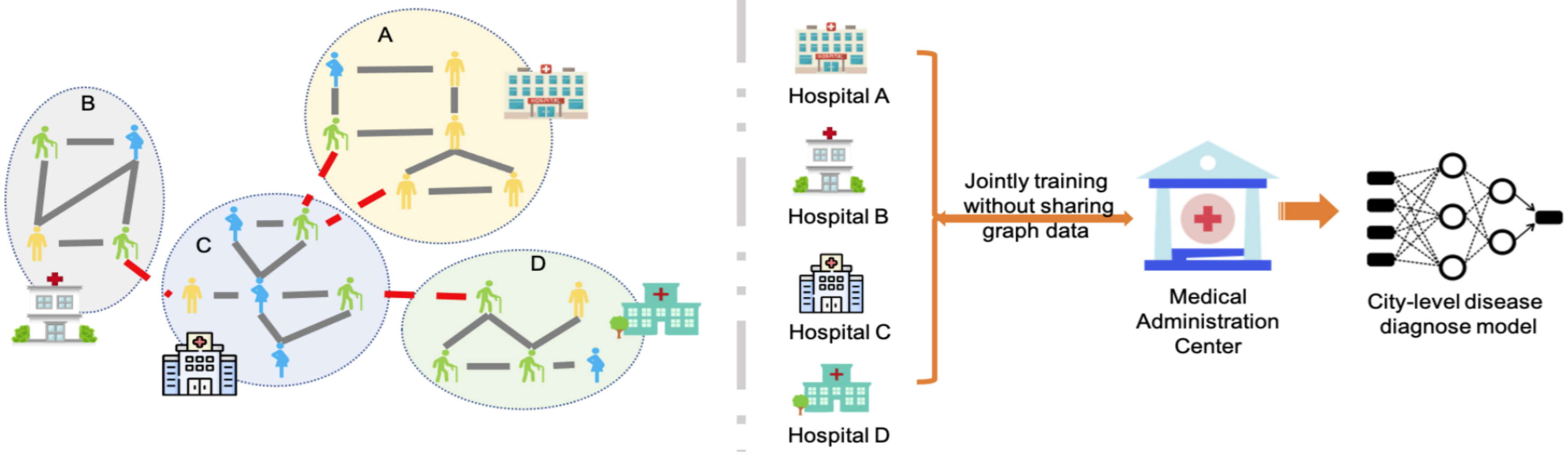
- ✓ Introduction
- ✓ Subgraph Federated Learning
- ✓ Federated Graph Learning with Non-IID Graphs
- ✓ Privacy-Preserving Federated Graph Learning
- ✓ Summary and Future Directions

Subgraph Federated Learning

➢ Background

❑ Problem Setting

- **Setting**: Each client **only holds a subgraph** (a local view) of the global graph and cannot share raw data due to privacy or communication constraints
- **Example:** Each hospital holds a **patient interaction subgraph**, where nodes represent patients and edges reflect contact or shared treatment. Using subgraph FL, hospitals can collaboratively train a disease prediction model **without sharing sensitive patient data**



Subgraph Federated Learning

➢ Background

❑ Problem Formulation

- Consider M clients. Each client $i \in [M]$ holds a local subgraph

$$G_i = \{V_i, E_i, X_i\} \subset G \ , i \in [M].$$

- Collaboratively learn models $\{\boldsymbol{f}(\cdot\,; \boldsymbol{\phi_i})\}_{[\boldsymbol{M}]}$ (GNNs) that minimizes the loss

$$\min_{\{\boldsymbol{\theta_i}\}_{[M]}} \sum_i \frac{|V_i|}{|V|} \boldsymbol{\mathcal{L}_i}(G_i; \boldsymbol{\phi_i}),$$

where $\boldsymbol{\mathcal{L}_i}$ and $\boldsymbol{\phi_i}$ denote the local objective function and model parameters

Subgraph Federated Learning

➢ Challenges in Subgraph Federated Learning

❑ Missing Cross-Client Links

- Training a separate graph mining model on each subgraph may **not capture the global data distribution** and is also prone to **overfitting**
- Due to privacy or siloed storage, the **cross-subgraph connections are unavailable**, leading to **incomplete neighborhoods** and degraded GNN performance

❑ Community Heterogeneity

- Subgraphs originate from **different communities** in the global graph, which can have **incompatible properties**
- Naïvely aggregating all local models leads to **knowledge collapse** — degradation due to incompatible updates

Subgraph Federated Learning

➢ Missing Cross-Client Links

❑ Joint Learning from Heterogeneous Subgraphs

- The global graph is distributed into a set of small subgraphs with **heterogeneous feature and structure** distributions
- Training locally may lead to **overfitting** and **poor generalization**

❑ Solution

- **FedSage = GraphSage + FedAvg**
- GraphSage: For a node $v \in V_i$ with features as $h_v^0 = x_v$, at each layer k,

$$h_v^k = \sigma\left(\phi^k \cdot \left(h_v^{k-1} \parallel \mathrm{AGG}\left(\left\{h_u^{k-1}, \forall u \in N_{G_i}(v)\right\}\right)\right)\right)$$

Subgraph Federated Learning

➢ Missing Cross-Client Links

❑ Cross-subgraph Connections are Unavailable during Deploying FedSage

- The inability to access the full ego-networks causes the neighborhood **aggregation to be biased**, violating GNN assumptions
- This results in **limited expressive power** and **suboptimal predictions**

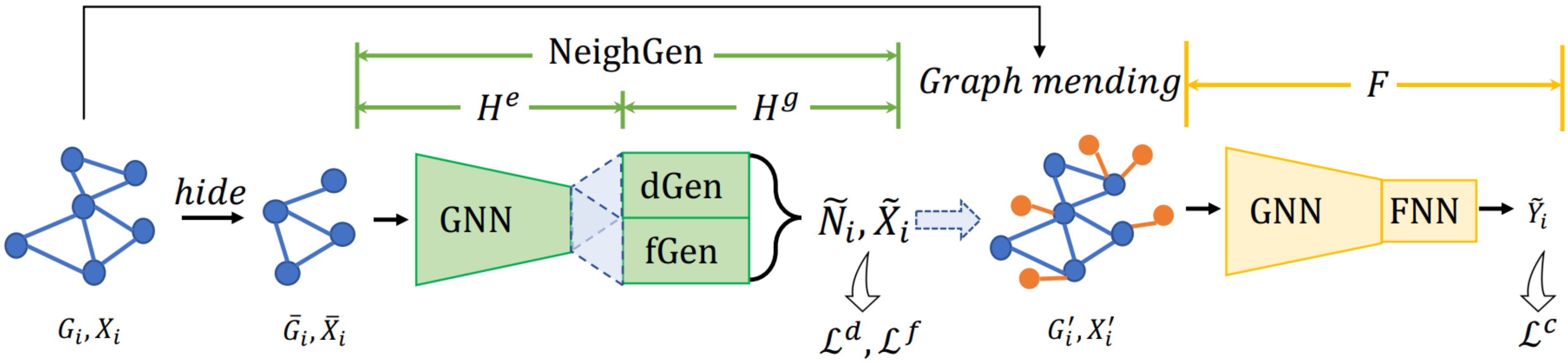
❑ Solution

- **FedSage+**: generating missing neighbors along FedSage

Subgraph Federated Learning

- Missing Cross-Client Links

- FedSage+
    - Each client first mends its subgraph by generating missing neighbors, then applies FedSage on the augmented subgraph



Subgraph Federated Learning

➢ Missing Cross-Client Links

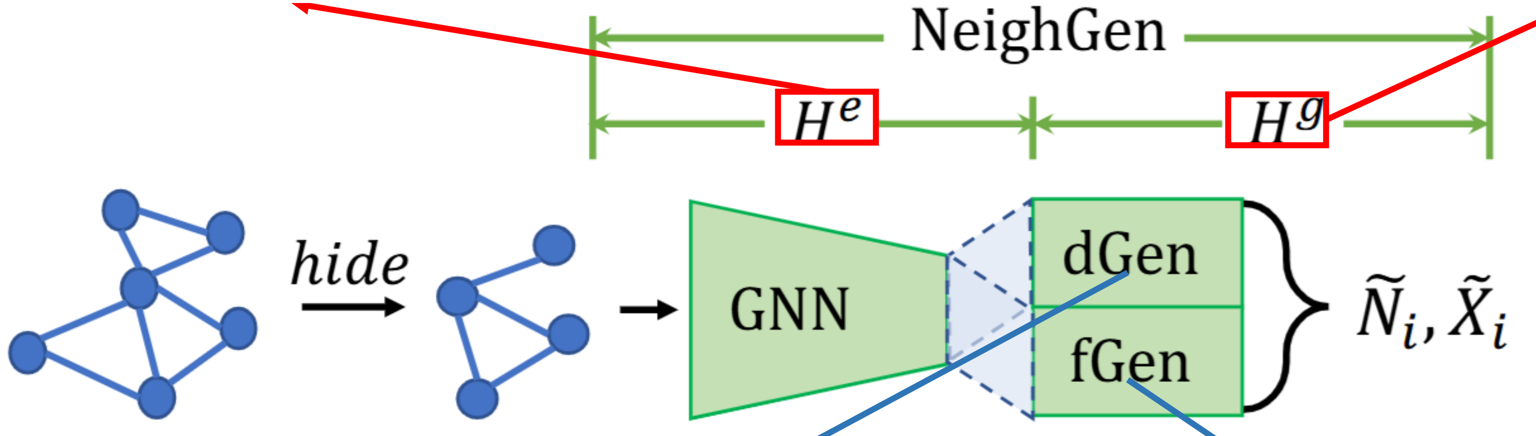
❑ FedSage+

- Missing Neighbor Generator (NeighGen)

H^e: a K-layer GraphSage encoder

$$Z_i = \{z_v | z_v = h_v^K, z_v \in \mathbb{R}^{d_z}, v \in V_i\}$$

H^g: a generative model (FNN) recovering missing neighbors



dGen: a linear regression model which predicts the numbers of missing neighbors

$$\tilde{N}_i = \{\tilde{n}_v | \tilde{n}_v \in \mathbb{N}, v \in V_i\}$$

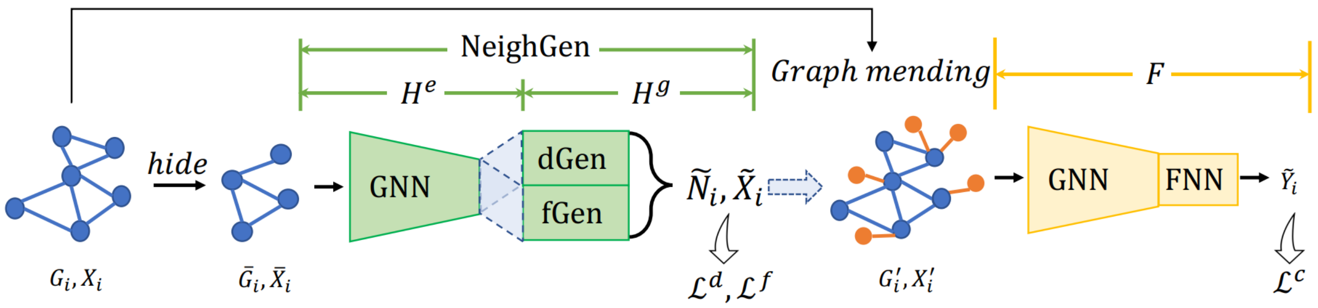
fGen: a feature generator which generates a set of $\tilde{N}_i$ feature vectors

$$\tilde{X}_i = \{\tilde{x}_v | \tilde{x}_v \in \mathbb{R}^{\tilde{n}_v \times d_x}, \tilde{n}_v \in \tilde{N}_i, v \in V_i\}$$

Subgraph Federated Learning

- Missing Cross-Client Links

- FedSage+
  - Missing Neighbor Generator (NeighGen)



Loss for dGen

Loss for fGen

$$\mathcal{L}^n = \lambda^d \mathcal{L}^d + \lambda^f \mathcal{L}^f = \lambda^d \boxed{\frac{1}{|\bar{V}_i|} \sum_{v \in \bar{V}_i} L_1^S(\tilde{n}_v - n_v)} + \lambda^f \boxed{\frac{1}{|\bar{V}_i|} \sum_{v \in \bar{V}_i} \sum_{p \in [\tilde{n}_v]} \min_{u \in \mathcal{N}_{G_i}(v) \cap V_i^h} (||\tilde{x}_v^p - x_u||_2^2)}$$

$\bar{V}_i$: the remaining node set in $\bar{G}_i$

$\tilde{n}_v$: the predicted number of v's missing neighbors

n_v: the ground-truth number of v's missing neighbors

L_1^S : smooth L1 distance

$\tilde{x}_v^p$: the p-th predicted feature

x_u: the feature of a v's missing neighbor

Subgraph Federated Learning

➢ Missing Cross-Client Links

❑ FedSage+

- Directly averaging NeighGen weights across clients **hurts personality**
- Solution: Local NeighGen + Cross-Subgraph Feature Matching

Loss for fGen: $$\frac{1}{|\bar{V}_i|}\sum_{v\in\bar{V}_i}\sum_{p\in[\tilde{n}_v]}\left(\min_{u\in\mathcal{N}_{G_i}(v)\cap V_i^h}(||\tilde{x}_v^p - x_u||_2^2) + \alpha\sum_{j\in[M]/i}\boxed{\min_{u\in V_j}(||H_i^g(z_v)^p - x_u||_2^2)}\right)$$

- Find the closest node in client j , to allow each NeighGen i to generate diverse neighbors
- Client j computes gradients and share with client i to update H^g
- Ensures **privacy** + enables **federated learning of diverse NeighGens**

Subgraph Federated Learning

➢ Experiments

❑ Datasets

- Four real-world datasets: Cora, Citeseer, PubMed and MSAcademic
- Synthesize the distributed subgraph system with the **Louvain** algorithm

❑ Baselines

- GlobSage (upper bound): the GraphSage model trained on the original global graph
- LocSage: one GraphSage model trained solely on each subgraph
- LocSage+: the GraphSage model + NeighGen model jointly trained solely on each subgraph

❑ Metric

- Node classification accuracy

Subgraph Federated Learning

➢ Experiments

❑ Main Results

- FedSage and FedSage+ have the relatively similar accuracy as GlobSage
- FedSage and FedSage+ have stable performance

Model	Cora			Citesser		
	M=3	M=5	M=10	M=3	M=5	M=10
LocSage	0.5762 (±0.0302)	0.4431 (±0.0847)	0.2798 (±0.0080)	0.6789 (±0.054)	0.5612 (±0.086)	0.4240 (±0.0859)
LocSage+	0.5644 (±0.0219)	0.4533 (±0.047)	0.2851 (±0.0080)	0.6848 (±0.0517)	0.5676 (±0.0714)	0.4323 (±0.0715)
FedSage	0.8656 (±0.0043)	0.8645 (±0.0050)	0.8626 (±0.0103)	0.7241 (±0.0022)	0.7226 ±0.0066)	0.7158 (±0.0053)
FedSage+	**0.8686** (±0.0054)	**0.8648** (±0.0051)	**0.8632** (±0.0034)	**0.7454** (±0.0038)	**0.7440** (±0.0025)	**0.7392** (±0.0041)
GlobSage	0.8701 (±0.0042)			0.7561 (±0.0031)		

Model	PubMed			MSAcademic		
	M=3	M=5	M=10	M=3	M=5	M=10
LocSage	0.8447 (±0.0047)	0.8039 (±0.0337)	0.7148 (±0.0951)	0.8188 (±0.0331)	0.7426 (±0.0790)	0.5918 (±0.1005)
LocSage+	0.8481 (±0.0041)	0.8046 (±0.0318)	0.7039 (±0.0925)	0.8393 (±0.0330)	0.7480 (±0.0810)	0.5927 (±0.1094)
FedSage	0.8708 (±0.0014)	0.8696 (±0.0035)	0.8692 (±0.0010)	0.9327 (±0.0005)	0.9391 (±0.0007)	0.9262 (±0.0009)
FedSage+	**0.8775** (±0.0012)	**0.8755** (±0.0047)	**0.8749** (±0.0013)	**0.9359** (±0.0005)	**0.9414** (±0.0006)	**0.9314** (±0.0009)
GlobSage	0.8776(±0.0011)			0.9681(±0.0006)		

Subgraph Federated Learning

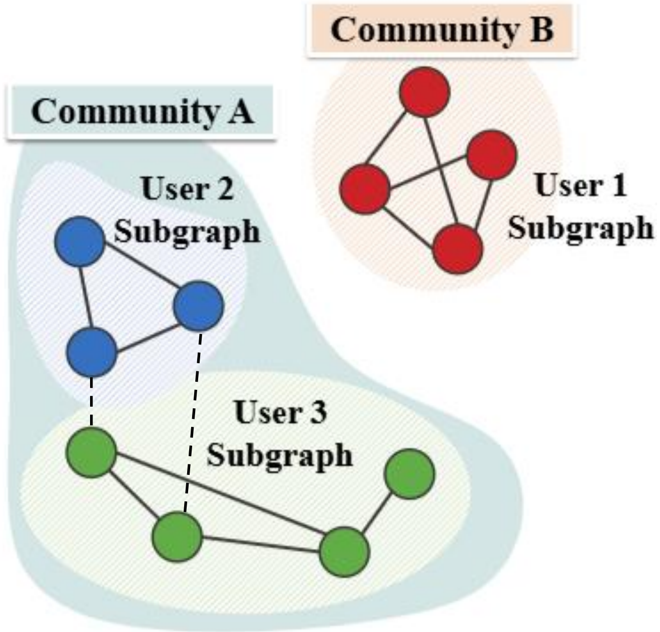
➢ Community Heterogeneity

❑ Heterogeneity of Subgraphs

- Subgraphs in different community can have **opposite properties**
- Naïvely aggregating all local models leads to **knowledge collapse**

❑ No Access to Subgraph Identities

- The server has **no visibility** into which client belongs to which community
- It's challenging to determine **which clients should share model parameters** or collaborate more closely

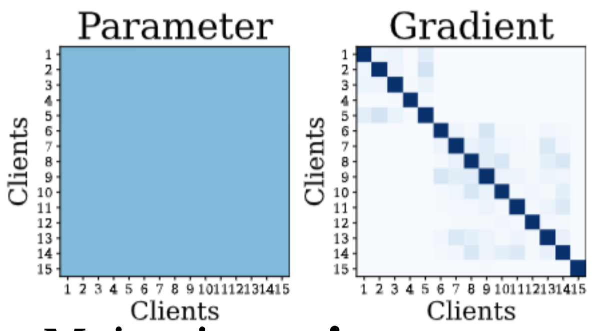


❑ Solution

- **FED-PUB:** Functional Similarity Matching + Personalized Parameter Masking

Subgraph Federated Learning

➢ Community Heterogeneity: FED-PUB



❑ Functional Embeddings for Subgraph Similarities

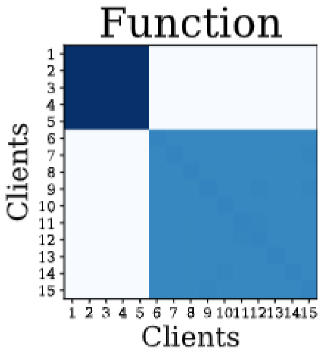
- Group clients with similar subgraphs (e.g., within the same community)
- Avoids **curse of dimensionality** + More **computationally efficient** + Maintains **privacy**

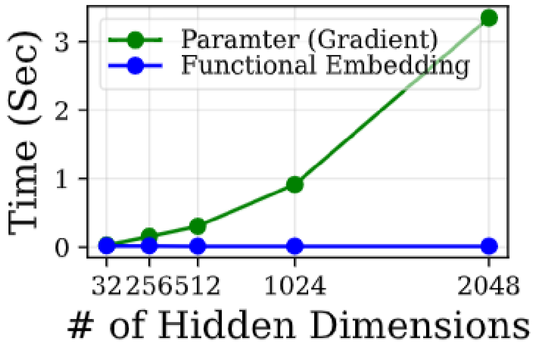
❑ Solution

- Measure **functional similarity** based on model outputs
- Use random graphs as shared GNN input and compare average embedding similarity

$$S(i,j) = \frac{\tilde{h}_i \cdot \tilde{\mathrm{h}}_j}{||\tilde{h}_i|| \cdot ||\tilde{\mathrm{h}}_j||}$$

$\tilde{h}_i$: averaged output of all node embeddings for random graph $\tilde{G}$





Subgraph Federated Learning

➢ Community Heterogeneity: FED-PUB

❑ Personalized Weight Aggregation

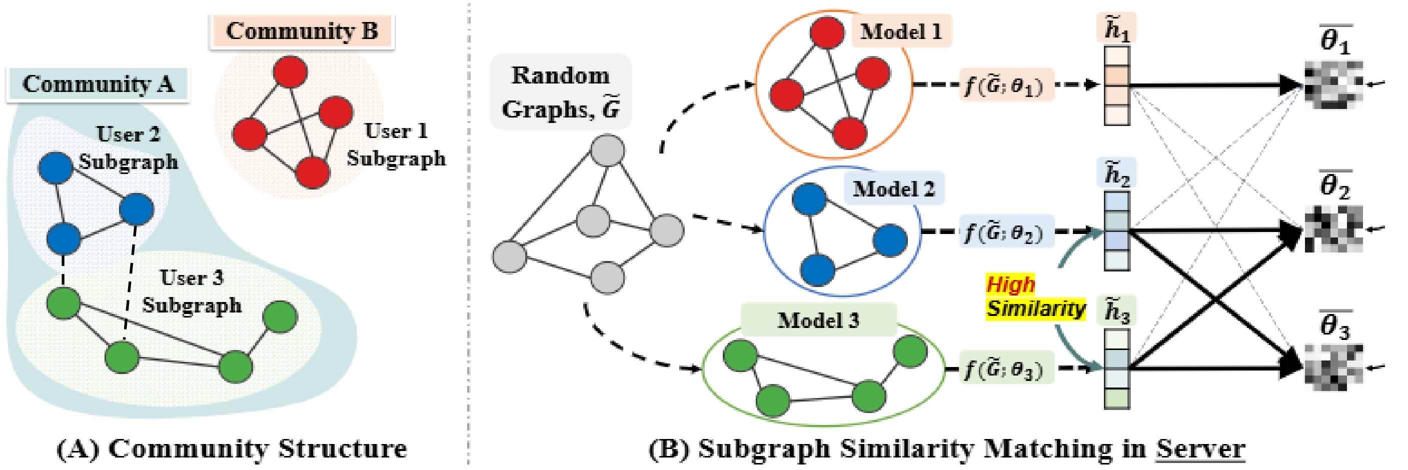
- Global model averaging can **collapse conflicting updates** from heterogeneous subgraphs
- Use functional similarity (via outputs on random graphs) to guide **personalized aggregation**

$$\bar{\boldsymbol{\theta}}_i \leftarrow \sum_{j=1}^{M} \alpha_{ij} \cdot \boldsymbol{\theta}_j, \qquad \alpha_{ij} = \frac{\exp(\tau \cdot S(i,j))}{\sum_k \exp(\tau \cdot S(i,k))}$$

$\bar{\boldsymbol{\theta}}_i$: aggregated personalized model weights

τ: hyperparameter for scaling

α_{ij}: normalized similarity between clients i and j



Subgraph Federated Learning

➢ Community Heterogeneity: FED-PUB

❑ Adaptive Weight Masking

- Even with functional similarity, scalar aggregation (α_{ij}) **can't tell which parameters are useful**
- Each client learns a sparse mask $\boldsymbol{\mu}_i$ for **fine-grained control**

$$\boldsymbol{\theta}_i = \boldsymbol{\mu}_i \circ \overline{\boldsymbol{\theta}}_i$$

- Final objective

$$\min_{\boldsymbol{\theta}_i, \boldsymbol{\mu}_i} \mathcal{L}(G_i; \boldsymbol{\theta}_i, \boldsymbol{\mu}_i) + \boxed{\lambda_1 ||\boldsymbol{\mu}_i||_1} + \boxed{\lambda_2 ||\boldsymbol{\theta}_i - \overline{\boldsymbol{\theta}}_i||_2^2}$$

Encourages sparsity

Prevent local overfitting

Subgraph Federated Learning

➢ Experiments

❑ Datasets

- **Citation graphs**: Cora, CiteSeer, Pubmed, ogbn-arxiv
- **Product graphs**: Amazon-Computer, Amazon-Photo
- Synthesize the distributed subgraph system with the **METIS** algorithm

❑ Baselines

- **Standard FL**: FedAvg, FedProx; **Personalized FL**: FedPer; **Subgraph FL**: FedGNN, FedSage+ **Graph-level FL**: GCFL; **Local**

❑ Metric

- Node classification accuracy

Subgraph Federated Learning

➢ Experiments

❑ Main Results

- FedSage+ fails due to **naive weight averaging** and ignoring community structure
- FedPer and GCFL alleviate knowledge collapse, but lack **community-aware aggregation**

Methods	Cora			CiteSeer			Pubmed			-
	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	-
Local	81.30 ± 0.21	79.94 ± 0.24	80.30 ± 0.25	69.02 ± 0.05	67.82 ± 0.13	65.98 ± 0.17	84.04 ± 0.18	82.81 ± 0.39	82.65 ± 0.03	-
FedAvg	74.45 ± 5.64	69.19 ± 0.67	69.50 ± 3.58	71.06 ± 0.60	63.61 ± 3.59	64.68 ± 1.83	79.40 ± 0.11	82.71 ± 0.29	80.97 ± 0.26	-
FedProx	72.03 ± 4.56	60.18 ± 7.04	48.22 ± 6.81	71.73 ± 1.11	63.33 ± 3.25	64.85 ± 1.35	79.45 ± 0.25	82.55 ± 0.24	80.50 ± 0.25	-
FedPer	81.68 ± 0.40	79.35 ± 0.04	78.01 ± 0.32	70.41 ± 0.32	70.53 ± 0.28	66.64 ± 0.27	85.80 ± 0.21	84.20 ± 0.28	84.72 ± 0.31	-
GCFL	81.47 ± 0.65	78.66 ± 0.27	79.21 ± 0.70	70.34 ± 0.57	69.01 ± 0.12	66.33 ± 0.05	85.14 ± 0.33	84.18 ± 0.19	83.94 ± 0.36	-
FedGNN	81.51 ± 0.68	70.12 ± 0.99	70.10 ± 3.52	69.06 ± 0.92	55.52 ± 3.17	52.23 ± 6.00	79.52 ± 0.23	83.25 ± 0.45	81.61 ± 0.59	-
FedSage+	72.97 ± 5.94	69.05 ± 1.59	57.97 ± 12.6	70.74 ± 0.69	65.63 ± 3.10	65.46 ± 0.74	79.57 ± 0.24	82.62 ± 0.31	80.82 ± 0.25	-
FED-PUB (Ours)	**83.70** ± 0.19	**81.54** ± 0.12	**81.75** ± 0.56	**72.68** ± 0.44	**72.35** ± 0.53	**67.62** ± 0.12	**86.79** ± 0.09	**86.28** ± 0.18	**85.53** ± 0.30	-

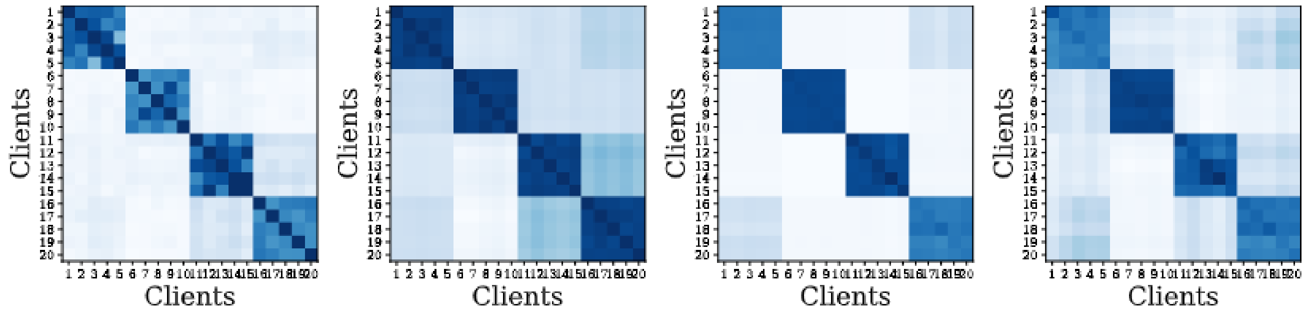
Methods	Amazon-Computer			Amazon-Photo			ogbn-arxiv			All
	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	Avg.
Local	89.22 ± 0.13	88.91 ± 0.17	89.52 ± 0.20	91.67 ± 0.09	91.80 ± 0.02	90.47 ± 0.15	66.76 ± 0.07	64.92 ± 0.09	65.06 ± 0.05	79.57
FedAvg	84.88 ± 1.96	79.54 ± 0.23	74.79 ± 0.24	89.89 ± 0.83	83.15 ± 3.71	81.35 ± 1.04	65.54 ± 0.07	64.44 ± 0.10	63.24 ± 0.13	74.58
FedProx	85.25 ± 1.27	83.81 ± 1.09	73.05 ± 1.30	90.38 ± 0.48	80.92 ± 4.64	82.32 ± 0.29	65.21 ± 0.20	64.37 ± 0.18	63.03 ± 0.04	72.84
FedPer	89.67 ± 0.34	89.73 ± 0.04	87.86 ± 0.43	91.44 ± 0.37	91.76 ± 0.23	90.59 ± 0.06	66.87 ± 0.05	64.99 ± 0.18	64.66 ± 0.11	79.94
GCFL	89.07 ± 0.91	90.03 ± 0.16	89.08 ± 0.25	91.99 ± 0.29	92.06 ± 0.25	90.79 ± 0.17	66.80 ± 0.12	65.09 ± 0.08	65.08 ± 0.04	79.90
FedGNN	88.08 ± 0.15	88.18 ± 0.41	83.16 ± 0.13	90.25 ± 0.70	87.12 ± 2.01	81.00 ± 4.48	65.47 ± 0.22	64.21 ± 0.32	63.80 ± 0.05	75.23
FedSage+	85.04 ± 0.61	80.50 ± 1.30	70.42 ± 0.85	90.77 ± 0.44	76.81 ± 8.24	80.58 ± 1.15	65.69 ± 0.09	64.52 ± 0.14	63.31 ± 0.20	73.47
FED-PUB (Ours)	**90.74** ± 0.05	**90.55** ± 0.13	**90.12** ± 0.09	**93.29** ± 0.19	**92.73** ± 0.18	**91.92** ± 0.12	**67.77** ± 0.09	**66.58** ± 0.08	**66.64** ± 0.12	**81.59**

Subgraph Federated Learning

➢ Experiments

❑ Ablation Study

- Functional embeddings are both **effective** and **privacy-preserving** for estimating subgraph similarity， outperforming parameter/gradient-based methods and matching the performance of privacy-sensitive label-based similarity



(a) Missing edges (b) Label similarity (c) Round at 5 (d) Round at 30

Figure 6: **Heatmaps of community structures** on overlapping node scenario with Cora (20 clients). Darker color indicates many missing edges between subgraphs (a) or high similarities in labels (b). (c) and (d) are functional similarities by FED-PUB.

Table 10: Results on varying the similarity calculation schemes: parameter, gradient, label, and our functional embedding, on the overlapping node scenario with 30 clients of the Cora dataset.

	Rounds			
Model	20	40	60	80
FedAvg	29.94	32.69	47.84	52.42
Parameter	29.94	35.89	47.03	52.28
Gradient	33.93	51.09	52.77	58.14
Label	65.97	74.31	76.50	76.82
Function (FED-PUB)	67.82	73.51	74.66	75.90

Subgraph Federated Learning

➢ References

❑ In this tutorial

- Zhang, Ke, et al. "Subgraph federated learning with missing neighbor generation.“ NeurIPS 2021.
- Baek, Jinheon, et al. "Personalized subgraph federated learning.“ ICML 2023.

❑ Related references

- Huang, Wenke, et al. "Federated graph semantic and structural learning." IJCAI 2023.
- Wan, Guancheng, et al. "Federated graph learning under domain shift with generalizable prototypes." AAAI 2024.
- Yu, Wentao, et al. "Modeling inter-intra heterogeneity for graph federated learning." AAAI 2025.

Outline

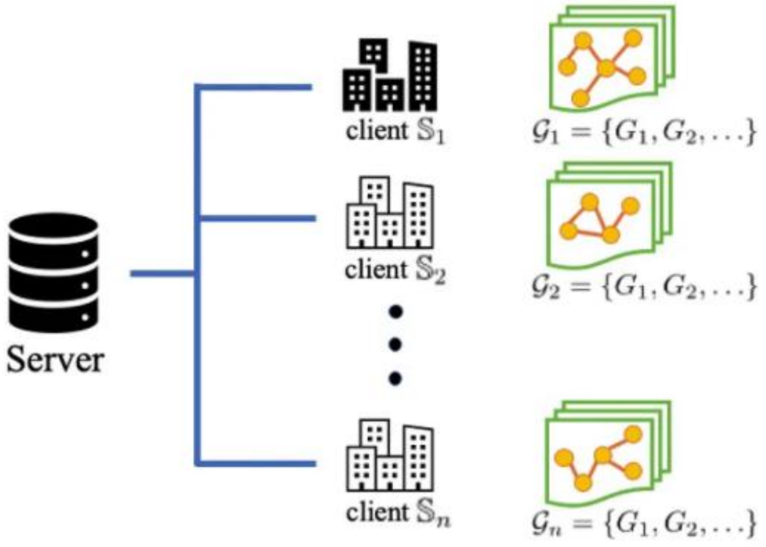
- ✓ Introduction
- ✓ Subgraph Federated Learning
- ✓ Federated Graph Learning with Non-IID Graphs
- ✓ Privacy-Preserving Federated Graph Learning
- ✓ Summary and Future Directions

Federated Graph Learning with Non-IID Graphs

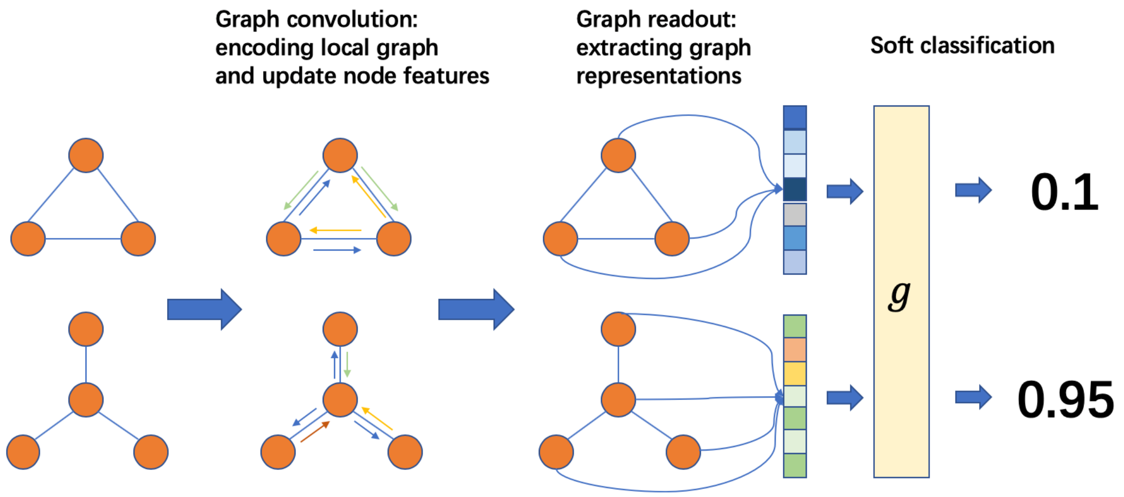
➢ Background

❑ Graph-level tasks in FGL

- Each client has multiple graphs (e.g. molecules, proteins, ……)
- The clients are interested in graph-level tasks (e.g., graph classification/regression)



Each client has multiple graphs



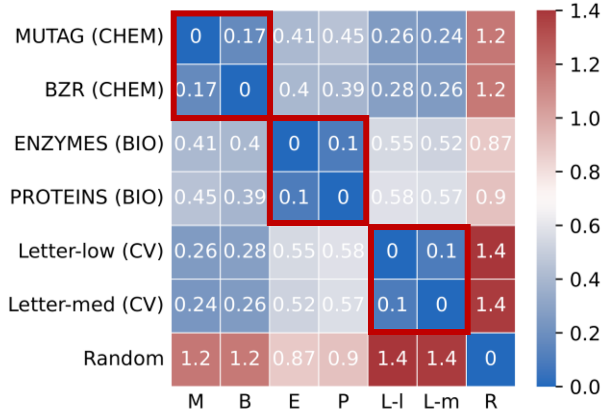
Graph Classification Process

Federated Graph Learning with Non-IID Graphs

➢ Challenges in FGL with Non-IID Graphs

❑ Cross-dataset structural knowledge sharing

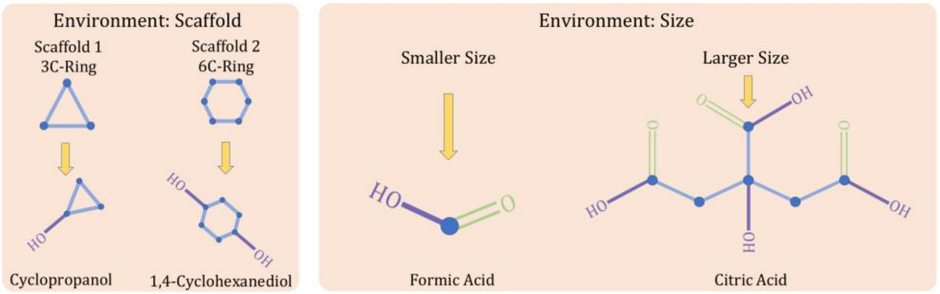
- Graph data from different datasets/domains may share common structural properties
- Sharing structural knowledge can enhance joint training



The JS divergence of degree distributions among six graph datasets and random graphs

❑ Distribution Shifts

- Graphs may be collected from different environments
- Toy example: graphs consisting of environment-invariant motifs and environment-varying bases
- Client-invariant causal subgraphs & client-varying non-causal subgraphs



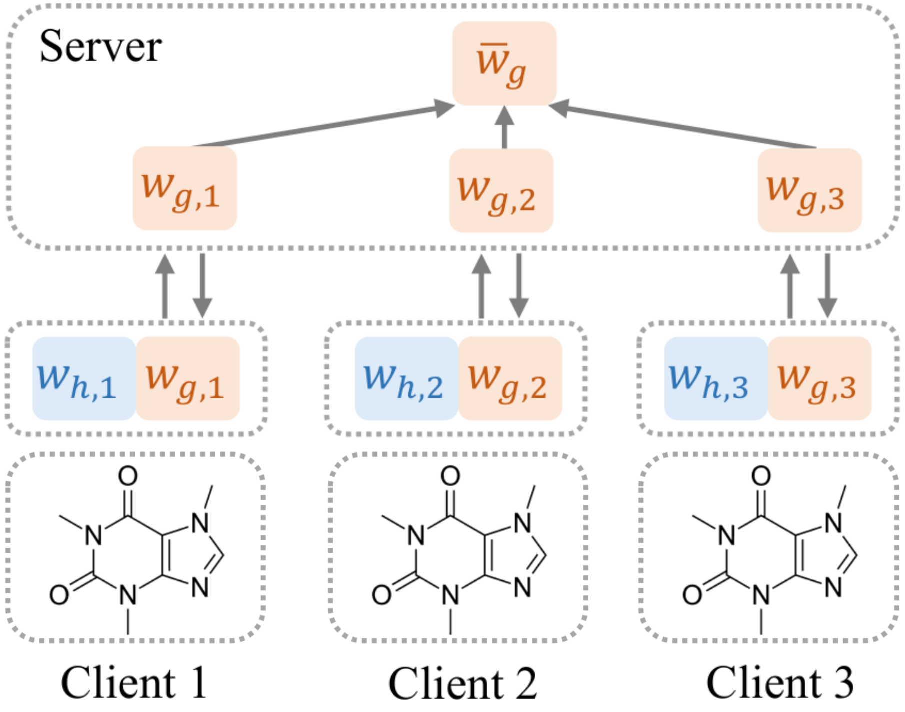
Molecular representation learning: graphs from environments by scaffold/size

Federated Graph Learning with Non-IID Graphs

➢ Cross-Dataset Structural Knowledge Sharing

❑ FedStar

- Share structure of graph data across homogeneous clients



$w_{h,i}$ **Feature encoder**

- Personalized model
- Trained locally

$w_{g,i}$ **Structure encoder**

- Global model
- Aggregated in the server

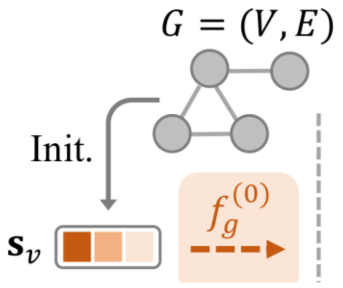
Federated Graph Learning with Non-IID Graphs

➢ Cross-Dataset Structural Knowledge Sharing

❑ FedStar: Structure Encoding

- Intuition: incorporates both local and global structural information

$$\mathrm{s}_v = \mathrm{concat}\left[\mathrm{s}_v^{\mathrm{DSE}}; \mathrm{s}_v^{\mathrm{RWSE}}\right]$$



- $\mathrm{s}_v^{\mathrm{DSE}}$: degree-based structure embedding (DSE)

$$\mathrm{s}_v^{\mathrm{DSE}} = [\mathbb{I}(d_v = 1), \mathbb{I}(d_v = 2), \cdots, \mathbb{I}(d_v \geq k_1)] \in \mathbb{R}^{k_1}$$

- $\mathrm{s}_v^{\mathrm{RWSE}}$: random walk-based structure embedding (RWSE)

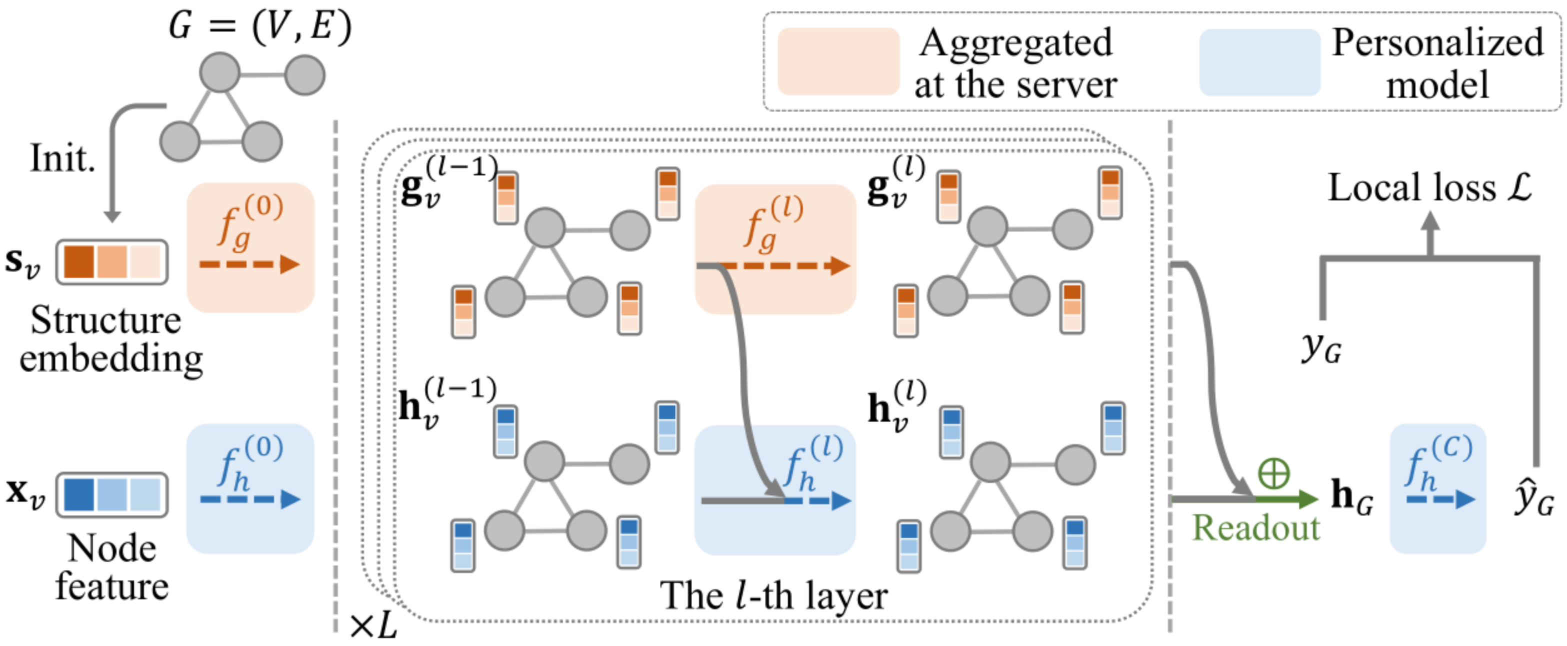
$$\mathrm{s}_v^{\mathrm{RWSE}} = [\mathrm{T}_{ii}, \mathrm{T}_{ii}^2, \cdots, \mathrm{T}_{ii}^{k_2}] \in \mathbb{R}^{k_2}$$

$\mathrm{T} = \mathrm{AD}^{-1}$ is a random walk transition matrix

Federated Graph Learning with Non-IID Graphs

➢ Cross-Dataset Structural Knowledge Sharing

❑ FedStar: Feature-Structure Decoupled GNN



Feature-Structure Decoupled GNN in FedStar

Federated Graph Learning with Non-IID Graphs

➢ Cross-Dataset Structural Knowledge Sharing

❑ FedStar: Structural Knowledge Sharing

- Intuition: share the learned structure encoders across clients
- Share $w_{g,m}$ with the FL framework while keeping $w_{h,m}$ being trained locally
- A global structure encoder w_g and personalized feature encoders $w_{h,m}$

$$\bar{w}_g = \sum_{m=1}^{M} \frac{|D_m|}{N} w_{g,m}$$

Federated Graph Learning with Non-IID Graphs

➢ Cross-Dataset Structural Knowledge Sharing

❑ Experiments

- Datasets

Molecules (CHEM)	Bioinformatics (BIO)	Social Networks (SN)	Computer Vision
MUTAG, PTC MR, COX2, DHFR, AIDS, NCI1, BZR	ENZYMES, DD, PROTEINS	COLLAB, IMDB-BINARY, IMDB-MULTI	Letter-low, Letter-high, Letter-med

- Backbone models: a three-layer GIN as the feature encoder, a three-layer GCN as the structure encoder
- Baselines: Local, FedAvg, FedProx, FedPer, FedSage, GCFL

Federated Graph Learning with Non-IID Graphs

➢ Cross-Dataset Structural Knowledge Sharing

❑ Experiments

- Main results

Setting (# domains)	CHEM(1)		BIO-CHEM(2)		BIO-CHEM-SN(3)		BIO-SN-CV(3)	
# datasets	7		10		13		9	
Accuracy	avg.	avg. gain	avg.	avg. gain	avg.	avg. gain	avg.	avg. gain
Local	75.38±2.26	-	71.09±1.21	-	69.37±3.05	-	66.91±2.84	-
FedAvg	75.26±2.00	-0.13	70.65±2.73	-0.44	68.92±2.12	-0.45	64.86±2.73	-2.05
FedProx	75.30±2.00	-0.08	70.75±2.26	-0.34	69.21±2.63	-0.16	65.18±2.01	-1.72
FedPer	77.09±3.36	1.70	71.97±1.97	0.88	69.37±2.92	-0.01	62.23±3.76	-4.67
FedSage	75.90±1.85	0.51	70.34±1.87	-0.74	69.55±2.15	0.18	67.95±1.87	1.04
GCFL	76.49±1.23	1.11	71.60±2.20	0.51	70.65±1.84	1.28	66.31±2.36	-0.60
FedStar (Ours)	**79.79**±2.44	**4.41**	**74.54**±2.50	**3.46**	**72.16**±2.43	**2.78**	**69.49**±1.81	**2.58**

Federated Graph Learning with Non-IID Graphs

➢ Cross-Dataset Structural Knowledge Sharing

❑ Experiments

- Analysis of decoupling and sharing mechanisms

Sharing	DC	Setting (# domains)		
		BIO-CHEM(2)	BIO-CHEM-SN(3)	BIO-SN-CV(3)
All	-	70.86±2.25	69.32±2.42	65.23±2.52
None	-	71.59±1.93	69.42±3.06	68.17±3.04
All	✓	71.97±2.14	69.85±2.43	65.78±4.25
None	✓	74.08±2.45	71.30±1.89	68.76±2.24
FE	✓	71.00±3.51	68.53±2.74	64.14±2.73
SE(Ours)	✓	**74.54±2.50**	**72.15±2.43**	**69.49±1.81**

Federated Graph Learning with Non-IID Graphs

➢ Cross-Dataset Structural Knowledge Sharing

❑ Experiments

- Analysis of varying structure embeddings

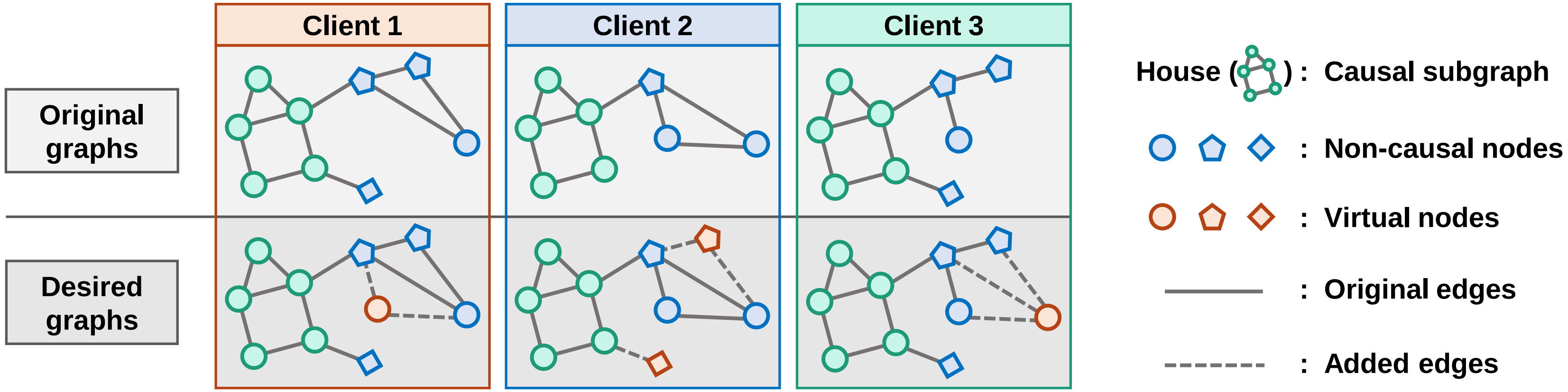
DSE	RWSE	Setting (# domains)		
		BIO-CHEM(2)	BIO-CHEM-SN(3)	BIO-SN-CV(3)
-	-	69.51±2.25	69.64±1.92	66.05±2.92
✓	-	74.42±3.15	72.05±2.82	69.25±2.41
-	✓	72.74±3.44	70.48±3.37	67.23±2.74
✓	✓	**74.54±2.50**	**72.15±2.43**	**69.49±1.81**

Federated Graph Learning with Non-IID Graphs

➢ Distribution Shifts

❑ Can We Train GNN Models over Identical Graphs?

- Original graphs → desired graphs

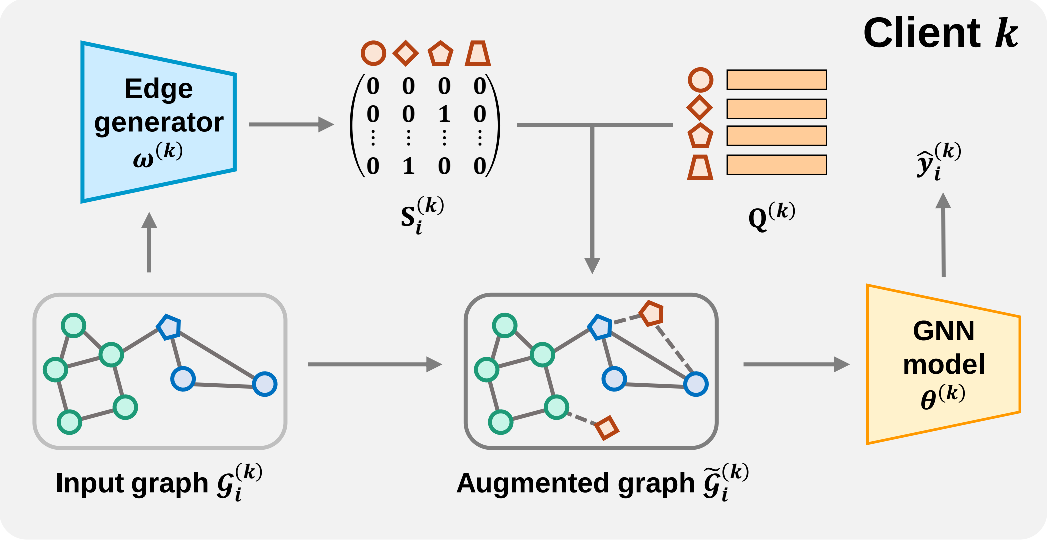


Federated Graph Learning with Non-IID Graphs

➢ Distribution Shifts

❑ Train GNN models over augmented graphs with virtual nodes

$$\mathcal{L}_S^{(k)} = \frac{1}{N^{(k)}} \sum_{i=1}^{N^{(k)}} \ell\left(f\left(\tilde{\mathcal{G}}_i^{(k)}; \theta\right), y_i^{(k)}\right)$$



❑ Personalized graph augmentation

- Added virtual node features: $\mathbf{Q}^{(k)} \in \mathbb{R}^{M \times d_x}$
- How they connect original graphs: $\mathbf{S}_i^{(k)} \in \mathbb{R}^{\left|\mathcal{V}_i^{(k)}\right| \times M}$

❑ Message passing

- Graph nodes: $\mathbf{h}_v^{(l)} = \mathrm{COMB}_{\mathrm{gn}}^{(l)}\left(\mathbf{h}_v^{(l-1)}, \mathrm{AGG}_{\mathrm{gn}}^{(l)}\left(\left\{\mathbf{h}_u^{(l-1)}: u \in \mathcal{N}(v)\right\}\right)\right) + \sum_{m=1}^{M} s_{v,m} \cdot \mathbf{h}_m^{(l-1)}$
- Virtual nodes: $\mathbf{h}_m^{(l)} = \mathrm{COMB}_{\mathrm{vn}}^{(l)}\left(\mathbf{h}_m^{(l-1)}, \mathrm{AGG}_{\mathrm{vn}}^{(l)}\left(\left\{s_{v,m} \cdot \mathbf{h}_v^{(l-1)}: v \in \mathcal{V}_i^{(k)}\right\}\right)\right)$

Federated Graph Learning with Non-IID Graphs

➢ Distribution Shifts

❑ Virtual node may collapse to fewer virtual nodes

- Decoupling loss

$$\mathcal{L}_V^{(k)} = \frac{1}{M^2}\|\Sigma\|_F^2$$

- Σ: the correlation matrix of $\mathbf{Q}$

❑ Similar intra-client edge patterns & dissimilar inter-client edge patterns

- Score-contrastive loss

$$\mathcal{L}_E^{(k)} = -\frac{1}{N^{(k)}}\sum_{i=1}^{N^{(k)}} \log \frac{e^{\mathrm{sim}\left(\tilde{\mathbf{s}}_i^{(k)}, \mathbf{s}_{local}^{(k)}\right)/\tau}}{e^{\mathrm{sim}\left(\tilde{\mathbf{s}}_i^{(k)}, \mathbf{s}_{local}^{(k)}\right)/\tau} + e^{\mathrm{sim}\left(\tilde{\mathbf{s}}_i^{(k)}, \mathbf{s}_{global}\right)/\tau}}$$

$$\tilde{\mathbf{s}}_i^{(k)} = \sum_{v\in\mathcal{V}_i^{(k)}} \mathbf{s}_v$$

$$\mathbf{s}_{local}^{(k)} = \frac{1}{N^{(k)}}\sum_{i=1}^{N^{(k)}} \tilde{\mathbf{s}}_i^{(k)}$$

$$\mathbf{s}_{global} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{s}_{local}^{(k)}$$

❑ Final objective function

$$\min_{\theta,\mathbf{Q},\omega^{(k)}} \mathcal{L}_S^{(k)} + \lambda_1\mathcal{L}_V^{(k)} + \lambda_2\mathcal{L}_E^{(k)}$$

Federated Graph Learning with Non-IID Graphs

➢ Distribution Shifts

❑ Local training

- Step 1: fix θ and $\mathbf{Q}$, update $\omega^{(k)}$ by

$$\omega^{(k)} \leftarrow \omega^{(k)} - \eta_{\omega} \nabla_{\omega} \left(\mathcal{L}_S^{(k)} + \lambda_2 \mathcal{L}_E^{(k)} \right)$$

- Step 2: fix $\omega^{(k)}$, update $\theta^{(k)}$ and $\mathbf{Q}^{(k)}$ by

$$\theta^{(k)} \leftarrow \theta^{(k)} - \eta_{\theta} \nabla_{\theta} \mathcal{L}_S^{(k)}$$

$$\mathbf{Q}^{(k)} \leftarrow \mathbf{Q}^{(k)} - \eta_{\mathbf{Q}} \nabla_{\mathbf{Q}} \left(\mathcal{L}_S^{(k)} + \lambda_1 \mathcal{L}_V^{(k)} \right)$$

❑ Global update

$$\theta = \sum_{k=1}^{K} \frac{N^{(k)}}{N} \theta^{(k)}, \qquad \mathbf{Q} = \sum_{k=1}^{K} \frac{N^{(k)}}{N} \mathbf{Q}^{(k)}$$

Federated Graph Learning with Non-IID Graphs

➢ Distribution Shifts

❑ Experiments

- Datasets: four datasets under five settings adapted from GOOD[1]

Dataset	Motif		CMNIST	ZINC	SST2
	Basis	Size	Color	Scaffold	Length
Data type	Synthetic		Synthetic	Molecule	Sentence
#(Clients)	5	5	5	10	7
#(Graphs)/client	1,000	1,000	1,000	1,000	1,000
Task	Classification		Classification	Regression	Classification
Metric	Accuracy		Accuracy	MAE	Accuracy

- GNN backbones: A three-layer GIN as the encoder and a two-layer MLP as the prediction head
- Baselines: Self-training, FedAvg, FedProx, FedBN, Ditto, FedRep, FedALA, GCFL+, FedStar
- Hyperparameters: GNN hidden size=100

[1] Gui, Shurui, et al. "Good: A graph out-of-distribution benchmark." NeurIPS 2022.

Federated Graph Learning with Non-IID Graphs

➢ Distribution Shifts

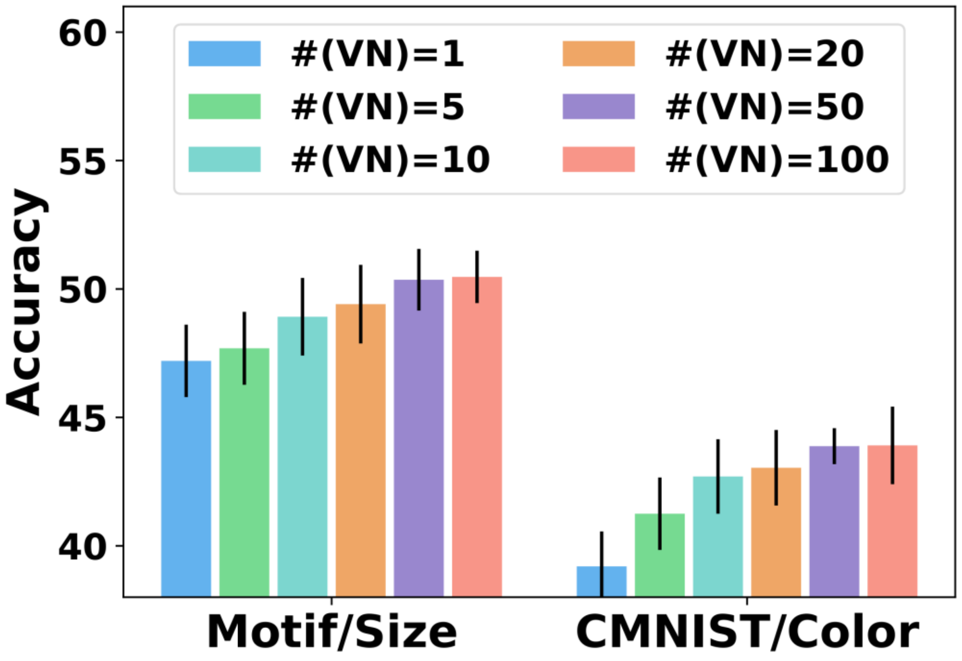
❑ Performance comparison

Dataset	Motif		CMNIST	ZINC	SST2
Metric	Accuracy ↑		Accuracy ↑	MAE ↓	Accuracy ↑
Partition setting	Basis	Size	Color	Scaffold	Length
Self-training	67.12±0.89	47.60±2.32	39.38±0.90	0.5442±0.0146	80.54±0.67
FedAvg	58.70±2.39	47.82±3.16	39.18±0.92	0.6235±0.0158	81.79±0.27
FedProx	57.90±1.36	47.88±4.08	39.78±0.68	0.6235±0.0165	81.74±0.33
FedBN	58.44±1.33	47.54±2.66	39.26±0.76	0.5129±0.0119	81.73±0.35
Ditto	63.38±0.89	47.48±3.20	39.00±0.94	0.5471±0.0146	81.69±0.67
FedRep	59.20±2.83	45.48±0.86	36.78±0.67	0.5220±0.0110	74.77±2.84
FedALA	59.92±1.14	48.52±3.34	39.22±1.12	0.5837±0.0159	81.77±0.61
GCFL+	57.36±2.00	49.34±2.70	38.82±1.11	0.6224±0.0147	81.39±0.45
FedStar	63.62±4.85	45.68±2.11	28.10±1.17	0.5963±0.0163	58.57±1.25
FedVN (Ours)	**75.72±1.85**	**50.41±1.17**	**43.67±1.25**	**0.4947±0.0174**	**83.13±0.79**

Federated Graph Learning with Non-IID Graphs

➢ Distribution Shifts

❑ Influence of VN numbers

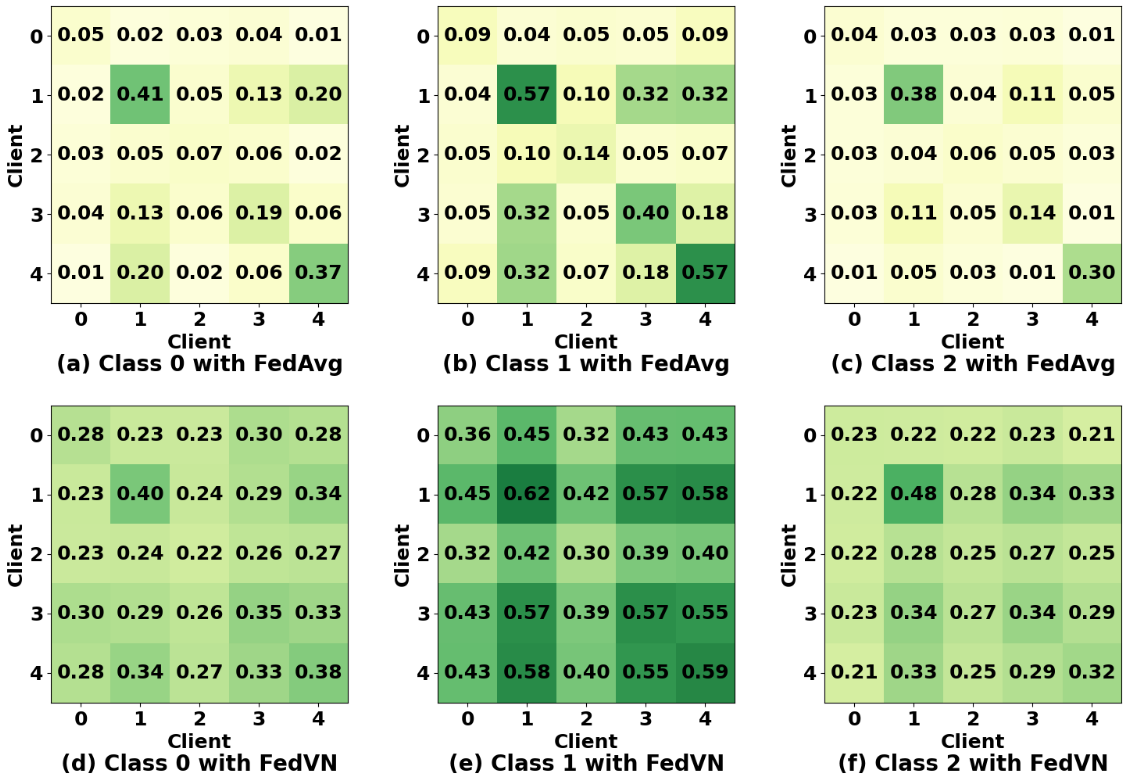


Performance of FedVN with different numbers of VNs

Federated Graph Learning with Non-IID Graphs

➢ Distribution Shifts

❑ Visualization of distribution shifts in FedAvg and FedVN



Cross-client cosine similarities of graph embeddings in each client on Motif/Basis

Federated Graph Learning with Non-IID Graphs

➢ References

❑ In this tutorial

- Tan, Yue, et al. "Federated learning on non-iid graphs via structural knowledge sharing." AAAI 2023.
- Fu, Xingbo, et al. "Virtual nodes can help: tackling distribution shifts in federated graph learning." AAAI 2025.

❑ Related references

- Tan, Zihan, et al. "FedSSP: federated graph learning with spectral knowledge and personalized preference." NeurIPS 2024.
- Wan, Guancheng, et al. "Federated graph learning under domain shift with generalizable prototypes." AAAI 2024.

Outline

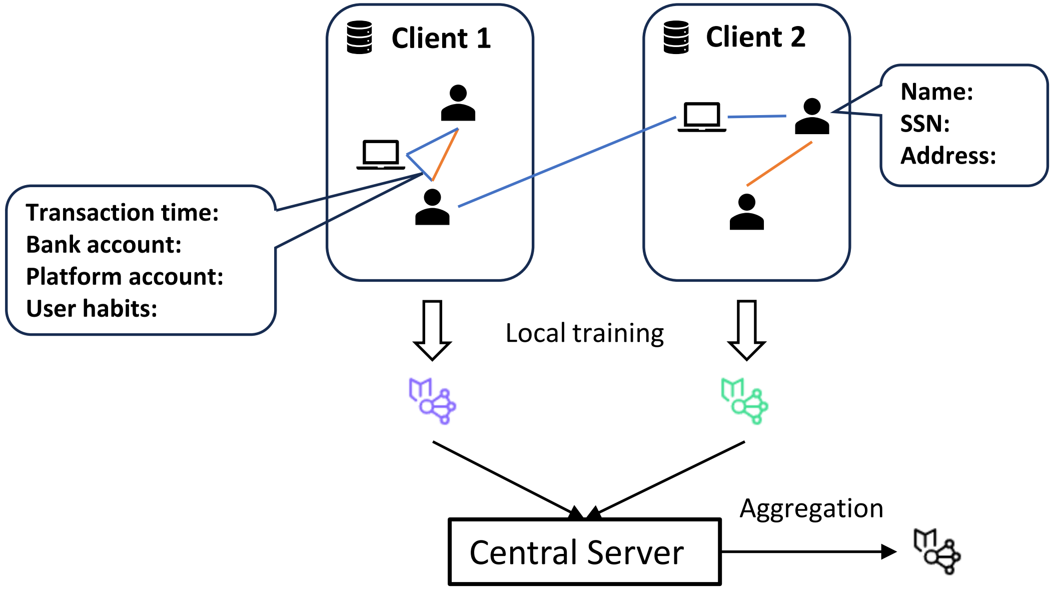
- ✓ Introduction
- ✓ Subgraph Federated Learning
- ✓ Federated Graph Learning with Non-IID Graphs
- ✓ Privacy-Preserving Federated Graph Learning
- ✓ Summary and Future Directions

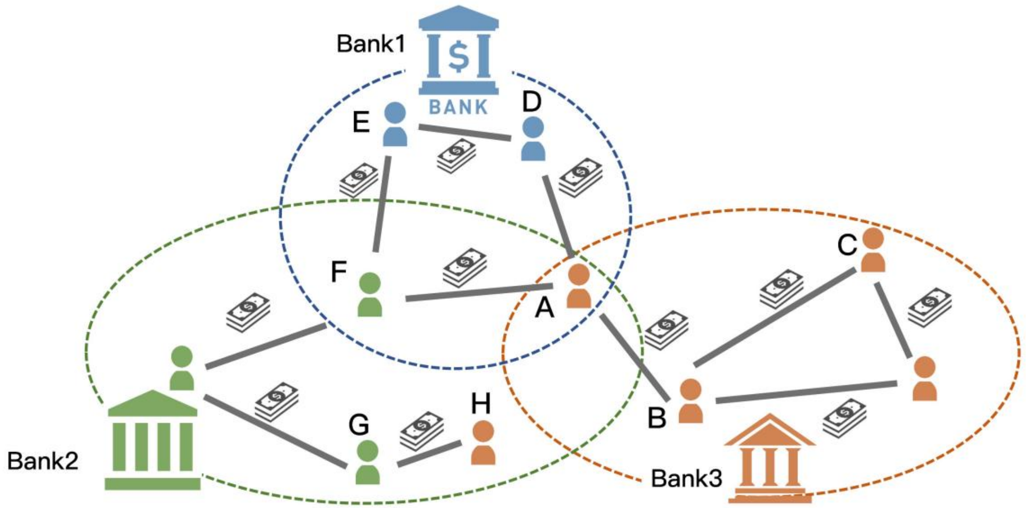
Privacy-Preserving Federated Graph Learning

➢ Background

❑ Private information in graph data

- Local information: graph structure and node features contain sensitive information
- Cross-client interactions





Privacy-Preserving Federated Graph Learning

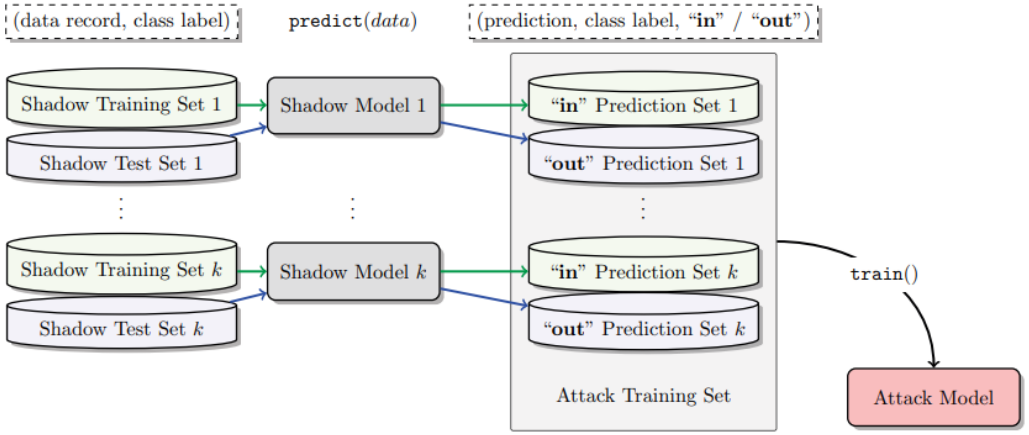
➢ Challenges in Privacy-Preserving FGL

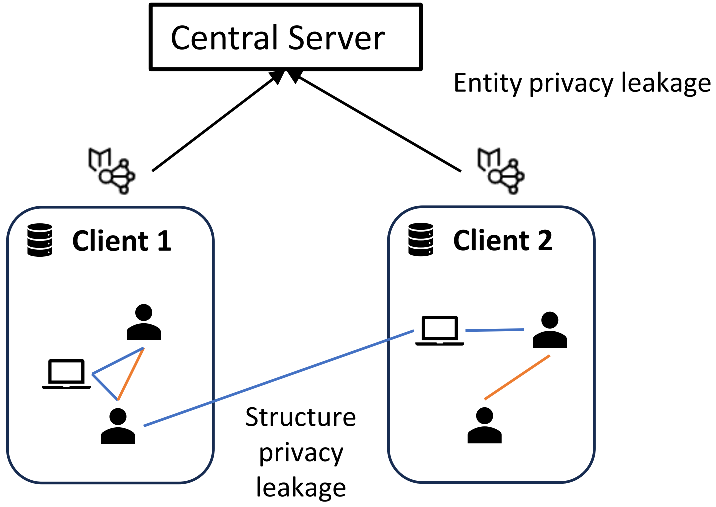
❑ Entity-level privacy protection (leakage from model updates)

- Entity feature inference
- Entity membership inference

❑ Structure-level privacy protection (leakage from graph structures)

- Cross-client neighbor leakage
- Boundary nodes leakage





Privacy-Preserving Federated Graph Learning

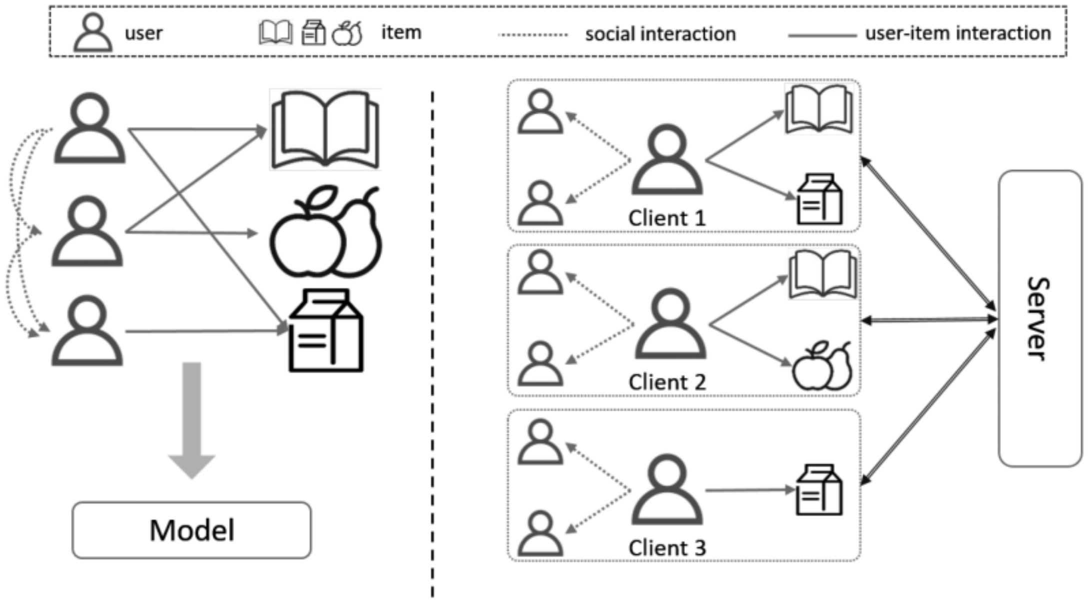
➢ Entity-Level Privacy Protection

❑ Hide local private information

- Differential privacy (DP) / Local differential privacy (LDP)
- Entity anonymization

❑ Application

- Recommendation systems
- Solution: FeSoG



(a) Centralized Learning (b) Federated Learning

Privacy-Preserving Federated Graph Learning

➢ FeSoG

❑ Federated Social recommendation with Graph neural network

- Social recommendation: Given user set U, item set T, rating matrix R, and social connection matrix S, complete the ratings of users to items.

Definition 1 (Client). A client c is defined as a local device storing the rating data and the social data. Each client c_n is associated with a user n, whose rating data and social data are $\mathbf{R}_{n\cdot}$ and $\mathbf{S}_{n\cdot}$, respectively.

Definition 2 (Server). A server is defined as a central device managing the coordination of multiple clients in training a model. It does not exchange raw data from clients but only requests necessary messages for updating the model.

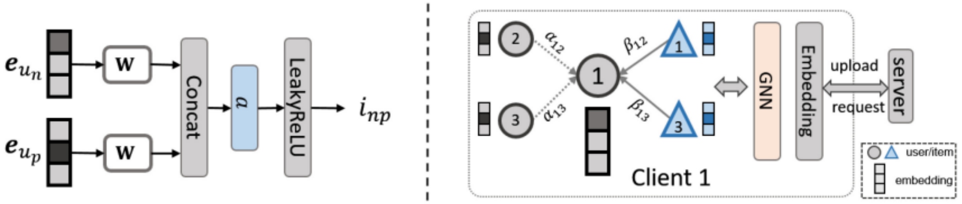
Definition 4 (Problem Definition). Given the local graphs $\{\mathcal{G}_n|_{n=1}^{N}\}$, can we collaboratively train a model to predict the attribute value for an unobserved edges (u_n, t^*) without access to the raw data of any local graphs?

Privacy-Preserving Federated Graph Learning

➢ FeSoG

❑ User and item embeddings

- Held by the central server
- Updated by aggregating the gradients from clients



(a) Attention Weights (b) Local Graph Neural Network

❑ Local GNN

- Relational GAT

User-user $o_{np} = \mathrm{LeakyReLU}\left(\mathbf{a}^\top\left[\mathbf{W}_1\mathbf{e}_{u_n}\|\mathbf{W}_1\mathbf{e}_{u_p}\right]\right),$

User-item $v_{nk} = \mathrm{LeakyReLU}\left(\mathbf{b}^\top\left[\mathbf{W}_2\mathbf{e}_{u_n}\|\mathbf{W}_2\mathbf{e}_{i_k}\right]\right),$

$$\alpha_{np} = \mathrm{softmax}_p(o_{np}) = \frac{\exp(o_{np})}{\sum_{i=1}^{P}\exp(o_{ni})},$$

$$\beta_{nk} = \mathrm{softmax}_k(v_{nk}) = \frac{\exp(v_{ni})}{\sum_{i=1}^{K}\exp(v_{ni})},$$

Neighbor aggr:

$$\mathbf{h}_u^{(n)} = \sum_{p=1}^{P}\alpha_{np}\mathbf{W}_h\mathbf{e}_{u_p}, \quad \mathbf{h}_t^{(n)} = \sum_{k=1}^{K}\beta_{nk}\mathbf{W}_h\mathbf{e}_{t_k},$$

Relational aggr:

User $$\gamma_u = \frac{\exp\left(\mathbf{c}^\top\left[\mathbf{h}_u^{(n)}\|\mathbf{v}_u\right]\right)}{\exp\left(\mathbf{c}^\top\left[\mathbf{h}_u^{(n)}\|\mathbf{v}_u\right]\right)+\exp\left(\mathbf{c}^\top\left[\mathbf{h}_t^{(n)}\|\mathbf{v}_t\right]\right)+\exp\left(\mathbf{c}^\top\left[\mathbf{h}_s^{(n)}\|\mathbf{v}_s\right]\right)},$$

Item $$\gamma_t = \frac{\exp\left(\mathbf{c}^\top\left[\mathbf{h}_t^{(n)}\|\mathbf{v}_t\right]\right)}{\exp\left(\mathbf{c}^\top\left[\mathbf{h}_u^{(n)}\|\mathbf{v}_u\right]\right)+\exp\left(\mathbf{c}^\top\left[\mathbf{h}_t^{(n)}\|\mathbf{v}_t\right]\right)+\exp\left(\mathbf{c}^\top\left[\mathbf{h}_s^{(n)}\|\mathbf{v}_s\right]\right)},$$

Self $$\gamma_s = \frac{\exp\left(\mathbf{c}^\top\left[\mathbf{h}_s^{(n)}\|\mathbf{v}_s\right]\right)}{\exp\left(\mathbf{c}^\top\left[\mathbf{h}_u^{(n)}\|\mathbf{v}_u\right]\right)+\exp\left(\mathbf{c}^\top\left[\mathbf{h}_t^{(n)}\|\mathbf{v}_t\right]\right)+\exp\left(\mathbf{c}^\top\left[\mathbf{h}_s^{(n)}\|\mathbf{v}_s\right]\right)},$$

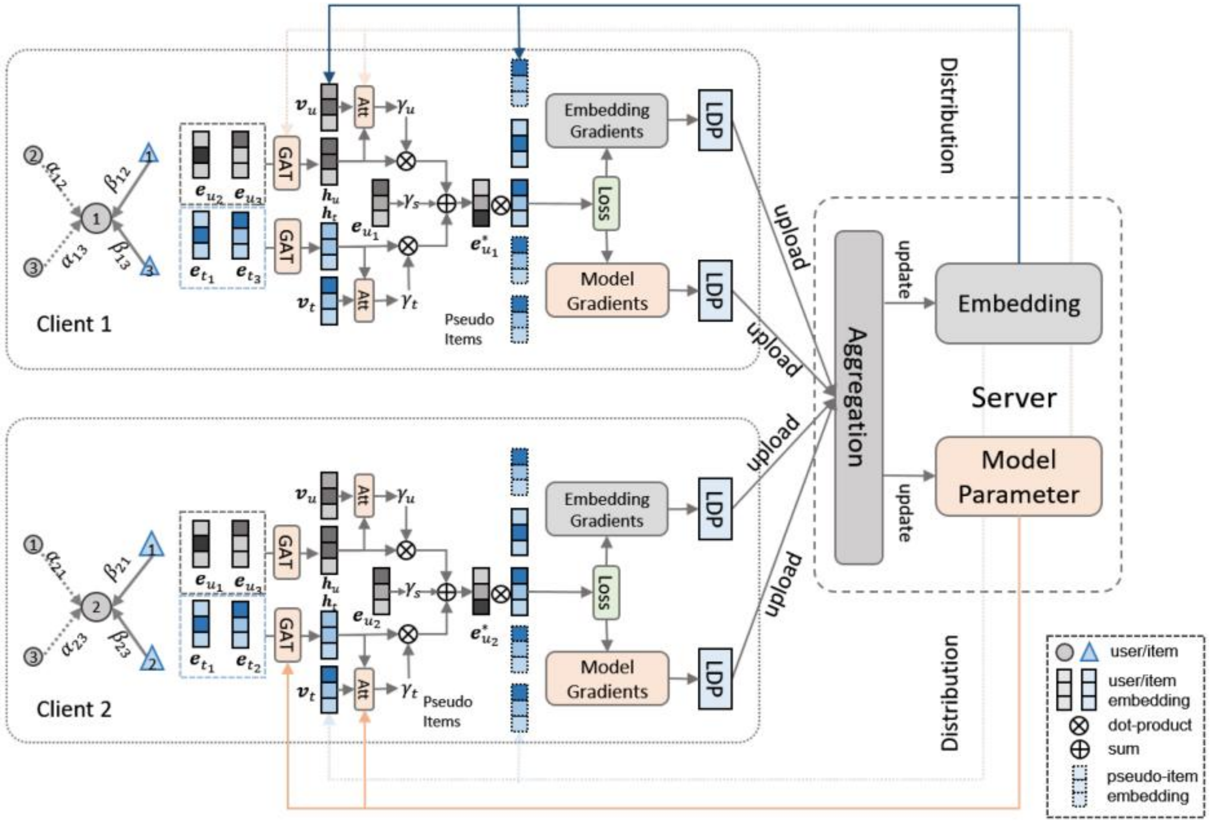
➢ FeSoG

❑ Privacy protection

- LDP

$$\tilde{g}^{(n)} = \text{clip}(g^{(n)}, \delta) + \text{Laplacian}(0, \lambda \cdot \text{mean}(g^{(n)})),$$

- Pseudo-item sampling
  - Sample q non-neighbor items as pseudo items
  - Compute ratings using the local GNN
  - Use rounded ratings as the labels for pseudo items



➢ FeSoG

❑ Experimental results

- FeSoG outperforms the SOTA federated recommender systems
- GNN-based models outperform MF-based models
- Federated learning impairs the performance compared with centralized learning

Table 4. Experiment Results Compared with Baseline Methods

Method	Ciao		Epinions		Filmtrust	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
SoRec	1.2024	0.8693	1.3389	1.0618	1.8094	1.4529
SoReg	1.0066	0.7595	1.0751	0.8309	1.7950	1.4413
SocialMF	1.0013	0.7535	1.0706	0.8264	1.8077	1.4557
GCMC+SN	1.0301	0.7970	1.1070	0.8480	1.8025	1.4325
GraphRec	1.0040	0.7591	1.0799	0.8219	<u>1.6775</u>	1.3194
CUNE	1.0002	0.7591	1.0681	0.8284	1.7675	1.4178
ConsisRec	<u>0.9722</u>	<u>0.7394</u>	<u>1.0495</u>	<u>0.8046</u>	1.7148	<u>1.3093</u>
FedMF	2.4216	2.0792	2.0685	1.5254	2.795	2.1713
FedGNN	2.02	1.58	1.8346	1.4238	2.13	1.65
FeSoG	**1.9136**	**1.4937**	**1.7969**	**1.3847**	**2.0942**	**1.5855**
Improvement	5.26%	5.46%	2.05%	2.74%	1.68%	3.9%

The best federated learning results are in bold, and the best results for non-federated learning methods are underlined. Improvement indicates the percent that FeSoG improves against the second-best federated learning result.

Matrix Factorization

— SoRec [32]: It co-factorizes user-item rating matrix and user-user social matrix.
— SoReg [33]: It develops a social regularization with social links to regularize on MF.
— SocialMF [18]: Compared with SoReg, social MF also considers social trust propagation.
— CUNE [64]: Collaborative user network embedding assumes users hold implicit social links from each other, and it tries to extract semantic and reliable social information by graph embedding method.

Graph Neural Network

— GCMC+SN [2]: GCMC is a GNN-based method. User nodes are initialized as vectors learned by node2vec [12] from the social graph to obtain social information. The dense representation learned upon the social graph can include more information than the random initialized feature.
— GraphRec [9]: Graph recommendation uses GNN to learn user embedding and item embedding from their neighbors and uses several fully connected layers as the rating predictor.
— ConsisRec [61]: It is the **state-of-the-art** (**SotA**) method in social recommendation. ConsisRec modifies GNN to mitigate the inconsistency problems in social recommendation.

Federated Learning

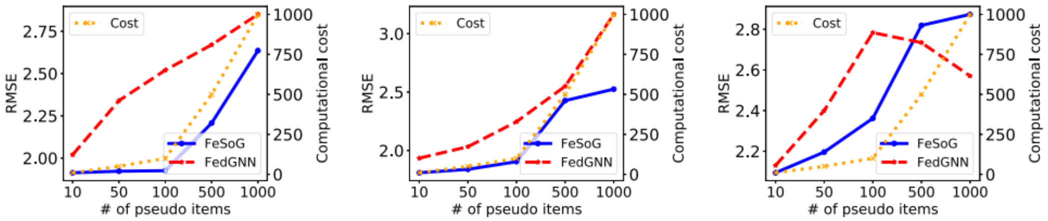
— FedMF [4]: It separates the MF computation to different users and uses an encryption method to avoid information leakage.
— FedGNN [53]: Federated GNN is the SotA federated recommendation method. It adopts local differential privacy methods to protect user's interaction with items.

Privacy-Preserving Federated Graph Learning

➢ FeSoG

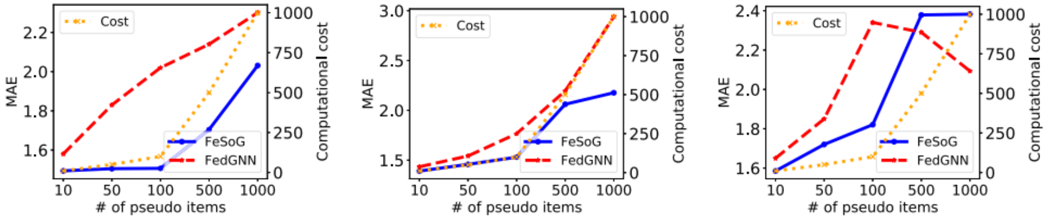
❑ Experimental results

- FeSoG outperforms the SOTA federated recommender systems
- If increasing # pseudo items, the error value increases for both methods
- If increasing # pseudo items, the extra computational cost increases linearly



(a) Ciao (b) Epinions (c) Filmtrust

Fig. 6. RMSE performance with respect to different pseudo item numbers on three datasets.



(a) Ciao (b) Epinions (c) Filmtrust

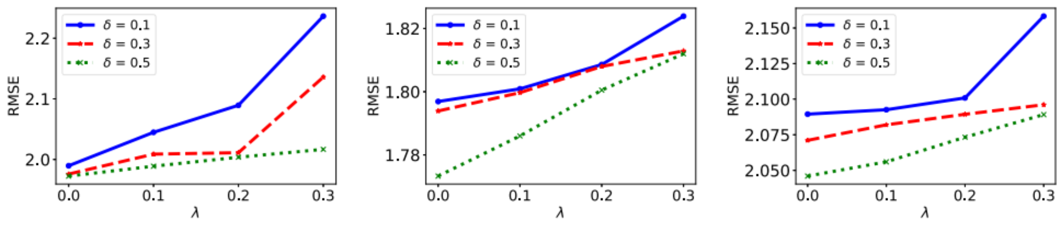
Fig. 7. MAE performance with respect to different pseudo item numbers on three datasets.

Privacy-Preserving Federated Graph Learning

➢ FeSoG

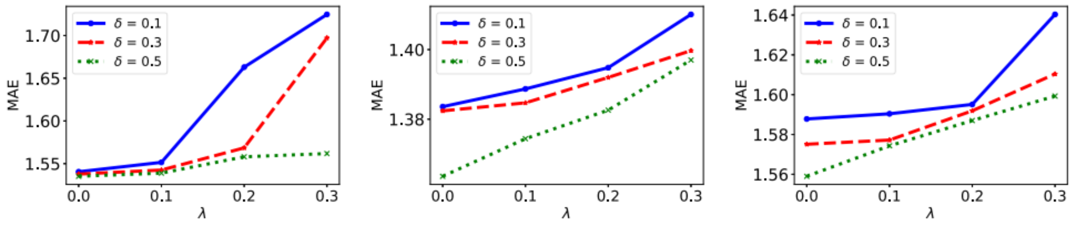
❑ Experimental results

- With a fixed λ, FeSoG performs better when increasing δ (reducing gradient clipping)
- With fixed δ, FeSoG performs worse when increasing λ (increasing noise)
- There is a tradeoff in selecting optimal values of λ and δ



(a) Ciao (b) Epinions (c) Filmtrust

Fig. 11. RMSE performance with respect to different δ and λ on three datasets.



(a) Ciao (b) Epinions (c) Filmtrust

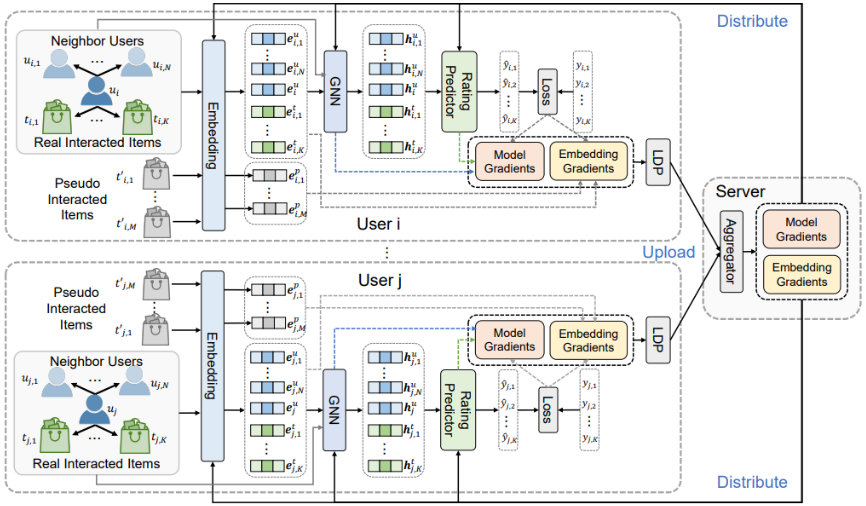
Fig. 12. MAE performance with respect to different δ and λ on three datasets.

$$\tilde{g}^{(n)} = \mathrm{clip}(g^{(n)}, \delta) + \mathrm{Laplacian}(0, \lambda \cdot \mathrm{mean}(g^{(n)})),$$

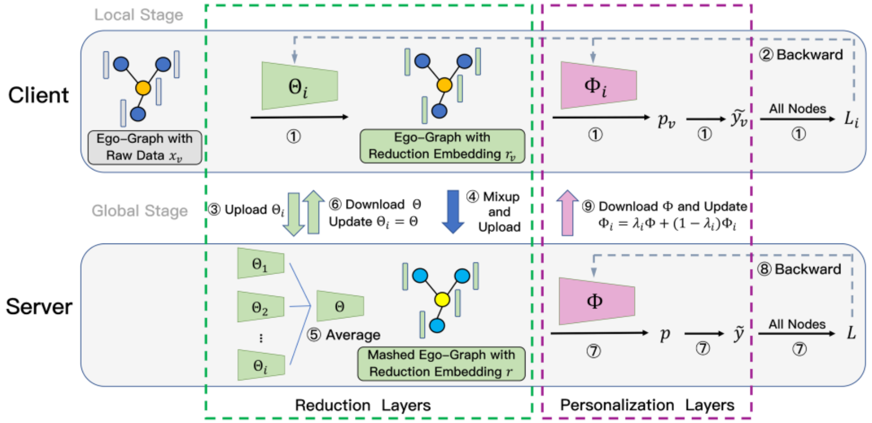
Privacy-Preserving Federated Graph Learning

➢ Structure-Level Privacy Protection

- ❑ Hide cross-client interaction
  - Privacy-preserving local neighbor expansion
  - Local neighbor generation
  - Local information mixup



- ❑ Solutions
  - FedGNN
  - FedEgo



Privacy-Preserving Federated Graph Learning

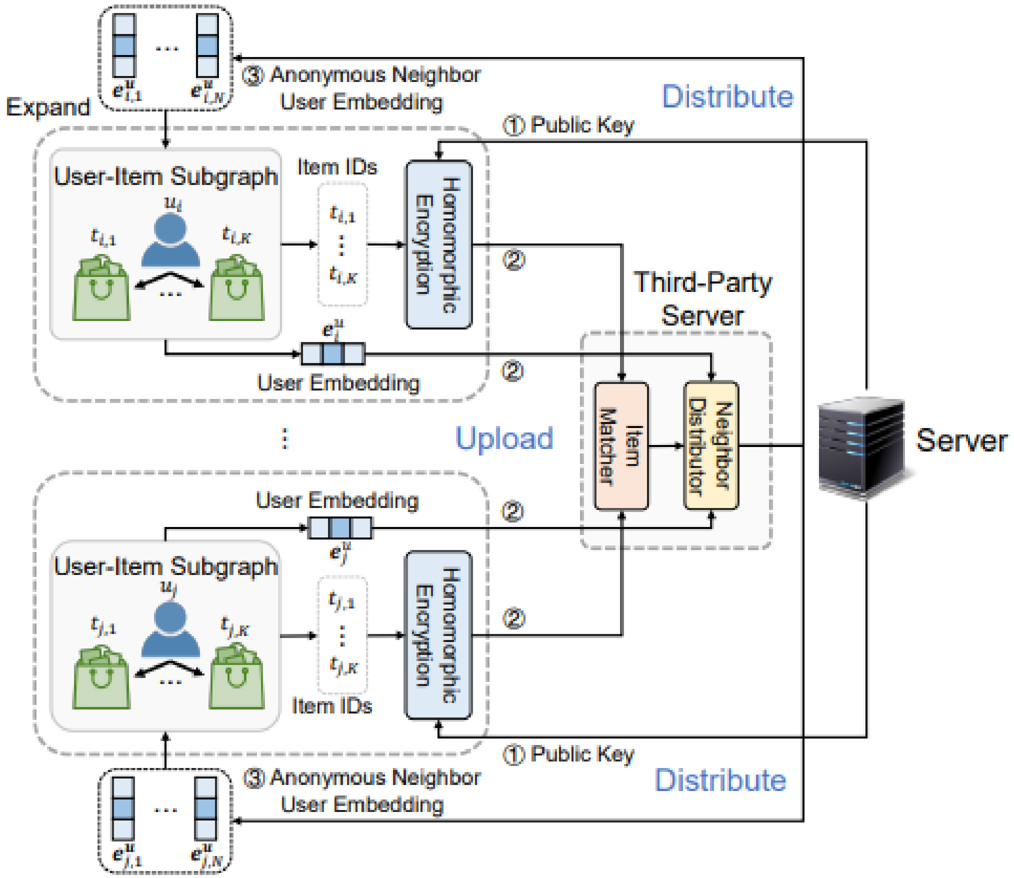
➢ FedGNN

❑ Privacy protection

- LDP with uniform Gaussian
- Pseudo item sampling with Gaussian-noise gradient
  - Sample q non-neighbor items as pseudo items
  - Generate gradients of pseudo items using Gaussian noise with the same mean and covariance as real items
- Privacy-preserving user-item graph expansion

Algorithm 2 privacy-preserving user-item graph expansion

1: **PrivacyPreservingGraphExpansion**():
2: Server sends a public key p to user clients
3: User clients encrypt item IDs with p
4: User clients upload the user embedding and encrypted item IDs to a third-party server
5: Third-party server distributes neighboring user embeddings to user clients
6: User clients extend local graphs



Privacy-Preserving Federated Graph Learning

➢ FedGNN

❑ Experimental results

- The performance of FedGNN is satisfactory on different GNN backbones
- Variants utilizing the high-order information by local neighbor expansion perform better than those without high-order information
- Using fixed neighbor user embeddings (trained in certain iterations) is better than using fully trainable ones (updated in each iteration)

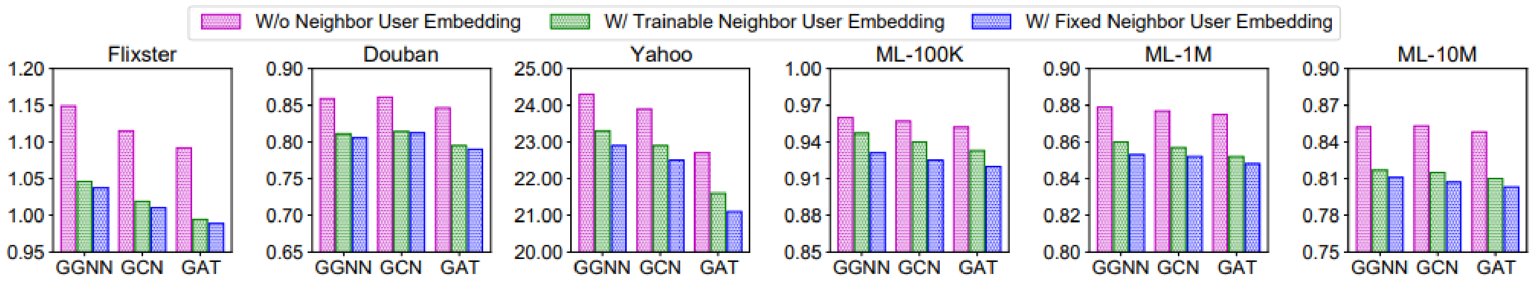


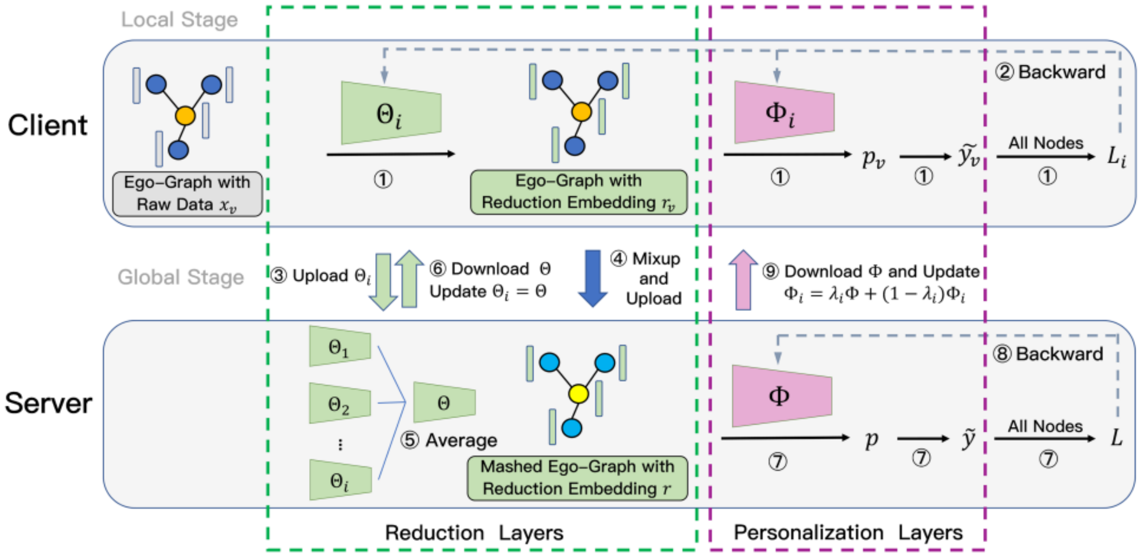
Figure 4: Influence of second-order information and different GNN architectures.

Privacy-Preserving Federated Graph Learning

➢ FedEgo

❑ Local information mixup

- Local stage:
  - Local ego graphs embedding
  - Personalized prediction
  - Ego graphs mixup



- Global stage:
  - Train personalized layers on local mashed ego graphs
  - Global parameter aggregation

Privacy-Preserving Federated Graph Learning

➢ FedEgo

❑ Local ego graph mixup

- Mixing up node embeddings and labels in the ego graphs in each batch
- Ego graphs are adopted as they are easily aligned for mixup (hiding private information while sharing)

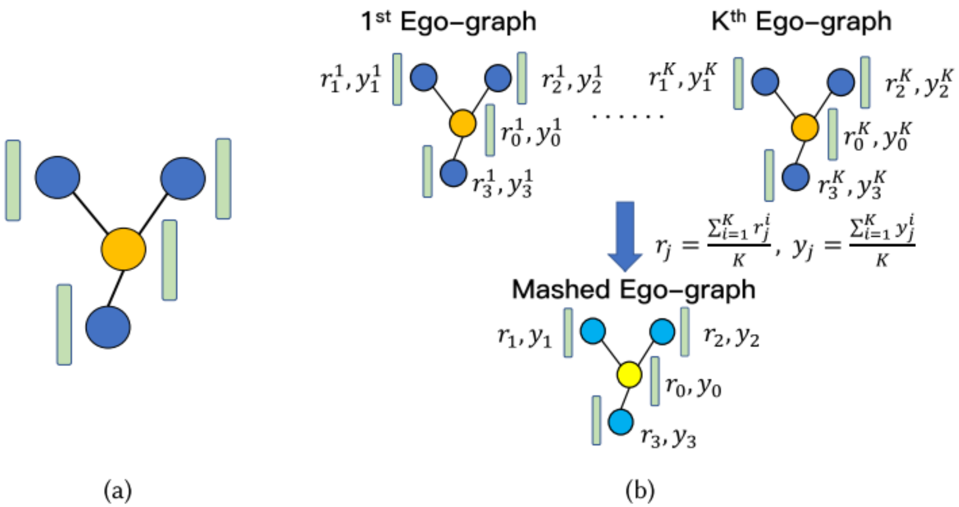


Fig. 2. (a) Illustration of 1 hop ego-graph. (b) Illustration of the alignment and Mixup among a batch of ego-graphs. The center nodes are aligned together and their neighbors are extended recursively. The reduction embedding r and one-hot label y are averaged according to the alignment.

➢ FedEgo

❑ Experimental results

- Fed methods benefit from the collaboration on all datasets and enhance the personalization ability of local models
- FedEgo consistently outperforms other methods and improves the generalization ability of local models
- The improvement indicates that FedEgo can facilitate client collaboration and generalize across non-IID local graph data

Table 2. F1 Score for Node Classification in the Local Test under Label-skew Scenarios

Dataset	Local Only	FedAvg	FedProx	GraphFL	D-FedGNN	FedGCN	FedSage	FedSage+	FedEgo
Cora	0.8437 (±0.0039)	0.9473 (±0.0012)	0.9483 (±0.0019)	0.867 (±0.0029)	0.9503 (±0.0017)	0.8784 (±0.0006)	0.9507 (±0.0009)	0.952 (±0.0008)	**0.9577** (±0.0012)
Citeseer	0.7617 (±0.0005)	0.918 (±0.0029)	0.918 (±0.0014)	0.755 (±0.0014)	0.9193 (±0.0005)	0.8967 (±0.0008)	0.913 (±0.0008)	0.9137 (±0.0005)	**0.9210** (±0.0024)
Wiki	0.8728 (±0.0141)	0.9258 (±0.0101)	0.9232 (±0.0096)	0.8088 (±0.0069)	0.92 (±0.0097)	0.817 (±0.0040)	0.9223 (±0.0083)	**0.9246** (±0.0075)	0.9191 (±0.0077)
CoraFull	0.6402 (±0.0002)	0.874 (±0.0010)	0.873 (±0.0009)	0.477 (±0.0017)	0.8837 (±0.0003)	0.8466 (±0.0025)	0.881 (±0.0003)	Out Of Memory	**0.8972** (±0.0008)

Table 3. F1 Score for Node Classification in the Global Test under Label-skew Scenarios

Dataset	Local Only	FedAvg	FedProx	GraphFL	D-FedGNN	FedGCN	FedSage	FedSage+	FedEgo
Cora	0.6985 (±0.0014)	0.7706 (±0.0033)	0.7697 (±0.0037)	0.7346 (±0.0027)	0.7865 (±0.0022)	0.6933 (±0.0007)	0.7926 (±0.0018)	0.7848 (±0.0026)	**0.8016** (±0.0019)
Citeseer	0.6125 (±0.0003)	0.6941 (±0.0058)	0.6924 (±0.0038)	0.6327 (±0.0070)	0.7049 (±0.0055)	0.6614 (±0.0009)	0.7055 (±0.0011)	0.7071 (±0.0012)	**0.7200** (±0.0015)
Wiki	0.696 (±0.0113)	0.7856 (±0.0020)	0.7851 (±0.0034)	0.7112 (±0.0061)	0.7960 (±0.0014)	0.4428 (±0.0310)	0.7839 (±0.0006)	0.7849 (±0.0001)	**0.8126** (±0.0100)
CoraFull	0.4905 (±0.0006)	0.5351 (±0.0045)	0.5336 (±0.0050)	0.3328 (±0.0032)	0.5615 (±0.0011)	0.4777 (±0.0005)	0.599 (±0.0006)	Out Of Memory	**0.6221** (±0.0006)

Privacy-Preserving Federated Graph Learning

➢ References

❑ In this tutorial

- Liu, Zhiwei, et al. "Federated social recommendation with graph neural network." ACM TIST 2022.
- Wu, Chuhan, et al. "Fedgnn: Federated graph neural network for privacy-preserving recommendation." FL-ICML'21.
- Zhang, Taolin, et al. "FedEgo: privacy-preserving personalized federated graph learning with ego-graphs." ACM TKDD 2023.

❑ Related references

- Yan, Bo, et al. "Federated heterogeneous graph neural network for privacy-preserving recommendation." WWW 2024.
- Tian, Changxin, et al. "Privacy-preserving cross-domain recommendation with federated graph learning." ACM TOIS 2024.

Outline

- ✓ Introduction
- ✓ Subgraph Federated Learning
- ✓ Federated Graph Learning with Non-IID Graphs
- ✓ Privacy-Preserving Federated Graph Learning
- ✓ Summary and Future Directions

Summary and Future Directions

➢ Summary

❑ FGL jointly trains graph learning models over distributed graph data

- Transmit model parameters while keeping graph data locally

❑ Key research topics in FGL

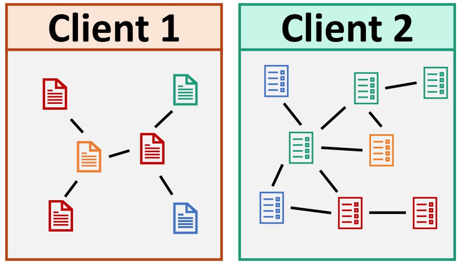
Research Topics	Challenges	Techniques	Representative Works
Subgraph Federated Learning	Missing cross-client links	Missing neighbor generator	FedSage+
	Community heterogeneity	Functional similarity matching + personalized parameter masking	Fed-PUB
FGL with Non-IID Graphs	Cross-dataset structural knowledge sharing	Structure knowledge sharing	FedStar
	Distribution shifts	Virtual node optimization	FedVN
Privacy-Preserving FGL	Entity-level privacy protection	(Local) differential privacy	FedSoG
	Structure-level privacy protection	Local information mixup	FedGNN, FedEgo

Summary and Future Directions

➢ Future Directions

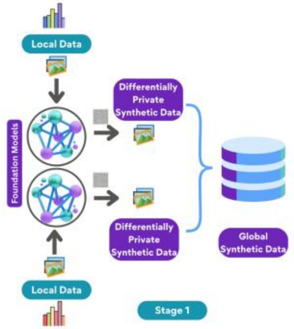
❑ FGL on text-attributed graphs (TAGs)

- Enhance modeling TAGs via LLMs



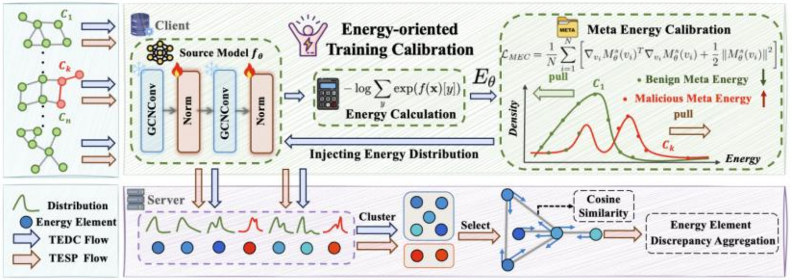
❑ FGL with graph foundation models (GFMs)

- Cross-dataset/domain graph data
- Personalized adaptation



❑ Backdoor attack & defense in FGL

- Topology knowledge injection



FedTGE (ICLR 2025 Oral)

The End

Thanks for listening!

Presenters: Xingbo Fu, Zihan Chen, Binchi Zhang, Jundong Li
University of Virginia

SDM 2025 Tutorial
May 2025