Carnegie Mellon University

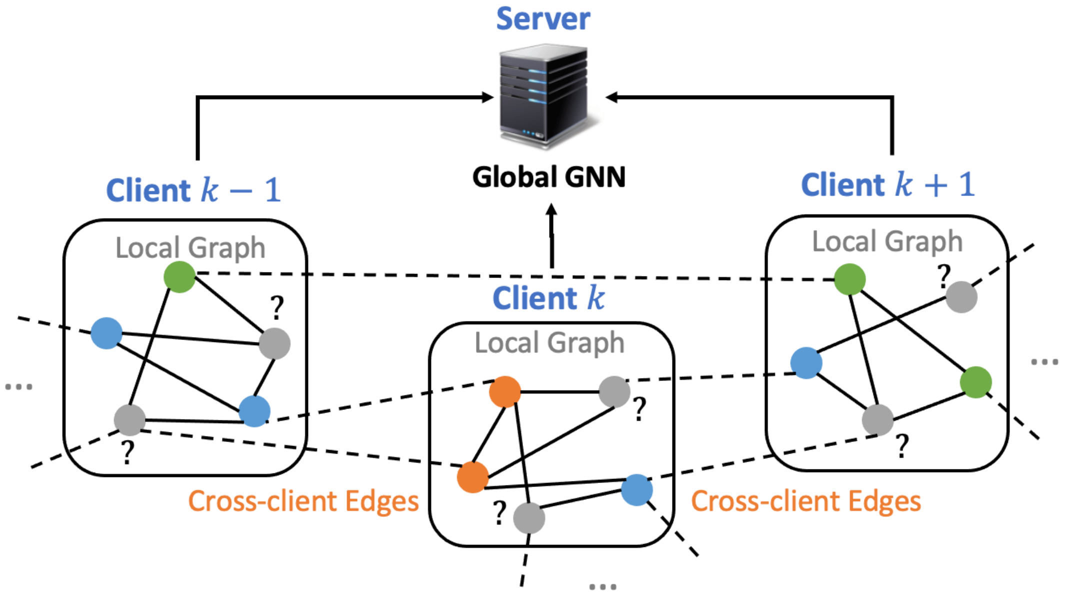
FedGCN: Convergence-Communication Tradeoffs in Federated Training of Graph Convolutional Networks

Presenter: Yuhang Yao

Author: Yuhang Yao, Weizhao Jin, Srivatsan Ravi, Carlee Joe-Wong

Federated Node Classification

- ➢ Nodes in a graph are partitioned across clients (e.g. private data across countries)
- ➢ Cross-client edges exist between nodes at different clients



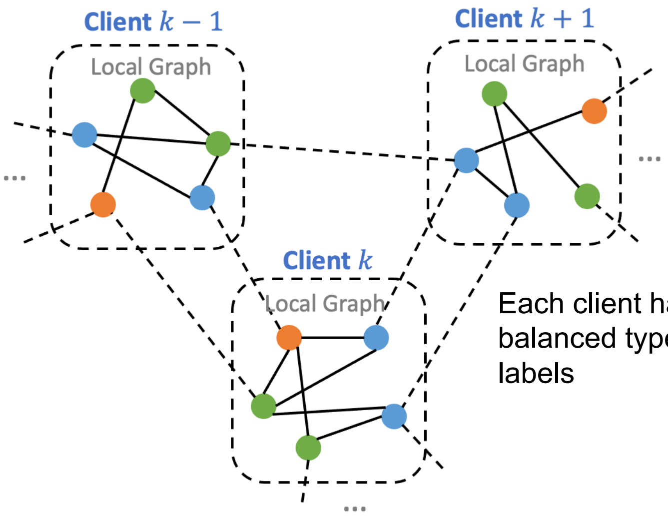
Each client knows
- ➢ Local graph structure
- ➢ Local node features
- ➢ Corresponding cross-client edges

Node classification requires node features stored in other clients

Carnegie Mellon University

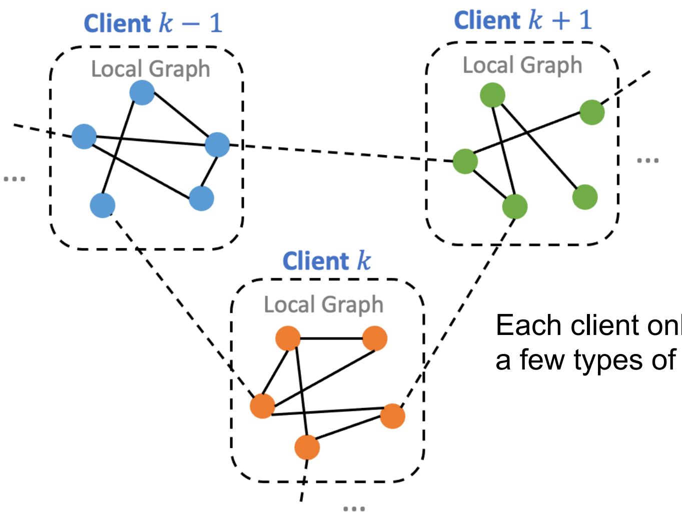
Edges in Heterogeneous Data Distribution

IID (Independently Identically Distributed)



Each client has balanced types of labels

Non-IID



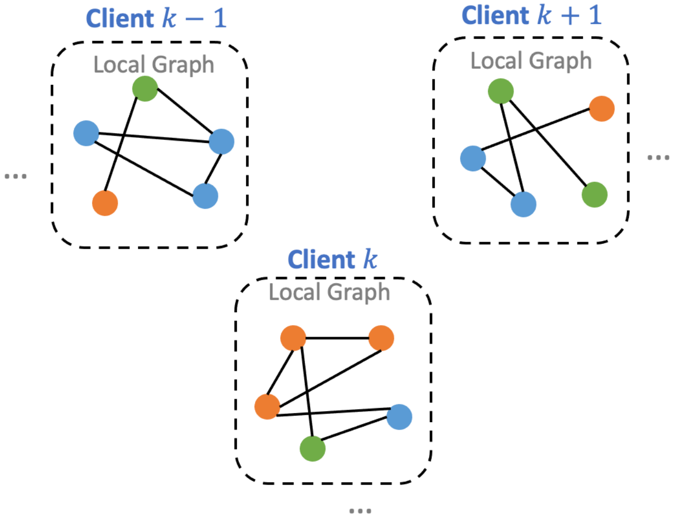
Each client only has a few types of labels

- Nodes connect more to nodes with the same label
- Non-IID may have fewer cross-client edges than IID
- More cross-client edges require more communication

Carnegie Mellon University

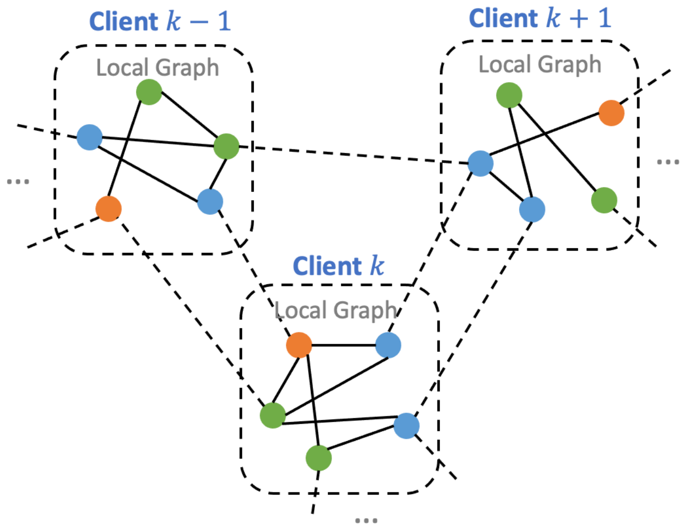
Limitation of Distributed Training

Ignore cross-client edges



Or

Send features and intermediate output at every round



Ignoring cross-client edges causes information loss

Sending features requires huge communication cost

[1] He, Chaoyang, et al. "Fedgraphnn: A federated learning system and benchmark for graph neural networks." *arXiv preprint arXiv:2104.07145* (2021).
[2] Wan, Cheng, et al. "BDS-GCN: Efficient full-graph training of graph convolutional nets with partition-parallelism and boundary sampling." (2020).

Carnegie Mellon University

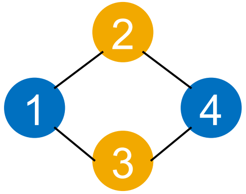
GCN in Federated Learning

In FL setting, nodes are stored in different clients

For each layer l

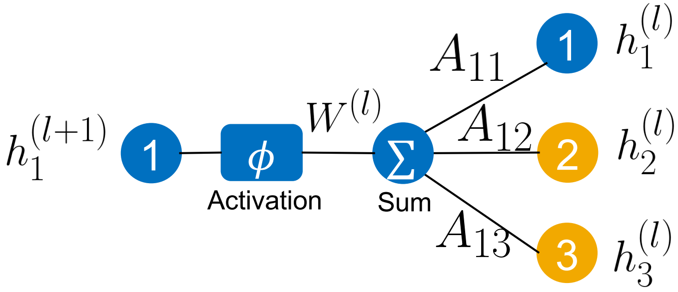
Node i in client $c(i)$ needs to aggregate information of nodes from $c(i)$ and other clients

Input Graph



Different color represents belonging of clients

$$\boldsymbol{h}_i^{(l+1)} = \phi\left(\sum_{j\in\mathcal{N}_i} A_{ij}\boldsymbol{h}_j^{(l)} W_{c(i)}^{(l)}\right)$$



Feature Aggregation of Node 1

A_{ij}: Weight of connections between node i and node j

$h_i^{(l)}$: Output of node i at layer l

$c_{(i)}$: index of the client that contains node i

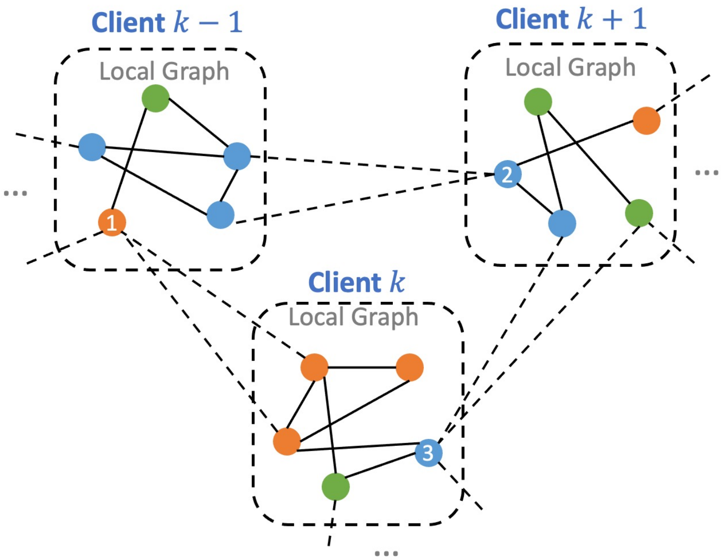
$W_{c(i)}^{(l)}$: Parameters of GCN at layer l at client $c_{(i)}$

$h_i^{(0)} = x_i$: Feature vector of node i at layer 0

Carnegie Mellon University

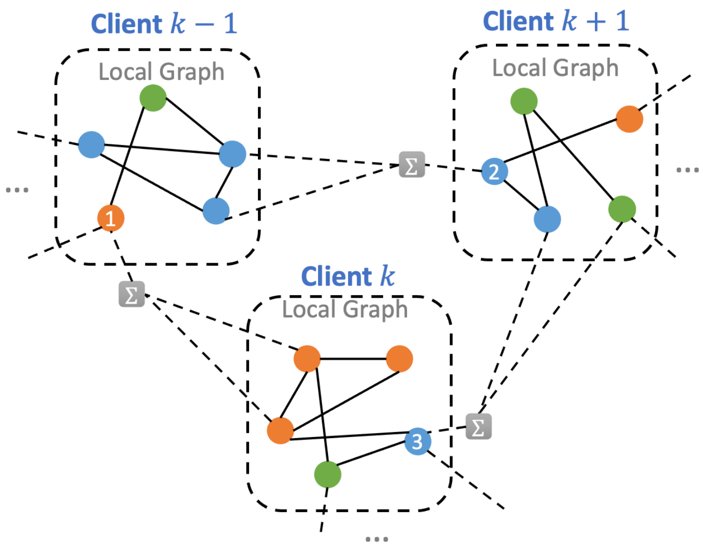
Feature Aggregation Instead of Sending Features

Send features and intermediate output at every training round



- High communication cost at every round

Send feature aggregations at initial round

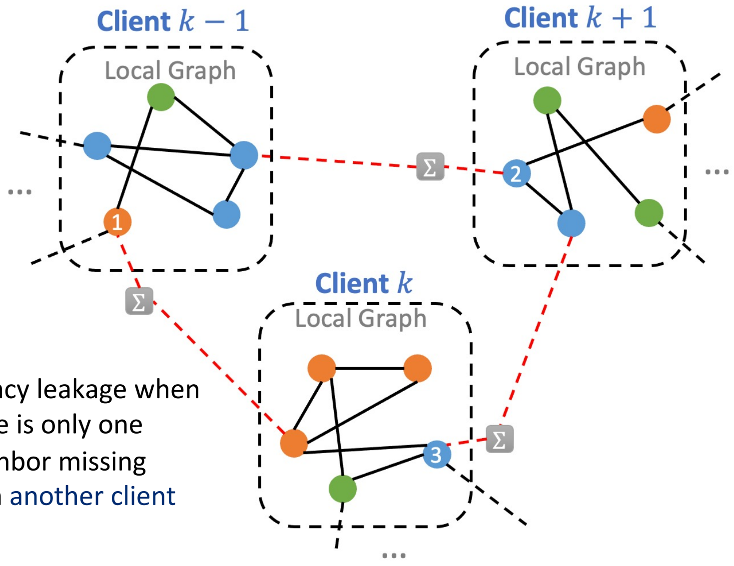


- Same computation
- Much lower communication cost

Carnegie Mellon University

Server Aggregation Instead of Clients Aggregation

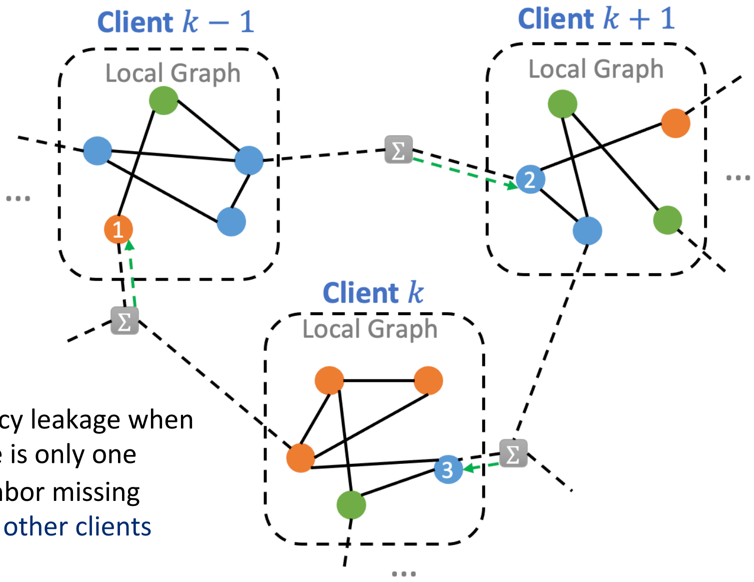
Clients Aggregation



Privacy leakage when there is only one neighbor missing from another client

$$\{\textstyle\sum_{j\in\mathcal{N}_i} \mathbb{I}_z(c(j)) A_{ij}\boldsymbol{x}_j\}_{z\in[K]}$$

Server Aggregation



Privacy leakage when there is only one neighbor missing from other clients

$$\sum_{j\in\mathcal{N}_i} A_{ij}\boldsymbol{x}_j = \sum_{k=1}^{K} \sum_{j\in\mathcal{N}_i} \mathbb{I}_k(c(j)) \cdot A_{ij}\boldsymbol{x}_j$$

Carnegie Mellon University

Secure Neighbor Feature Aggregation

To guarantee privacy during the aggregation process of accumulated features, we leverage Fully Homomorphic Encryption (FHE)

$$\left[\left[\sum_{j \in \mathcal{N}_i} A_{ij} \boldsymbol{x}_j\right]\right] = \sum_{k=1}^{K} \left[\left[\sum_{j \in \mathcal{N}_i} \mathbb{I}_k(c(j)) \cdot A_{ij} \boldsymbol{x}_j\right]\right]$$

1. All clients agree on and initialize a FHE keypair
2. Each client encrypts the local neighbor feature array and sends it to the server
3. Upon receiving all encrypted neighbor feature arrays from clients, the server performs secure neighbor feature aggregation

Carnegie Mellon University

FedGCN with Three Types of Communication

- **No Communication(0-hop)**: Use feature aggregation at the same client
- **1-hop Communication**: Communicate feature aggregation of 1-hop neighbors at all clients
- **2-hop Communication**: Communicate feature aggregation of 2-hop neighbors at all clients

  - More hop means higher communication costs but with less information loss
  - 2-hop communication does not have information loss for 2-layer GCN

Carnegie Mellon University

Training Process of FedGCN

Communication at initial round

Normal federated training process

Algorithm 1 FedGCN Federated Training for Graph Convolutional Network

// Communication Round
for each client $k \in [K]$ **do in parallel**
  Send $[\![\{\sum_{j \in \mathcal{N}_i} \mathbb{I}_k(c(j)) \cdot A_{ij}\boldsymbol{x}_j\}_{i \in V_k}]\!]$ to the server
end
// Server Operation
for $i \in V$ **do in parallel**
  $[\![\sum_{j \in \mathcal{N}_i} A_{ij}\boldsymbol{x}_j]\!] = \sum_{k=1}^{K} [\![\sum_{j \in \mathcal{N}_i} \mathbb{I}_k(c(j)) \cdot A_{ij}\boldsymbol{x}_j]\!]$
end
for each client $k \in [K]$ **do in parallel**
  if 1-hop **then**
    Receive $[\![\{\sum_{j \in \mathcal{N}_i} A_{ij}\boldsymbol{x}_j\}_{i \in V_k}]\!]$ and decrypt it
  end
  if 2-hop **then**
    Receive $[\![\{\sum_{j \in \mathcal{N}_i} A_{ij}\boldsymbol{x}_j\}_{i \in \mathcal{N}_{V_k}}]\!]$ and decrypt it
  end
end
// Training Round
for $t = 1, \ldots, T$ **do**
  for each client $k \in [K]$ **do in parallel**
    Receive $[\![\boldsymbol{w}^{(t)}]\!]$ and decrypt it
    Set $\boldsymbol{w}_k^{(t,1)} = \boldsymbol{w}^{(t)}$,
    for $e = 1, \ldots, \tau$ **do**
      Set $\boldsymbol{g}_{\boldsymbol{w}_k}^{(t,e)} = \nabla_{\boldsymbol{w}_k} f_k(\boldsymbol{w}_k^{(t,e)}; G_k)$
      $\boldsymbol{w}_k^{(t,e+1)} = \boldsymbol{w}_k^{(t,e)} - \eta\, \boldsymbol{g}_{\boldsymbol{w}_k}^{(t,e)}$ // Update Parameters
    end
    $\boldsymbol{\Delta}_{\boldsymbol{w}_k}^{(t,\tau)} = \boldsymbol{w}_k^{(t,\tau+1)} - \boldsymbol{w}_k^{(t,1)}$
    Send $[\![\boldsymbol{\Delta}_{\boldsymbol{w}_k}^{(t,\tau)}]\!]$ to the server
  end
  // Server Operations
  $[\![\boldsymbol{\Delta}_{\boldsymbol{w}}^{(t)}]\!] = \frac{1}{K}\sum_{k=1}^{K} [\![\boldsymbol{\Delta}_{\boldsymbol{w}_k}^{(t,\tau)}]\!]$ // Difference[6] Aggregation
  $[\![\boldsymbol{w}^{(t+1)}]\!] = [\![\boldsymbol{w}^{(t)}]\!] - [\![\boldsymbol{\Delta}_{\boldsymbol{w}}^{(t)}]\!]$ and broadcast to local clients // Update Global Models
end

Carnegie Mellon University

Experiment Setups

Citation Network Datasets

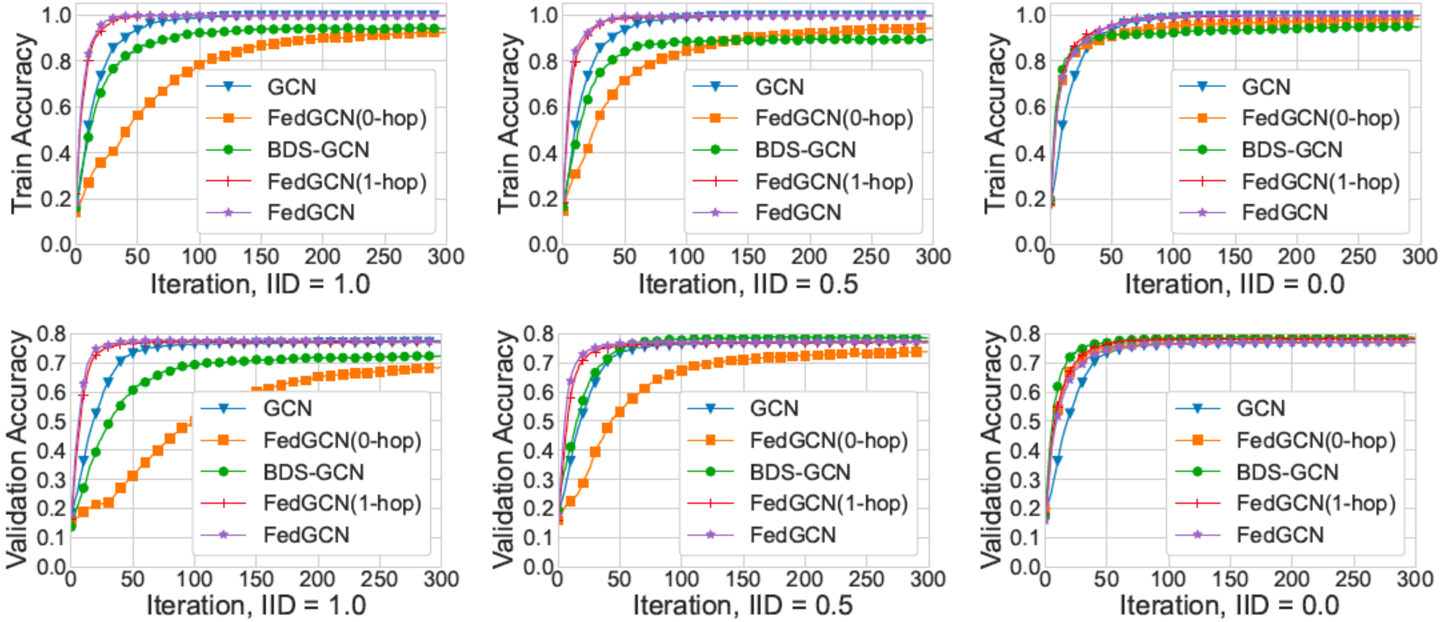
Dataset	Nodes	Edges	Features	Classes
Cora	2,708	5,429	1,433	7
Citeseer	3,327	4,732	3,703	6
Pubmed	19,717	44,338	500	3
Ogbn-Arxiv	169,343	1,166,243	128	40

Compared methods

- Centralized GCN
- FedGCN(0-hop), FedGraphnn[1]
- BDS-GCN: Randomly samples cross-client edges
- FedSage+: Approximates 1-hop neighbors [2]
- FedGCN(1-hop)
- FedGCN(2-hop)

[1] He, Chaoyang, et al. "Fedgraphnn: A federated learning system and benchmark for graph neural networks." *arXiv* (2021).
[2] Zhang, Ke, et al. "Subgraph federated learning with missing neighbor generation." *NeurIPS* (2021).

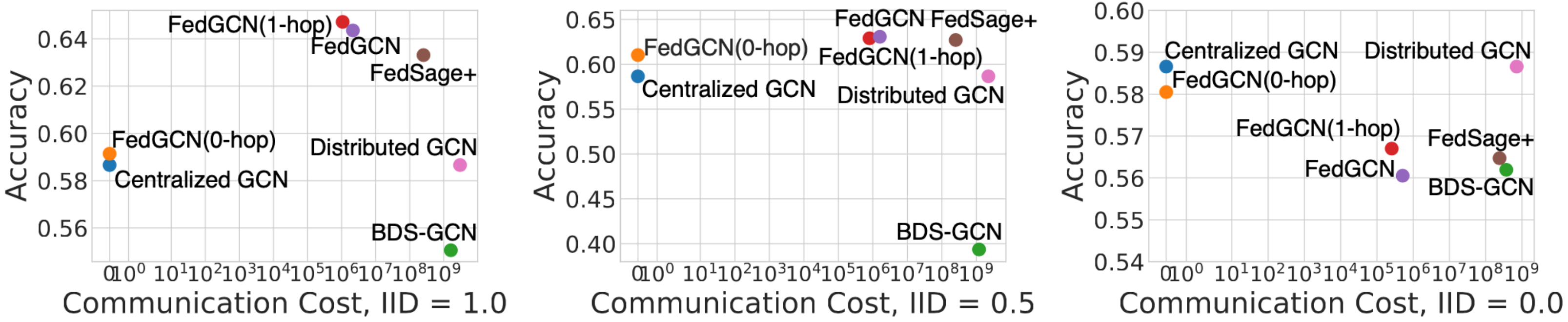
Training and Test Accuracy on Cora



- FedGCN converges much faster and has a higher test and training accuracy in all settings.
- Under the extreme non-i.i.d. setting, FedGCN (0-hop) has sufficient information to train a good model.

Carnegie Mellon University

Test Accuracy vs Communication Cost on OGBN-ArXiv



- ➢ FedGCN (0-, 1-, and 2-hop) requires little communication with high accuracy
- ➢ FedGCN (0-hop) requires much less communication, but has lower accuracy due to information loss in the i.i.d. and partial-i.i.d. settings

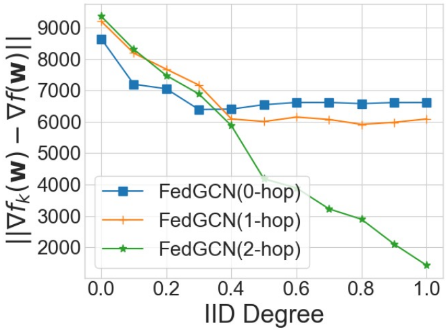
Convergence Rate

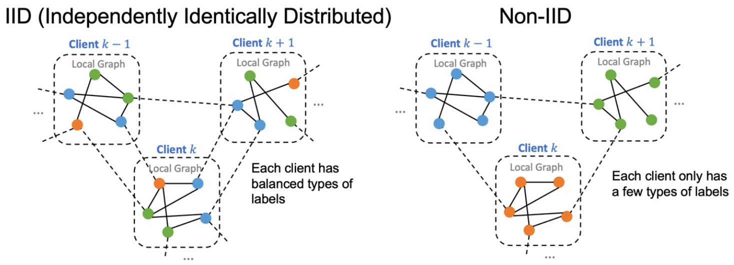
	0-hop	1-hop	2-hop
i.i.d. (SBM)	$(1-\frac{1}{K^4})\frac{N^5}{K^5}\|B^4\|$	$(1-\frac{1}{K^4}(1+c_\alpha+c_\mu)^2)\frac{N^5}{K^5}\|B^4\|$	$(1-\frac{1}{K^4}(1+c_\alpha+c_\mu)^6)\frac{N^5}{K^5}\|B^4\|$

In i.i.d., the 2-hop method has faster convergence rate

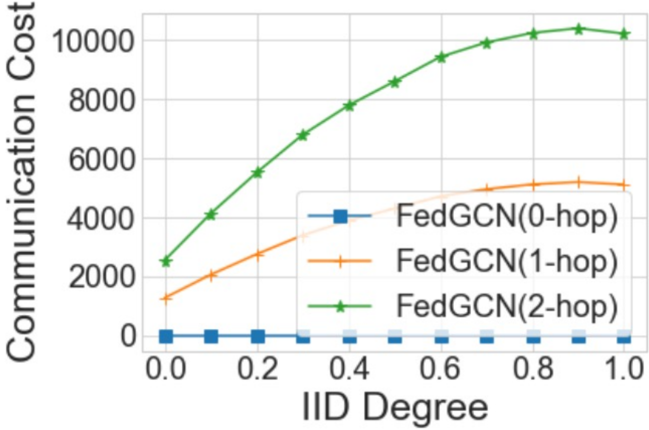
	0-hop	1-hop	2-hop
Non-i.i.d. (SBM)	$(1-\frac{1}{K^4})\frac{N^5}{K^5}\|B^4\|+\sigma$	$(1-\frac{1}{K^4}(1+c_\mu)^2)\frac{N^5}{K^5}\|B^4\|+\sigma$	$(1-\frac{1}{K^4}(1+c_\mu)^6)\frac{N^5}{K^5}\|B^4\|+\sigma$

In extremely Non-i.i.d., the 1-hop and 2-hop methods help little.





Communication Cost



2-hop requires more communication but helps the convergence

Data Distribution	0-hop	1-hop	2-hop
Generic Graph	0	$\sum_{i \in V} \lvert c(\mathcal{N}_i) \rvert d + Nd$	$\sum_{i \in V} \lvert c(\mathcal{N}_i) \rvert d + \sum_{k=1}^{K} \lvert \mathcal{N}_{V_k} \rvert d$
Non-i.i.d. (SBM)	0	$(c_\mu + 2)Nd$	$2(c_\mu + 1)Nd$
Partial-i.i.d. (SBM)	0	$(c_\alpha p + c_\mu + 2)Nd$	$2(c_\alpha p + c_\mu + 1)Nd$
i.i.d. (SBM)	0	$(c_\alpha + c_\mu + 2)Nd$	$2(c_\alpha + c_\mu + 1)Nd$

Carnegie Mellon University

Conclusion & Next Step

Conclusion

- Cross-client edges affect the model performance (convergence rate and test accuracy).
- Proposed FedGCN helps recover information on cross-client edges and only requires communication at the initial step
- Tradeoffs exist between convergence and communication under different data distributions.

Next Steps

- Large-scale experiments and more compared methods
- System deployment and library development