Carnegie Mellon University

USC

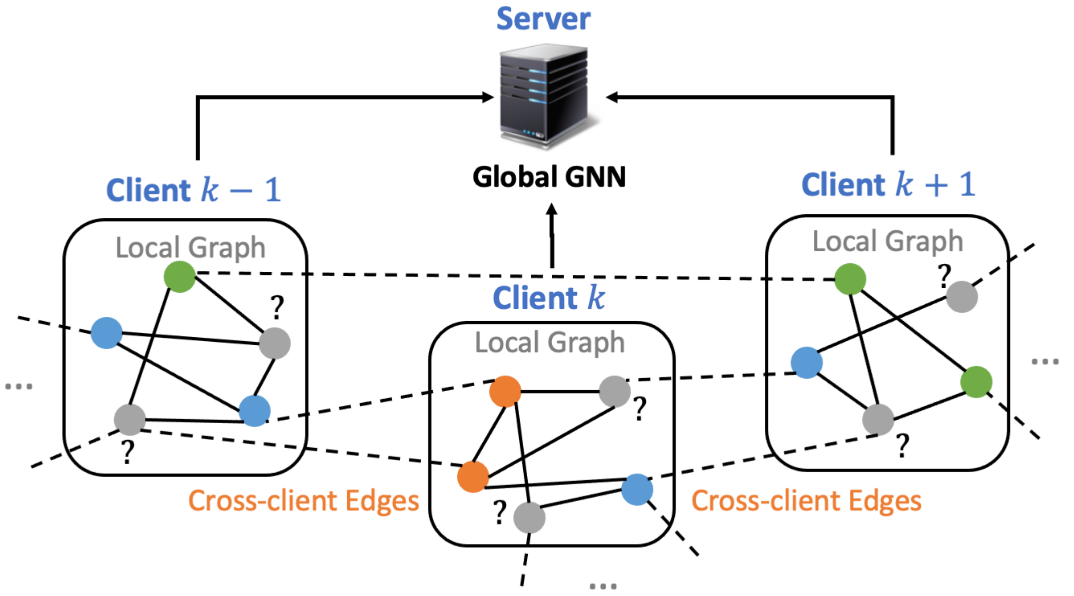
FedGCN: Convergence-Communication Tradeoffs in Federated Training of Graph Convolutional Networks

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

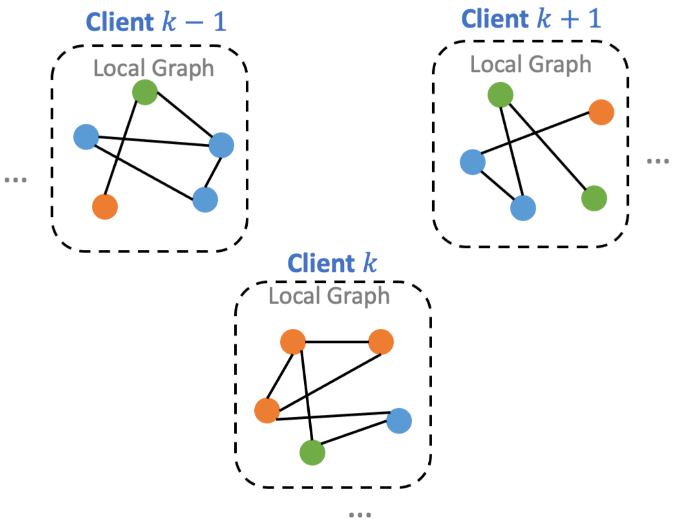


Each client knows
- ➢ Local graph structure
- ➢ Local node features
- ➢ Corresponding cross-client edges

Node classification requires node features stored in other clients

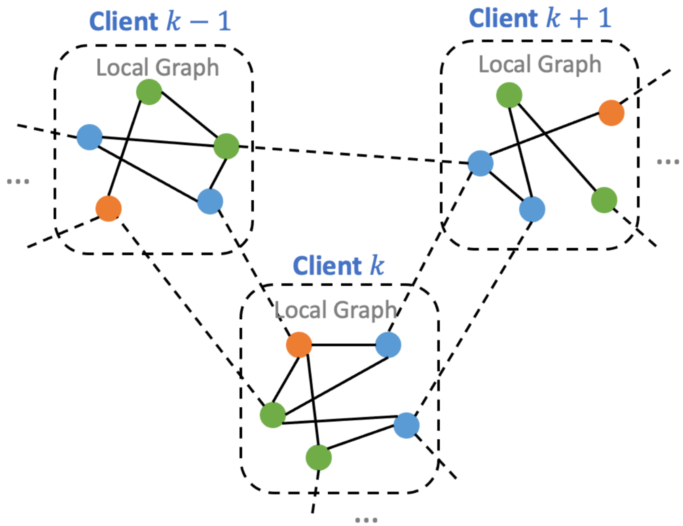
Limitation of Distributed Training

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).

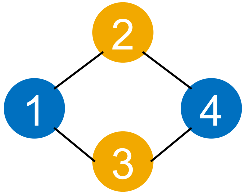
GCN in Federated Learning

In FL setting, nodes are stored in different clients

For each layer l

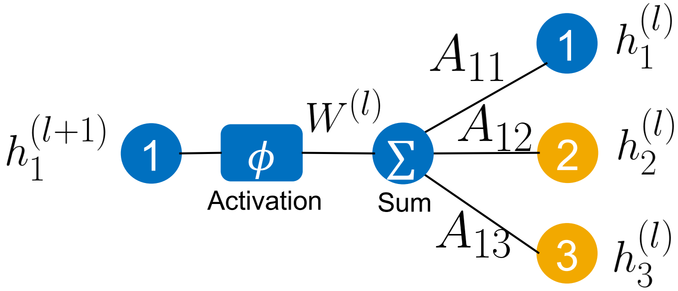
Node i in client $c(i)$ needs to aggregate information of nodes from $c(i)$ and other clients

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

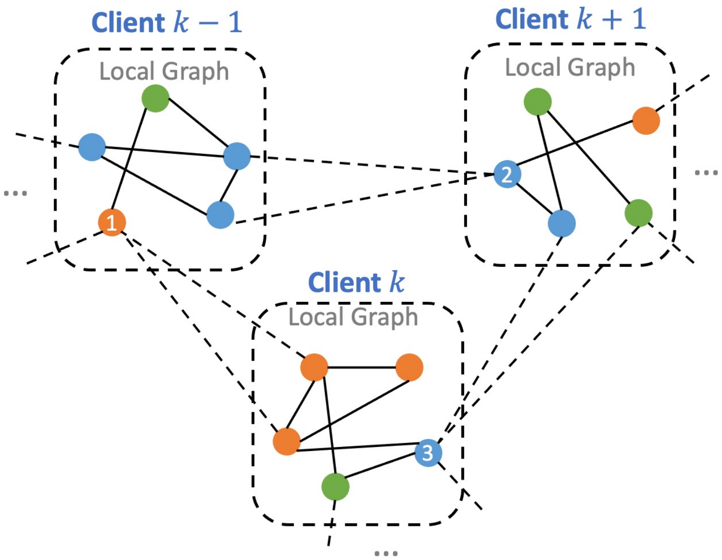
$W_{c(i)}^{(l)}$: Parameters of GCN at layer l at client $c_{(i)}$

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

Carnegie Mellon University

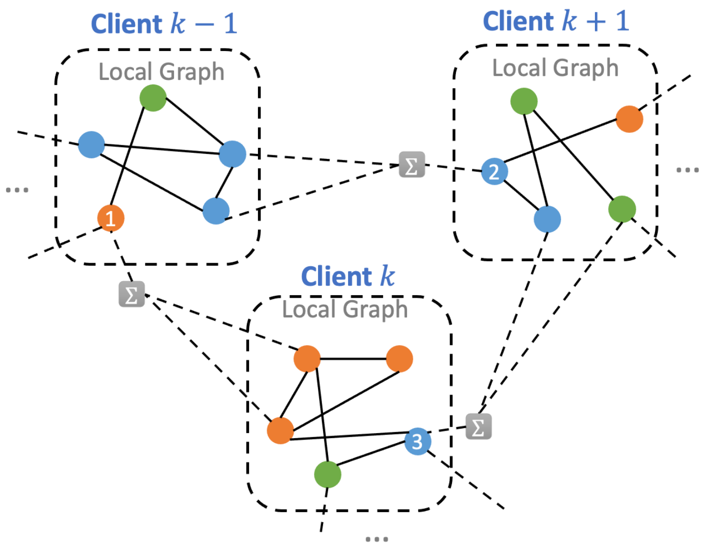
Feature Aggregation Instead of Sending Features

Send features and intermediate output at every training round



- ➢ High communication cost at every round

Send feature aggregations at initial round

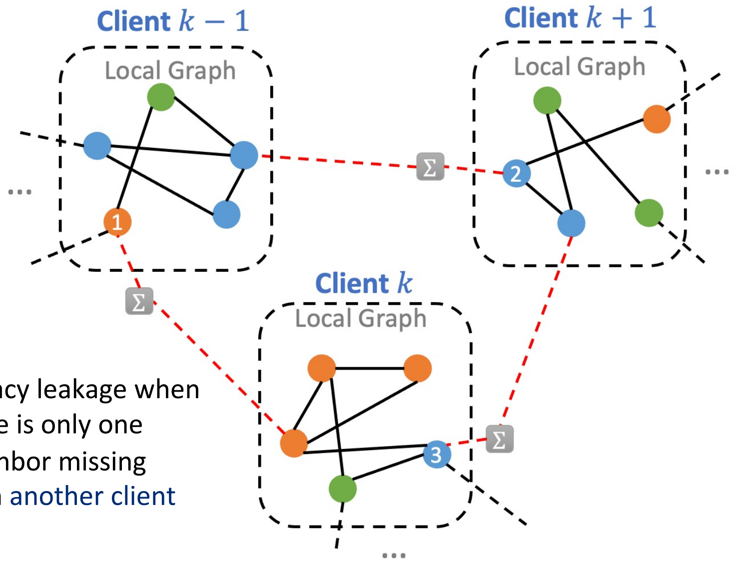


- ➢ Same computation
- ➢ Much lower communication cost

Carnegie Mellon University

Server Aggregation Instead of Clients Aggregation

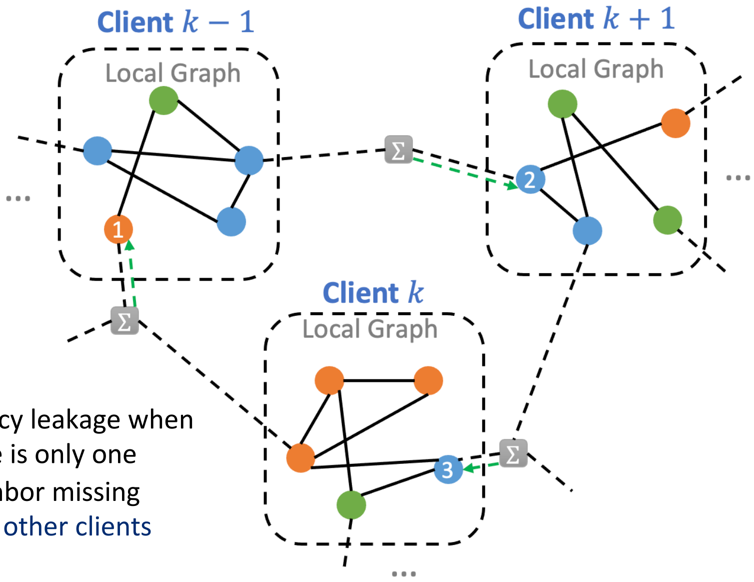
Clients Aggregation



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

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

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)

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

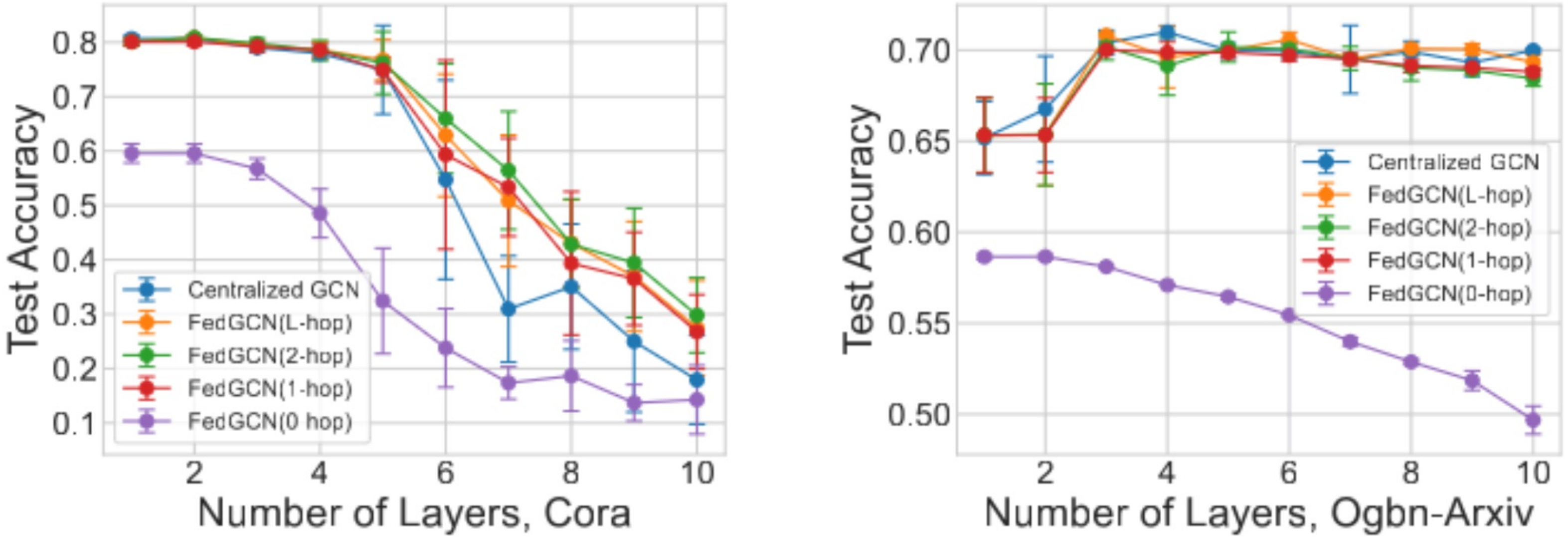
1. All clients agree on and initialize a FHE keypair
2. Each client encrypts the local neighbor feature array and sends it to the server
3. Upon receiving all encrypted neighbor feature arrays from clients, the server performs secure neighbor feature aggregation

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 and perform the aggregation of L-layer GCNs

More hop means higher communication costs but with less information loss

Why Not L-hop Communication?



Adding more layers and hop communication does not increase model accuracy for ≥3-layer GCN layers and ≥2-hop communication

Carnegie Mellon University

Training Process of FedGCN

Communication at initial round

Normal federated training process

Algorithm 1 FedGCN Federated Training for Graph Convolutional Network

// Pre-Training Communication Round
for each client $k \in [K]$ **do in parallel**
  Send $[\![\{\sum_{j \in \mathcal{N}_i} \mathbb{I}_k(c(j)) \cdot \boldsymbol{A}_{ij}\boldsymbol{x}_j\}_{i \in \mathcal{V}_k}]\!]$ to the server
end
// Server Operation
for $i \in \mathcal{V}$ **do in parallel**
  $[\![\sum_{j \in \mathcal{N}_i} \boldsymbol{A}_{ij}\boldsymbol{x}_j]\!] = \sum_{d=1}^{C} [\![\sum_{j \in \mathcal{N}_i} \mathbb{I}_k(c(j)) \cdot \boldsymbol{A}_{ij}\boldsymbol{x}_j]\!]$
end
for each client $k \in [K]$ **do in parallel**
  if 1-hop **then**
    Receive $[\![\{\sum_{j \in \mathcal{N}_i} \boldsymbol{A}_{ij}\boldsymbol{x}_j\}_{i \in \mathcal{V}_k}]\!]$ and decrypt it
  end
  if 2-hop **then**
    Receive $[\![\{\sum_{j \in \mathcal{N}_i} \boldsymbol{A}_{ij}\boldsymbol{x}_j\}_{i \in \mathcal{N}_{\mathcal{V}_k}}]\!]$ and decrypt it
  end
end
// Training Rounds
for $t = 1, \ldots, T$ **do**
  for each client $k \in [K]$ **do in parallel**
    Receive $[\![\boldsymbol{w}^{(t)}]\!]$ and decrypt it
    Set $\boldsymbol{w}_k^{(t,0)} = \boldsymbol{w}^{(t)}$,
    for $e = 1, \ldots, E$ **do**
      Set $\boldsymbol{g}_{\boldsymbol{w}_k}^{(t,e)} = \nabla_{\boldsymbol{w}_k} f_k(\boldsymbol{w}_k^{(t,e-1)}; \mathcal{G}_k)$
      $\boldsymbol{w}_k^{(t,e)} = \boldsymbol{w}_k^{(t,e-1)} - \eta\, \boldsymbol{g}_{\boldsymbol{w}_k}^{(t,e)}$ // Update Parameters
    end
    Send $[\![\boldsymbol{w}_k^{(t,E)}]\!]$ to the server
  end
  // Server Operations
  $[\![\boldsymbol{w}^{(t+1)}]\!] = \frac{1}{K} \sum_{d=1}^{C} [\![\boldsymbol{w}_k^{(t,E)}]\!]$ // Update Global Models
  Broadcast $[\![\boldsymbol{w}^{(t+1)}]\!]$ to local clients
end

Convergence Rate

First theoretical analysis on the convergence rate of FGL with cross-client edges

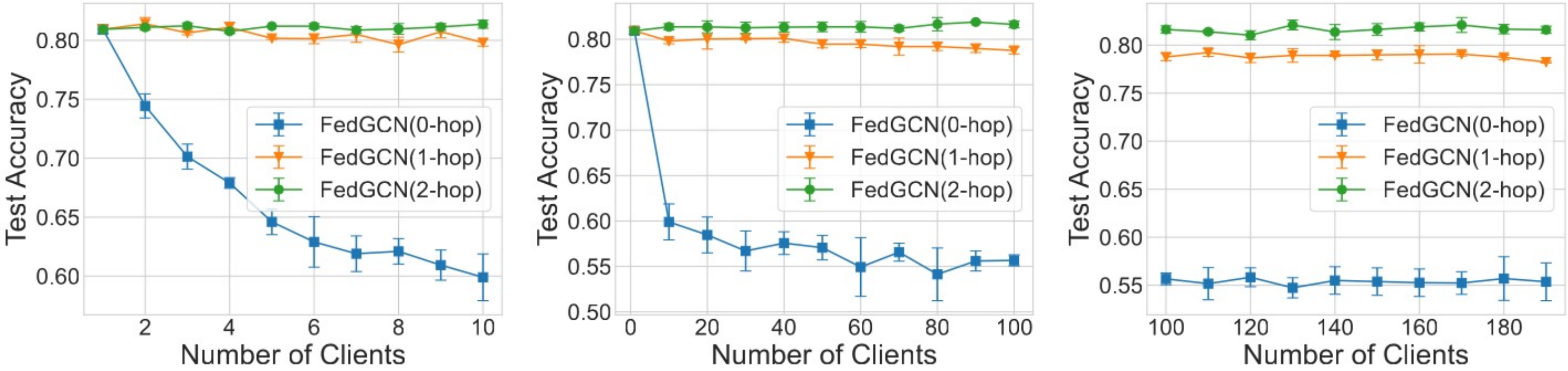
	Non-i.i.d.	i.i.d.
0-hop	$(1-\frac{1}{K^4})\frac{N^5}{C^5}\|B^4\|+(1-\frac{1}{C})^{\frac{5}{2}}(1-p)^5$	$(1-\frac{1}{K^4})\frac{N^5}{C^5}\|B^4\|$
1-hop	$(1-\frac{1}{K^4}(1+c_\alpha p+c_\mu)^2)\frac{N^5}{C^5}\|B^4\|+(1-\frac{1}{C})^{\frac{5}{2}}(1-p)^5$	$(1-\frac{1}{K^4}(1+c_\alpha+c_\mu)^2)\frac{N^5}{C^5}\|B^4\|$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

- Faster convergence with more communication hops
- More hops are needed for cross-device FL
- One-hop is sufficient for cross-silo FL

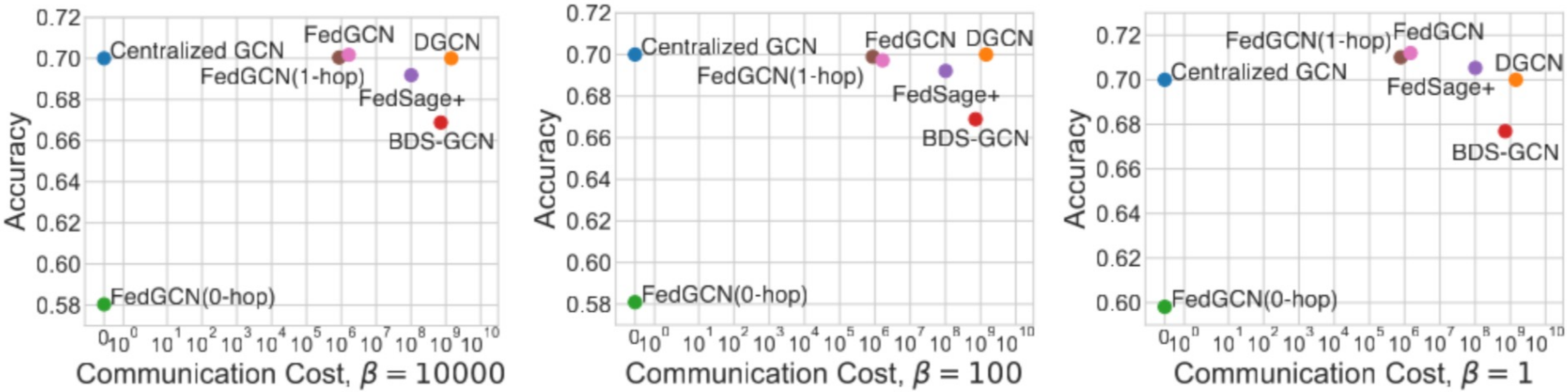
Carnegie Mellon University

Empirical Validation

- Faster convergence with more communication hops
- More hops are needed for cross-device FL
- One-hop is sufficient for cross-silo FL



Test Accuracy vs Communication Cost on OGBN-ArXiv



- ➢ FedGCN (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

Carnegie Mellon University

Test Accuracy on four datasets

	Cora, 10 clients			Citeseer, 10 clients		
Centralized GCN	0.8069±0.0065			0.6914±0.0051		
	$\beta = 1$	$\beta = 100$	$\beta = 10000$	$\beta = 1$	$\beta = 100$	$\beta = 10000$
FedGCN(0-hop)	0.6502±0.0127	0.5958±0.0176	0.5992±0.0226	0.617±0.0118	0.5841±0.0168	0.5841±0.0138
BDS-GCN	0.7598±0.0143	0.7467±0.0117	0.7479±0.018	0.6709±0.0184	0.6596±0.0128	0.6582±0.01
FedSage+	0.8026±0.0054	0.7942±0.0075	0.796±0.0075	0.6977±0.0097	0.6856±0.0121	0.688±0.0086
FedGCN(1-hop)	**0.81±0.0066**	0.8009±0.007	0.8009±0.0077	**0.7006±0.0071**	0.6891±0.0067	0.693±0.0069
FedGCN(2-hop)	0.8064±0.0043	**0.8084±0.0051**	**0.8087±0.0061**	0.6933±0.0067	**0.6953±0.0069**	**0.6948±0.0032**
	Ogbn-Arxiv, 10 clients			Ogbn-Products, 5 clients		
Centralized GCN	0.7±0.0082			0.7058±0.0008		
	$\beta = 1$	$\beta = 100$	$\beta = 10000$	$\beta = 1$	$\beta = 100$	$\beta = 10000$
FedGCN(0-hop)	0.5981±0.0094	0.5809±0.0017	0.5804±0.0015	0.6789±0.0031	0.658±0.0008	0.658±0.0008
BDS-GCN	0.6769±0.0086	0.6689±0.0024	0.6688±0.0015	0.6996±0.0019	0.6952±0.0012	0.6952±0.0009
FedSage+	0.7053±0.0073	0.6921±0.0014	0.6918±0.0024	0.7044±0.0017	0.7047±0.0009	0.7051±0.0006
FedGCN(1-hop)	0.7101±0.0078	**0.6989±0.0038**	0.7004±0.0031	0.7049±0.0016	**0.7057±0.0003**	**0.7057±0.0004**
FedGCN(2-hop)	**0.712±0.0075**	0.6972±0.0075	**0.7017±0.0081**	**0.7053±0.002**	**0.7057±0.0009**	0.7055±0.0006

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

- Library development: pip install fedgraph

Distributed Trainig Code: https://github.com/yh-yao/FedGCN

Another Dataset Paper in NeurIPS: Wyze Rule: Federated Rule Dataset for Rule Recommendation Benchmarking

Carnegie Mellon University