MLSys

FedGraphNN: A Federated Learning System and Benchmark for Graph Neural Networks

Chaoyang He*, Keshav Balasubramanian*, Emir Ceyani*, Yu Rong, Peilin Zhao, Junzhou Huang, Murali Annavaram, Salman Avestimehr

Overview

Graph Neural Networks (GNN)

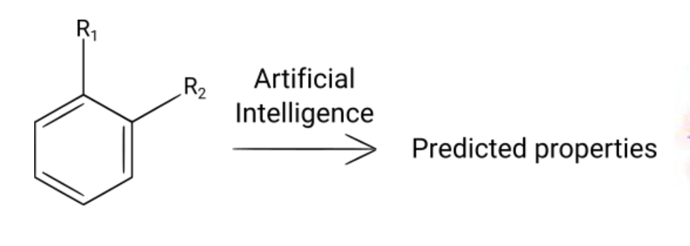
- SoTA methods for graph representations
- Many real-world graph data are **decentralized**

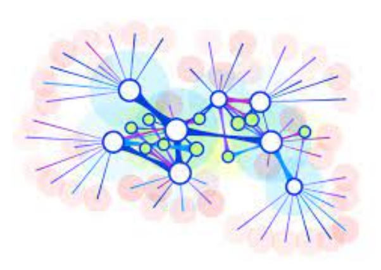
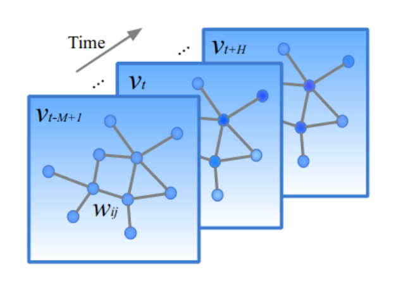
Federated Learning (FL)

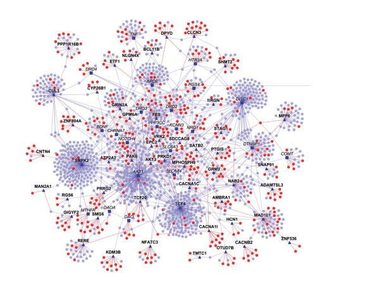
- Decentralized learning under privacy
- Federated GNNs are **ill-defined.**

FedGraphNN

Motivating Examples



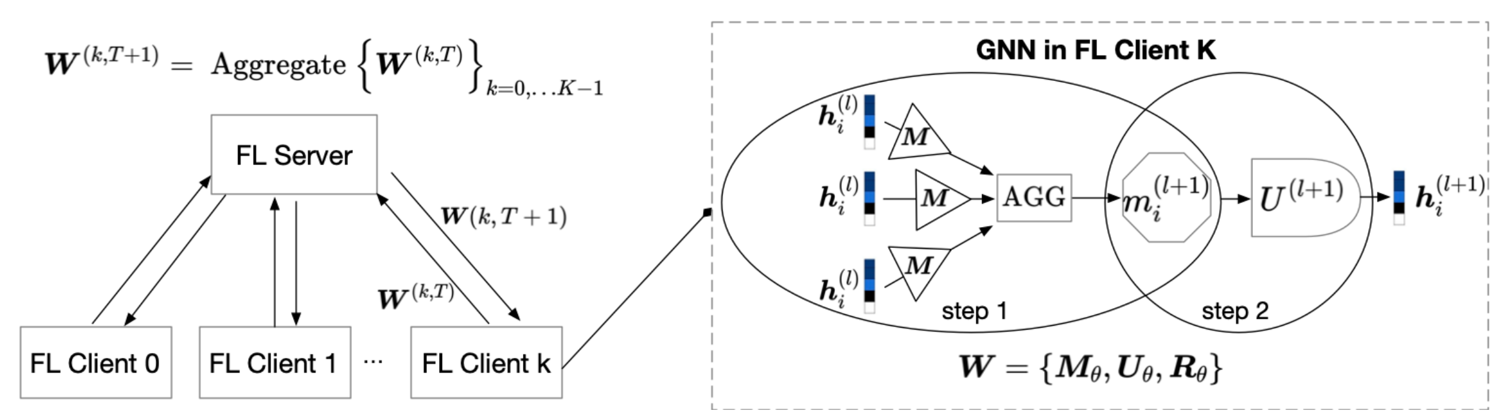


Molecular Property Prediction	Knowledge Graphs	Spatio-temporal Forecasting	Protein-Protein Interactions
↕	↕	↕	↕
Graph-Level	Sub-graph Level	Node Level	Link Level

Contributions:

- An open-source federated learning system for GNNs, namely **FedGraphNN**
- A large-scale federated molecular dataset (hERG) for further research exploration.

Problem Formulation: Federated GNN's



Say, k^{th} client owns a dataset $\mathscr{D}^{(k)} := \left\{\left(G_i^{(k)}, y_i^{(k)}\right)\right\}_{i=1}^{N^{(k)}}$, where $G_i^{(k)} = (\mathscr{V}_i^{(k)}, \mathscr{E}_i^{(k)})$ is the i^{th} graph sample in $\mathscr{D}^{(k)}$ with node & edge feature sets $X^{(k)} = \{x_m^{(k)}\}_{m\in\mathscr{V}_i^{(k)}}$ & $Z^{(k)} = \{e_{m,n}^{(k)}\}_{m,n\in\mathscr{V}_i^{(k)}}$, $y_i^{(k)}$ is the multi-class label of $G_i^{(k)}$. Each client also owns a L-layer MPNN formalized as :

$$m_i^{(k,\ell+1)} = \text{AGG}\left(\left\{M_\theta^{(k,\ell+1)}\left(h_i^{(k,\ell)}, h_j^{(k,\ell)}, e_{i,j}\right) \mid j \in \mathscr{N}_i\right\}\right)$$

$$h_i^{(k,\ell+1)} = U_\theta^{(k,\ell+1)}\left(h_i^{(k,\ell)}, m_i^{(k,\ell+1)}\right)$$

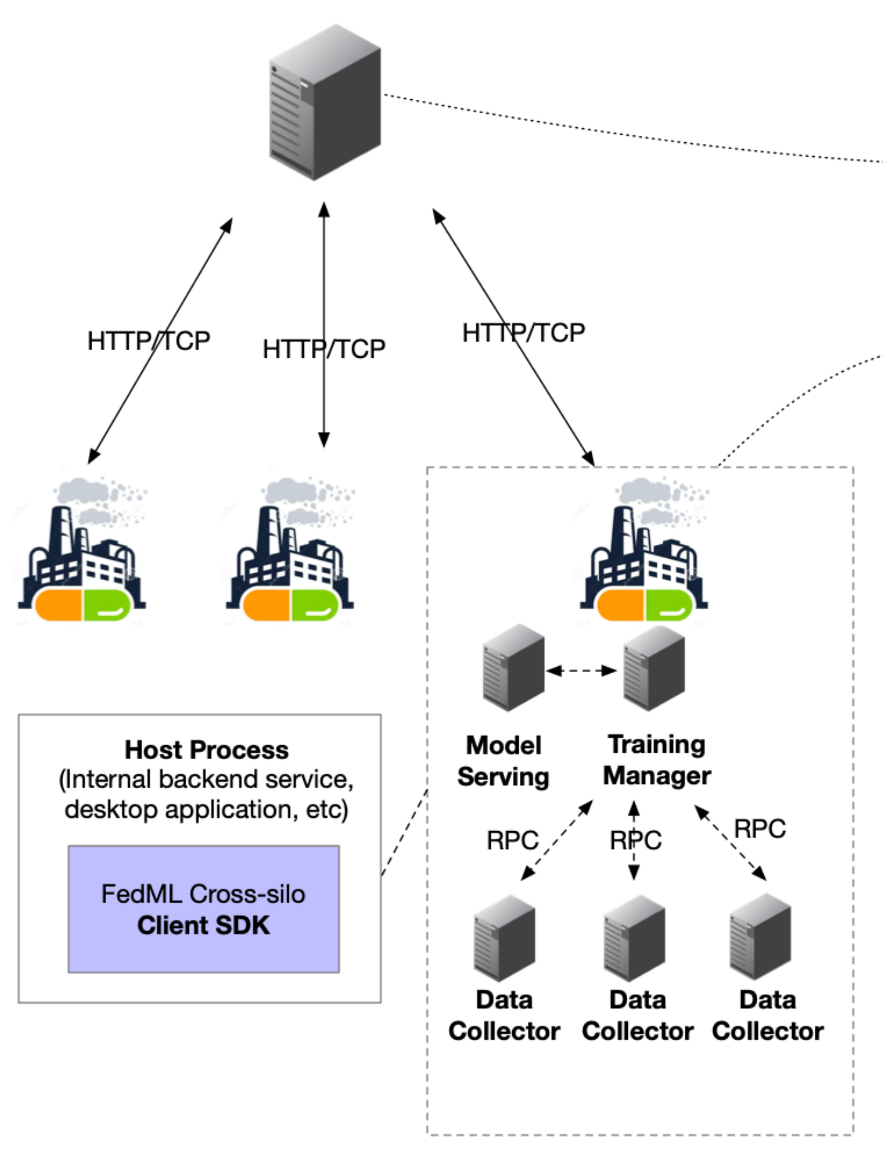
$$\hat{y}_i^{(k)} = R_\theta\left(\left\{h_j^{(k,L)} \mid j \in \mathscr{V}_i^{(k)}\right\}\right)$$

We formulate GNN-based FL as a distributed optimization problem as follows:

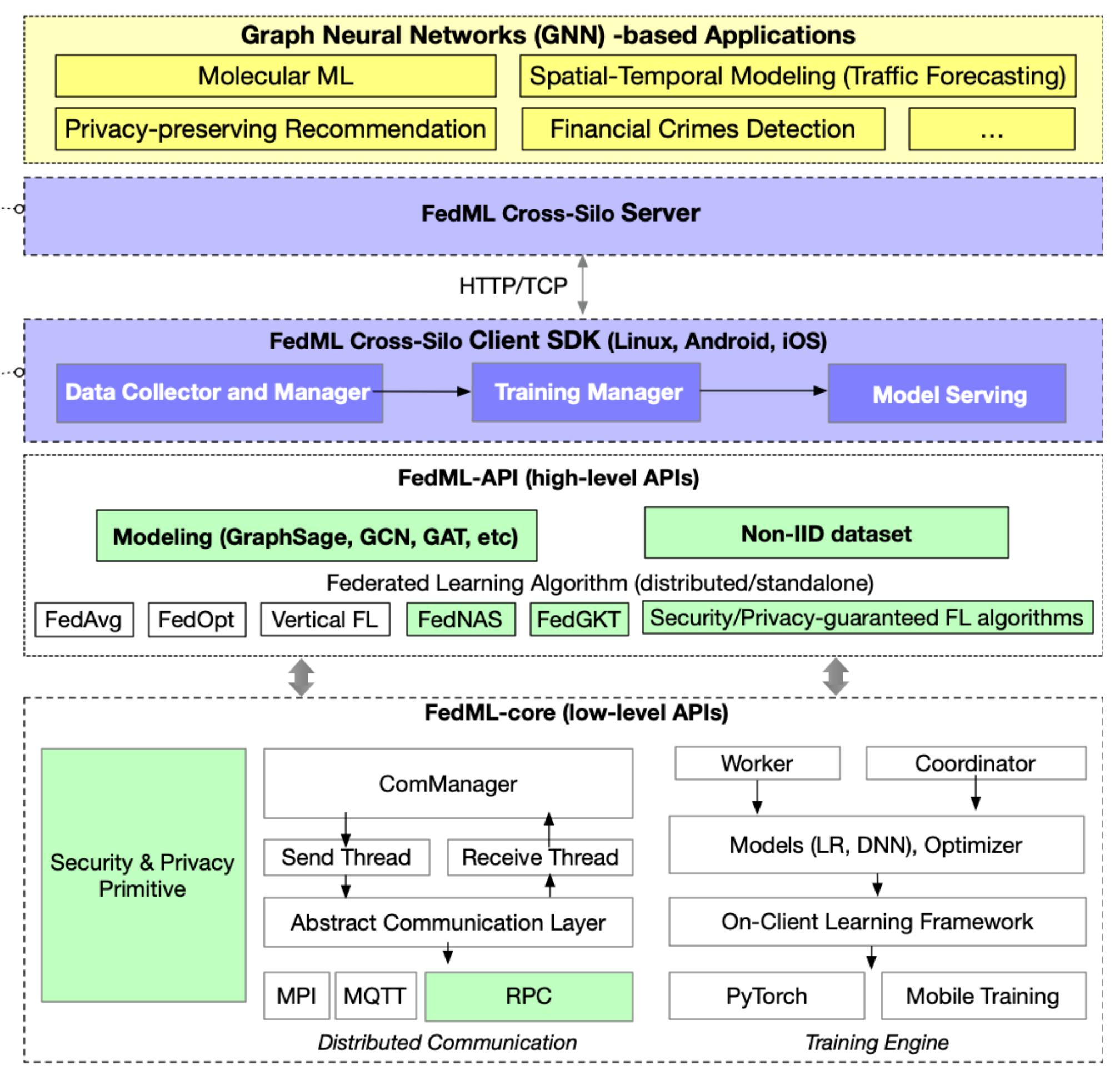
$$\min_W F(W) = \min_W \sum_{k=1}^{K} \frac{N^{(k)}}{N} \cdot f^{(k)}(W),$$

where $f^{(k)}(W) = \frac{1}{N^{(k)}} \sum_{i=1}^{N^{(k)}} \mathscr{L}(W; X_i^{(k)}, Z_i^{(k)}, y_i^{(k)})$ is the k^{th} client's local objective function measuring the local empirical risk over dataset $\mathscr{D}^{(k)}$.

FedGraphNN System Design



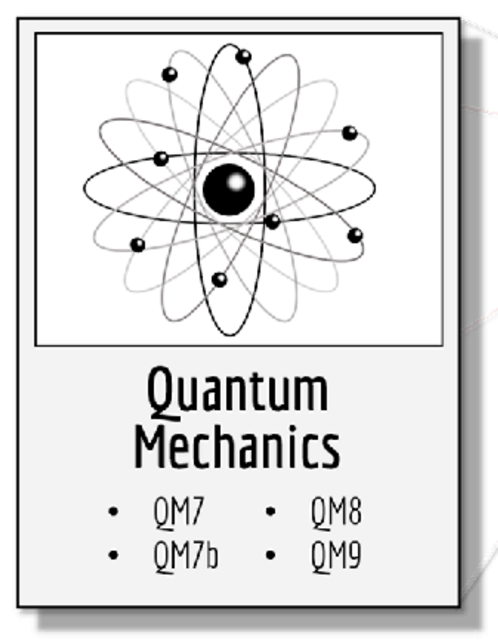
Cross-silo Federated Learning Infrastructure
(e.g., pharmaceutical factory)

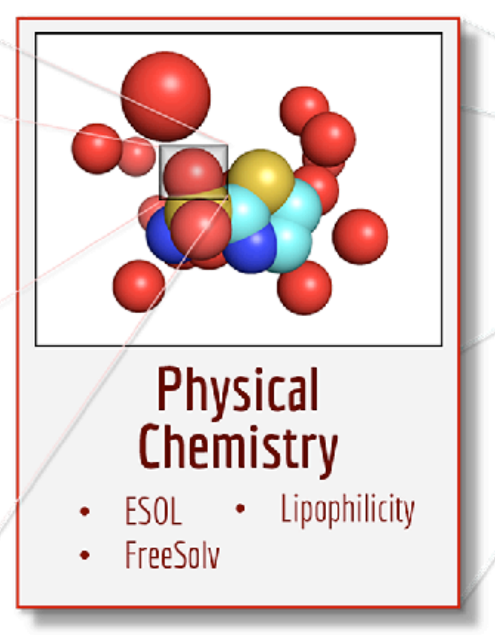

Graph Neural Networks (GNN) -based Applications
Molecular ML
Spatial-Temporal Modeling (Traffic Forecasting)
Privacy-preserving Recommendation
Financial Crimes Detection
...
FedML Cross-Silo Server
HTTP/TCP
FedML Cross-Silo Client SDK (Linux, Android, iOS)
Data Collector and Manager
Training Manager
Model Serving
FedML-API (high-level APIs)
Modeling (GraphSage, GCN, GAT, etc)
Non-IID dataset
Federated Learning Algorithm (distributed/standalone)
FedAvg
FedOpt
Vertical FL
FedNAS
FedGKT
Security/Privacy-guaranteed FL algorithms
FedML-core (low-level APIs)
Security & Privacy Primitive
ComManager
Send Thread
Receive Thread
Abstract Communication Layer
MPI
MQTT
RPC
Distributed Communication
Worker
Coordinator
Models (LR, DNN), Optimizer
On-Client Learning Framework
PyTorch
Mobile Training
Training Engine

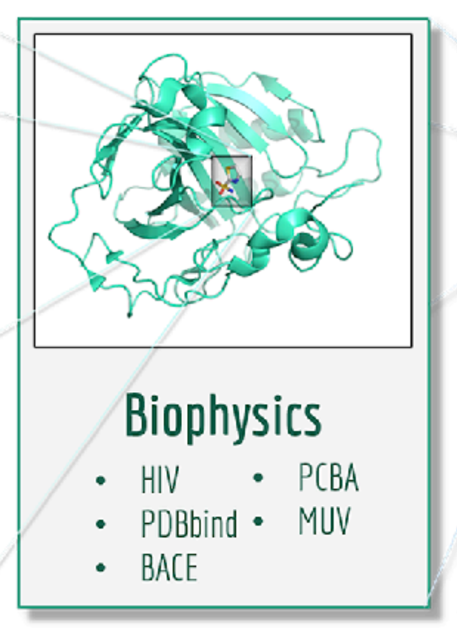
FedGraphNN System Architecture Design
(Based on FedML.ai, the blocks with colors are newly upgraded modules for FedGraphNN)

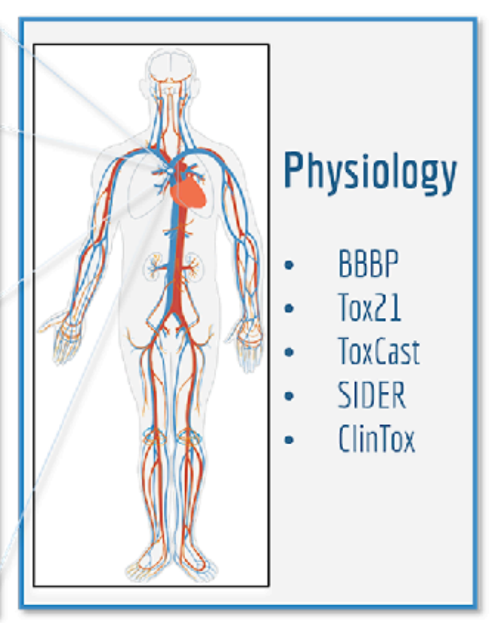
FedGraphNN Benchmark

FedGraphNN supports the MoleculeNet datasets:





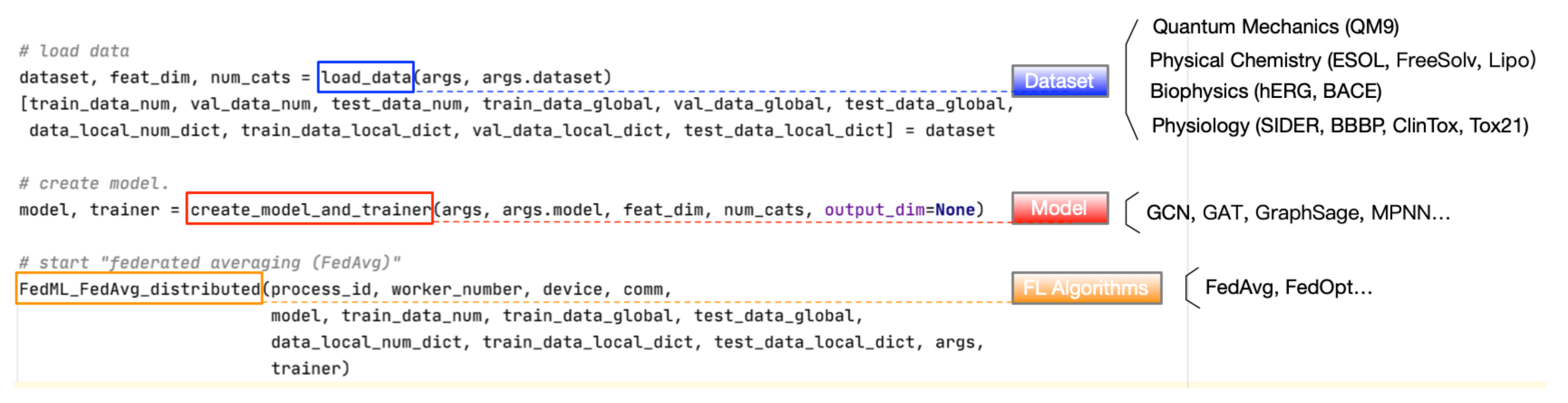




FedGraphNN currently supports the following GNN models & FL optimizers:

- GCN
- GAT
- GraphSage

- FedAvg
- Split Learning
- FedOpt
- FedNova



```
# load data
dataset, feat_dim, num_cats = load_data(args, args.dataset)
[train_data_num, val_data_num, test_data_num, train_data_global, val_data_global, test_data_global,
 data_local_num_dict, train_data_local_dict, val_data_local_dict, test_data_local_dict] = dataset

# create model.
model, trainer = create_model_and_trainer(args, args.model, feat_dim, num_cats, output_dim=None)

# start "federated averaging (FedAvg)"
FedML_FedAvg_distributed(process_id, worker_number, device, comm,
                         model, train_data_num, train_data_global, test_data_global,
                         data_local_num_dict, train_data_local_dict, test_data_local_dict, args,
                         trainer)
```

Dataset
Quantum Mechanics (QM9)
Physical Chemistry (ESOL, FreeSolv, Lipo)
Biophysics (hERG, BACE)
Physiology (SIDER, BBBP, ClinTox, Tox21)
Model
GCN, GAT, GraphSage, MPNN...
FL Algorithms
FedAvg, FedOpt...

Figure 3: Example code for benchmark evaluation with `FedGraphNN`

Experimental Results

Table 2. Classification results (higher is better)

Dataset (samples)	Non-I.I.D. Partition Method	GNN Model	Federated Optimizer	Performance Metric	MoleculeNet Results	Score on Centralized Training	Score on Federated Training
SIDER (1427)	LDA with $\alpha = 0.2$ 4 clients	GCN	FedAvg	ROC-AUC	0.638	0.6476	0.6266 (↓ 0.0210)
		GAT				0.6639	0.6591 (↓ 0.0048)
		GraphSAGE				0.6669	0.6700 (↑ 0.0031)
BACE (1513)	LDA with $\alpha = 0.5$ 4 clients	GCN	FedAvg	ROC-AUC	0.806	0.7657	0.6594 (↓ 0.1063)
		GAT				0.9221	0.7714 (↓ 0.1507)
		GraphSAGE				0.9266	0.8604 (↓ 0.0662)
Clintox (1478)	LDA with $\alpha = 0.5$ 4 clients	GCN	FedAvg	ROC-AUC	0.832	0.8914	0.8784 (↓ 0.0130)
		GAT				0.9573	0.9129 (↓ 0.0444)
		GraphSAGE				0.9716	0.9246 (↓ 0.0470)
BBBP (2039)	LDA with $\alpha = 2$ 4 clients	GCN	FedAvg	ROC-AUC	0.690	0.8705	0.7629 (↓ 0.1076)
		GAT				0.8824	0.8746 (↓ 0.0078)
		GraphSAGE				0.8930	0.8935 (↑ 0.0005)
Tox21 (7831)	LDA with $\alpha = 3$ 8 clients	GCN	FedAvg	ROC-AUC	0.829	0.7800	0.7128 (↓ 0.0672)
		GAT				0.8144	0.7186 (↓ 0.0958)
		GraphSAGE				0.8317	0.7801 (↓ 0.0516)

*Note: to reproduce the result, please use the same random seeds we set in the library.

Table 4. Training time with FedAvg on GNNs (Hardware: 8 x NVIDIA Quadro RTX 5000 GPU (16GB/GPU); RAM: 512G; CPU: Intel Xeon Gold 5220R 2.20GHz).

		SIDER	BACE	Clintox	BBBP	Tox21	FreeSolv	ESOL	Lipo	hERG	QM9
Wall-clock Time	GCN	5m 58s	4m 57s	4m 40s	4m 13s	15m 3s	4m 12s	5m 25s	16m 14s	35m 30s	6h 48m
	GAT	8m 48s	5m 27s	7m 37s	5m 28s	25m 49s	6m 24s	8m 36s	25m 28s	58m 14s	9h 21m
	GraphSAGE	2m 7s	3m 58s	4m 42s	3m 26s	14m 31s	5m 53s	6m 54s	15m 28s	32m 57s	5h 33m
Average FLOP	GCN	697.3K	605.1K	466.2K	427.2K	345.8K	142.6K	231.6K	480.6K	516.6K	153.9K
	GAT	703.4K	612.1K	470.2K	431K	347.8K	142.5K	232.6K	485K	521.3K	154.3K
	GraphSAGE	846K	758.6K	1.1M	980K	760.6K	326.9K	531.1K	1.5M	1.184M	338.2K
Parameters	GCN	15.1K	13.5K	13.6K	13.5K	14.2K	13.5K	13.5K	13.5K	13.5K	14.2K
	GAT	20.2K	18.5K	18.6K	18.5K	19.2K	18.5K	18.5K	18.5K	18.5K	19.2K
	GraphSAGE	10.6K	8.9K	18.2K	18.1K	18.8K	18.1K	18.1K	269K	18.1K	18.8K

*Note that we use the distributed training paradigm where each client's local training uses one GPU. Please refer our code for details.

Research Questions & Future Directions

Our key findings:

1. **How to mitigate the accuracy gap** on graph datasets with non-I.I.D.ness?
   1. Can we personalize the model for each user?
2. **How to deal with limited labels** for real-world graph data?
   1. **How to leverage** semi or self-supervised learning into GNN-based FL?
   2. **What if we do not have labels at the edge?**
3. **How to design** efficient GNN-based FL algorithms for **sub-graph, node and edge levels?**

Future Directions:

1. **Integrate more domains:**
   1. Recommendation Systems
   2. Spatiotemporal Forecasting
   3. Knowledge Graphs
2. **Enable GNN models with edge information**

Code Release

https://github.com/FedML-AI/FedGraphNN