FEDGRAPHNN: A FEDERATED LEARNING SYSTEM AND BENCHMARK FOR GRAPH NEURAL NETWORKS

Chaoyang He [*1] **Keshav Balasubramanian** [*1] **Emir Ceyani** [*1] **Yu Rong** [2] **Peilin Zhao** [2] **Junzhou Huang** [2]
Murali Annavaram [1] **Salman Avestimehr** [1]

ABSTRACT

Graph Neural Network (GNN) research is rapidly growing thanks to the capacity of GNNs to learn representations from graph-structured data. However, centralizing a massive amount of real-world graph data for GNN training is prohibitive due to user-side privacy concerns, regulation restrictions, and commercial competition. Federated learning (FL), a trending distributed learning paradigm, aims to solve this challenge while preserving privacy. Despite recent advances in vision and language domains, there is no suitable platform for the federated training of GNNs. To this end, we introduce `FedGraphNN`, an open research federated learning system and a benchmark to facilitate GNN-based FL research. `FedGraphNN` is built on a unified formulation of federated GNNs and supports commonly used datasets, GNN models, FL algorithms, and flexible APIs. We also contribute a new molecular dataset, hERG, to promote research exploration. Our experimental results present significant challenges in federated GNN training: federated GNNs perform worse in most datasets with a non-I.I.D split than centralized GNNs; the GNN model that attains the best result in centralized setting may not hold its advantage in the federated setting. These results imply that more research efforts are needed to unravel the mystery of federated GNN training. Moreover, our system performance analysis demonstrates that the `FedGraphNN` system is computationally affordable to most research labs with limited GPUs. `FedGraphNN` will be regularly updated and welcomes inputs from the community.

1 INTRODUCTION

Graph Neural Networks (GNN) are state-of-the-art models that learn representations from complex graph-structured data in various domains such as drug discovery (Rong et al., 2020b), social network recommendation (Wu et al., 2018a; Sun et al., 2019; He et al., 2019b), and traffic flow modeling (Wang et al., 2020b; Cui et al., 2019). However, for reasons such as user-side privacy, regulation restrictions, and commercial competition, there are surging real-world cases in which graph data is decentralized, limiting cilents' data size .For example, in the AI-based drug discovery industry, pharmaceutical research institutions would significantly benefit from other institutions' private data, but neither cannot afford to disclose their private data for commercial reasons. Federated Learning (FL) is a distributed learning paradigm that addresses this data isolation problem. In FL, training is an act of collaboration between multiple clients without requiring centralized local data while providing a certain degree of user-level privacy (McMahan et al., 2017; Kairouz et al., 2019; He et al., 2019a).

Despite FL being successfully applied in domains like computer vision (Liu et al., 2020; Hsu et al., 2020) and natural language processing (Hard et al., 2018; Ge et al., 2020), it has yet to be widely adopted in the domain of graph machine learning. There are multiple reasons for this:

1. Most existing FL libraries, as summarized by (He et al., 2020b), do not support GNNs. Given the complexity of graph data, the dynamics of training GNNs in a federated setting may be different from training vision or language models. A fair and easy-to-use benchmark is essential to distinguish the advantages of different GNN models and FL algorithms;

2. The definition of federated GNNs is vague in current literature. This vagueness makes it difficult for researchers who focus on SGD-based federated optimization algorithms to understand challenges in federated GNNs ;

3. Applying existing FL algorithms to GNNs is nontrivial and requires significant engineering effort to transplant and reproduce existing algorithms to GNN models and

[*]Equal contribution [1]Viterbi School of Engineering , University of Southern California, California, USA [2]Machine Learning Center, Tencent AI Lab. Correspondence to: Chaoyang He <chaoyang.he@usc.edu>.

Proceedings of the 4^{th} MLSys Conference, San Jose, CA, USA, 2021.

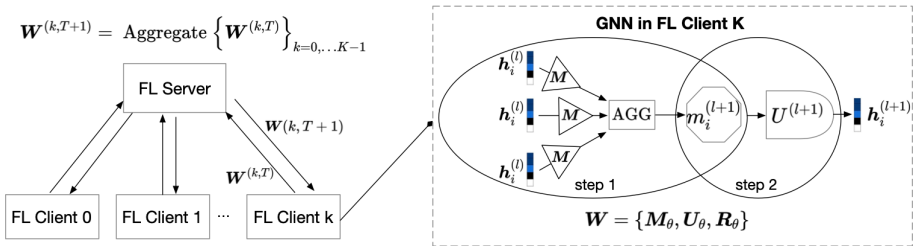


Figure 1. Formulation of `FedGraphNN` (Federated Graph Neural Network)

graph datasets. Recent works (Wang et al., 2020a; Meng et al., 2021; Wu et al., 2021), only use the naive FedAvg algorithm (McMahan et al., 2017), which we demonstrate is sub-optimal in many cases.

To address these issues, we present an open-source federated learning system for GNNs, namely `FedGraphNN`, that enables the training of a variety of GNN models effectively and efficiently in a federated setting as well as benchmarks in non-I.I.D. graph datasets (e.g., molecular graphs). We first formulate federated graph neural networks to provide a unified framework for federated GNNs (Section 3). Under this formulation, we design a federated learning system to support federated GNNs with a curated list of FL algorithms and provide low-level APIs for algorithmic research customization and deployment (Section 4). We then provide a benchmark on commonly used molecular datasets and GNNs. We also contribute a large-scale federated molecular dataset named hERG for further research exploration (Section 5).

2 Related Works

Federated Graph Neural Networks (`FedGraphNN`) lies at the intersection of graph neural networks (GNNs) and federated learning. We mainly discuss related works that train GNNs using decentralized datasets. (Suzumura et al., 2019) and (Mei et al., 2019) use computed graph statistics for information exchange and aggregation to avoid node information leakage. (Zhou et al., 2020) utilize Secure Multi-Party Computation (SMPC) and Homomorphic Encryption (HE) into GNN learning for node classification. (Zheng et al., 2020) train a global GNN model for privacy-preserving node-classification under non-IID data using Shamir's secret sharing. (Jiang et al., 2020) propose a secure aggregation method to learn dynamic representations from multi-user graph sequences. More recently, (Wang et al., 2020a) use a hybrid method of federated learning and meta-learning to solve the semi-supervised graph node classification problem in decentralized social network datasets. (Meng et al., 2021) attempts to protect the node-level privacy using an edge-cloud partitioned GNN model for spatio-temporal forecasting tasks using node-level traffic sensor datasets. Finally, (Wu et al., 2021) proposes a federated recommendation system with GNNs.

3 Formulation: Federated Graph Neural Networks

We consider a federated learning setting where graph datasets are dispersed over multiple edge servers that cannot be centralized for training due to privacy or regulation restrictions. For instance, compounds in molecular trials (Rong et al., 2020b) or knowledge graphs for recommendation systems (Chen et al., 2020) may not be shared across entities because of intellectual property concerns. Under this setting, we assume that there are K clients in the FL network, and the k^{th} client has its own dataset $\mathcal{D}^{(k)} := \left\{\left(G_i^{(k)}, y_i^{(k)}\right)\right\}_{i=1}^{N^{(k)}}$, where $G_i^{(k)} = (\mathcal{V}_i^{(k)}, \mathcal{E}_i^{(k)})$ is the i^{th} graph sample in $\mathcal{D}^{(k)}$ with node feature set $\boldsymbol{X}^{(k)} = \left\{\boldsymbol{x}_m^{(k)}\right\}_{m\in\mathcal{V}_i^{(k)}}$ and edge feature set $\boldsymbol{Z}^{(k)} = \left\{\boldsymbol{e}_{m,n}^{(k)}\right\}_{m,n\in\mathcal{V}_i^{(k)}}$, $y_i^{(k)}$ is the corresponding multiclass label of $G_i^{(k)}$, $N^{(k)}$ is the sample number in dataset $\mathcal{D}^{(k)}$, and $N = \sum_{k=1}^{K} N^{(k)}$. Each client owns a Graph Neural Network (GNN) model to learn graph or node-level representations. Multiple clients are interested in collaborating through a server to improve their GNN models without necessarily revealing their graphs.

We illustrate the formulation of Federated Graph Neural Network (`FedGraphNN`) in Figure 1. Without loss of generality, we use a Message Passing Neural Network (MPNN) framework (Gilmer et al., 2017; Rong et al., 2020c). Most of the spatial-based GNN models (Kipf & Welling, 2016; Veličković et al., 2018; Hamilton et al., 2017) can be unified into this framework, where the forward pass has two phases: a message-passing phase and a readout phase. The message passing phase contains two steps: First, the model gathers and transforms the neighbors' messages. Then, the model uses aggregated messages to update node hidden states. Mathematically, for client k and for layer indices $\ell = 0, \dots, L-1$, a L-layer MPNN is formalized as follows:

$$\boldsymbol{m}_i^{(k,\ell+1)} = \text{AGG}\left(\left\{\boldsymbol{M}_\theta^{(k,\ell+1)}\left(\boldsymbol{h}_i^{(k,\ell)}, \boldsymbol{h}_j^{(k,\ell)}, \boldsymbol{e}_{i,j}\right) \mid j \in \mathcal{N}_i\right\}\right) \quad (1)$$

$$\boldsymbol{h}_i^{(k,\ell+1)} = \boldsymbol{U}_\theta^{(k,\ell+1)}\left(\boldsymbol{h}_i^{(k,\ell)}, \boldsymbol{m}_i^{(k,\ell+1)}\right) \quad (2)$$

where $\boldsymbol{h}_i^{(k,0)} = \boldsymbol{x}_i^{(k)}$ is the k^{th} client's node features, ℓ is the layer index, `AGG` is the aggregation function (e.g., in

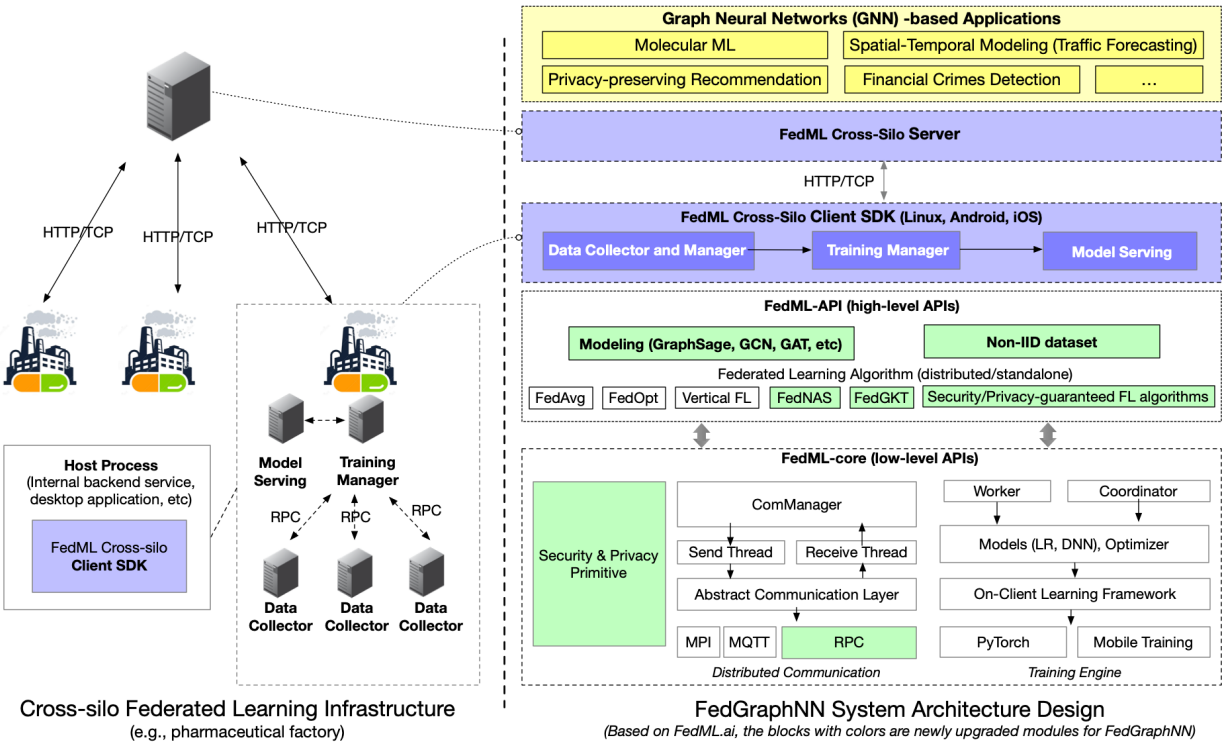


Figure 2. Overview of `FedGraphNN` System Architecture Design

the GCN model, the aggregation function is a simple `SUM` operation), $\mathcal{N}_i$ is the neighborhood set of node i (e.g., 1-hop neighbors), and $\boldsymbol{M}_\theta^{(k,\ell+1)}(\cdot)$ is the message generation function which takes the hidden state of current node $\boldsymbol{h}_i$, the hidden state of the neighbor node $\boldsymbol{h}_j$ and the edge features $\boldsymbol{e}_{i,j}$ as inputs. $\boldsymbol{U}_\theta^{(k,\ell+1)}(\cdot)$ is the state update function receiving the aggregated feature $\boldsymbol{m}_i^{(k,\ell+1)}$. After propagating through an L-layer MPNN, the readout phase computes a feature vector for downstream tasks (node-level or graph-level). For example, we can obtain the whole graph representation using some readout function $\boldsymbol{R}_\theta(\cdot)$ according to:

$$\hat{y}_i^{(k)} = \boldsymbol{R}_\theta\left(\left\{h_j^{(k,L)} \mid j \in \mathcal{V}_i^{(k)}\right\}\right) \tag{3}$$

To formulate GNN-based FL, we define $\boldsymbol{W} = \{\boldsymbol{M}_\theta, \boldsymbol{U}_\theta, \boldsymbol{R}_\theta\}$ as the overall learnable weights in client k. In general, $\boldsymbol{W}$ is independent of graph structure (i.e., GNN models are normally inductive and generalize to unseen graphs). Consequently, we formulate GNN-based FL as a distributed optimization problem as follows:

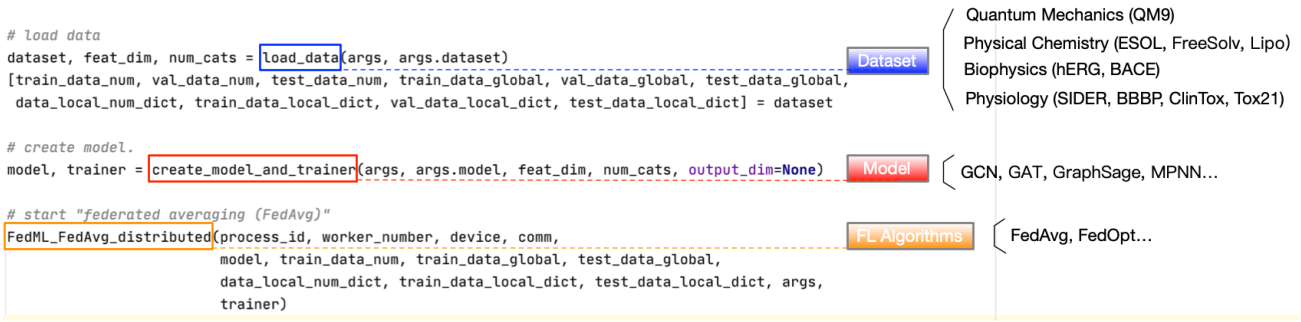
$$\min_{\boldsymbol{W}} F(\boldsymbol{W}) \overset{\text{def}}{=} \min_{\boldsymbol{W}} \sum_{k=1}^{K} \frac{N^{(k)}}{N} \cdot f^{(k)}(\boldsymbol{W}) \tag{4}$$

where $f^{(k)}(\boldsymbol{W}) = \frac{1}{N^{(k)}} \sum_{i=1}^{N^{(k)}} \mathcal{L}(\boldsymbol{W}; \boldsymbol{X}_i^{(k)}, \boldsymbol{Z}_i^{(k)}, y_i^{(k)})$ is the kth client's local objective function that measures the local empirical risk over the heterogeneous graph dataset $\mathcal{D}^k$. $\mathcal{L}$ is the loss function of the global GNN model. To solve this problem, we utilize FedAvg (McMahan et al., 2017). Other advanced algorithms such as FedOPT (Reddi et al., 2020) and FedGKT (He et al., 2020a) can also be applied.

4 FedGraphNN Federated Learning System

We develop an open-source federated learning system for GNNs, named `FedGraphNN`, which includes implementations of standard baseline datasets, models, and federated learning algorithms for GNN-based FL research. `FedGraphNN` aims to enable efficient and flexible customization for future exploration.

As shown Figure 2, `FedGraphNN` is built based on `FedML` research library (He et al., 2020b) which is a widely used FL library, but without any GNN support as yet. To distinguish FedGraphNN over FedML, we color-coded the modules that specific to `FedGraphNN`. In the lowest layer, `FedGraphNN` reuses `FedML-core` APIs but further supports tensor-aware RPC (remote procedure call), which enables the communication between servers located at different data centers (e.g., different pharmaceutical vendors). Enhanced security and privacy primitive modules are added




```
# 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)
```

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

Dataset	Dataset	# Tasks	Task Type	# Compounds	Average # of Nodes	Average # of Edges	Rec - Metric
Quantum Mechanics	QM9 (Gaulton et al., 2012)	12	Regression	133885	8.80	27.60	MAE
Physical Chemistry	ESOL (Delaney, 2004)	1	Regression	1128	13.29	40.65	RMSE
	FreeSolv(Mobley & Guthrie, 2014)	1	Regression	642	8.72	25.60	RMSE
	Lipophilicity (Gaulton et al., 2012)	1	Regression	4200	27.04	86.04	RMSE
Biophysics	hERG(Gaulton et al., 2016; Kim et al., 2021)	1	Regression	10572	29.39	94.09	RMSE
	BACE (Subramanian et al., 2016)	1	Classification	1513	34.09	36.89	ROC-AUC
Physiology	BBBP (Martins et al., 2012)	1	Classification	2039	24.03	25.94	ROC-AUC
	SIDER (Kuhn et al., 2016)	27	Classification	1427	33.64	35.36	ROC-AUC
	ClinTox (Gayvert et al., 2016)	2	Classification	1478	26.13	27.86	ROC-AUC
	Tox21 (tox)	12	Classification	7831	18.51	25.94	ROC-AUC

Table 1. Summary of Molecular Machine Learning Datasets

to support techniques such as secure aggregation in upper layers. The layer above supports plug and play operation of common GNN models such as GraphSage and GAT. Given that graphs are likely to exhibit strong non-IID behavior, we provide dedicated modules to handle non-IID split algorithms and data loaders. Users can either reuse our data distribution or manipulate the non-IIDness by setting hyperparameters. Example code is shown in Figure 3. We introduce details of `FedML Client SDK` in the Appendix A.

5 FedGraphNN Benchmark: Datasets, Models, and Algorithms

Non-I.I.D. Datasets. To facilitate GNN-based FL research, we plan to support various graph datasets with non-IIDness in different domains such as molecule machine learning, knowledge graph, and recommendation system. In the latest release, we use MoleculeNet (Wu et al., 2018b), a molecule machine learning benchmark, as the data source to generate our non-I.I.D. benchmark datasets. Specially, we use the unbalanced partition algorithm Latent Dirichlet Allocation (LDA) (He et al., 2020b) to partition datasets in the MoleculeNet benchmark. Besides, we provide a new dataset, named hERG, related to cardiac toxicity and collected from (Kim et al., 2021; Gaulton et al., 2017) with data cleaning. Table 5 in the Appendix summarizes all datasets we used in experiments. Figure 4 in the Appendix shows each dataset's non-IID distribution. More details and their specific preprocessing details can be found in Appendix B.1 & B.2.

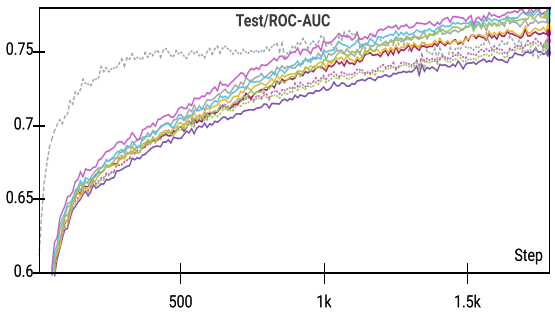
GNN Models and Federated Learning Algorithms. In the latest release, `FedGraphNN` supports GCN (Kipf & Welling, 2016), GAT (Veličković et al., 2018), and GraphSage (Hamilton et al., 2017) as the GNN models. The readout function currently supported is a simple Multilayer Perceptron (MLP). Users can easily plug their customized GNN models and readout functions into our framework. For FL algorithms, besides FedAvg (McMahan et al., 2017), other advanced algorithms such as FedOPT (Reddi et al., 2020) and FedGKT (He et al., 2020a) are also supported. We refer to Appendix B.4 for the details on FL algorithms and GNN models.

6 Experiments

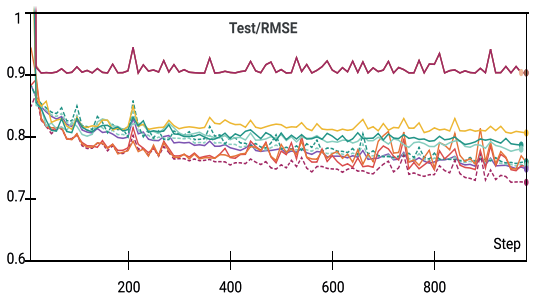
Implementation and Hyper-parameters. Experiments were conducted on a GPU server equipped with 8 NVIDIA Quadro RTX 5000 (16GB GPU memory). We built the benchmark with FedAvg algorithm for three GNN models (GCN, GAT, and GraphSage) on various MoleculeNet datasets with different scales of sample numbers. The hyper-parameters we used in experiments can be found in the Appendix C.

6.1 Result of Model Accuracy on Non-I.I.D.

We run experiments on both classification tasks and regression tasks. Hyper-parameters are tuned (sweeping) by grid search (see Section D for the search space). Figures 4(a) and 4(b) use GraphSage on Tox21 and hERG as examples to show the test score curve during sweeping. We maintain



(a) Tox21: test score during sweeping



(b) hERG: test score during sweeping

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 GAT GraphSAGE	FedAvg	ROC-AUC	0.638	0.6476 0.6639 0.6669	0.6266 (↓ 0.0210) 0.6591 (↓ 0.0048) 0.6700 (↑ 0.0031)
BACE (1513)	LDA with $\alpha = 0.5$ 4 clients	GCN GAT GraphSAGE	FedAvg	ROC-AUC	0.806	0.7657 0.9221 0.9266	0.6594 (↓ 0.1063) 0.7714 (↓ 0.1507) 0.8604 (↓ 0.0662)
Clintox (1478)	LDA with $\alpha = 0.5$ 4 clients	GCN GAT GraphSAGE	FedAvg	ROC-AUC	0.832	0.8914 0.9573 0.9716	0.8784 (↓ 0.0130) 0.9129 (↓ 0.0444) 0.9246 (↓ 0.0470)
BBBP (2039)	LDA with $\alpha = 2$ 4 clients	GCN GAT GraphSAGE	FedAvg	ROC-AUC	0.690	0.8705 0.8824 0.8930	0.7629 (↓ 0.1076) 0.8746 (↓ 0.0078) 0.8935 (↑ 0.0005)
Tox21 (7831)	LDA with $\alpha = 3$ 8 clients	GCN GAT GraphSAGE	FedAvg	ROC-AUC	0.829	0.7800 0.8144 0.8317	0.7128 (↓ 0.0672) 0.7186 (↓ 0.0958) 0.7801 (↓ 0.0516)

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

Table 3. Regression results (lower is better)

Dataset	Non-I.I.D. Partition Method	GNN Model	Federated Optimizer	Performance Metric	MoleculeNet Result	Score for Centralized Training	Score for Federated Training
FreeSolv (642)	LDA with $\alpha = 0.5$ 4 clients	GCN GAT GraphSAGE	FedAvg	RMSE	1.40 ± 0.16	1.5787 1.2175 1.3630	2.7470 (↑ 1.1683) 1.3130 (↑ 0.0925) 1.6410 (↑ 0.2780)
ESOL (1128)	LDA with $\alpha = 2$ 4 clients	GCN GAT GraphSAGE	FedAvg	RMSE	0.97 ± 0.01	1.0190 0.9358 0.8890	1.4350 (↑ 0.4160) 0.9643 (↑ 0.4382) 1.1860 (↑ 0.2970)
Lipo (4200)	LDA with $\alpha = 2$ 8 clients	GCN GAT GraphSAGE	FedAvg	RMSE	0.655 ± 0.036	0.8518 0.7465 0.7078	1.1460 (↑ 0.2942) 0.8537 (↑ 0.2575) 0.7788 (↑ 0.0710)
hERG (10572)	LDA with $\alpha = 3$ 8 clients	GCN GAT GraphSAGE	FedAvg	RMSE	-	0.7257 0.6271 0.7132	0.7944 (↑ 0.0687) 0.7322 (↑ 0.1051) 0.7265 (↑ 0.0133)
QM9 (133885)	LDA with $\alpha = 3$ 8 clients	GCN GAT GraphSAGE	FedAvg	MAE	2.35	14.78 12.44 13.06	21.075 (↑ 6.295) 23.173 (↑ 10.733) 19.167 (↑ 6.107)

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

such intermediate results for all datasets in our source code. After hyper-parameter tuning, we report all results in Table 2 and Table 3. For each result, the optimal hyper-parameters can be found in the Appendix C.

There are multiple takeaways from these results:

1. When graph datasets are small, FL accuracy is on par with (or even better than) centralized learning.

2. But when dataset sizes grow, FL accuracy becomes worse than the centralized approach. In larger datasets, the non-I.I.D. nature of graphs leads to an accuracy drop.

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.

3. The dynamics of training GNNs in a federated setting are different from training federated vision or language models. Our findings show that the best model in the centralized setting may not necessarily be the best model in the non-I.I.D. federated setting. Interestingly, we find that GAT suffers the most considerable performance compromise on 5 out of 9 datasets. This may be due to the sensibility of the attention calculation on the non-IID settings.

Hence, additional research is needed to understand the nuances of training GNNs in a federated setting and bridge this gap.

6.2 System Performance Analysis

We also present system performance analysis when using MPI as the communication backend. The results are summarized in Table 4. Even on large datasets, Federated training can be completed under 1 hour using only 4 GPUs, except the QM9 dataset, which requires hours to finish training. `FedGraphNN` thus provides an efficient mapping of algorithms to the underlying resources, thereby making it attractive for deployment.

The training time using RPC is also evaluated. Its evaluation result is similar to that of using MPI. More details can be found in our source code. Note that RPC is useful for realistic deployment when GPU/CPU-based edge devices can only be accessed via public IP addresses due to locating in different data centers. We will provide test result in such a scenario in our future work.

7 Discussion

Federated Graph Neural Networks (`FedGraphNN`) is still in its early stage. Our vision is that `FedGraphNN` should cover four types of GNN-based federated learning:

1. *Graph level*. We believe molecular machine learning is a paramount application in this setting, where many small graphs are distributed between multiple edge devices; 2. *Sub-graph level*. This scenario typically pertains to social networks or knowledge graphs that need to be partitioned into many small sub-graphs due to data barriers between different departments in a giant company, as demonstrated in (Wu et al., 2021). 3. *Node level*. When the privacy of a specific node in a graph is important, node-level GNN-based FL is useful in practice. The IoT setting is a good example (Zheng et al., 2020); 4. *Link level* is also a promising direction that is relevant when the privacy of edges (eg: connections in a social network) is of importance.

Although the current version of `FedGraphNN` only contains graph-level GNN-based FL, other scenarios are also in our plan. We welcome contribution from the machine learning community to enrich our benchmark.

8 Future Works and Conclusion

Here we highlight some future research directions for consideration: 1. Supporting more graph datasets and GNN models for diverse applications; 2. Optimizing the system to accelerate the training speed for large-scale graph datasets; 3. Proposing advanced FL algorithms or GNN models to mitigate the accuracy gap on datasets with non-IIDness; 4. Real-world graph data often has limited labels. However, existing FL algorithms are mainly for supervised learning. Exploring semi-supervised or self-supervised learning methods is essential for realistic GNN-based FL applications.

In this paper, we design a federated learning (FL) system and benchmark for federated graph neural networks (GNN), named `FedGraphNN`. `FedGraphNN` includes implementations of baseline datasets, models, and federated learning algorithms. Our system performance analysis shows that GNN-based FL research is affordable to most research labs. We hope `FedGraphNN` can serve as an easy-to-follow research platform for researchers to explore vital problems at the intersection of federated learning and graph neural networks.

REFERENCES

Tox21 challenge. https://tripod.nih.gov/tox21/challenge/. Accessed: 2021-03-01.

Bemis, G. W. and Murcko, M. A. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

Chen, C., Cui, J., Liu, G., Wu, J., and Wang, L. Survey and open problems in privacy preserving knowledge graph: Merging, query, representation, completion and applications. *arXiv preprint arXiv:2011.10180*, 2020.

Cui, Z., Henrickson, K., Ke, R., Pu, Z., and Wang, Y. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting, 2019.

Delaney, J. S. Esol: estimating aqueous solubility directly from molecular structure. *Journal of chemical information and computer sciences*, 44(3):1000–1005, 2004.

Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

Gaulton, A., Hersey, A., Nowotka, M., Bento, A. P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrián-Uhalte, E., Davies, M., Dedman, N., Karlsson, A., Magariños, M. P., Overington, J. P., Papadatos, G., Smit, I., and Leach, A. R. The ChEMBL database in 2017. *Nucleic Acids Research*, 45(D1):D945–D954, 11 2016. ISSN 0305-1048. doi: 10.1093/nar/gkw1074. URL https://doi.org/10.1093/nar/gkw1074.

Gaulton, A., Hersey, A., Nowotka, M., Bento, A. P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrián-Uhalte, E., et al. The chembl database in 2017. *Nucleic acids research*, 45(D1):D945–D954, 2017.

Gayvert, K. M., Madhukar, N. S., and Elemento, O. A data-driven approach to predicting successes and failures of clinical trials. *Cell chemical biology*, 23(10):1294–1301, 2016.

Ge, S., Wu, F., Wu, C., Qi, T., Huang, Y., and Xie, X. Fedner: Privacy-preserving medical named entity recognition with federated learning. *arXiv e-prints*, pp. arXiv–2003, 2020.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pp. 1263–1272. PMLR, 2017.

Gupta, O. and Raskar, R. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8, 2018.

Hagberg, A. A., Schult, D. A., and Swart, P. J. Exploring network structure, dynamics, and function using networkx. In Varoquaux, G., Vaught, T., and Millman, J. (eds.), *Proceedings of the 7th Python in Science Conference*, pp. 11 – 15, Pasadena, CA USA, 2008.

Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017. URL http://arxiv.org/abs/1706.02216.

Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

He, C., Tan, C., Tang, H., Qiu, S., and Liu, J. Central server free federated learning over single-sided trust social networks. *arXiv preprint arXiv:1910.04956*, 2019a.

He, C., Xie, T., Rong, Y., Huang, W., Huang, J., Ren, X., and Shahabi, C. Cascade-bgnn: Toward efficient self-supervised representation learning on large-scale bipartite graphs. 2019b.

He, C., Annavaram, M., and Avestimehr, S. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems*, 33, 2020a.

He, C., Li, S., So, J., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., Shen, L., Zhao, P., Kang, Y., Liu, Y., Raskar, R., Yang, Q., Annavaram, M., and Avestimehr, S. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020b.

Hsu, T.-M. H., Qi, H., and Brown, M. Federated visual classification with real-world data distribution. *arXiv preprint arXiv:2003.08082*, 2020.

Jiang, M., Jung, T., Karl, R., and Zhao, T. Federated dynamic gnn with secure aggregation. *arXiv preprint arXiv:2009.07351*, 2020.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., et al. Pubchem in 2021: new data content and improved

web interfaces. *Nucleic Acids Research*, 49(D1):D1388–D1395, 2021.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL http://arxiv.org/abs/1609.02907.

Kuhn, M., Letunic, I., Jensen, L. J., and Bork, P. The sider database of drugs and side effects. *Nucleic acids research*, 44(D1):D1075–D1079, 2016.

Landrum, G. et al. Rdkit: Open-source cheminformatics. 2006.

Li, Q., Han, Z., and Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Liu, Y., Huang, A., Luo, Y., Huang, H., Liu, Y., Chen, Y., Feng, L., Chen, T., Yu, H., and Yang, Q. Fedvision: An online visual object detection platform powered by federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13172–13179, 2020.

Markowitz, E., Balasubramanian, K., Mirtaheri, M., Abu-El-Haija, S., Perozzi, B., Steeg, G. V., and Galstyan, A. Graph traversal with tensor functionals: A meta-algorithm for scalable learning, 2021.

Martins, I. F., Teixeira, A. L., Pinheiro, L., and Falcao, A. O. A bayesian approach to in silico blood-brain barrier penetration modeling. *Journal of chemical information and modeling*, 52(6):1686–1697, 2012.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

Mei, G., Guo, Z., Liu, S., and Pan, L. Sgnn: A graph neural network based federated learning approach by hiding structure. In *2019 IEEE International Conference on Big Data (Big Data)*, pp. 2560–2568. IEEE, 2019.

Meng, C., Rambhatla, S., and Liu, Y. Cross-node federated graph neural network for spatio-temporal data modeling, 2021. URL https://openreview.net/forum?id=HWX5j6Bv_ih.

Mobley, D. L. and Guthrie, J. P. Freesolv: a database of experimental and calculated hydration free energies, with input files. *Journal of computer-aided molecular design*, 28(7):711–720, 2014.

Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., and Huang, J. Self-supervised graph transformer on large-scale molecular data, 2020a.

Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., and Huang, J. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33, 2020b.

Rong, Y., Xu, T., Huang, J., Huang, W., Cheng, H., Ma, Y., Wang, Y., Derr, T., Wu, L., and Ma, T. Deep graph learning: Foundations, advances and applications. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '20, pp. 3555–3556, New York, NY, USA, 2020c. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3406474. URL https://doi.org/10.1145/3394486.3406474.

Subramanian, G., Ramsundar, B., Pande, V., and Denny, R. A. Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches. *Journal of chemical information and modeling*, 56(10):1936–1949, 2016.

Sun, M., Zhao, S., Gilvary, C., Elemento, O., Zhou, J., and Wang, F. Graph convolutional networks for computational drug development and discovery. *Briefings in Bioinformatics*, 21(3):919–935, 06 2019. ISSN 1477-4054. doi: 10.1093/bib/bbz042. URL https://doi.org/10.1093/bib/bbz042.

Suzumura, T., Zhou, Y., Baracaldo, N., Ye, G., Houck, K., Kawahara, R., Anwar, A., Stavarache, L. L., Watanabe, Y., Loyola, P., et al. Towards federated graph learning for collaborative financial crimes detection. *arXiv preprint arXiv:1909.12946*, 2019.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks, 2018.

Vepakomma, P., Gupta, O., Swedish, T., and Raskar, R. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018.

Wang, B., Li, A., Li, H., and Chen, Y. Graphfl: A federated learning framework for semi-supervised node classification on graphs. *arXiv preprint arXiv:2012.04187*, 2020a.

Wang, X., Ma, Y., Wang, Y., Jin, W., Wang, X., Tang, J., Jia, C., and Yu, J. *Traffic Flow Prediction via Spatial Temporal Graph Neural Network*, pp. 1082–1092. Association for Computing Machinery, New York, NY, USA, 2020b. ISBN 9781450370233. URL https://doi.org/10.1145/3366423.3380186.

Wu, C., Wu, F., Cao, Y., Huang, Y., and Xie, X. Fedgnn: Federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925*, 2021.

Wu, L., Sun, P., Hong, R., Fu, Y., Wang, X., and Wang, M. Socialgcn: An efficient graph convolutional network based model for social recommendation. *CoRR*, abs/1811.02815, 2018a. URL http://arxiv.org/abs/1811.02815.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018b.

Yang, Q., Liu, Y., Chen, T., and Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2), January 2019. ISSN 2157-6904. doi: 10.1145/3298981. URL https://doi.org/10.1145/3298981.

Zheng, L., Zhou, J., Chen, C., Wu, B., Wang, L., and Zhang, B. Asfgnn: Automated separated-federated graph neural network. *arXiv preprint arXiv:2011.03248*, 2020.

Zhou, J., Chen, C., Zheng, L., Zheng, X., Wu, B., Liu, Z., and Wang, L. Privacy-preserving graph neural network for node classification. *arXiv preprint arXiv:2005.11903*, 2020.

A MORE DETAILS OF SYSTEM DESIGN

To address these deployment challenges, we plan to develop `FedML Client SDK`, which has three key modules, `Data Collector and Manager`, `Training Manager`, and `Model Serving`, as shown in Figure 2. In essence, the three modules inside `FedML Client SDK` builds up a pipeline that manages a model's life cycle, from federated training to personalized model serving (inference). Unifying three modules of a pipeline into a single SDK can simplify the system design. Any subsystem in an institute can integrate `FedML Client SDK` with a host process, which can be the backend service or desktop application. We can create multiple replicas on multiple servers in the institute. More specially, `Data Collector and Manager` is a distributed computing system that can collect scattered datasets or features from multiple servers to `Training Manager`. Such collection can also keep the raw data in the original server with RPCs (remote procedure call), which can only access the data during training. After obtaining all necessary datasets for federated training, `Training Manager` will start federated training using algorithms supported by `FedML-API`. Once training has been completed, `Model Serving` can request the trained model to deploy for inference. Under this SDK abstraction, we plan to address the challenges mentioned above (1) and (2) within the `Data Collector and Manager`. As for challenge (3), we plan to make `FedML Client SDK` compatible with any operating systems (Linux, Android, iOS) with a cross-platform abstraction interface design. Overall, we hope `FedML Client SDK` could be a lightweight and easy-to-use SDK for federated learning among diverse cross-silo institutes.

B BENCHMARK DETAILS

B.1 Molecular Dataset Details

Dataset	Dataset	# Tasks	Task Type	# Compounds	Average # of Nodes	Average # of Edges	Rec - Metric
Quantum Mechanics	QM9 (Gaulton et al., 2012)	12	Regression	133885	8.80	27.60	MAE
Physical Chemistry	ESOL (Delaney, 2004)	1	Regression	1128	13.29	40.65	RMSE
	FreeSolv(Mobley & Guthrie, 2014)	1	Regression	642	8.72	25.60	RMSE
	Lipophilicity (Gaulton et al., 2012)	1	Regression	4200	27.04	86.04	RMSE
Biophysics	hERG(Gaulton et al., 2016; Kim et al., 2021)	1	Regression	10572	29.39	94.09	RMSE
	BACE (Subramanian et al., 2016)	1	Classification	1513	34.09	36.89	ROC-AUC
Physiology	BBBP (Martins et al., 2012)	1	Classification	2039	24.03	25.94	ROC-AUC
	SIDER (Kuhn et al., 2016)	27	Classification	1427	33.64	35.36	ROC-AUC
	ClinTox (Gayvert et al., 2016)	2	Classification	1478	26.13	27.86	ROC-AUC
	Tox21 (tox)	12	Classification	7831	18.51	25.94	ROC-AUC

Table 5. Summary of Molecular Machine Learning Datasets

Table 5 summarizes the necessary information of benchmark datasets (Wu et al., 2018b). The details of each dataset are listed below:

Molecular Classification Datasets

- `BBBP` (Martins et al., 2012) involves records of whether a compound carries the permeability property of penetrating the blood-brain barrier.
- `SIDER` (Kuhn et al., 2016), or Side Effect Resource, dataset consists of marketed drugs with their adverse drug reactions. The available
- `ClinTox` (Gayvert et al., 2016) includes qualitative data of drugs both approved by the FDA and rejected due to the toxicity shown during clinical trials.
- `BACE` (Subramanian et al., 2016) is collected for recording compounds which could act as the inhibitors of human β-secretase 1 (BACE-1) in the past few years.
- `Tox21`(tox) is a dataset which records the toxicity of compounds.

Molecular Regression Datasets

- `QM9` (Ramakrishnan et al., 2014) is a subset of GDB-13, which records the computed atomization energies of stable and synthetically accessible organic molecules, such as HOMO/LUMO, atomization energy, etc. It contains various molecular structures such as triple bonds, cycles, amide, and epoxy.

- `hERG` (Gaulton et al., 2017; Kim et al., 2021) is a dataset that records the gene (KCNH2) that codes for a protein known as Kv11.1 responsible for its contribution to the electrical activity of the heart to help the coordination of the heart's beating.
- `ESOL` (Delaney, 2004) is a small dataset documenting the water solubility(log solubility in mols per litre) for common organic small molecules.
- `Lipophilicity` (Gaulton et al., 2012) which records the experimental results of octanol/water distribution coefficient for compounds.
- `FreeSolv` (Mobley & Guthrie, 2014) contains the experimental results of hydration free energy of small molecules in water.

Dataset Splitting. We apply random splitting as advised in (Wu et al., 2018b). Dataset partition is 80% training, 10% validation, and 10% test. We plan to support the scaffold splitting (Bemis & Murcko, 1996) specifically for molecular machine learning datasets as future work.

B.2 Feature Extraction Procedure for Molecules

The feature extraction is in two steps:

1. Atom-level feature extraction and Molecule object construction using RDKit (Landrum et al., 2006).
2. Constructing graphs from molecule onbjects using NetworkX (Hagberg et al., 2008).

Atom features, shown in Table 6, are the atom features we used exactly same as in (Rong et al., 2020a).

Features	Size	Description
atom type	100	Representation of atom (e.g., C, N, O), by its atomic number
formal charge	5	An integer electronic charge assigned to atom
number of bonds	6	Number of bonds the atom is involved in
chirality	5	Number of bonded hydrogen atoms
number of H	5	Number of bonded hydrogen atoms
atomic mass	1	Mass of the atom, divided by 100
aromaticity	1	Whether this atom is part of an aromatic system
hybridization	5	SP, SP2, SP3, SP3D, or SP3D2

Table 6. Atom features

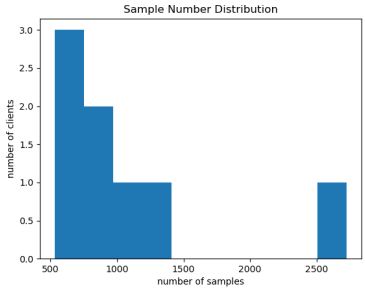
B.3 Non-I.I.D. Partition

The alpha value for latent Dirichlet allocation (LDA) in each non-IID graph dataset can be found in Table 2 and 3. The data distribution for each dataset is illustrated in Figure 4.

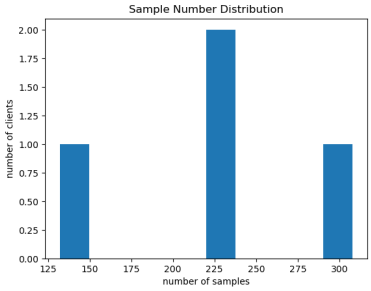
B.4 Details of Supported Models and Algorithms

Graph Neural Network Architectures

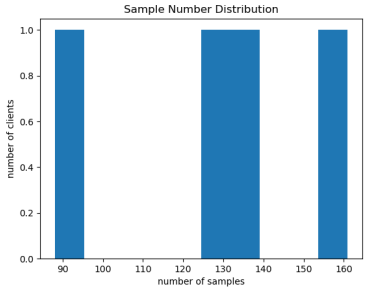
- **Graph Convolutional Networks** (Kipf & Welling, 2016) is a GNN model which is a 1^{st} order approximation to spectral GNN models. (Markowitz et al., 2021)
- **GraphSAGE** (Hamilton et al., 2017) is a general inductive GNN framework capable of generating node-level representations for unseen data.
- **Graph Attention Networks** (Veličković et al., 2018) is the first attention-based GNN model. Attention is computed in a message-passing fashion.



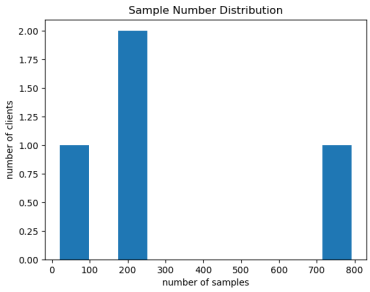
(c) hERG (#clients: 4, alpha: 3)



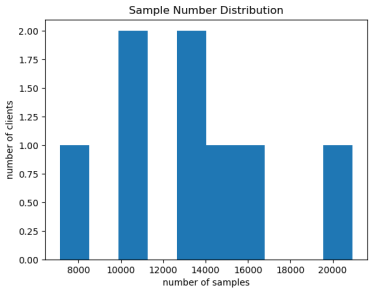
(d) ESOL (#clients: 4, alpha: 2)



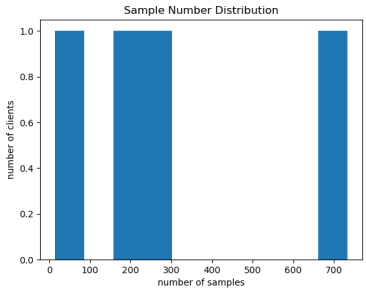
(e) FreeSolv (#clients: 4, alpha: 0.5)



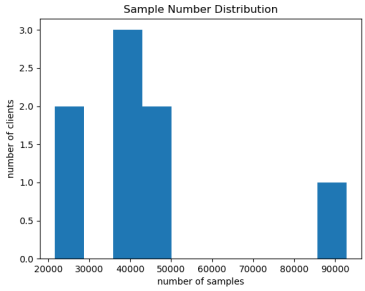
(f) BACE (#clients: 4, alpha: 0.5)



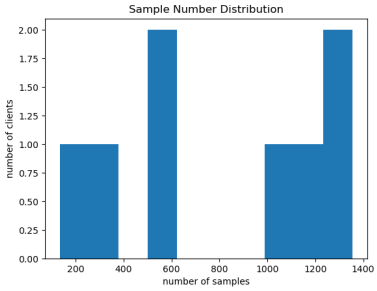
(g) QM9 (#clients: 8, alpha: 3)



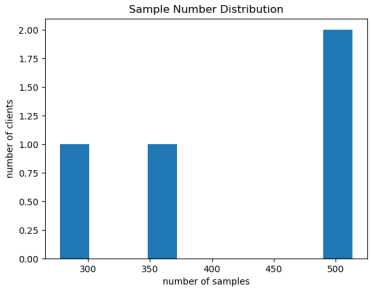
(h) Clintox (#clients: 4, alpha: 0.5)



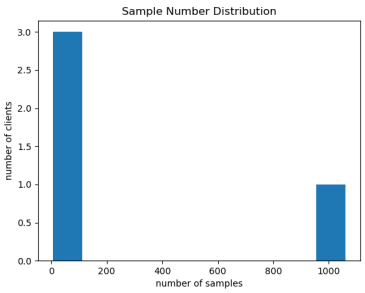
(i) PCBA (#clients: 8, alpha: 3)



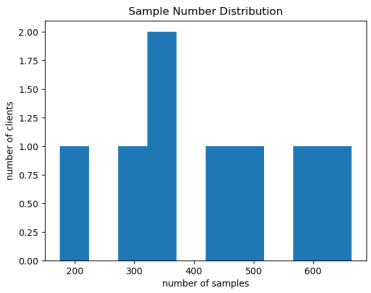
(j) Tox21 (#clients: 8, alpha: 3)



(k) BBBP (#clients: 4, alpha: 2)



(l) SIDER (#clients: 4, alpha: 0.2)



(m) LIPO (#clients: 8, alpha: 2)

Figure 4. Unbalanced Sample Distribution (Non-I.I.D.) for Molecular Datasets

Federated Learning Algorithms

- **Federated Averaging (FedAvg).** FedAvg (McMahan et al., 2017) is a standard federated learning algorithm that is normally used as a baseline for advanced algorithm comparison. Each worker trains its local model for several epochs,

then updates its local model to the server. The server aggregates the uploaded client models into a global model by weighted coordinate-wise averaging (the weights are determined by the number of data points on each worker locally), and then synchronizes the global model back to all workers.

Vertical Federated Learning (VFL). VFL or feature-partitioned FL (Yang et al., 2019) is applicable to the cases where all participating parties share the same sample space but differ in the feature space. VFL is the process of aggregating different features and computing the training loss and gradients in a privacy-preserving manner to build a model with data from all parties collaboratively.

Split Learning. Split learning is computing and memory-efficient variant of FL introduced in (Gupta & Raskar, 2018; Vepakomma et al., 2018) where the model is split at a layer and the parts of the model preceding and succeeding this layer are shared across the worker and server, respectively. Only the activations and gradients from a single layer are communicated in split learning, as against that the weights of the entire model are communicated in federated learning.

C HYPER-PARAMETERS

For each task, we utilize grid search to find the best results. Table 7 & 8 list all the hyper-parameters ranges used in our experiments. All hyper-parameter tuning is run on a single GPU. The best hyperparameters for each dataset and model are listed in Table 9,10,11, & 12 For molecule tasks ,batch-size is kept fixed since the molecule-level task requires us to have mini-batch is equal to 1. Also, number of GNN layers were fixed to 2 because having too many GNN layers result in over-smoothing phenomenon as shown in (Li et al., 2018). For all experiments, we used Adam optimizer.

Table 7. Hyper-parameter Range for Centralized Training

hyper-parameter	Description	Range
learning rate	Rate of speed at which the model learns.	$[0.00015, 0.0015, 0.015, 0.15]$
dropout rate	Dropout ratio	$[0.2, 0.3, 0.5, 0.6]$
node embedding dimension	Dimensionality of the node embedding	$[16, 32, 64, 128, 256]$
hidden layer dimension	Hidden layer dimensionality	$[16, 32, 64, 128, 256]$
readout embedding dimension	Dimensionality of the readout embedding	$[16, 32, 64, 128256]$
graph embedding dimension	Dimensionality of the graph embedding	$[16, 32, 64, 128, 256]$
attention heads	Number of attention heads required for GAT	1-7
alpha	LeakyRELU parameter used in GAT model	0.2

Table 8. Hyper-parameter Range for Federated Learning

hyper-parameter	Description	Range
learning rate	Rate of speed at which the model learns.	$[0.00015, 0.0015, 0.015, 0.15]$
dropout rate	Dropout ratio	$[0.3, 0.5, 0.6]$
node embedding dimension	Dimensionality of the node embedding	64
hidden layer dimension	Hidden layer dimensionality	64
readout embedding dimension	Dimensionality of the readout embedding	64
graph embedding dimension	Dimensionality of the graph embedding	64
attention heads	Number of attention heads required for GAT	1-7
alpha	LeakyRELU parameter used in GAT model	0.2
rounds	Number of federating learning rounds	[10, 50, 100]
epoch	Epoch of clients	1
number of clients	Number of users in a federated learning round	4-10

D MORE EXPERIMENTAL DETAILS

The hyper-parameters reported in Section C are based on the hyper-parameter sweeping (grid search). We further provide the curve of test score (accuracy) during training for each dataset with a specific model. We hope these visualized training

Table 9. Hyperparameters for Molecular Classification Task

Dataset	Score & Parameters	GCN	GAT	GraphSAGE
BBBP	ROC-AUC Score	0.8705	0.8824	**0.8930**
	learning rate	0.0015	0.015	0.01
	dropout rate	0.2	0.5	0.2
	node embedding dimension	64	64	64
	hidden layer dimension	64	64	64
	readout embedding dimension	64	64	64
	graph embedding dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
BACE	ROC-AUC Score	0.9221	0.7657	**0.9266**
	learning rate	0.0015	0.001	0.0015
	dropout rate	0.3	0.3	0.3
	node embedding dimension	64	64	16
	hidden layer dimension	64	64	64
	readout embedding dimension	64	64	64
	graph embedding dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
Tox21	ROC-AUC Score	0.7800	0.8144	**0.8317**
	learning rate	0.0015	0.00015	0.00015
	dropout rate	0.4	0.3	0.3
	node embedding dimension	64	128	256
	hidden layer dimension	64	64	128
	readout embedding dimension	64	128	256
	graph embedding dimension	64	64	128
	attention heads	None	2	None
	alpha	None	0.2	None
SIDER	ROC-AUC Score	0.6476	0.6639	**0.6669**
	learning rate	0.0015	0.0015	0.0015
	dropout rate	0.3	0.3	0.6
	node embedding dimension	64	64	16
	hidden layer dimension	64	64	64
	readout embedding dimension	64	64	64
	graph embedding dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
ClinTox	ROC-AUC Score	0.8914	0.9573	**0.9716**
	learning rate	0.0015	0.0015	0.0015
	dropout rate	0.3	0.3	0.3
	node embedding dimension	64	64	64
	hidden layer dimension	64	64	64
	readout embedding dimension	64	64	64
	graph embedding dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None

results can be a useful reference for future research exploration.

Table 10. Hyperparameters for Federated Molecular Classification Task

Dataset	Score & Parameters	GCN + FedAvg	GAT + FedAvg	GraphSAGE + FedAvg
BBBP	ROC-AUC Score	0.7629	0.8746	**0.8935**
	number of clients	4	4	4
	learning rate	0.0015	0.0015	0.015
	dropout rate	0.3	0.3	0.6
	Node Embedding Dimension	64	64	64
	Hidden Layer Dimension	64	64	64
	Readout Embedding Dimension	64	64	64
	Graph Embedding Dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
BACE	ROC-AUC Score	0.6594	0.7714	**0.8604**
	Number of Clients	4	4	4
	Learning Rate	0.0015	0.0015	0.0015
	Dropout Rate	0.5	0.3	0.5
	Node Embedding Dimension	64	64	16
	Hidden Layer Dimension	64	64	64
	Readout Embedding Dimension	64	64	64
	Graph Embedding Dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
Tox21	ROC-AUC Score	0.7128	0.7171	**0.7801**
	Number of Clients	4	4	4
	Learning Rate	0.0015	0.0015	0.00015
	Dropout Rate	0.6	0.3	0.3
	Node Embedding Dimension	64	64	64
	Hidden Layer Dimension	64	64	64
	Readout Embedding Dimension	64	64	64
	Graph Embedding Dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
SIDER	ROC-AUC Score	0.6266	0.6591	**0.67**
	Number of Clients	4	4	4
	Learning Rate	0.0015	0.0015	0.0015
	Dropout Rate	0.6	0.3	0.6
	Node Embedding Dimension	64	64	16
	Hidden Layer Dimension	64	64	64
	Readout Embedding Dimension	64	64	64
	Graph Embedding Dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
ClinTox	ROC-AUC Score	0.8784	0.9160	**0.9246**
	Number of Clients	4	4	4
	Learning Rate	0.0015	0.0015	0.015
	Dropout Rate	0.5	0.6	0.3
	Node Embedding Dimension	64	64	64
	Hidden Layer Dimension	64	64	64
	Readout Embedding Dimension	64	64	64
	Graph Embedding Dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None

Table 11. Hyperparameters for Molecular Regression Task

Dataset	Score &Parameters	GCN	GAT	GraphSAGE
Freesolv	RMSE Score	0.8705	0.8824	**0.8930**
	learning rate	0.0015	0.015	0.01
	dropout rate	0.2	0.5	0.2
	node embedding dimension	64	64	64
	hidden layer dimension	64	64	64
	readout embedding dimension	64	64	64
	graph embedding dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
ESOL	RMSE Score	0.8705	0.8824	**0.8930**
	learning rate	0.0015	0.015	0.01
	dropout rate	0.2	0.5	0.2
	node embedding dimension	64	64	64
	hidden layer dimension	64	64	64
	readout embedding dimension	64	64	64
	graph embedding dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
Lipophilicity	RMSE Score	0.8521	0.7415	**0.7078**
	learning rate	0.0015	0.001	0.001
	dropout rate	0.3	0.3	0.3
	node embedding dimension	128	128	128
	hidden layer dimension	64	64	64
	readout embedding dimension	128	128	128
	graph embedding dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
hERG	RMSE Score	0.7257	**0.6271**	0.7132
	learning rate	0.001	0.001	0.005
	dropout rate	0.3	0.5	0.3
	node embedding dimension	64	64	64
	hidden layer dimension	64	64	64
	readout embedding dimension	64	64	64
	graph embedding dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
QM9	RMSE Score	0.8705	0.8824	**0.8930**
	learning rate	0.0015	0.015	0.01
	dropout rate	0.2	0.5	0.2
	node embedding dimension	64	64	64
	hidden layer dimension	64	64	64
	readout embedding dimension	64	64	64
	graph embedding dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None

Table 12. Hyperparameters for Federated Molecular Regression Task

Dataset	Parameters	GCN + FedAvg	GAT + FedAvg	GraphSAGE + FedAvg
FreeSolv	RMSE Score	2.747	3.108	**1.641**
	Number of Clients	4	8	4
	learning rate	0.0015	0.00015	0.015
	dropout rate	0.6	0.5	0.6
	node embedding dimension	64	64	64
	hidden layer dimension	64	64	64
	readout embedding dimension	64	64	64
	graph embedding dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
ESOL	RMSE Score	1.435	**1.028**	1.185
	Number of Clients	4	4	4
	learning rate	0.0015	0.0015	0.0015
	dropout rate	0.5	0.3	0.3
	node embedding dimension	64	256	64
	hidden layer dimension	64	64	64
	readout embedding dimension	64	64	64
	graph embedding dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
Lipophilicity	RMSE Score	1.146	1.004	**0.7788**
	Number of Clients	4	4	4
	learning rate	0.0015	0.0015	0.0015
	dropout rate	0.3	0.3	0.3
	node embedding dimension	64	64	256
	hidden layer dimension	64	64	256
	readout embedding dimension	64	64	256
	graph embedding dimension	64	64	256
	attention heads	None	2	None
	alpha	None	0.2	None
hERG	RMSE Score	0.7944	0.7322	**0.7265**
	Number of Clients	8	8	8
	learning rate	0.0015	0.0015	0.0015
	dropout rate	0.3	0.3	0.6
	node embedding dimension	64	64	64
	hidden layer dimension	64	64	64
	readout embedding dimension	64	64	64
	graph embedding dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None
QM9	MAE Score	0.8705	3.508	15.824
	Number of Clients	8	8	8
	learning rate	0.0015	0.00015	0.15
	dropout rate	0.2	0.5	0.3
	node embedding dimension	64	256	64
	hidden layer dimension	64	128	64
	readout embedding dimension	64	256	64
	graph embedding dimension	64	64	64
	attention heads	None	2	None
	alpha	None	0.2	None