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

Chaoyang He, Keshav Balasubramanian & Emir Ceyani *
Viterbi School of Engineering
University of Southern California
Los Angeles, CA, USA
{chaoyang.he,keshavba,ceyani}@usc.edu

Yu Rong, Peilin Zhao & Junzhou Huang
Machine Learning Center
Tencent AI Lab
Shenzhen, China

Murali Annavaram & Salman Avestimehr
Viterbi School of Engineering
University of Southern California
Los Angeles, CA, USA

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 the 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 include a new molecular dataset, hERG, to promote research exploration. Our experimental results present significant challenges from federated GNN training: federated GNNs perform worse in most datasets with a non-I.I.D split than centralized GNNs; the GNN model that performs the best in centralized training may not hold its advantage in the federated setting. These results imply that more research effort is needed to unravel the mystery of federated GNN training. Moreover, our system performance analysis demonstrates that the `FedGraphNN` system is affordable to most research labs with a few GPUs. We maintain the source code at https://github.com/FedML-AI/FedGraphNN.

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 restriction, and commercial competition, there are surging real-world cases in which graph data is decentralized, limiting the data size of a single party (client). For example, in the AI-based drug discovery industry, pharmaceutical research institutions would significantly benefit from the private data of another institution, but neither cannot afford to disclose their private data for commercial reasons. Federated Learning (FL) is a distributed learning paradigm with provable privacy guarantees (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), FL has yet to be widely

*Equal contribution

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 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 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 (Section 2). 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 3). 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 C). Our experiments show that the straightforward deployment of FL algorithms for GNNs is sub-optimal (Section 4). Finally, we highlight future directions for federated GNNs (Section 5).

2 Formulation: Federated Graph Neural Networks

We consider a *graph level GNN-based federated learning*[1] setting, depicted in Figure 1, 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 & edge feature sets $\boldsymbol{X}^{(k)} = \left\{\boldsymbol{x}_m^{(k)}\right\}_{m\in\mathcal{V}_i^{(k)}}$ & $\boldsymbol{Z}^{(k)} = \left\{\boldsymbol{e}_{m,n}^{(k)}\right\}_{m,n\in\mathcal{V}_i^{(k)}}$, $y_i^{(k)}$ 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 GNN model (an L-layer MPNN(Gilmer et al., 2017; Rong et al., 2020c)) to learn graph-level representations. Multiple clients are interested in collaborating through a server to improve their GNN models without necessarily revealing their graphs.

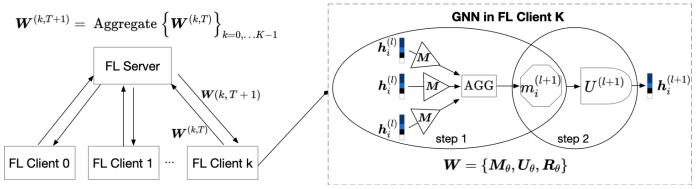


Figure 1: Formulation of `FedGraphNN` (Federated Graph Neural Network)

MPNN performs the forward pass in 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,

[1] Different types of GNN-based FL are explained in Appendix A

for client k, a L-layer MPNN is formalized as:

$$\boldsymbol{m}_i^{(k,\ell+1)} = \text{AGG}\left(\left\{\boldsymbol{M}_\theta^{(k,\ell+1)}\left(\boldsymbol{h}_i^{(k,\ell)}, \boldsymbol{h}_j^{(k,\ell)}, \boldsymbol{e}_{i,j}\right) \mid j \in \mathcal{N}_i\right\}\right), \ell = 0, \dots L-1 \tag{1}$$

$$\boldsymbol{h}_i^{(k,\ell+1)} = \boldsymbol{U}_\theta^{(k,\ell+1)}\left(\boldsymbol{h}_i^{(k,\ell)}, \boldsymbol{m}_i^{(k,\ell+1)}\right), \ell = 0, \dots L-1 \tag{2}$$

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 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 (e.g. 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:

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

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). It is important to note here that in FedAvg, the aggregation function on the server merely averages model parameters. We use GNNs inductively, i.e. the model is independent of the structure of the graphs it is trained on. Thus, no topological information about graphs on any client is required on the server during parameter aggregation. Other advanced algorithms such as FedOPT (Reddi et al., 2020) and FedGKT (He et al., 2020a) can also be applied.

3 FedGraphNN System Design

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 in Figure 2 in the appendix, `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). An enhanced security and privacy primitive modules are added to support techniques in upper layers. The layer above supports plug and play operation of common GNN models such as GraphSage and GAT. We provide dedicated data loaders and splitters to handle non-I.I.D. nature of graph datasets. Users can either reuse our data distribution or manipulate the non-I.I.D.ness by setting hyperparameters. For details of the system design and benchmark details, we refer the readers to Appendix B and C.

4 Benchmark and Experiments

Benchmarking Dataset: 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 using the partition algorithm Latent Dirichlet Allocation (LDA) (He et al., 2020b). In addition, we provide

a new dataset, named hERG, related to the cardiac toxicity and collected from (Kim et al., 2021; Gaulton et al., 2017) with data cleaning. Table 2 summarizes all datasets we used in experiments. Figure 4 shows each dataset's distribution.

Result of Model Accuracy on Non-I.I.D. Partitioning: 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). We run experiments on both classification and regression tasks. Hyper-parameters are tuned (sweeping) by grid search (see Section D for the search space). After hyper-parameter tuning, we report all results in Table 1 and Table 5. For each result, the optimal hyper-parameters can be found in the Appendix E. For more details, please refer to the Appendix D.

There are multiple takeaways: 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. 3. Our results show that the best model in the centralized setting may not be the best for the non-I.I.D. federated setting. Interestingly, we find that GAT suffers the largest performance compromise on 5 out of 9 datasets. This may be due to the sensibility of the attention calculation on the non-I.I.D. settings. Hence, additional research is needed to understand the nuances of training GNNs in a federated setting and bridge this gap.

Table 1: 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.

System Performance Analysis : We also present system performance analysis when using Message Passing Interface (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, but results similar to using MPI. 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 detailed test results in such a scenario in our future work.

5 Conclusion and Future Works

In this paper, we designed a federated learning (FL) system and benchmark for federated graph neural networks (GNN), named `FedGraphNN` including implementations of common 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 exciting problems in the intersection of federated learning and graph neural networks. Here we highlight some future research directions that deserve more efforts: 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 toward realistic GNN-based FL applications.

REFERENCES

Tox21 challenge. https://tripod.nih.gov/tox21/challenge/, 2017.

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

Chaochao Chen, Jamie Cui, Guanfeng Liu, Jia Wu, and Li Wang. Survey and open problems in privacy preserving knowledge graph: Merging, query, representation, completion and applications. *arXiv preprint arXiv:2011.10180*, 2020.

Zhiyong Cui, Kristian Henrickson, Ruimin Ke, Ziyuan Pu, and Yinhai Wang. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting, 2019.

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

Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

Anna Gaulton, Anne Hersey, Michał Nowotka, A. Patrícia Bento, Jon Chambers, David Mendez, Prudence Mutowo, Francis Atkinson, Louisa J. Bellis, Elena Cibrián-Uhalte, Mark Davies, Nathan Dedman, Anneli Karlsson, María Paula Magariños, John P. Overington, George Papadatos, Ines Smit, and Andrew R. Leach. 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.

Anna Gaulton, Anne Hersey, Michał Nowotka, A Patricia Bento, Jon Chambers, David Mendez, Prudence Mutowo, Francis Atkinson, Louisa J Bellis, Elena Cibrián-Uhalte, et al. The chembl database in 2017. *Nucleic acids research*, 45(D1):D945–D954, 2017.

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

Suyu Ge, Fangzhao Wu, Chuhan Wu, Tao Qi, Yongfeng Huang, and Xing Xie. Fedner: Privacy-preserving medical named entity recognition with federated learning. *arXiv e-prints*, pp. arXiv–2003, 2020.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pp. 1263–1272. PMLR, 2017.

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

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

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

Chaoyang He, Conghui Tan, Hanlin Tang, Shuang Qiu, and Ji Liu. Central server free federated learning over single-sided trust social networks. *arXiv preprint arXiv:1910.04956*, 2019a.

Chaoyang He, Tian Xie, Yu Rong, W. Huang, Junzhou Huang, X. Ren, and C. Shahabi. Cascade-bgnn: Toward efficient self-supervised representation learning on large-scale bipartite graphs. *arXiv preprint arXiv:1906.11994*, 2019b.

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

Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020b.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. *arXiv preprint arXiv:2003.08082*, 2020.

Meng Jiang, Taeho Jung, Ryan Karl, and Tong Zhao. Federated dynamic gnn with secure aggregation. *arXiv preprint arXiv:2009.07351*, 2020.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem in 2021: new data content and improved web interfaces. *Nucleic Acids Research*, 49(D1):D1388–D1395, 2021.

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

Michael Kuhn, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. The sider database of drugs and side effects. *Nucleic acids research*, 44(D1):D1075–D1079, 2016.

Greg Landrum. Rdkit: Open-source cheminformatics, 2006. URL http://www.rdkit.org.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. 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.

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

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

Guangxu Mei, Ziyu Guo, Shijun Liu, and Li Pan. 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.

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

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

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data, 2020a.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33, 2020b.

Yu Rong, Tingyang Xu, Junzhou Huang, Wenbing Huang, Hong Cheng, Yao Ma, Yiqi Wang, Tyler Derr, Lingfei Wu, and Tengfei Ma. Deep graph learning: Foundations, advances and applications. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery &; 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.

Govindan Subramanian, Bharath Ramsundar, Vijay Pande, and Rajiah Aldrin Denny. Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches. *Journal of chemical information and modeling*, 56(10):1936–1949, 2016.

Mengying Sun, Sendong Zhao, Coryandar Gilvary, Olivier Elemento, Jiayu Zhou, and Fei Wang. 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.

Toyotaro Suzumura, Yi Zhou, Natahalie Baracaldo, Guangnan Ye, Keith Houck, Ryo Kawahara, Ali Anwar, Lucia Larise Stavarache, Yuji Watanabe, Pablo Loyola, et al. Towards federated graph learning for collaborative financial crimes detection. *arXiv preprint arXiv:1909.12946*, 2019.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

Binghui Wang, Ang Li, Hai Li, and Yiran Chen. Graphfl: A federated learning framework for semi-supervised node classification on graphs. *arXiv preprint arXiv:2012.04187*, 2020a.

Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. *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.

Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. Fedgnn: Federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925*, 2021.

Le Wu, Peijie Sun, Richang Hong, Yanjie Fu, Xiting Wang, and Meng Wang. Socialgcn: An efficient graph convolutional network based model for social recommendation. *CoRR*, abs/1811.02815, 2018a. URL http://arxiv.org/abs/1811.02815.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018b.

Longfei Zheng, Jun Zhou, Chaochao Chen, Bingzhe Wu, Li Wang, and Benyu Zhang. Asfgnn: Automated separated-federated graph neural network. *arXiv preprint arXiv:2011.03248*, 2020.

Jun Zhou, Chaochao Chen, Longfei Zheng, Xiaolin Zheng, Bingzhe Wu, Ziqi Liu, and Li Wang. Privacy-preserving graph neural network for node classification. *arXiv preprint arXiv:2005.11903*, 2020.

A 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 graph neural networks 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. Recently, (Wang et al., 2020a) use the hybrid method of federated learning and meta-learning to solve the semi-supervised graph node classification problem in decentralized social network dataset. (Meng et al., 2021) attempt to protect the node-level privacy using a edge-cloud partitioned GNN model for spatio-temporal forecasting task using node-level traffic sensor datasets. Finally, (Wu et al., 2021) propose a federated recommendation system with GNNs.

Our library 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.

B More Details of System Design

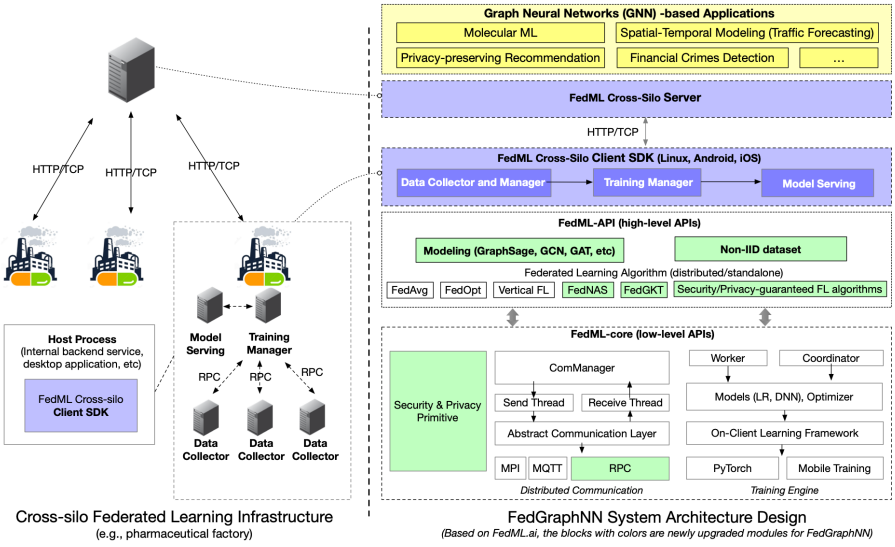


Figure 2: Overview of `FedGraphNN` System Architecture Design

Deploying federated learning algorithms to existing internal systems in cross-silo institutes faces several challenges:

1. Both different institutes and different subsystems in an institute have heterogeneous data schemes (different feature space, different labels for the same data point, different formats);
2. Datasets or features are scattered in different subsystems in an institute;
3. The FL client software should be compatible to existing system (OS platform, system architecture, API design pattern).

In general, frequent and large-scale deployment of updates, monitoring, and debugging is challenging; running ML workloads on an edge server is hampered by the lack of a portable, fast, small footprint, and flexible runtime engine for on-device training (Kairouz et al., 2019, Section 7).

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`

Category	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
	BBBP (Martins et al., 2012)	1	Classification	2039	24.03	25.94	ROC-AUC
Physiology	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, 2017)	12	Classification	7831	18.51	25.94	ROC-AUC

Table 2: Summary of Molecular Machine Learning Datasets

`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 be deployed for inference. Under this SDK abstraction, we plan to address the aforementioned challenges (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.

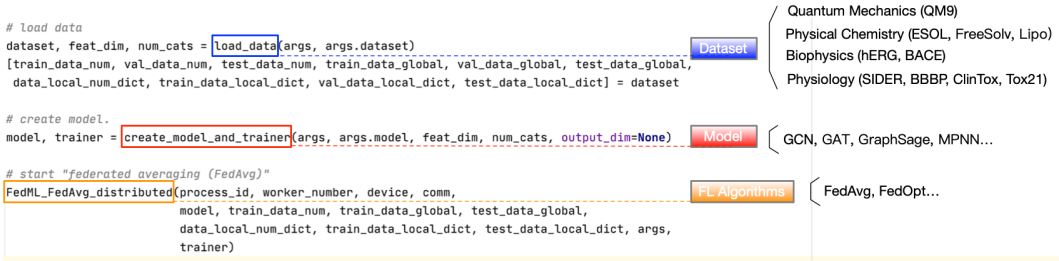


Figure 3: Example code for benchmark evaluation with `FedGraphNN`

C FEDGRAPHNN BENCHMARK: DATASETS, MODELS, AND ALGORITHMS

Non-I.I.D. Datasets. To facilitate the research for GNN-based federated learning, we plan to support various graph datasets with nonIIDness 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. In addition, we provide a new dataset, named hERG, which is related to the cardiac toxicity and collected from (Kim et al., 2021; Gaulton et al., 2017) with data cleaning. Table 2 summarizes all datasets we used in experiments. Figure 4 shows each dataset's distribution.

C.1 MOLECULAR DATASET DETAILS

Table 2 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, 2017) 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, epoxy, etc .
- `hERG` (Gaulton et al., 2017; Kim et al., 2021) is a dataset which 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) specificaslly for moelcular machine learning datasets as a future work.

C.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). 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. GNN algortihms are listed as follows:

- **Graph Convolutional Networks** (Kipf & Welling, 2016) is a GNN model which is a 1^{st} order approximation to spectral GNN models.
- **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.3 Feature Extraction Procedure for Molecules

The feature extraction is in two steps:

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

Atom features, shown in Table 3, 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 3: Atom features

C.4 Non-I.I.D. Partition

The alpha value for latent Dirichlet allocation (LDA) in each non-IID graph dataset can be found in Table 1 and 5. The data distribution for each dataset is illustrated in Figure 4.

D More Experimental Details

The hyper-parameters reported in Section E 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 results can be an useful reference for future research exploration.

E Hyper-parameters

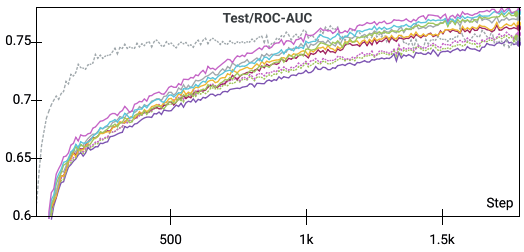


Figure 5: Tox21: test score during sweeping

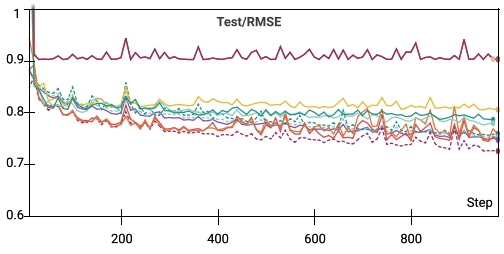
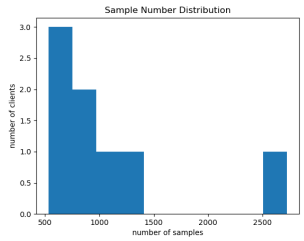


Figure 6: hERG: test score during sweeping

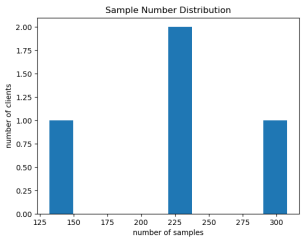
Table 4: Training time with FedAvg on GNNs (Hardware: 8 x NVIDIA Quadro RTX 5000 GPU (16GB/GPU); RAM: 512G; CPU: Intel Xeon Gold 5220R 2.20GHz).

		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

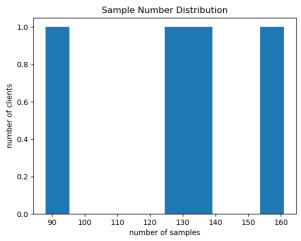
*Note that we use the distributed training paradigm where each client's local training uses one GPU. Please refer our code for details.



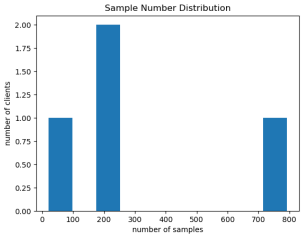
(a) hERG (#clients: 4, alpha: 3)



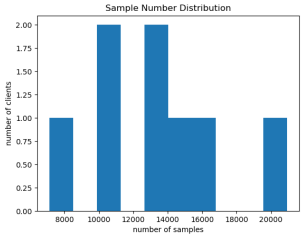
(b) ESOL (#clients: 4, alpha: 2)



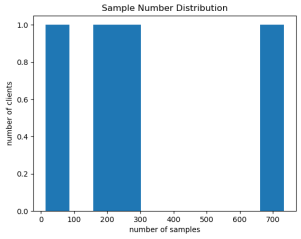
(c) FreeSolv (#clients: 4, alpha: 0.5)



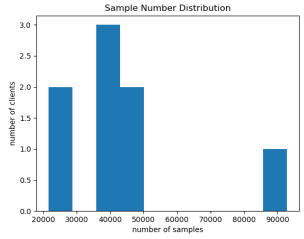
(d) BACE (#clients: 4, alpha: 0.5)



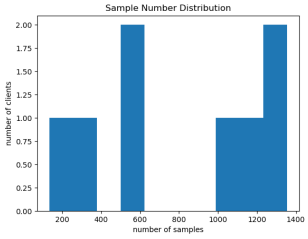
(e) QM9 (#clients: 8, alpha: 3)



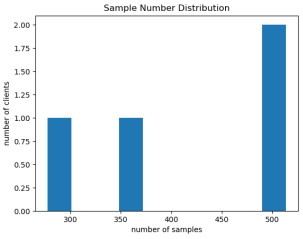
(f) Clintox (#clients: 4, alpha: 0.5)



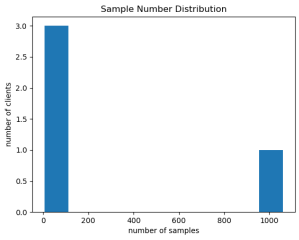
(g) PCBA (#clients: 8, alpha: 3)



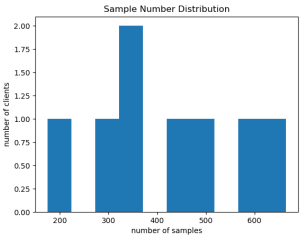
(h) Tox21 (#clients: 8, alpha: 3)



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



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



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

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

For each task, we utilize grid search to find the best results. Table 6 & 7 lists all the hyper-parameters range used in our experiments. All hyper-parameter tuning is run on a single GPU. The best hyper-parameters for each dataset and model Table 8,9,10, & 11 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 5: 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	FedAvg	RMSE	1.40 ± 0.16	1.5787	2.7470 (↑ 1.1683)
		GAT				1.2175	3.1080 (↑ 1.8905)
		GraphSAGE				1.3630	1.6410 (↑ 0.2780)
ESOL (1128)	LDA with $\alpha = 2$ 4 clients	GCN	FedAvg	RMSE	0.97 ± 0.01	1.0190	1.4350 (↑ 0.4160)
		GAT				0.9358	1.3740 (↑ 0.4382)
		GraphSAGE				0.8890	1.1860 (↑ 0.2970)
Lipo (4200)	LDA with $\alpha = 2$ 8 clients	GCN	FedAvg	RMSE	0.655 ± 0.036	0.8518	1.1460 (↑ 0.2942)
		GAT				0.7465	1.0040 (↑ 0.2575)
		GraphSAGE				0.7078	0.7788 (↑ 0.0710)
hERG (10572)	LDA with $\alpha = 3$ 8 clients	GCN	FedAvg	RMSE	-	0.7257	0.7944 (↑ 0.0687)
		GAT				0.6271	0.7322 (↑ 0.1051)
		GraphSAGE				0.7132	0.7265 (↑ 0.0133)
QM9 (133885)	LDA with $\alpha = 3$ 8 clients	GCN	FedAvg	MAE	2.35	14.78	21.075 (↑ 6.295)
		GAT				12.44	23.173 (↑ 10.733)
		GraphSAGE				13.06	19.167 (↑ 6.107)

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

Table 6: 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 7: 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

Table 8: 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

Table 9: 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	0.7629	**0.8604**
	Number of Clients	4	4	4
	Learning Rate	0.0015	0.0015	0.0015
	Dropout Rate	0.5	0.3	0.6
	Node Embedding Dimension	64	1.05 ± 0.10	16
	Hidden Layer Dimension	64	1.05 ± 0.10	64
	Readout Embedding Dimension	64	1.05 ± 0.10	64
	Graph Embedding Dimension	64	1.05 ± 0.10	64
	attention heads	None	2	None
	alpha	None	0.2	None
Tox21	ROC-AUC Score	0.7128	0.7714	**0.7801**
	Number of Clients	4	4	4
	Learning Rate	0.0015	0.15	0.00015
	Dropout Rate	0.6	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
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 10: 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	14.78	**12.44**	13.06
	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 11: Hyperparameters for Federated Molecular Regression Task

Dataset	Parameters	GCN + FedAvg	GAT + FedAvg	GraphSAGE + FedAvg
FreeSolv	RMSE Score	2.747	3.508	**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	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
ESOL	RMSE Score	1.435	1.374	**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	128	64
	readout embedding dimension	64	256	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	0.6
	node embedding dimension	64	256	256
	hidden layer dimension	64	128	256
	readout embedding dimension	64	256	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	0.2	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
QM9	MAE Score	21.075	23.173	**19.167**
	Number of Clients	8	8	8
	learning rate	0.0015	0.00015	0.0015
	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