Federated Contrastive Learning of Graph-Level Representations

Xiang Li*, Gagan Agrawal†, Rajiv Ramnath* and Ruoming Jin§
*The Ohio State University, Columbus OH 43210, USA
†University of Georgia, Athens GA 30602, USA
‡Kent State University, Kent OH 44242, USA

2024 IEEE International Conference on Big Data (BigData) | 979-8-3503-6248-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/BigData62323.2024.10825209

***Abstract*—**
Graph-level representations (and clustering/classification based on these representations) are required in a variety of applications. Examples include identifying malicious network traffic, prediction of protein properties, and many others. Often, data has to stay in isolated local systems due to a variety of considerations like privacy concerns, lack of trust between the parties, regulations, or simply because the data is too large to be shared sufficiently quickly. This points to the need for federated learning for graph-level representations, a topic that has not been explored much, especially in an unsupervised setting.

Addressing this problem, this paper presents a new framework we refer to as Federated Contrastive Learning of Graph-level Representations (FCLG). Our approach builds on contrastive learning. However, what is unique is that we apply contrastive learning at two levels. The first application is for local unsupervised learning of graph representations. The second level is to address the challenge associated with data distribution variation (i.e. the "Non-IID issue") when combining local models. Through extensive experiments on the downstream task of graph-level clustering, we demonstrate FCLG outperforms baselines with significant margins.

I. Introduction

Attributed graphs have lately become backbone of countless systems [1] as they capture information about individual entities (i.e., the *node features*) as well as the interactions between them (i.e. the *edges*). There has also been a growing interest in *graph-level representation learning*, where the goal is to learn embedding of an entire graph.

Motivation for Unsupervised Federated Learning: In many machine learning scenarios, including those involving graphs, data cannot be stored and analyzed centrally due to reasons like privacy concerns. Alternatively, in some cases, data is too large to be shared, or at least cannot be shared sufficiently quickly to provide timely analysis results [2], [3], [4], [5]. Such scenarios have led to interest within the graph analytics community in the well-known paradigm of *Federated Learning* [6], which involves training centralized models while keeping data decentralized – but by sharing and updating model parameters [7]. This paper focuses on *unsupervised* federated learning. Unsupervised learning is an alternative to supervised learning, which in general, and more specifically in the context of graphs, requires labeling that ranges from being expensive and/or time-consuming to simply being impossible [8].

As a background, to address the challenges associated with attributed graphs, researchers have built on the success of Graph Neural Networks (GNNs) [9]. GNN based models have achieved advanced performance on *graph-level* representation learning, especially in a supervised manner [10], [11], [12], [13], [14], [15], [16], [17]. There has also been an interest in unsupervised learning approaches for graph-level representations [18], [19], [8], though for the cases when the data is centrally available. This includes the works based on *Graph Kernels* [20], [21] and limited GNN based works [8], [22].

Contributions of this Paper: This paper presents a federated and unsupervised graph-level learning framework. The proposed method is inspired by the recent success of Contrastive Learning (CL) in various unsupervised learning tasks [23], [8], [24]. To illustrate the type of challenges we need to address – it is well known that Federated Learning (FL) performance is hindered by the "Non-IID" (Independent and Identical Distribution) issue, i.e. data at different sites follows different distributions [7]. Thus, both the local models and the global model generated simply by averaging parameters of these models may be far from the global optima [25]. Previous work [7], [26], [27] has studied the impact of this Non-IID property especially in supervised learning where label information is explicitly utilized during training. In such scenarios, skewed label distribution on each client is encoded into the respective local model.

Addressing this problem, we propose **F**ederated **C**ontrastive **L**earning on **G**raphs (FCLG), a federated graph embedding method with a novel *two-level contrastive learning mechanism*. Specifically, the two levels in our approach are: 1) *intra-contrasting* during local training on each client or site, specifically where we contrast augmented views of encoded graph-level representations to make each individual graph sample more distinguishable from others; 2) *inter-contrasting* between local and global models, ensuring that the global model captures the common patterns underlying graphs from different clients, to achieve better generalization performance than any local model. We also show how contrastive learning can be seen as an improvement over the approaches derived from the area of Knowledge Distillation (KD) [28], [29].

The core contributions of this work can be summarized as:

- To our best knowledge, FCLG is the first framework to learn graph-level representations in an unsupervised manner under federated settings.

- FCLG introduces a novel two-level contrastive mechanism to alleviate issues caused by the non-IID situations that arise in federated learning.
- Through comprehensive empirical studies we show that FCLG consistently outperforms other baselines in the downstream graph-level clustering task.

II. Methodology

A. Problem Statement

Suppose there are K separate clients $C_1, C_2, \cdots, C_K$, where client C_i has a local graph dataset S^i containing a set of graphs. Each graph can be represented as $G = (\mathcal{V}, \mathcal{E}, X)$, where $\mathcal{V}$ is the set of n vertices, $\mathcal{E}$ is the edge set, $X \in \mathbb{R}^{n \times F}$ is the feature matrix where each node is associated with a F-dimensional feature vector. Let A be the $n \times n$ adjacency matrix capturing all edges in the graph G. For simplicity, we can also denote each graph as $G(A, X)$.

Our objective is to learn a model $\psi_\omega : G \rightarrow \mathbb{R}^d$, where ω is the corresponding parameter set. More specifically, we have a federated learning problem where our goal is to learn a parameter set ω_g using the datasets $S \triangleq \bigcup_{i \in [K]} S^i$. The individual parameter set at the client C_i is denoted as ω_i. In other words, the d-dimensional graph-level embedding $\psi_\omega(G)$ must be learnt collaboratively from a set of clients, each of which stores a distinct set of graphs, while inter-client raw graph data transfer is disallowed.

As additional background, we introduce the first basic FL algorithm FedAvg [6] which is commonly used as the starting point for developing more advanced FL frameworks [7]. In FedAvg, the global model on the server will aggregate local model parameters transmitted from clients and distribute the aggregated parameters back to all clients. During each communication round, each client downloads the model from the server $\omega_i \leftarrow \omega_g^{(t)}$ and trains the model ω_i with its own data S^i for E_{local} epochs. Each client S^i will send locally updated parameters $\omega_i^{(t)}$ to the server, then the global model $\omega_g^{(t+1)}$ will be updated by a simple weighted average:

$$\omega_g^{(t+1)} = \sum_{i=1}^{K} \frac{|S^i|}{|S|} \omega_i^{(t)} \tag{1}$$

where $|S^i|$ denotes the size of data on the client S^i and $|S|$ is the total size of all data samples distributed over all clients. The server will broadcast newly updated parameters $\omega_g^{(t+1)}$ to clients for the next round of training.

B. Basic Ideas

Let us say we have a local model f_c from a client and the global model f_s on the server. On the server, we wish f_s to encode all samples into a representation where each sample is more distinguishable from others. In other words, the overall model training will push apart $f_s(x_i)$ and $f_s(x_j)$, where different graph samples x_i, x_j can be from different clients. One way of viewing federated learning is that we are trying to distill knowledge from multiple *teacher* models trained on isolated clients into a single global student model on the server. An earlier federated learning effort [28] has applied a knowledge distillation (KD) technique for model fusion in image classification tasks. A general objective function of KD is the Kullback-Leibler (KL) divergence loss between the softened probability distributions of the teacher models and the student model [28].

$$\begin{aligned} & l_{KL}\Big(\sigma(f_c^t(x)), \sigma(f_s(x))\Big) \\ & = \sigma(f_s(x)) \cdot \Big(\log \sigma(f_s(x)) - \log \sigma(f_c^t(x))\Big) \end{aligned} \tag{2}$$

where $\sigma(\cdot)$ denotes the softmax function.

KL divergence loss has achieved considerable success, which can be attributed to its ability to control the softness of targets via the temperature-scaling hyperparameter τ. Specifically, utilizing a larger value makes the softmax vectors smooth over latent classes [29], [30], [31]. Moving further, it has been observed that the logit matching is positively correlated to the performance improvement in KD [29]. These authors considered the mean squared error (MSE) between the logit vectors such that the student model can directly learn the logit of the teacher model.

$$l_{MSE}\Big(f_c^t(x), f_s(x)\Big) = ||f_c^t(x) - f_s(x)||_2^2 \tag{3}$$

For the KD loss functions like KL-divergence in Eq.2 and MSE in Eq.3, we are pushing closer the representations $f_s(x_i)$ and $f_c(x_i)$. This alleviates the local drift by aligning the local logits with that of the server model during training.

In our work, what we discover is that the paradigm of *Contrastive Learning* can help significantly improve both unsupervised graph-level representation on individual sites as well as in federated settings in a unified fashion. From a knowledge distillation perspective, the inter-contrasting mechanism can be taken as an advanced KD, more specifically an ensembling distillation technique.

$$l_{inter} = \log\Big(1 + \exp(f_c^t(x) \cdot f_c^{t-1}(x) - f_c^t(x) \cdot f_s(x))\Big) \tag{4}$$

The idea here is that the local drift will be hindered by using more historical information of local training. We not only try to push closer the in-situ local model $f_c^t(x)$ and $f_s(x)$, but also aim to push apart the current representations $f_c^t(x)$ away from its previous round $f_c^{t-1}(x)$.

C. Technical Details

Prior graph-level learning approaches, such as InfoGraph, [8], [22] are based on mutual information maximization between multi-scale graph structures like graph-level and subgraph or node-level representations. Specifically, they consider

the summarized patch representation centered at each node as a *positive example*. This positive example is compared with the global representation of the entire graph, based on mutual information. There are also negative samples that arise from *fake* graphs' local representations. In this paper, we produce positive samples of a given graph by *randomized augmentations* – a powerful and proven approach to produce similar pairs for deep learning models [32] that has not yet been fully used for learning graph-level representations. Basically, we create augmented similar and dissimilar graphs and then apply an instance-wise contrasting objective [23].

Moreover, in federated learning setting, this approach can directly compare graphs available at each client against each other to sharpen the characteristics of each individual graph sample. Thus, one of the novel parts of our approach is that we apply contrastive learning at two levels. The first application (*intra-contrasting*) is for local learning of graph representations – specifically, for contrasting augmented views of graphs within each client as described above. The second level (*inter-contrasting*) is to contrast between the global model from the server and the local model on each client, as already introduced through Equation 4.

Intra-contrasting Details: As stated previously, at this level we contrast graph-level representations of different graphs from multiple views. Suppose we are given graph-level representations U of a batch of graphs. After generating the set of augmented views V, the contrastive mechanism [23] works as follows. We denote $u \in U$ as the representation of a single graph in the view U and $v \in V$ as the augmented counterpart of u. Under this contrasting strategy, v forms the only positive sample of u and any other graph representation in these two views, i.e. $U \bigcup V$, is regarded as a negative sample. Thus, we get the contrastive loss:

$$L(u,v) = \log \left\{ \sum_{\substack{z \in U \bigcup V \\ z \neq u}} \exp \left\{ \frac{u \cdot z}{\tau} \right\} / \exp \left\{ \frac{u \cdot v}{\tau} \right\} \right\} \quad (5)$$

where τ is the temperature parameter [23]. Suppose we are training the local model on a batch consisting of B graphs, the overall training loss to be minimized is defined as the average agreement L over all positive pairs as follows:

$$l_{intra} = \frac{1}{2B} \sum_{u \in U} [L(u,v) + L(v,u)] \quad (6)$$

Intra-contrasting will force the local model to capture robust characteristics behind all the graphs stored on each client. The resulting graph embeddings are expected to make each graph sample more distinguishable from others so as to ease the downstream graph-level clustering task.

Inter-contrasting Details: In the method described above, each client can only conduct intra-contrasting for its own data. When there is a skewed or non-identical distribution among the clients, the lost opportunity of contrasting with data samples from other clients can have a significant impact on the generalization performance of the resulting model. This is analogous to how, in federated supervised learning, the local model can explicitly encode the local label distribution and thereby drift away from the global optimum during training. As already stated above, we draw inspiration from a model-level contrasting mechanism proposed by the federated image classification framework proposed in MOON [25]. This work verified that the skewed local data distribution can cause a *drift* in the local updates leading the local model to learn worse representations than the global model. We extended their idea to the graph learning domain in order to control local training *drift* and bridge the gap between graph representations learned by the local model and the global model. This, when applied in an unsupervised manner, is the second part of our novel two-level contrasting framework. Rewriting Equation 4 in our specific context, our goal is to decrease the distance between U^t (representations learned by the current local model) and U_s (representations learned by the global model) while increasing the distance between U^t and U_c^{t-1} (representations learned by the local model with parameters from the previous local epoch). Through this process, we pull U^t and U_c^{t-1} apart to prevent the local model from drifting along skewed subset of data after iterations of local training. Meanwhile, we push U^t and U_s closer to keep optimizing the generalization performance of the global model. Specifically, we minimize the contrastive loss as:

$$l_{inter} = \log \frac{\exp(sim(U^t, U_s)/\tau') + \exp(sim(U^t, U_c^{t-1})/\tau')}{\exp(sim(U^t, U_s)/\tau')} \quad (7)$$

where $sim(\cdot,\cdot)$ is a cosine similarity function and τ' is the temperature for inter-contrasting.

Next, we also construct a model variant by contrasting node-level representations H in place of graph-level representations U by minimizing:

$$l_{inter}^{H} = \log \frac{exp(sim(H^t, H_s)/\tau') + exp(sim(H^t, H_c^{t-1})/\tau')}{exp(sim(H^t, H_s)/\tau')} \quad (8)$$

where H^t, H_c^{t-1}, and H_s are the node-level counterpart representations of U^t, U_c^{t-1}, and U_s, respectively.

D. FCLG: Overall Framework and Algorithm

Framework Architecture. Based on the two-level contrastive learning idea, our framework first learns a powerful graph embedding in a self-supervised manner from decentralized data. The resulting embedding can then be used for downstream graph-level clustering tasks. Figure 1 presents an overview of FCLG architecture. Given a batch of input graph data $G_b(X, A)$, a graph diffusion precomputing procedure [33] is performed to generate the augmentation view $G_d(X, A)$ (following [22]). The subsequent graph embedding generation module comprises a GIN [10] based encoder f. This encoder f involves a set of parameters denoted by ω and a non-parametric graph pooling function g. ω thus also denotes the

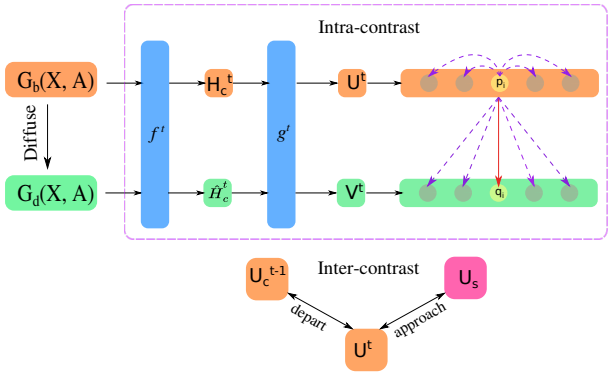

Fig. 1: Schematic of FCLG framework architecture. Given one batch of graph data $G_b(X, A)$, graph diffusion is used to generate an augmented counterpart $G_d(X, A)$. These two views are fed to the subsequent GIN based encoder f to generate latent node-level representations (H_c^t and $\widehat{H}_c^t$, respectively). The graph pooling function g will further summarize (H_c^t, $\widehat{H}_c^t$) by summing up all features for each graph along the columns, yielding (U^t, V^t), on which intra-contrastive learning is performed (the superscript t indicates the local training epoch). The inter-contrastive learning step will pull apart the representations (U_c^{t-1}, U^t) between adjacent local training epochs on each client, and bring closer (U^t, U_s) to improve the generalization performance of U_s.

model parameters of FCLG. We use the GIN [10] due to its advanced performance in graph-level tasks [7] and apply a summing-up graph pooling function as described in [22], [8], [12], [11]. Within this module, f containing stacked L-layers of GIN, will first abstract information from graphs into a series of latent representations $\{h^k\}_{k=1}^{L}$. Inspired by [34], we summarize representations at all depths of neural networks through concatenation along the feature dimension, with the goal of capturing local information centered at every node:

$$H = CONCAT(\{h^k\}_{k=1}^{L}), \quad U = POOL(H)$$

Here, H indicates the node-level latent representations and U is the pooled graph-level representation of the input graph $G_b(X, A)$. Similarly, V is the pooled graph-level representation of the augmented view $G_d(X, A)$. The model will be optimized in a self-supervised manner via contrastive learning objectives.

Complete Algorithm. The entire training process is summarized in **Algorithm 1**. During each communication round, the global model on the server will send model parameters to clients, receive the local model from each client, and update the global model by Eq.1 (on line 2-9). During local training, each client C_i first downloads the most updated model parameters from the server (on line 11). Next, we perform the training on each batch of graphs G_b locally stored on each client. Augmented view of graphs G_d will be generated through graph diffusion mechanism summarized earlier (on line 14). The GIN [10] based encoder f and graph pooling function g abstract graph data into latent graph-level representations (on line 15-16): 1) U^t by the current local model w_i^t encoding G_b; 2) V^t by the current local model w_i^t encoding the augmented view G_d; 3) U_s by the global model w^j encoding G_b; 4) U_c^{t-1} by the previous epoch of local model w_i^{t-1} encoding G_b. These obtained representations will be used to compute intra-contrasting and inter-contrasting losses (on line 17). The total training loss can be computed by summing two contrastive losses up (on line 18) by:

$$l_{tot} = l_{inter} + l_{intra} \tag{9}$$

TABLE I: Statistics of datasets for unsupervised graph-level representation learning.

Domain	Dataset	Graphs	Avg. Nodes	Avg. Edges	class
Proteins	ENZYMES [35]	600	32.63	62.14	6
	PROTEINS [35]	1113	39.06	72.82	2
Molecules	DHFR [36]	467	42.43	44.54	2
	NCI1 [37]	4110	29.87	32.30	2

Algorithm 1 FCLG training for graph-level representations

Input: local training epochs E, communication rounds T, number of clients K and γ the fraction of clients participating in each round.
Output: learned model parameters ω^T

1: **Server:**
2: initialize ω^0
3: **for** $j = 0, 1, \cdots, T-1$ **do**
4: Randomly sample a subset of K_γ clients
5: **for** $i = 1, 2, \cdots, K_\gamma$ in parallel **do**
6: send global model ω^j to C_i
7: $\omega_i^j \leftarrow$ **Client**(i, ω^j)
8: $\omega^{j+1} \leftarrow \sum_i^{K_\gamma} \frac{|S^i|}{|S|} \omega_i^j$
9: return ω^T
10: **Client**(i, ω^j)**:**
11: $\omega_i^0 \leftarrow \omega^j$
12: **for** $t = 1, 2, \cdots, E$ **do**
13: **for** each batch of graphs $G_b \in S^i$ **do**
14: $G_d \leftarrow$ graph diffusion on G_b
15: $U^t \leftarrow g(f_{\omega_i^t}(G_b))$; $V^t \leftarrow g(f_{\omega_i^t}(G_d))$
16: $U_s \leftarrow g(f_{\omega^j}(G_b))$; $U_c^{t-1} \leftarrow g(f_{\omega_i^{t-1}}(G_b))$
17: Calculate l_{intra} from Eq.6 and l_{inter} from Eq.7
18: $l_{tot} \leftarrow l_{inter} + l_{intra}$
19: SGD update on ω_i^t
20: return ω_i^E

III. Experimental Results

A. Experimental Settings

Datasets and Baselines: we use four datasets [38] for the graph-level clustering task covering multiple domains as in **Table I**. We divide graphs of each dataset into 6 clients for federated training. To build baselines, we follow similar settings

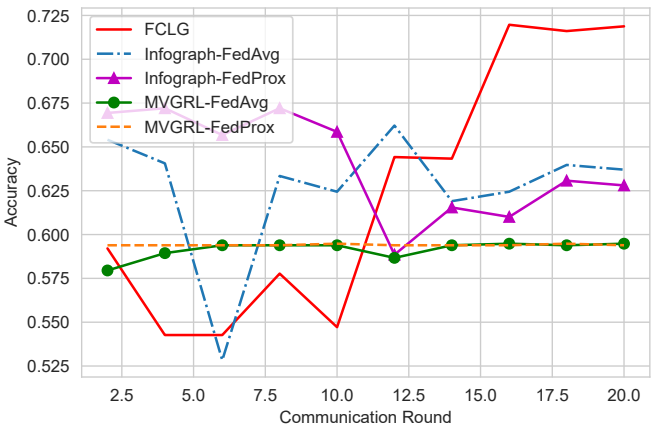

Fig. 2: Accuracy Vs. Communication Round on PROTEINS

from recent efforts [7], [27]. We combine different prior federated learning strategies (FedAvg [6] and FedProx [39]) with the graph-level representation learning methods (*InfoGraph* [8] and *MVGRL* [22]).

Metrics: We evaluate the quality of learned graph-level embeddings through its performance on the downstream graph-level clustering, specifically using two metrics: accuracy and macro F1-score. To quantify non-identical distributions for different (versions of) datasets, we apply an average Earthmover's Distance (EMD) [40] metric. Following [41], we take the discrete graph class distribution $\mathbf{q_i}$ for each client. The class distribution across the total population including datasets from all clients is denoted as $\mathbf{p}$. Hence, the non-identical distribution of a dataset is calculated as average distances between the clients and the population in a weighted fashion:

$$\sum_i \frac{|S_i|}{|S|} Dist(\mathbf{q_i}, \mathbf{p})$$

As indicated earlier, $|S_i|$ indicates the data size on each client and $|S| = \sum_i |S_i|$. Dist$(\cdot, \cdot)$ is a specific distance metric, where in our particular case we use the EMD$(\mathbf{q}, \mathbf{p}) \triangleq ||\mathbf{q} - \mathbf{p}||_\mathbf{1}$ and EMD$(\mathbf{q}, \mathbf{p}) \in [\mathbf{0}, \mathbf{2}]$. Hence a larger EMD implies a more skewed distribution.

B. Graph-level Clustering Results

We evaluate the clustering performance following the clustering evaluation protocol as taken by previous works [22]. Specifically, we set the number of clusters to the number of ground-truth graph-level classes and perform K-Means algorithm [42] on the resulting graph-level embedding. Table II shows the graph-level clustering results on four datasets with non-IID client data – each experiment is performed for 10 times with both the average and the range reported. We provide the EMD values in alignment with each dataset name.

On top of FCLG, we also tested the variant model **FCLG-H**, where intra-contrasting is conducted between node-level representations H via Eq. 8. More variants are based on different ensembling distillation strategies: **Intra-FedAvg** by removing the term l_{inter} (Eq.9) and only keeping intra-contrastive loss; **Intra-KL** by replacing l_{inter} with l_{KL} as in Eq.2; **Intra-MSE** by replacing l_{inter} with l_{MSE} as in Eq.3. Our FCLG based models obtained state-of-the-art graph-level clustering metrics across all datasets with a significant clustering gain 2-7%. Figure 2 shows graph-level clustering accuracy after a varying number of communication rounds on the dataset PROTEINS , where FCLG starts achieving a clearly superior performance after a few rounds.

IV. Conclusions

In this paper, we have presented and evaluated a new federated learning framework referred to as FCLG. This framework has overcome two difficulties in graph-level representations learning under federated settings: 1) inaccessibility of graph-level ground-truth labels; 2) poor generalization performance of federated learning when different sites see distinct distributions (the non-IID issue). With a unique two-level contrastive learning mechanism, FCLG has attained high robustness with non-IID and obtained high quality graph-level representations. Our extensive evaluation has shown FCLG achieved state-of-the-art graph-level clustering accuracy compared with the baselines we constructed, specifically, 2-7% clustering accuracy gain in non-IID settings.

References

[1] R. Angles and C. Gutierrez, "Survey of graph database models," *ACM Comput. Surv.*, vol. 40, no. 1, feb 2008. [Online]. Available: https://doi.org/10.1145/1322432.1322433

[2] T. Suzumura, Y. Zhou, N. Barcardo, G. Ye, K. Houck, R. Kawahara, A. Anwar, L. L. Stavarache, D. Klyashtorny, H. Ludwig, and K. Bhaskaran, "Towards federated graph learning for collaborative financial crimes detection," *CoRR*, vol. abs/1909.12946, 2019. [Online]. Available: http://arxiv.org/abs/1909.12946

[3] C. Wu, F. Wu, Y. Cao, Y. Huang, and X. Xie, "Fedgnn: Federated graph neural network for privacy-preserving recommendation," 2021.

[4] K. Zhang, C. Yang, X. Li, L. Sun, and S. M. Yiu, "Subgraph federated learning with missing neighbor generation," 2021.

[5] C. Ke and J. Honorio, "Federated myopic community detection with one-shot communication," 2021.

[6] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017.

[7] H. Xie, J. Ma, L. Xiong, and C. Yang, "Federated graph classification over non-iid graphs," *ArXiv*, vol. abs/2106.13423, 2021.

[8] F.-Y. Sun, J. Hoffman, V. Verma, and J. Tang, "Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *International Conference on Learning Representations*, 2020.

[9] K. N. Thomas and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[10] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=ryGs6iA5Km

[11] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," ser. AAAI'18/IAAI'18/EAAI'18, 2018.

[12] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Curran Associates Inc., 2018, p. 4805–4815.

[13] B. Pang, Y. Fu, S. Ren, Y. Wang, Q. Liao, and Y. Jia, "Cgnn: Traffic classification with graph neural network," 2021.

TABLE II: FCLG graph-level clustering performance for Non-IID client data.

Model	Metric	PROTEINS (0.5774)	ENZYMES (1.2667)	DHFR (0.5694)	NCI1 (0.5995)
InfoGraph-FedAvg	Accuracy	62.40 ± 1.59	26.00 ± 1.23	56.59 ± 0.05	57.82 ± 0.19
	F1-macro	61.58 ± 1.19	25.85 ± 1.34	55.51 ± 0.05	57.59 ± 0.20
InfoGraph-FedProx	Accuracy	60.97 ± 2.39	27.93 ± 0.15	57.54 ± 0.00	59.15 ± 0.56
	F1-macro	60.47 ± 2.19	27.89 ± 0.18	57.35 ± 0.00	58.85 ± 0.85
MVGRL-FedAvg	Accuracy	59.35 ± 0.10	17.33 ± 0.00	61.11 ± 0.00	50.07 ± 0.00
	F1-macro	37.25 ± 0.04	6.12 ± 0.00	38.25 ± 0.00	33.41 ± 0.00
MVGRL-FedProx	Accuracy	59.30 ± 0.20	17.50 ± 0.00	61.11 ± 0.00	50.07 ± 0.00
	F1-macro	37.22 ± 0.08	6.45 ± 0.00	38.25 ± 0.00	33.41 ± 0.00
FCLG	Accuracy	69.90 ± 2.59	32.45 ± 0.45	67.01 ± 0.83	61.42 ± 0.07
	F1-macro	69.12 ± 1.97	30.61 ± 0.40	66.59 ± 0.76	60.03 ± 0.07
FCLG-H	Accuracy	69.95 ± 2.57	32.03 ± 1.22	65.38 ± 0.84	61.45 ± 0.06
	F1-macro	68.98 ± 2.08	29.76 ± 1.53	64.93 ± 0.90	60.05 ± 0.07
Intra-FedAvg	Accuracy	70.41 ± 2.35	30.50 ± 0.07	60.22 ± 1.19	61.55 ± 0.10
	F1-macro	69.29 ± 1.93	29.96 ± 0.07	59.43 ± 1.18	60.23 ± 0.10
Intra-KL	Accuracy	66.58 ± 1.14	31.00 ± 0.00	61.44 ± 0.25	61.31 ± 0.14
	F1-macro	66.07 ± 0.99	29.95 ± 0.00	58.59 ± 0.24	59.95 ± 0.17
Intra-MSE	Accuracy	64.80 ± 2.17	29.83 ± 0.99	61.48 ± 0.05	61.52 ± 0.07
	F1-macro	64.72 ± 2.09	29.24 ± 1.19	59.70 ± 0.04	60.25 ± 0.08

[14] A. Garcia-Garcia, B. S. Zapata-Impata, S. Orts, P. Gil, and J. G. Rodríguez, "Tactilegcn: A graph convolutional network for predicting grasp stability with tactile sensors," *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2019.

[15] D. Jiang, Z. Wu, C. Hsieh, G. Chen, B. Liao, Z. Wang, C. Shen, D. Cao, J. Wu, and T. Hou, "Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models." *J Cheminform*, vol. 13, no. 1, 02 2021.

[16] X. Ji and Q. Meng, "Traffic classification based on graph convolutional network," in *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)*, 2020, pp. 596–601.

[17] V. Gligorijević, P. D. Renfrew, T. Kosciolek, J. K. Leman, D. Berenberg, T. Vatanen, C. Chandler, B. C. Taylor, I. M. Fisk, H. Vlamakis, R. J. Xavier, R. Knight, K. Cho, and R. Bonneau, "Structure-based protein function prediction using graph convolutional networks," *Nature Communications*, vol. 12, no. 1, p. 3168, May 2021.

[18] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal, "graph2vec: Learning distributed representations of graphs," *arXiv preprint arXiv:1707.05005*, 2017.

[19] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," *arXiv preprint arXiv:1905.12265*, 2019.

[20] N. Pržulj, "Biological network comparison using graphlet degree distribution," *Bioinformatics*, vol. 23, no. 2, pp. e177–e183, 2007.

[21] F. Orsini, P. Frasconi, and L. De Raedt, "Graph invariant kernels," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[22] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proceedings of International Conference on Machine Learning*, 2020, pp. 3451–3461.

[23] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[24] A. Mnih and K. Kavukcuoglu, "Learning word embeddings efficiently with noise-contrastive estimation," *Advances in neural information processing systems*, vol. 26, 2013.

[25] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[26] H.-Y. Chen and W.-L. Chao, "Fedbe: Making bayesian model ensemble applicable to federated learning," *arXiv preprint arXiv:2009.01974*, 2020.

[27] F. Zhang, K. Kuang, Z. You, T. Shen, J. Xiao, Y. Zhang, C. Wu, Y. Zhuang, and X. Li, "Federated unsupervised representation learning," *arXiv preprint arXiv:2010.08982*, 2020.

[28] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 2351–2363. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/18df51b97ccd68128e994804f3eccc87-Paper.pdf

[29] T. Kim, J. Oh, N. Kim, S. Cho, and S.-Y. Yun, "Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conferences on Artificial Intelligence Organization.

[30] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1607–1616.

[31] J. Tang, R. Shivanna, Z. Zhao, D. Lin, A. Singh, E. H. Chi, and S. Jain, "Understanding and improving knowledge distillation," *arXiv preprint arXiv:2002.03532*, 2020.

[32] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "Gcc: Graph contrastive coding for graph neural network pre-training," ser. KDD '20, 2020, p. 1150–1160.

[33] J. Klicpera, S. Weißenberger, and S. Günnemann, "Diffusion improves graph learning," *arXiv preprint arXiv:1911.05485*, 2019.

[34] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *International conference on machine learning*. PMLR, 2018, pp. 5453–5462.

[35] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. V. N. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, 06 2005.

[36] J. J. Sutherland, L. A. O'Brien, and D. F. Weaver, "Spline-fitting with a genetic algorithm: a method for developing classification structure-activity relationships," *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 6, pp. 1906–1915, Nov. 2003.

[37] N. Wale, I. A. Watson, and G. Karypis, "Comparison of descriptor spaces for chemical compound retrieval and classification," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 347–375, Mar 2008.

[38] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "Tudataset: A collection of benchmark datasets for learning with graphs," *arXiv preprint arXiv:2007.08663*, 2020.

[39] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2020.

[40] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. IEEE, 1998, pp. 59–66.

[41] T.-M. H. Hsu, H. Qi, and M. Brown, "Federated visual classification with real-world data distribution," in *European Conference on Computer Vision*. Springer, 2020, pp. 76–92.

[42] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.