

Federated Graph Neural Networks: Overview, Techniques and Challenges

Rui Liu and Han Yu

School of Computer Science and Engineering, Nanyang Technological University, Singapore
 {rui.liu, han.yu}@ntu.edu.sg

Abstract

With its powerful capability to deal with graph data widely found in practical applications, graph neural networks (GNNs) have received significant research attention. However, as societies become increasingly concerned with data privacy, GNNs face the need to adapt to this new normal. This has led to the rapid development of federated graph neural networks (FedGNNs) research in recent years. Although promising, this interdisciplinary field is highly challenging for interested researchers to enter into. The lack of an insightful survey on this topic only exacerbates this problem. In this paper, we bridge this gap by offering a comprehensive survey of this emerging field. We propose a unique 3-tiered taxonomy of the FedGNNs literature to provide a clear view into how GNNs work in the context of Federated Learning (FL). It puts existing works into perspective by analyzing how graph data manifest themselves in FL settings, how GNN training is performed under different FL system architectures and degrees of graph data overlap across data silo, and how GNN aggregation is performed under various FL settings. Through discussions of the advantages and limitations of existing works, we envision future research directions that can help build more robust, dynamic, efficient, and interpretable FedGNNs.

1 Introduction

Graph neural networks (GNNs) are powerful tools for dealing with graph-structured data [Wu *et al.*, 2020]. Graph-structured data are data samples connected by a graph topology. For example, molecule data are graph-structured data with atoms as the nodes in the graph and the bonds connecting them as the edges in the graph. GNNs can improve the quality of node embedding by considering neighborhood information extracted from the underlying graph topology. They have been widely adopted in diverse applications including drug discovery [Gilmer *et al.*, 2017], neuroscience [Ding *et al.*, 2021], social networks [Hamilton *et al.*, 2017], knowledge graphs

[Chen *et al.*, 2020], recommender systems [Ying *et al.*, 2018] and traffic flow prediction [Cui *et al.*, 2019].

A well-trained GNN requires a large amount of training graph data, which may be distributed in different institutes or data owners in practice. Due to privacy concerns [GDPR, 2018], these data owners may not be willing to share the data, which leads to the problem of data isolation. Furthermore, the distributions of the graph data stored by different data owners are often non-identically distributed (non-IID), which adds to the challenge of data isolation. Such non-IID properties can manifest as differences in graph structures or node feature distributions across data owners. Subgraphs in a data owner can also be biased compared to the global graph distribution.

Federated learning (FL), a distributed collaborative machine learning paradigm, is a promising approach to deal with this data isolation challenge. It enables local models to benefit from each other, while keeping local data private [Kairouz *et al.*, 2021]. In addition, the problem of learning personalized FL models in the presence of non-IID data has been extensively studied [Tan *et al.*, 2021]. In FL, only model parameters or embedding features are shared by FL participants without exposing potentially sensitive local data. This architectural design, combined with various cryptographic techniques, can provide effective protection of local data privacy.

The confluence of these two trends of development has inspired the emergence of the field of federated graph neural networks (FedGNNs) in recent years [Jiang *et al.*, 2020], which has witnessed rapid development over the year of 2021. A position paper [Zhang *et al.*, 2021b] summarized this trend by envisioning four possible FedGNN scenarios based on how graph data are partitioned among data owners. As technical research for these envisioned scenarios was not widely found at the time, there are overlaps in the boundaries separating these scenarios. For example, the “horizontal intra-graph FL” can be confused with “graph-structured FL” when the missing links between graphs stored by different data owners are reconstructed. It can also be confused with “inter-graph FL” if there is only one graph per data owner. To date, there is no insightful survey into the interdisciplinary topic of FedGNN to guide interested researchers to enter into this promising but challenging field.

In this paper, we bridge this gap by providing a comprehen-

sive review of the existing literature on FedGNN. We propose a multi-tiered taxonomy which first divides existing works according to the relationship between graph data to the FL data owner, then based on how GNN training is performed under different FL system architectures and degrees of graph data overlap across data silo, and finally based on how GNN aggregation is performed under various FL settings. It offers a closely integrated view to help readers understand how these two fields can complement each other. We analyse the advantages and limitations of existing FedGNN approaches, and discuss promising future research directions that can lead to more robust, dynamic, efficient, and interpretable FedGNNs.

2 Terminology and Taxonomy

In this section, we explain key terminologies used by the GNN and FL fields and introduce the proposed 3-tiered FedGNN taxonomy.

2.1 Terminology

In GNNs, a graph, which consists of *nodes* and *edges*, is represented by an *adjacency matrix* $\mathbf{A} \in \mathbb{R}^{N \times N}$. Nodes can contain *node features* $\mathbf{X} \in \mathbb{R}^{N \times f}$.

In FL, data owners with sensitive local data can be referred to *clients* if they are coordinated by a central entity referred to as the *server*. This setting is referred to *centralized FL*. In situations in which data owners communicate with each other directly without a central server, the setting is referred to *decentralized FL*.

GNN and FL both involve an “aggregation” operation. Aggregation in the context of GNN updates the embedding of a given node by aggregating information from its neighboring nodes. The aggregation operation could be the mean, weighted average, or max/min pooling methods. Aggregation in the context of FL updates model parameters in the server with local model parameters uploaded by data owners by following a given FL algorithm (e.g., FedAvg [McMahan *et al.*, 2017]). To disambiguate the aggregation operations in GNN and FL, we refer to them as “*GNN aggregation*” and “*FL aggregation*”, respectively in this paper.

2.2 The Proposed 3-Tiered FedGNN Taxonomy

The proposed 3-tiered FedGNN taxonomy is shown in Figure 1. In the first tier, we divide existing works into two categories according to the location of the graph: 1) data owners related by a graph, and 2) data owners not related by a graph.

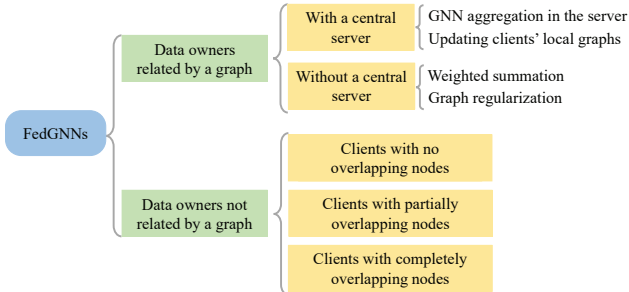


Figure 1: The Proposed 3-Tiered FedGNN Taxonomy.

In the first category, data owners are related by a graph topology. For example, a bank can be a data owner which holds many accounts as the nodes in the graph and transactions between accounts as the edges. The bank can be related to other banks through transaction edges among accounts held by different banks. Note that as long as the data owners are related by a graph topology, the format of the local data does not necessarily need to be a graph. Works in this category can be further divided into two scenarios in the second tier of our taxonomy according to whether a central server is involved in the FL model training process: 1) with a central server, and 2) without a central server. The central server has a global view of the inter-client graph topology. It can leverage this view to: 1) train a GNN model in the server to improve FL aggregation; and 2) help clients update their local graphs (the third tier of our taxonomy). This global view also enables the server to estimate the graph topology during training, instead of requiring the graph topology to be known in advance. Without a central server, the inter-data-owner graph topology must be given in advance so that FL data owners can find their neighbors. In this setting, GNN aggregation is performed in two ways among neighboring data owners: 1) weighted summation, and 2) graph regularization (the third tier of our taxonomy).

In the second category, data owners are not related by a graph topology. Instead, the graph topology only exists within each data owner’s local dataset. For example, multiple e-commerce companies each having user-item browsing data represented as graphs collaborate to train a recommender system model leveraging FedGNN. Existing works under this category all belong to the centralized FL setting. They can be further divided into three specific scenarios according to the overlap of nodes among clients: 1) clients with no overlapping nodes, 2) clients with partially overlapping nodes, and 3) clients with completely overlapping nodes. In the following sections, we discuss existing works in FedGNN based on the proposed taxonomy.

3 Data Owners Related by a Graph

In this section, we introduce the category of FedGNN research that assumes data owners are related by a graph. Under this setting, it is commonly assumed data owners’ local data distributions are non-IID, and those who are closely related on the graph are likely to share similar data distributions. For example, traffic monitoring sensors installed nearby on a road network tend to record similar traffic conditions.

3.1 FedGNNs with a Central Server

Figure 2 illustrates the situation of centralized FedGNNs. Clients’ local data do not necessarily need to be graph data. The central server coordinates clients (indicated by the arrow lines in Figure 2) based on their relationships in the graph (indicated by the dashed lines in Figure 2). The server performs two coordination activities. Firstly, it performs GNN aggregation based on the graph topology. Secondly, it helps clients update their local graphs by estimating the missing edges connecting nodes in different clients based on the graph topology.

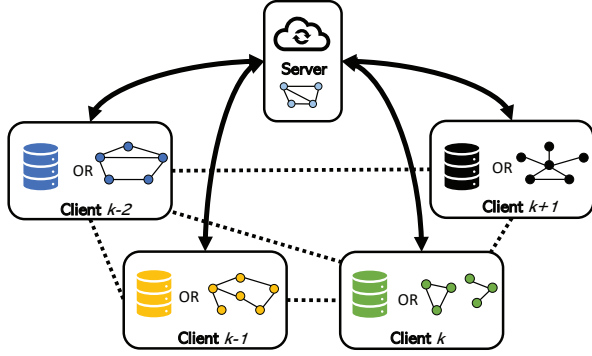


Figure 2: Illustration of a FedGNN with a central server.

GNN Aggregation by the FL Server

In [Meng *et al.*, 2021], inductive federated learning scheme is proposed to deal with spatio-temporal data leveraging an alternating optimization procedure. It disentangles the spatio-temporal information via split learning by extracting temporal features with a Gated Recurrent Unit in a client and encoding the connections among clients using their node embeddings with a GNN model in the server. To increase training efficiency, the server also performs FL aggregation via FedAvg based on the clients’ uploaded model parameters.

In [Xing *et al.*, 2021], the FedGNN is modeled as a bi-level optimization problem. Compared to [Meng *et al.*, 2021] which uses a summation of local objective functions for training, this work trains client models with local task objective functions in the inner loop, and trains the GNN model in the server with a separate contrastive learning objective function in the outer loop. The server collects model parameters from clients as node features for the Graph Convolutional Network (GCN) [Kipf and Welling, 2016] model. The model parameters are updated based on the graph topology among the clients. As the first work modeling this problem from the bi-level optimization perspective, it provides a theoretical analysis of convergence under this framework.

Updating Clients’ Local Graphs

Under this setting, it is assumed that each client’s local data are in the form of a sub-graph of an entire graph. The relationship graph topology connecting clients is leveraged to help them update their local graphs, which benefits their local GNN model training and mitigates the non-IID data problem. In general, the server applies FedAvg to aggregated uploaded client model parameters.

In [Zhang *et al.*, 2021a], the server reconstructs the missing edges between clients with Gaussian randomly generated edges and broadcasts them to all clients to update their local graphs. The local graphs are used in the local attention-based spatial-temporal GNN model training to learn graph embeddings from temporal graph data.

The server in [Chen *et al.*, 2021a] generates a global pseudo graph with node embeddings uploaded by FL clients, and distributes it to the clients to update their local graphs for GCN model training. Besides, the server also generates global pseudo node labels based on the uploaded node predictions, which are incorporated into the self-supervised learning loss of the clients to mitigate the non-IID data problem.

In [Zhang *et al.*, 2021c], a missing edge generator is trained for each client based on the local sub-graph data and closely proximal neighboring clients. The missing edge generator repairs missing edges among clients’ local sub-graph data. Once the sub-graph is updated, each client trains the GraphSage [Hamilton *et al.*, 2017], an inductive GNN model, and uploads the model parameters to the server for aggregation. [Peng *et al.*, 2021b] also proposed a missing edge generator based on a Generative Adversarial Network (GAN). Compared to [Zhang *et al.*, 2021c], it removes nodes in a client with breadth-first search, instead of random walk, during the generator training phase. A population graph generation method is also proposed to transform neuro-imaging data into graph-structured data to leverage FedGNN for disease prediction. Each subject is regarded as a node in the population graph, and the similarity between the images and phenotypes of pairs of subjects are used to compute edge weights.

In [Caldarola *et al.*, 2021], the authors assume that client local data consists of images from multiple domains, and the server can generate and update the domain-relationship graph topology during the training process. The server generates the adjacency matrix of a fully connected graph among domains from all clients based on the uploaded domain-specific layer parameters, and shares it with the clients. Each client clusters local images into domain-specific clusters in the unsupervised teach-student classifier training iterations, followed by a two-part CNN-based classification model to classify the local samples. The first part consists of domain-agnostic layers to extract common features. The second part is domain-specific layers tailored for each client, whose model parameters are updated via GNN aggregation in the clients.

In [Wu *et al.*, 2021], the focus of research starts to shift towards data privacy protection in FedGNN. A privacy-preserving user-item graph expansion method is proposed. It is executed on a third-party server to expand the clients’ local user-item graph. Then, each client embeds items and user information with a GNN model. To protect data privacy, clients’ local model gradients are encrypted before being sent to the server. In addition, pseudo interaction items are added to each client to protect user identities in the local graph data.

3.2 FedGNNs without a Central Server

Figure 3 illustrates the situation of decentralized FedGNNs. There is no server in the system to coordinate data owners, which hold either normal data or graph data. Data owners communicate with their neighbors directly (i.e., the graph topology among clients is known in advance) for FL aggregation. There are two ways for data owners to update the local model parameters by leveraging the inter-data-owner graph

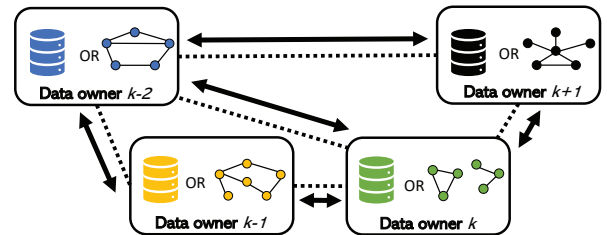


Figure 3: Illustration of a FedGNN without a central server.

topology: 1) updating the FL model parameters via weighted summation, and 2) updating the FL model parameters via graph regularization. Due to the distributed setting, the eventual aggregated model in each data owner is a personalized FL model. This applies to all works under this section.

Weighted Summation of FL Model Parameters

In this setting, a data owner communicates with its neighbors and updates its local FL model by aggregating their model parameters based on the graph topology:

$$\mathbf{w}_i^{(t+1)} = \sum_{j \in N(i)} a_{ij} \cdot [\mathbf{w}_j^{(t)}] \quad (1)$$

where $\mathbf{w}_i^{(t+1)} \in \mathbb{R}^p$ denotes the model parameters of data owner i at iteration $(t + 1)$. $[\cdot]$ is the encryption operation for data privacy protection. a_{ij} is the $[i$ -th row, j -th column] element in the adjacency matrix \mathbf{A} of the graph, which is assumed to reflect the local data distribution similarity between i and j . $N(i)$ is neighborhood of i (including itself). All works under this section apply Eq. (1) once per round (i.e., a data owner only aggregates models from its 1-hop neighbors).

[Lalitha *et al.*, 2019] is the first work that considers the graph connectivity among data owners in a decentralized FL setting. Data owners share their local Bayesian model weights with their neighbors to learn a model that fits the observations over the entire network. It provides a theoretical guarantee of the lower bound on the number of training samples required to ensure a target error probability. Nevertheless, it has only been evaluated with two data owners each having synthetic data, which lacks realism.

Both [Scardapane *et al.*, 2020] and [Pei *et al.*, 2021] model the decentralized FedGNN training as a distributed optimization problem, and provide theoretical analysis on the convergence and generalization error rates. To enhance performance, [Scardapane *et al.*, 2020] leverages a 2-layer GCN and [Pei *et al.*, 2021] leverages a multi-layer GNN to handle complex local graph data. Nevertheless, these multiple-layer local models decrease communication efficiency.

To enhance communication efficiency, [Rizk and Sayed, 2021] proposed a multi-server FedGNN architecture. It assumes that there are multiple servers in the network related by a fixed and well-known graph topology. Clients under each server conduct FL model training following the classic centralized FL protocol. Once all the servers have aggregated their own clients' model updates, they perform inter-server model aggregation following Eq. (1) among themselves. There is no central server coordinating the network of servers. It is a kind of clustering-based FL. Instead of data owners directly communicating with each other, they only communicate with their respective servers.

Graph Regularization on FL Model Parameters

In this setting, each data owner incorporates a graph Laplacian regularization into the objective function to make model parameters from neighboring clients similar in order to address the non-IID data problem [Ortega *et al.*, 2018]:

$$R(\mathbf{W}, \mathbf{L}) = \text{tr}(\mathbf{W}^T \mathbf{L} \mathbf{W}) = \frac{1}{2} \sum_{ij} a_{ij} \|\mathbf{w}_i - \mathbf{w}_j\|^2 \quad (2)$$

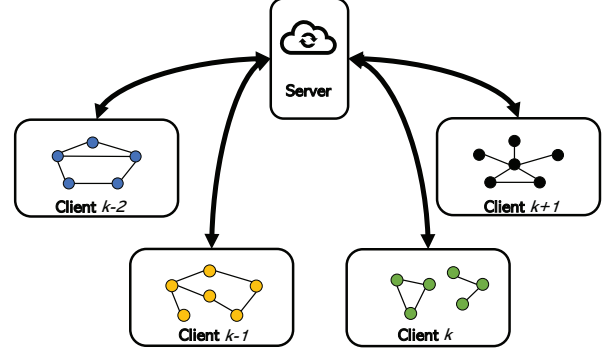


Figure 4: An illustration of clients with no overlapping nodes.

where $\mathbf{W} \in \mathbb{R}^{n \times p}$ denotes the model weights of neighboring clients. $\text{tr}(\cdot)$ is the trace operation. $\mathbf{L} \in \mathbb{R}^{n \times n}$ is the Laplacian matrix of the graph topology between neighboring clients. a_{ij} is the edge weight in the adjacency matrix connecting data owner i and j . $\mathbf{w}_i \in \mathbb{R}^p$ denotes the model parameters in data owner i .

[Dinh *et al.*, 2021a; Dinh *et al.*, 2021b] proposed a federated multi-task learning (FMTL) scheme with Laplacian regularization to fit different models to the local data in order to address the non-IID data problem. The assumptions are that each data owner focuses on one task, and there exists a fully connected graph among data owners with fixed edge weights known in advance. Each data owner updates the local deep model for multiple rounds, and sends the updated model to its neighbors to perform model regularization using the graph Laplacian regularization in Eq. (2).

[He *et al.*, 2021b] also proposed a FMTL scheme to solve the non-IID problem for distributed multi-labeled graph data. The difference is that it assumes each data owner has multiple tasks. Instead of applying Laplacian regularization with the relationship graph among data owners, it applies Laplacian regularization with a task relationship graph among tasks, which is the inverse of the covariance matrix between related tasks, represented by their classifier model parameters. A client trains a Message-Passing-based GNN model locally to produce a graph embedding. Then, the graph embedding is inputted into a task classifier with a task-relationship graph regularization. The task classifier model parameters are shared among neighboring data owners for FL aggregation.

4 Data Owners not Related by a Graph

In this section, we introduce FedGNN research works which assume data owners are not related by a graph. Under this setting, graph typologies and GNN models only exist in clients' local data. The assumption is that graphs from different domains are stored by different data owners. Thus, sharing graph information among data owners via the FL server can be beneficial. According to the degrees of overlap in graph nodes across clients, works in this category can be divided into three categories: 1) clients with no overlapping nodes, 2) clients with partially overlapping nodes, and 3) clients with completely overlapping nodes.

4.1 Clients with No Overlapping Nodes

Figure 4 illustrates the situation in which FedGNN clients hold no overlapping nodes (denoted by different colors). Clients train local GNN models with local graph data, and upload model parameters to the server for FL aggregation.

[Jiang *et al.*, 2020] predicts object positions in distributed video surveillance data with FedGNN. Each camera is a client. Objects in the videos, detected by Convolutional Neural Networks, are regarded as nodes in a graph. Graph edge weights denote the Euclidean distance between objects. In each client, the spatial patterns are captured by a dynamic GNN model trained with a self-supervised loss. The local models from devices are uploaded to the server for FL aggregation. To guard against inference attacks, it incorporates a cryptographic method in the FL aggregation step.

In [Bayram and Rekik, 2021], a FedGNN framework is proposed to learn connective brain templates (CBT) for hospitals. Each hospital (i.e., client) has a set of graphs. It trains a GNN model with a Subject Normalization Loss (SNL) function. The features from the last layer are used to generate a CBT. The server aggregates local models using a temporary-weighted averaging method in a layer-wise manner, and distributes the updated model to all clients. The proposed framework helps hospitals share their knowledge and improve the representativeness of learned CBTs.

[Xie *et al.*, 2021] proposed a FedGNN scheme to classify molecule properties in distributed graph datasets. The server performs top-down bipartition clustering to dynamically cluster clients based on the gradients of the Graph Isomorphism Networks model [Xu *et al.*, 2018] from each client, and performs FL aggregation with FedAvg in a cluster-wise manner. It also reduces fluctuations in gradients across different rounds by using dynamic time warping with gradients from multiple training rounds. [Zhu *et al.*, 2021] proposed a FedGNN for molecule properties classification. It solves the non-IID data problem by reweighting an instance based on its prediction confidence. Higher weights are placed on samples with low prediction confidence in the local loss function to make the local training more consistent across clients.

In [Zheng *et al.*, 2021] and [Chen *et al.*, 2021b], FedGNN schemes are proposed to classify papers in a distributed citation network with different focus. [Zheng *et al.*, 2021] mainly focuses on the non-IID data problem among different clients by adding an extra model parameter updating step in the clients after FL aggregation. It refines the global model parameters based on the JS-divergence between a client’s dataset distribution and the server’s dataset distribution. This work also leverages Bayesian optimization to automatically tune the hyper-parameters of all clients’ models as an outer loop optimization implemented in the server. On the other hand, [Chen *et al.*, 2021b] focuses on the sampling strategy for clients with large-scale graphs. It designs a sampling policy for the server based on reinforcement learning to refine the clients’ sampling strategies in each round. Then, each client applies its sampling strategy on the local large-scale graph to reduce the computation over in GCN model training, and uploads model parameters to the server for FL aggregation.

In [Liu *et al.*, 2021], FedGNN is being studied in social recommendation settings. A personalized relational-

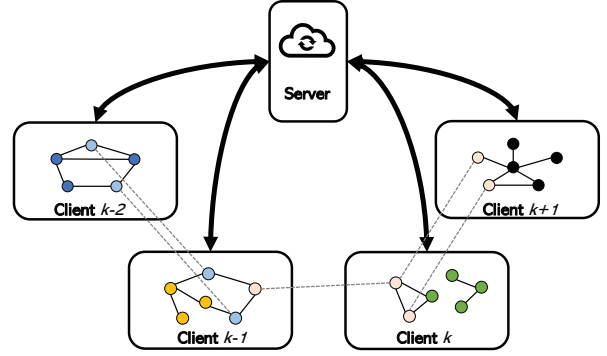


Figure 5: An illustration of clients with partially overlapping nodes.

attention-based GNN model is designed to handle heterogeneous user-user graphs and user-item graphs in the clients. They upload encrypted gradients to the server for FL aggregation. FedGNN-based sleep stage classification has been studied in [Lou *et al.*, 2021]. A graph generation method is designed to transform non-graph-structured temporal data into graph-structured data. By regarding each channel as one node in the graph, the edge weights are calculated based on the correlation between pairs of channels.

4.2 Clients with Partially Overlapping Nodes

Figure 5 illustrates the situation in which the nodes in clients’ local datasets partially overlap. Overlapping nodes in different clients are painted with the same color and indicated by dashed lines. These overlapping nodes are treated in different ways by existing FedGNN approaches.

In [Wang *et al.*, 2020], the effect of overlapping nodes on FedGNN is studied without considering the corresponding relationships among overlapping nodes. A meta-learning-inspired FedGNN framework for semi-supervised node classification is proposed. It consists of two stages: 1) learning a global model on the server following the MAML [Finn *et al.*, 2017] training scheme to mitigate the non-IID data problem, and 2) leveraging the FL scheme to further update the global model with local models to improve generalization ability on test data with new labels. Each client trains a GCN model with a self-training method to leverage unlabeled nodes with pseudo labels for its local task.

[Shan *et al.*, 2021] replaced the weighted summation operation of neighboring nodes’ features in the GNN aggregation step with a max/min pooling operation to avoid duplicate calculations in the overlapping nodes. It proposed a split-learning-based FedGNN scheme for node classification. The execution of GNN is split on a per-layer basis between each client and the server. The node embeddings from the client and the server are updated in the forward propagation step. The gradients of local model parameters are updated in the back propagation step. To protect privacy, the server applies a secure global pooling FL aggregation mechanism.

The overlapping nodes in paired knowledge graphs (KGs) are leveraged to translate knowledge embedding across clients [Peng *et al.*, 2021a]. A GAN-based FedGNN framework is proposed to improve KG embedding quality from different domains. Each client holds a KG from one domain. It trains a GNN locally to produce the embedding. Then, a

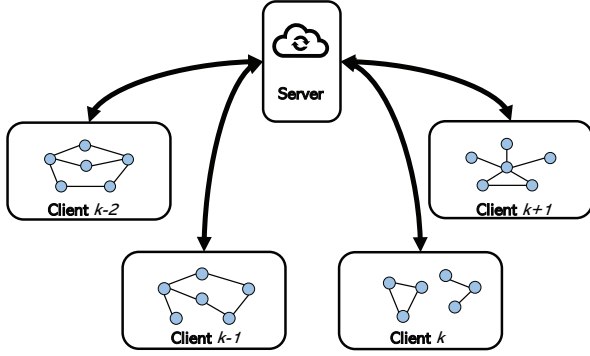


Figure 6: An illustration of clients with fully overlapping nodes.

revised GAN [Jordon *et al.*, 2018] is applied to the aligned overlapping nodes and relations in the paired KGs to translate the embedding across clients. If the paired KGs are improved, the refined embeddings are broadcast to other KGs.

4.3 Clients with Completely Overlapping Nodes

Figure 6 illustrates the situation in which clients hold completely overlapping nodes. Each client only holds part of the node features, and only some of the clients hold the labels for the learning task. All clients hold the same set of nodes, they upload node embeddings instead of model parameters to the server for FL aggregation. Existing works focus on the vertically partitioned citation network data with two clients.

In [Zhou *et al.*, 2020], a FedGNN node classification scheme is proposed. An initial node embedding module in the server integrates encrypted node features from different clients and shares them among the clients. Then, each client trains its local GNN model to update the initial node embeddings, and uploads them to the server for FL aggregation via mean, concatenation, or regression calculations. Once the training is completed, the client with the labels receives the global node embeddings and the final hidden layer from the server to perform node prediction. [Ni *et al.*, 2021] addresses the same problem. The key difference is that it does not require the initial node embedding module.

5 Promising Future Research Directions

As an emerging field, FedGNN research is starting to gain traction. Nevertheless, in order for this technology to be able to deal with challenges in real-world applications, many problems remain to be addressed. Here, we highlight six of them which hold promising opportunities.

Robust FedGNNs against malicious attacks. By sharing node embeddings, graph topology and model parameters, FedGNNs have large attack surfaces. Although some works attempt to address this issue by leveraging differential privacy [Peng *et al.*, 2021a; Zhang *et al.*, 2021a; Wu *et al.*, 2021] or cryptographic methods, such as Secure Multi-Party Computation [Chen *et al.*, 2021b; Zhou *et al.*, 2020], Homomorphic encryption [Ni *et al.*, 2021], Diffie-HellmanKey exchange [Pei *et al.*, 2021] or Secret Sharing [Rizk and Sayed, 2021; Zheng *et al.*, 2021], they are designed to guard against only semi-honest attackers. Additional research is needed to ex-

plore how FedGNNs can be made more robust in the face of malicious privacy attacks.

FedGNNs for dynamic graph data. The graph topology or node features in dynamic graph data can change over time. In such cases, temporal information needs to be considered during GNN training. Works that extract temporal features from the dynamic graph data within each FL client using a Gated Recurrent Unit are starting to emerge [Meng *et al.*, 2021; Zhang *et al.*, 2021a]. However, in the settings of FL clients related by a graph, the relationships among clients can also evolve over time. Thus, FedGNN approaches in which the edge weights and the connectivity among FL clients are learnable need to be explored.

Efficient FedGNNs for large-scale graph data. Existing FedGNNs are generally studied with small-scale distributed datasets. Thus, communication efficiency has not yet been adequately considered. However, in order to scale FedGNNs up to large-scale graph data (e.g., knowledge graphs), communication overhead can be an important bottleneck since the data owners often adopt multi-layer GNN models with a large number of model parameters to be transmitted.

Explainable FedGNNs to improve interpretability. Works on the explainability of GNNs [Yuan *et al.*, 2020] and FL [Li *et al.*, 2021] are starting to emerge. FedGNN involves complex model structures and training processes. Thus, achieving explainability under this setting is even more challenging. The incorporation of explainability into FedGNN needs to jointly consider the needs for interpretability by the stakeholders involved, while balancing the goals of preserving privacy and training models efficiently.

Multi-hop neighborhood aggregation in decentralized FedGNNs. In existing decentralized FedGNN research, only the 1-hop neighbors’ model parameters are aggregated to produce a personalized FL model for each data owner. Although such an approach simplifies the model structure, it limits the ability of FedGNN to leverage the rich neighborhood information in an inter-data-owner graph. New techniques that enable FedGNNs to move beyond this limitation while still keeping the model structure and the training process reasonably simple are desirable.

Realistic distributed graph datasets for benchmarking. Existing FedGNN research works are mostly evaluated with synthetic distributed graph data. These are generated from GNN benchmark datasets, such as Cora, PubMed and Cite-seer. To re-context them into the FL setting, the current practice is to partition the entire graph into multiple sub-graphs, which are then assigned to different data owners. The sizes of the subgraph assigned to each data owner in this way tend to be small. In [He *et al.*, 2021a], an open-source platform supporting 3 GNN models and 2 FL aggregation methods has been proposed. It has collected 36 graph datasets and partitioned them into distributed silos, forming a promising FedGNN benchmarking tool. Nevertheless, the long-term development of the FedGNN field still requires realistic and large-scale federated graph datasets to be established to support experimental evaluation under settings close to practical applications. Real-world graph datasets, such as brain con-

nectomic datasets, molecule datasets, recommender systems and knowledge graphs, can be useful starting points.

References

- [Bayram and Rekik, 2021] Hızır Can Bayram and Islem Rekik. A federated multigraph integration approach for connectional brain template learning. In *International Workshop on Multimodal Learning for Clinical Decision Support*, pages 36–47. Springer, 2021.
- [Caldarola et al., 2021] Debora Caldarola, Massimiliano Mancini, Fabio Galasso, Marco Ciccone, Emanuele Rodolà, and Barbara Caputo. Cluster-driven graph federated learning over multiple domains. In *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’21)*, pages 2749–2758, 2021.
- [Chen et al., 2020] Yu Chen, Lingfei Wu, and Mohammed J Zaki. Toward subgraph guided knowledge graph question generation with graph neural networks. *arXiv preprint arXiv:2004.06015*, 2020.
- [Chen et al., 2021a] Chuan Chen, Weibo Hu, Ziyue Xu, and Zibin Zheng. FedGL: Federated graph learning framework with global self-supervision. *arXiv preprint arXiv:2105.03170*, 2021.
- [Chen et al., 2021b] Fahao Chen, Peng Li, Toshiaki Miyazaki, and Celimuge Wu. Fedgraph: Federated graph learning with intelligent sampling. *IEEE Transactions on Parallel and Distributed Systems*, 33(8):1775–1786, 2021.
- [Cui et al., 2019] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4883–4894, 2019.
- [Ding et al., 2021] Yi Ding, Neethu Robinson, Qiuhaio Zeng, and Cuntai Guan. Lggnnet: learning from local-global-graph representations for brain-computer interface. *arXiv preprint arXiv:2105.02786*, 2021.
- [Dinh et al., 2021a] Canh T Dinh, Tung T Vu, Nguyen H Tran, Minh N Dao, and Hongyu Zhang. Fedu: A unified framework for federated multi-task learning with laplacian regularization. *arXiv preprint arXiv:2102.07148*, 2021.
- [Dinh et al., 2021b] Canh T Dinh, Tung T Vu, Nguyen H Tran, Minh N Dao, and Hongyu Zhang. A new look and convergence rate of federated multi-task learning with laplacian regularization. *arXiv preprint arXiv:2102.07148*, 2021.
- [Finn et al., 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML’17)*, pages 1126–1135, 2017.
- [GDPR, 2018] GDPR. General data protection regulation. <https://gdpr-info.eu/>, 2018. Accessed: 2021-12-08.
- [Gilmer et al., 2017] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning (ICML’17)*, pages 1263–1272, 2017.
- [Hamilton et al., 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS’17)*, 2017.
- [He et al., 2021a] Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Carl Yang, Han Xie, Lichao Sun, Lifang He, Liangwei Yang, Philip S Yu, Yu Rong, et al. Fedgraphnn: A federated learning system and benchmark for graph neural networks. *arXiv preprint arXiv:2104.07145*, 2021.
- [He et al., 2021b] Chaoyang He, Emir Ceyani, Keshav Balasubramanian, Murali Annamaram, and Salman Avestimehr. Spreadgnn: Serverless multi-task federated learning for graph neural networks. *arXiv preprint arXiv:2106.02743*, 2021.
- [Jiang et al., 2020] Meng Jiang, Taeho Jung, Ryan Karl, and Tong Zhao. Federated dynamic gnn with secure aggregation. *arXiv preprint arXiv:2009.07351*, 2020.
- [Jordon et al., 2018] James Jordon, Jinsung Yoon, and Michaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.
- [Kairouz et al., 2021] Peter Kairouz, H. Brendan McMahan, et al. *Foundations and Trends in Machine Learning*, 14(1-2):1–210, 2021.
- [Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [Lalitha et al., 2019] Anusha Lalitha, Osman Cihan Kilinc, Tara Javidi, and Farinaz Koushanfar. Peer-to-peer federated learning on graphs. *arXiv preprint arXiv:1901.11173*, 2019.
- [Li et al., 2021] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, page doi:10.1109/TKDE.2021.3124599, 2021.
- [Liu et al., 2021] Zhiwei Liu, Liangwei Yang, Ziwei Fan, Hao Peng, and Philip S Yu. Federated social recommendation with graph neural network. *arXiv preprint arXiv:2111.10778*, 2021.
- [Lou et al., 2021] Guannan Lou, Yuze Liu, Tiejia Zhang, and Xi Zheng. Stfl: A temporal-spatial federated learning framework for graph neural networks. *arXiv preprint arXiv:2111.06750*, 2021.
- [McMahan et al., 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguerre y Arcaas. Communication-efficient learning of deep networks

- from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS'17)*, pages 1273–1282, 2017.
- [Meng *et al.*, 2021] Chuizheng Meng, Sirisha Rambhatla, and Yan Liu. Cross-node federated graph neural network for spatio-temporal data modeling. *arXiv preprint arXiv:2106.05223*, 2021.
- [Ni *et al.*, 2021] Xiang Ni, Xiaolong Xu, Lingjuan Lyu, Changhua Meng, and Weiqiang Wang. A vertical federated learning framework for graph convolutional network. *arXiv preprint arXiv:2106.11593*, 2021.
- [Ortega *et al.*, 2018] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.
- [Pei *et al.*, 2021] Yang Pei, Renxin Mao, Yang Liu, Chaoran Chen, Shifeng Xu, Feng Qiang, and Blue Elephant Tech. Decentralized federated graph neural networks. In *International Workshop on Federated and Transfer Learning for Data Sparsity and Confidentiality in Conjunction with IJCAI (FTL-IJCAI'21)*, 2021.
- [Peng *et al.*, 2021a] Hao Peng, Haoran Li, Yangqiu Song, Vincent Zheng, and Jianxin Li. Differentially private federated knowledge graphs embedding. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM'21)*, pages 1416–1425, 2021.
- [Peng *et al.*, 2021b] Liang Peng, Nan Wang, Nicha Dvornek, Xiaofeng Zhu, and Xiaoxiao Li. Fedni: Federated graph learning with network inpainting for population-based disease prediction. *arXiv preprint arXiv:2112.10166*, 2021.
- [Rizk and Sayed, 2021] Elsa Rizk and Ali H Sayed. A graph federated architecture with privacy preserving learning. *arXiv preprint arXiv:2104.13215*, 2021.
- [Scardapane *et al.*, 2020] Simone Scardapane, Indro Spinelli, and Paolo Di Lorenzo. Distributed training of graph convolutional networks. *IEEE Transactions on Signal and Information Processing over Networks*, 7:87–100, 2020.
- [Shan *et al.*, 2021] Chuanqiang Shan, Huiyun Jiao, and Jie Fu. Towards representation identical privacy-preserving graph neural network via split learning. *arXiv preprint arXiv:2107.05917*, 2021.
- [Tan *et al.*, 2021] Alysa Ziyang Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *arXiv preprint arXiv:2103.00710*, 2021.
- [Wang *et al.*, 2020] 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*, 2020.
- [Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.
- [Wu *et al.*, 2021] 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.
- [Xie *et al.*, 2021] Han Xie, Jing Ma, Li Xiong, and Carl Yang. Federated graph classification over non-iid graphs. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS'21)*, 2021.
- [Xing *et al.*, 2021] Pengwei Xing, Songtao Lu, Lingfei Wu, and Han Yu. BiG-Fed: Bilevel optimization enhanced graph-aided federated learning. *FL-ICML*, 2021.
- [Xu *et al.*, 2018] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.
- [Ying *et al.*, 2018] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'18)*, pages 974–983, 2018.
- [Yuan *et al.*, 2020] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *arXiv preprint arXiv:2012.15445*, 2020.
- [Zhang *et al.*, 2021a] Chenhan Zhang, Shuyu Zhang, JQ James, and Shui Yu. Fastgnn: A topological information protected federated learning approach for traffic speed forecasting. *IEEE Transactions on Industrial Informatics*, 2021.
- [Zhang *et al.*, 2021b] Huanding Zhang, Tao Shen, Fei Wu, Mingyang Yin, Hongxia Yang, and Chao Wu. Federated graph learning—a position paper. *arXiv preprint arXiv:2105.11099*, 2021.
- [Zhang *et al.*, 2021c] Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. Subgraph federated learning with missing neighbor generation. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS'21)*, 2021.
- [Zheng *et al.*, 2021] Longfei Zheng, Jun Zhou, Chaochao Chen, Bingzhe Wu, Li Wang, and Benyu Zhang. Asfgnn: Automated separated-federated graph neural network. *Peer-to-Peer Networking and Applications*, 14(3):1692–1704, 2021.
- [Zhou *et al.*, 2020] Jun Zhou, Chaochao Chen, Longfei Zheng, Huiwen Wu, Jia Wu, Xiaolin Zheng, Bingzhe Wu, Ziqi Liu, and Li Wang. Vertically federated graph neural network for privacy-preserving node classification. *arXiv preprint arXiv:2005.11903*, 2020.
- [Zhu *et al.*, 2021] Wei Zhu, Andrew White, and Jiebo Luo. Federated learning of molecular properties in a heterogeneous setting. *arXiv preprint arXiv:2109.07258*, 2021.