Federated Graph Learning

Nhu Ngoc Hoang
Barcelona School of Informatics
Universitat Politècnica de Catalunya
Barcelona, Spain
nhu.ngoc.hoang@estudiantat.upc.edu

I. Introduction

In recent years, graph neural networks have emerged as a powerful paradigm for learning from graph-structured data, achieving state-of-the-art results across many domains such as biomedical and healthcare, recommendation, social networks, finance, etc. Graph neural networks typically require centralized access to large graph datasets, which is becoming increasingly infeasible due to privacy regulations such as the European Union's General Data Protection Regulation [1], data ownership concerns and policies, as well as the sheer volume of data distributed across multiple organizations. Federated graph learning addresses this challenging by combining graph neural networks with federated learning, another powerful paradigm that enables collaborative training of machine learning models without sharing raw data, upholding data privacy and ownership.

Unlike federated learning for tabular, image, or textual data, the graph-structured nature of the domain introduces multiple novel challenges, including:

- **Data and structural heterogeneity**: Local subgraphs vary greatly in terms of feature distributions, graph topology, degree distribution, and homophily, causing standard aggregation methods in traditional federated learning to degrade global convergence.
- **Missing cross-client connectivity**: One of the core mechanisms of graph neural networks is the message passing among connected nodes. In federated graph learning, message propagation is truncated at client boundaries, leading to biased or under-informed node representations.
- **Graph-level privacy risks**: Beyond node attributes, edge existence and subgraph membership may reveal sensitive information, further necessitating privacy-preserving techniques.

This paper presents a state-of-the-art review of federated graph learning by categorizing and analyzing existing body of literature across key dimensions corresponding to the aforementioned core challenges. Domain applications and open research directions are also discussed, highlighting both progress and gaps pointing to potential future development in this high-impact area.

The rest of this paper is organized as follows. Section II provides a general formulation of the domain, setting up contexts for federated learning, graph neural networks, as well as federated graph learning. Due to the variances in graph models and data partitioning methods, federated graph learning problems can be categorized according to multiple dimensions. Section III addresses this multi-dimensional nature of taxonomy by discussing some of the major categorizations of federated graph learning. Section IV presents the three core challenges of federated graph learning while Section V discusses more in-depth the various approaches and architectures that have been developed to tackle the core challenges. Section VI discusses some of the areas in which federated graph learning has been utilized with promising results, while Section VII presents the remaining open challenges and directions for future development. Finally, Section VIII concludes the paper.

II. Problem formulation

A. Federated learning

Federated learning, first coined in 2017 [2], is a machine learning paradigm where multiple clients collaboratively train a model without exchanging raw data, under the orchestration of a central coordinating server. Clients participating in federated learning can be IoT devices, personal devices, or private servers of different organizations. The collaborative and decentralized nature of federated learning facilitates compliance with data privacy regulations and policies.

[2] formulates federated learning as including multiple rounds of updates conducted by a set of M clients $\mathcal{C} = \{c_1, c_2, ..., c_M\}$ where each client c_k owns a private dataset $\mathcal{P}_k = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_{N_k}, y_{N_k})\}$ where N_k is the number of samples in this dataset. The total number of data samples across all clients is then defined as $N = \sum_{k=1}^{M} n_k$.

Similar to the traditional machine learning configuration, the goal of federated learning is to optimize some global objective function. Let f_k denote the loss function on a client c_k parameterized by w and F_k the average loss over the entire private dataset of c_k. Federated learning can then be formulated as the optimization of the loss function across all clients while keeping the data private to each client:

$$\min_w \sum_{k=1}^{M} \frac{N_k}{N} F_k(w) = \min_w \frac{1}{N} \sum_{k=1}^{M} \sum_{i=1}^{N_k} f_k(\mathbf{x}_i, y_i; w)$$

Federated learning involves multiple rounds of updates during which model parameters are transmitted between the central server and clients. At round r, the server selects a subset of clients among $\mathcal{C}$ to participate in that round.

Then, the server sends to each selected client the current parameters of the global model w^r. Client c_k then conducts local computations to update its set of parameters w_k^r using its private dataset $\mathcal{P}_k$ via some methods such as stochastic gradient descent [3]. The updated parameters from the clients are communicated back to the server and used to obtain the updated global parameters w^{r+1} which will be used for the next update round.

B. Graph neural networks

Graphs have increasingly gained traction as a data format suitable for modeling complex data across many real-world domains such as bioinformatics, healthcare, transportation, recommendation systems, etc. [3]. Graph neural networks have naturally emerged as powerful models for handling graph data [4]. Graph neural networks are a class of deep learning model designed to encode both structural and feature-based information in graphs via neighborhood propagation. Their ability to encode such rich contextual and topological information leads to high performance in tasks on the graph level (e.g. graph property prediction), node level (e.g. node classification), or edge level (e.g. missing edge prediction).

Graph neural networks produce embeddings of graph attributes (node embeddings, edge embeddings, etc.) given a graph and its associated node features as inputs. A graph $\mathcal{G} = (V, E)$ consists of a set of nodes (vertices) V and a set of edges E. The node features can be represented as a matrix $\mathbf{X} \in \mathbb{R}^{|V| \times d_x}$ where d_x denotes the dimension of the feature vector associated with a node. Graph neural networks update the embedding of a given node through two main steps:

1) Aggregating information from its neighbors: $a_v^{(l)} = \text{AGGREGATE}^{(l)}(\{h_u^{(l-1)} | u \in \mathcal{N}(v)\})$ where $h_v^{(l)}$ is the representation of node v at layer l; $h_v^{(0)} = \mathbf{X}_v$ (the initial raw features of node v) and $\mathcal{N}(v)$ gives the set of neighbors of node v. There are multiple options for the AGGREGATE function such as mean, weighted average, min/max pooling.
2) Updating the node's representation: $h_v^{(l)} = \text{UPDATE}^{(l)}(h_v^{(l-1)}, a_v^{(l)})$

Many variants and models of graph neural networks have been developed over the year, with some of the most prominent being graph convolutional networks [5], graph attention networks [6], and GraphSAGE [7].

C. Federated graph learning

Combining graph neural networks with the federated learning paradigm, federated graph learning facilitates the training of graph neural networks across multiple data-owning clients without sharing raw graphs, ensuring decentralization and privacy of data. Federated graph learning has been utilized on a wide range of graph types (e.g. property graphs, knowledge graphs, user-item graphs) to bring about new approaches in many domains, including bioinformatics and drug discovery, social network classification, or recommendation systems.

III. Taxonomy

A. Data partitioning taxonomy

[8] puts forth a categorization of federated graph learning from the perspective of the distribution of graph data among clients in a federated learning settings: **inter-graph federated learning**, **intra-graph federated learning**, and **graph-structured federated learning**.

1) Inter-graph federated learning: In this category, the decentralized data belonging to individual clients are graph data, where each client owns a set of whole graphs. The field where inter-graph federated learning is most common is the biomedical field for tasks such as molecular property prediction. In this example, a graph can be constructed to represent a molecule with a set of atoms (nodes) and a set of chemical bonds (edges), a client can be a pharmaceutical company which can owns multiple confidential molecule structures. Federated graph learning can boost cross-company collaboration while still maintaining each company's privacy to their own data.

2) Intra-graph federated learning: In this settings, it is assumed that there exists a global graph and each client owns a part of it. Intra-graph federated learning is further divided into horizontal and vertical categories based on the specific nature of the partitioning, which will be discussed shortly. Under this settings, it can also be assumed that there exist some adjacent nodes in the global graph that are distributed to different clients in the system, and the edges connecting them are subsequently lost. This leads to the removal of potentially critical cross-client information, which will be further discussed in Sections IV and VI.

3) Graph-structured federated learning: While the previous two types assume some graph-structured nature on the data level, this category assumes the graph-structured nature of the system topology, that is, participating clients are considered nodes and the relationships among them are considered edges comprising a global graph. On the data level, the clients can own graph or non-graph data. In this settings, the server utilizes graph neural networks for the aggregation of updates conducted at the local models of clients.

B. Intra-graph taxonomy

The the intra-graph federated learning settings, where some global graph is assumed to be distributed among the clients, can be further divided into the horizontal and vertical settings.

1) Horizontal federated graph learning: Under this settings, subgraphs owned by different clients overlap in the feature space but differ in the ID space. In other words, nodes belonging to different clients have the same set of features but the nodes themselves are distinct from client to client. An example of this can be a scenario in which a university has a unified graph schema for storing student information, but the data is distributed among different campuses where each campus owns only the data of students enrolled at that campus.

2) Vertical federated graph learning: On the contrary, under this settings, subgraphs of different clients overlap in the ID space but differ in the feature space. In other words, clients hold disjoint features for heavily overlapping nodes. An example scenario is one where patients goes to different hospitals for different checkups, so the same patients can exist in multiple subgraphs with different hospital-specific features.

Figure 1 visualizes the data partitioning taxonomy with the intra-graph federated learning configuration expanded into horizontal and vertical federated graph learning. In the horizontal intra-graph settings, dashed edges connecting nodes from different clients represent cross-client latent edges that are missing when data is distrbuted to individual clients. In the vertical intra-graph settings, dashed edges connect nodes with overlapping IDs that belong to different clients.

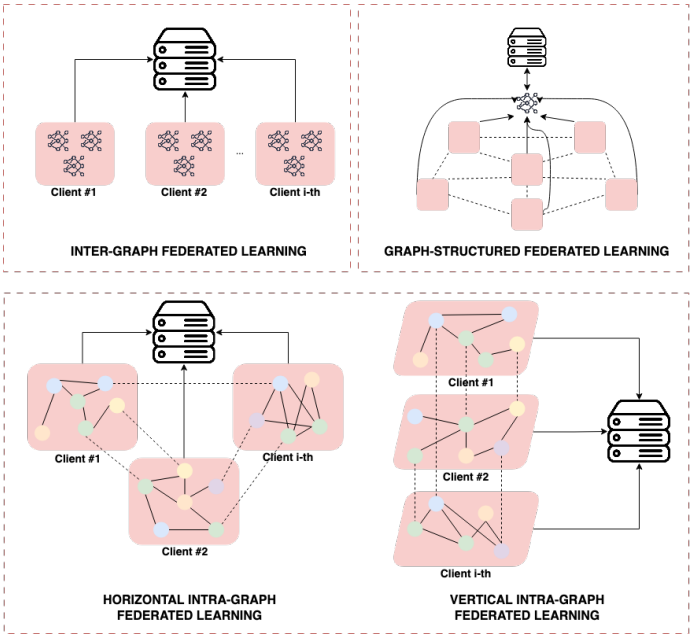

Fig. 1. Data partitioning taxonomy and intra-graph taxonomy

C. Centralization taxonomy

[4] puts forth an additional categorization based on the communication patterns of entities participating in the training: centralized and decentralized federated graph learning. The centralized settings includes the existence of a central server which coordinates the training among the clients, while the decentralized settings assumes client-to-client communication for joining training the model without the intervention of a server.

IV. Core challenges

This section discusses the core challenges in federated graph learning within the scope of the intra-graph configuration, where clients own varying subgraphs of a presumed global graph. If some approach targets the graph-structured federated learning settings, it will explicitly be said so.

A. Data heterogeneity

A core challenge of federated learning in general arising from the intra-graph settings is the unbalanced and non-IID (identically and independently distributed) partitioning of data across the clients [9]. The heterogeneous and imbalanced nature of data in federated learning may manifest in many forms of deviation, including in terms of imbalanced features, labels, volumes, or other attributes of data samples. For federated graph learning, the heterogeneity originates from not only the data (feature heterogeneity), but also the structural information of subgraphs owned by different clients. Subgraphs differ not only in size but also in topology e.g. node degrees, clustering coefficients, connectedness, and homophily levels which vary markedly across clients.

The two sources of heterogeneity together lead to greatly divergent gradients obtained through the local learning process of different clients and overall degrade the global model [3]. The feature heterogeneity complicates the process of learning the message passing functions in the clients (which are essential for the update mechanism of graph neural networks, as previously discussed) while the structure heterogeneity further hampers the emergence of coherent global graph representations [10]. As a result, standard federated learning mechanism fails to converge robustly in the graph settings, necessitating specialized mechanisms to align both feature and structural learning across clients.

B. Missing cross-client information

As clients own subgraphs of the global graph, a common scenario that can arise is one where some nodes in one client are connected via direct edges with other nodes belonging to a different client. Due to the data privacy requisite underpinning federated learning, the learning process at a client can only aggregate information from a node's neighbors that are located at the same client, excluding cross-client edges. Consequently, node representations are computed from an incomplete neighborhood, introducing biased and suboptimal embeddings that degrade downstream performance [3]. Specifically, the missing neighbor information issue can manifest in various forms of degraded performances, such as impeded message passing for graph neural networks, truncated degree distribution in local subgraphs, or amplified biases for minority label class. This challenge draws a distinction between federated graph learning and the conventional federated learning settings with non-graph data.

C. Privacy and structural confidentiality

One of the fundamental values of federated learning is privacy on the basis of data minimization and decentralization. Participating clients own confidential raw datasets that are not transmitted or shared during the entirety of the training process, rather, only model updates (gradients) are communicated to the server. Additionally, the gradients are only kept provisionally by the server for incremental updates to the global model [9]. Such principles provide the baseline privacy guarantee for clients participating in federated learning compared

to, for example, centralized training. However, without further formal privacy-preserving techniques and mechanisms beyond this baseline, federated learning systems remain vulnerable to sophisticated privacy leakages and attacks, namely inference attacks, reconstruction attacks, inversion attacks, among others [11].

Federated graph learning introduces unique privacy risks due to the combination of graph-structured data vulnerabilities and distributed training dynamics. The following primary threat models emerge from the interaction between clients and servers.

1) Server-side threats:

- Honest-but-curious servers: This category represents a seemingly non-intrusive group of threats originating from passively observing servers which, while adhering to the federated learning protocols [9], have the capacity to analyze aggregated gradients to infer private graph structures or node attributes.
- Malicious servers: This category represents a stronger threat model with servers acting as active adversaries that tamper with the training process by modifying model parameters to expose client-specific subgraphs or inject backdoors.

2) Client-side threats:

- Gradient inversion attacks: Adversaries seek to reconstruct local graph data from shared model gradients.
- Membership inference: Adversaries attempt to determine if a data sample participated in the training process based on analysis of model output. The inference attempts can be made on various components of a graph including nodes, edges, attributes, or the whole graph [12].

V. Technical approaches and architectures

A. Data heterogeneity

Many approaches have been proposed to tackle the issue of non-IID data partitioning in federated graph learning focusing on developing personalized models that are better adapted to the clients, ensuring better performance than having a single global model [3]. This section discusses some of the most prominent methods, dividing them into two groups: **clustering-based aggregation** and **global alignment-based calibration**. The underlying assumptions are different for the two groups: clustering requires some notion of similarity, while alignment methods assume global knowledge is available for calibration.

1) Clustering-based aggregation: Instead of training separate models on individual clients with potentially highly divergent gradients, **GCFL+** [10] is proposed with the main idea of identifying clusters of similar clients in terms of graph structures and node features. The core mechanism of GCFL+ is the dynamic clustering of clients based on local computed gradients with a view to maximizing the collaboration of homogeneous clients. Specifically, the notion of similarity among clients is built upon the gradient series obtained from multiple rounds of federated training. The coordinating server maintains a time-series matrix Q where each row is the gradient series of a given client. Upon new gradients being sent to the server after a round of training, Q is updated using the norms of the new gradients. A distance matrix β is then built to record the distances between pairs of gradient series, computed using dynamic time warping [13]. The distance matrix is used to construct fully connected graph where the clients are nodes and the distances are edge weights, and the Stoer-Wagner algorithm [14] is applied to bi-partition the graphs, effectively obtaining the clusters. The gradients are aggregated by the server cluster-wise, instead of across all participating clients.

While GCFL+ falls into the inter-graph category, **FedCG** [15] is proposed as a cluster-driven method designed to mitigate heterogeneity in graph-structured federated learning. Rather than assuming a single global model, FedCG first identifies latent domains through unsupervised clustering, performed locally at clients using a teacher-student domain classifier. Each domain is assigned its own set of domain-specific parameters, and clients are soft-assigned to one or more domains. To enable cross-domain knowledge sharing, FedCG builds a graph where nodes correspond to domains, and edges encode similarity between domain-specific parameters. A graph convolutional network [5] is then used to update these domain parameters jointly, allowing information to flow across clusters during training. This architecture effectively combines clustering and representation smoothing to reduce the impact of statistical heterogeneity.

2) Global alignment-based calibration: Recent works emphasize calibrating graph structural bias, employing techniques such as graph structure distillation to align local adjacency statistics with global ones. The following methods operate by aligning local models to globally shared structures, which can be embeddings, structural proxies, or parameter decompositions.

FGSSL [16] decouples heterogeneity into node-level semantic bias and graph-level structural bias and proposes two complementary mechanisms to tackle them. On the node-level, local models are inclined to learn biased node representations due to limited class coverage or skewed distribution. FGSSL uses supervised contrastive learning to align local node embeddings with class-consistent global representations, effectively countering semantic drift caused by non-IID class distributions across clients. On the graph level, local graphs may be structurally incomplete and contain misleading neighborhood signals. To tackle this, FGSSL aligns similarity distributions (rather than raw structure) of node neighborhoods from global and local models. In other words, this approach utilizes the aggregated neighborhood information available on the global level to calibrate the local biased graph structure.

FedSpray [17] tackles the data heterogeneity in label distribution where nodes of the minority class in a client may aggregate adverse neighborhood information. Instead of relying on potentially incomplete or biased local neighborhoods, FedSpray introduces class-wise structure proxies, which are representative embeddings capturing class-specific structural information aggregated from all clients. These proxies act as

soft references, offering clients an external, unbiased sense of how nodes in each class are typically structured. During local training, each client's personalized model is regularized by aligning its predictions with the soft labels derived from these structure proxies (obtained through a feature-structure encoder taking the structure proxies and raw node features as inputs), guiding it to conform to global relational patterns while maintaining local adaptability.

FedHGN [18] tackles heterogeneity in federated learning of heterogeneous graph neural networks (HGNNs) by decoupling schema-specific and schema-agnostic knowledge, enabling aligned model updates without revealing private graph schemas. In real-world federated HGNN settings, clients oftentimes have distinct node and edge type definitions i.e. different graph schemas due to varied data construction and storage practices (for example, the same entity type may be called "paper" in one client and "article" in another), leading to schema mismatches and non-overlapping feature space. Through schema-weight decoupling, each client can factorize its HGNN parameters into basis weights that are shareable with other clients and confidential coefficients, thus preserving schema privacy while also enabling cross-client learning.

B. Missing cross-client information

To tackle the missing cross-client information issue, multiple approaches have been proposed to create augmented subgraphs through constructing the missing edges, ensuring higher quality of node representations and, to some extent, mitigating the non-IID data partitioning problem [4]. This section highlights some representative approaches with a tentative division into three categories: **local missing neighbors augmentation**, **cross-client graph extension**, and **knowledge graph embedding-based alignment**.

1) Local missing neighbors augmentation: The methods in this category address the challenge by directly generating or imputing plausible missing neighbors within each client's subgraph, either before or during the graph neural network training. These techniques enhance model expressiveness and mitigate cross-client information loss without requiring direct data sharing.

FedSage+ [19] proposes the addition of a missing neighbor generator on each client. From the original graph, each client creates an impaired graph by randomly holding out a number of existing nodes and related edges. The Gaussian-based generator then learns first to predict the number of missing neighbors of a given node, then to reconstruct the features of those missing neighbors. After augmenting the subgraph, the client continues regular GraphSage [7] training and shares updated model weights. While a foundational model, FedSage+ has a number of limitations such as the use of a simple node generation mechanism with random masking. Such drawbacks are addressed in later approaches building upon FedSage+.

FedNI [20] refines neighbor augmentation by moving from simple reconstruction to inpainting utilizing generative adversarial networks (GANs) [21], specifically spectral normalized GAN [22]. Instead of random masking, FedNI removes nodes along breadth-first search paths to simulate realistic missing neighborhoods. After creating the impaired graph with masked nodes, a graph convolutional network [5] is used to obtain embeddings of the remaining nodes, which are then used as inputs to a multilayer perceptron (MLP) to predict the number of missing neighbors. At the same time, another MLP decoder uses the node embeddings to generate realistic features for the missing neighbors. An discriminator is implemented to guide the node feature generator, ensuring that the generated features as closely resemble the true masked neighbors as possible. As FedNI was implemented for the domain of medical imaging, to complement the complex nature of the disease population network, an additional predictor is used to obtain phenotypic information (i.e. human observable traits such as gender and height), which is further used to predict the weights of edges connecting generated missing nodes and known nodes.

FedDEP [23] advances upon the basis of FedSage+ by generating multi-hop neighbors in embedding space, rather than feature space, allowing richer context reconstruction. Node embeddings are generated using a graph neural network encoder in combination with a Bernoulli sampling mechanism to help predicting representations of missing neighbors. It further introduces prototype-based pseudo-federated learning to reduce communication overhead and ensures privacy with noise-free edge-level differential privacy. These enhancements offer deeper, more efficient, and privacy-aware augmentation, proving more robust than prior single-hop, feature-level generators.

2) Cross-client graph extension: Rather than relying solely on local augmentation, cross-client graph extension methods enable clients to supplement their subgraphs with structural information obtained from other participants while preserving privacy. These approaches facilitate cross-client collaboration by extending graphs or sharing global structure insights in a controlled, privacy-aware manner, effectively improving representation quality under heterogeneity. This section discusses a number of representative approaches with regards to their mechanisms for mending the local subgraphs, while the privacy-preserving techniques employed will be discussed in the next section.

DP-FedRec [24] enhances local graphs by privately expanding connectivity using cross-client edges. DP-FedRec explores the domain of recommendation systems where each client owns a bipartite graph representing the purchasing interactions among users and items (nodes) and the users' ratings of items (edge weights). The goal of cross-client graph extension is to privately expand the subgraph with edges related to overlapping users between different clients. Specifically, clients perform Private Set Intersection (PSI) to identify shared users across subgraphs and selectively extend their graphs with k-hop neighborhoods around these intersected users. The shared edge information complements each client's sparse view of the global user-item graph, enabling a more accurate aggregation of structural signals across clients.

FedGL [25] adopts a different strategy by generating a global pseudo-graph at the server level through self-supervised

learning. Each client uploads local node embeddings and prediction outputs, which the server uses to infer pseudo-edges and pseudo-labels, effectively constructing a synthetic global structure that reflects inter-client relationships without needing raw data. FedGL relies on distilled global information to indirectly guide clients toward a more coherent, less biased understanding of the global graph. This approach is particularly effective in semi-supervised or low-label settings, where the pseudo-labels and connections serve to supplement both structural and semantic gaps in local data.

FedGNN [26] and **FedPerGNN** [27] address the challenge of missing user-item interactions by enabling clients to recover high-order collaborative signals via a secure embedding-sharing mechanism. Both of these approaches involve a third-party server which handles graph expansion for clients. The graph expansion protocol involves the following steps: (1) the coordinating server issues a public key to all clients, (2) each client encrypts its item IDs and uploads them along with its user embedding to the third-party server, (3) the server performs ciphertext matching to find overlapping items and constructs anonymous high-order neighborhoods, and finally (4) it returns the embeddings of these neighbors to the original client to further facilitate local model training.

3) Knowledge graph embedding-based alignment: In federated settings involving knowledge graphs (KGs), clients typically hold entity-relation triples that are locally complete but globally fragmented, and often partially overlapping. To enable learning meaningful global representations while preserving structural semantics and client privacy, knowledge graph embedding (KGE)-based methods align and aggregate embedding spaces across clients. These methods focus on coordinating entity and relation embeddings under varying degrees of overlap and sensitive structure, often using specialized alignment architectures or privacy-preserving protocols.

FKGE [28] tackles the problem of embedding alignment across distributed KGs by proposing a privacy-preserving adversarial translation (PPAT) framework, built upon a revised GAN architecture. It focuses on clients that share overlapping entities and relations, and learns to translate embeddings from one KG to another across these alignments. For each pair of aligned KGs G_k and G_s, the generator takes an entity embedding from G_k and maps it into the embedding space of G_s. A student discriminator and multiple teacher discriminators, each trained on local KGs, work together to distinguish synthesized from real embeddings in G_s. This adversarial mechanism forces the generator to produce high-quality translations, ensuring consistent semantics across KGs. FKGE incorporates differential privacy mechanisms in the discriminator training to prevent inference on sensitive entities or relations. The resulting system allows federated KGs to collaboratively refine embedding quality without revealing raw triples or embedding parameters.

FedE [29] focuses on aligning entity embeddings across clients by introducing a shared entity table at the server that records all unique entity embeddings received from clients. During training, each client maintains local KG embeddings, which are aligned to this global entity table. The server follows the standard federated learning protocol to update global entity embeddings based on local updates from participating clients. After each round, the updated embeddings are redistributed to clients, enabling consistency across learning. To enhance the quality of the entity representations, FedE also incorporates self-training with a contrastive learning loss, encouraging each client to refine its embeddings by distinguishing between positive and negative entity-relation pairs.

FedR [30] addresses privacy risks in federated KGE by proposing to aggregate relation embeddings instead of entity embeddings, based on the observation that relation embeddings carry less identifying information. Clients locally train KG models and then upload their relation embeddings to the server. To preserve privacy, these embeddings are protected with Secure Aggregation, and the server constructs a relation table using Private Set Union (PSU) to identify and align shared relations. Since relation embeddings cannot be easily used to reconstruct entity-level structure, FedR provides stronger privacy guarantees than entity-sharing approaches. This design makes FedR particularly suitable for sensitive domains where disclosing even partial entity structure could lead to information leakage.

C. Privacy and structural confidentiality

This section discusses some of the most prominent privacy-preserving techniques and mechanisms aiming to combat the vulnerabilities presented in Section IV. Additionally, many of the models that have been discussed up to this point will also be re-evaluated under the privacy-preserving lens. In this discussion, the privacy-preserving techniques are grouped into three categories: **perturbation**, **encryption**, and **obfuscation**.

1) Perturbation-based methods: Perturbation techniques in federated graph learning rely on the addition of randomized noise or data mixing to disguise sensitive information in model updates, providing a layer of protection against structural and attribute inference attacks without requiring heavy cryptographic overhead. In the graph learning settings, perturbation targets not only node features but also topology and neighborhood structure, which are increasingly vulnerable in federated settings. The most commonly used specific techniques within this category are local differential privacy (LDP) and global differential privacy (DP) which introduce carefully calibrated noise to model gradients or data, reducing the ability of adversaries to reverse-engineer sensitive relationships from updates.

FedNI [20] train two models under a federated settings: the missing node generator and the graph convolutional network for node classifier. Differential privacy is applied, with Gaussian noise of mean 0 and standard deviation 0.01 being added to the parameters of both models before they are sent to the coordinating server. FedDEP [23] claims guaranteed noise-free differential privacy on the edge level. This privacy guarantee is attributed to the two phases of random sampling, the first one being the random neighborhood sampling at each layer of the graph convolutional network, and the second one

being the Bernoulli sampling mechanism in the neighbor embedding generation process. DP-FedRec [24] first guarantees the privacy of nodes that are not in the intersected user set through the usage of Private Set Intersection. For the nodes that are in the intersected set, Laplacian noise is added to both the adjacency matrix and the edge weights with a view to preserving privacy on both topology and edge information. FedGNN [26] and FedPerGNN [27] apply local differential privacy to locally computed gradients by adding zero-mean Laplacian noise after clipping gradients based on a threshold of L1-norm. FKGE [28] provides differential privacy guarantee for generated gradients via the implementation of PATE-GAN [31] in the student-teachers discriminators.

Perturbation-based approaches, particularly those leveraging local differential privacy, offer practical defenses against privacy attacks by adding calibrated noise to graph data or model updates. As a trade-off between privacy and utility, however, the intensity of noise directly impacts model quality. As demonstrated in studies of general federated learning with local differential privacy (e.g. [32]), reducing the privacy parameter ϵ (i.e. increasing privacy) typically incurs a notable drop in utility, converging more slowly and achieving lower accuracy at stringent privacy levels. Therefore, a balance must be established between the strength of privacy guarantee and the acceptable utility decrease to preserve both accuracy and structural confidentiality in node embeddings and other downstream tasks.

2) Encryption-based methods: Encryption-based methods in federated graph learning utilize cryptographic schemes to securely process sensitive graph elements during collaborative learning. Privacy-preserving techniques within this category can be further divided into homomorphic encryption-based methods and secure multiparty computation-based methods.

Homomorphic encryption [33] allows arithmetic operations (addition or multiplication) to be performed directly on ciphertexts. This ensures that sensitive embeddings or adjacency vectors remain encrypted during transit and computation. FedGNN [26] and FedPerGNN [27] are two representative models that utilize homomorphic encryption as a privacy-preserving mechanism. These approaches follow a protocol with three steps:

1) Public key distribution: The coordinating (learning) server issues a public key to all participating clients
2) Client-side encryption: Each client encrypts the sensitive graph components (which, in the case of FedGNN and FedPerGNN, are the private item IDs) and send to a third-party server
3) Encrypted matching: The third-party server performs matching based on ciphertexts sent by clients in order to return anonymized structural information (neighbor embeddings in the case of FedGNN and FedPerGNN) without decrypting data

While homomorphic encryption ensures strong confidentiality, it incurs significant computational and communication costs and typically requires trusted key infrastructures. FedGNN and FedPerGNN both operate under the assumption that the third-party server handling graph expansion is trustworthy and cannot infer confidential information from encrypted item IDs. However, if this server colludes with the coordinating server by sharing the private key and the global item table, the privacy guarantee will be broken and private user information is at risk of being leaked.

Secure multiparty computation enables multiple parties to evaluate a function over their inputs without revealing them, often using techniques like secret sharing, garbled circuits, or set operations. DP-FedRec [24] utilizes Private Set Intersection [34] which allows two clients to obtain the set of shared users while preserving the privacy of the users that do not belong to the intersection set. FedR [30] applies Private Set Union [35] on the relations received from clients in order to maintain a global relation table without revealing relation ownership of specific clients.

In graph-based settings, secure multiparty computation enables secure computation of unions or matches of subgraph elements (e.g. nodes, edges, neighborhoods) without disclosing client-specific data. When paired with encryption and differential privacy (e.g. in the case of DP-FedRec, differential privacy is also used), such methods offer strong aggregate privacy with manageable overhead.

3) Obfuscation-based methods: Obfuscation strategies aim to conceal sensitive structural or interaction information through selective data sampling or aggregation procedures. Unlike perturbation methods which rely on noise addition or encryption-based methods which use cryptographic primitives, obfuscation mechanisms achieve privacy by hiding the presence or identity of sensitive graph elements. They are highly effective in hiding which nodes or edges are present in a user's subgraph without necessarily altering raw data distributions. Some prominent techniques include pseudo-node sampling (clients introduce synthetic elements e.g. pseudo-interacted items indistinguishable from real ones), ego-graph hybridization (combining real neighborhoods with decoy structures), or structure masking (a client mixes true edge connections with random candidate links to obscure relationship identities). Such strategies obscure structural features while still enabling meaningful graph neural network training, and are particularly salient in recommendation systems or social graph scenarios, where revealing linkages is most sensitive.

FedGNN [26] and FedPerGNN [27] employ pseudo-interacted item sampling as an obfuscation method. In the user-item graph settings of these systems, the model update gradients can inadvertently reveal sensitive user history information, since for a training round, only the items that a user has interacted with have non-zero embedding gradients, allowing the coordinating server to infer the interaction history of a user. To uphold privacy, FedGNN and FedPerGNN randomly sample an additional set of non-interacted items and generate their embeddings using a Gaussian distribution mimicking the mean and co-variance of the embeddings of real items. While this approach provides good privacy guarantee, the impact on performance is also to be taken into consideration.

Empirical results from [27] show that performance peaks when no pseudo-interacted items are added and declines as sampling is conducted since the generated gradients impact the accuracy of item gradients. However, it was also observed that a larger number of pseudo-interacted items leads to improvement since their gradients are better counteracted after aggregation.

D. Summary

Table I presents a summary of the approaches that have been discussed thus far, including the core challenge addressed, the categorization of the approach, the privacy-preserving mechanisms employed, as well as some additional information on the data type and corresponding downstream task.

VI. Applications

Federated learning and federated learning on graph-structured data is emerging across diverse domains. This section discusses some of the most promising applications of federated graph learning in fields such as healthcare, recommendation systems, computer vision, and finance.

A. Biomedical and healthcare

Biomedical data are often represented as graphs (e.g. protein-protein interactions, patient similarity networks, molecular structures). Such datasets, especially those involving patients, are also extremely sensitive and private, necessitating the isolated ownership of data by different hospitals, medical institutes, and research centers [3]. On the other hand, scientific research can benefit immensely from an extended, collaborative network utilizing cross-institute data. Therefore, biomedical research and healthcare becomes a prominent field for utilizing federated graph learning.

Federated patient graphs (with privacy constraints) are being studied for disease prediction and classification tasks utilizing networks of patients data and electronic health records (EHRs). [36] federates across hospitals using fMRI spatio-temporal graphs where client devices process each patient's temporal brain signal and share embeddings to a central graph convolutional network to predict diseases. [20] also explores the task of disease prediction using cross-institutional population graph, with experiments conducted on neuroimaging data combined with phenotypic information. [37] proposed a federated graph learning settings using Protein-Protein Interaction networks to represent unique patients for disease classification. [38] tackles graph classification using temporal-spatial medical data, specifically polysomnography recordings with different channels being formulated as nodes. [39] explores the model-agnostic meta-learning paradigm for training multi-task deep learning models on federated EHRs. [40] utilizes federated graph learning to improve local connectional brain template representations by integrating cross-institutional multi-view brain connectomic datasets. Drug discovery also immensely benefits from federated graph learning, where multiple hospitals or pharmaceutical companies could jointly train graph neural networks on on molecular graphs without sharing proprietary compounds. [41] integrates GAN and graph neural networks in a federated manner to generate highly novel and diverse molecular graphs. [42] jointly trains a graph neural network for molecular property prediction utilizing molecular graphs from multiple research labs.

B. Recommendation systems

Many recommendation models include multiple collaborating clients possessing user-item bipartite graphs. Federated graph learning allows a natural framework for multiple vendors to collaboratively improve recommendation quality. As previously discussed, DP-FedRec [24], FedGNN [26], and FedPerGNN [27] explore graph expansion strategies for improving recommendations using user-item subgraphs. [43] complements user-item interaction data with social links to propose a recommendation system for social recommendation and personalization. [44] proposes a vertical federated graph neural network-based recommender system which allows joint training when different parties hold disjoint features e.g. one company has user graph, another has item graph. [45] explores cross-domain recommendation allowing federated learning across e-commerce domains (e.g. different product categories) to transfer "positive knowledge" (embeddings) between domains while filtering negative transfer.

C. Computer vision

Numerous applications in computer vision can be classified as graph-structured federated learning where the clients (or domains of data) are constructed as nodes with edges representing semantic connections among them [3], with [15] being an example. In addition to image classification, applications in video-based trajectory prediction are also emerging. [46] proposes learning representations of objects from graph sequences which represent inter-object relationships from video frames.

D. Finance

Graph-structured financial data (transaction networks, user-vendor bipartite graphs, account graphs, etc.) often reside across institutions. Federated graph learning enables banks or retailers to collaboratively detect fraud without revealing raw transactions. [47] explores the applicability of federated graph learning in detecting money laundering activities using a transaction graph representing money transfers among different banks and a party relationship graph representing social relationships of customers. [48] leverages graph-structured data of digital currency transactions to identify malicious transactions where the federated learning paradigm helps to uphold user privacy.

VII. Open challenges and future directions

This section presents some challenges in federated graph learning that are open to improvement as well as some subsequent future research directions.

- **Scalability**: Real-world graphs can have millions of nodes, while existing literature usually rely on graphs of smaller scale to simulate the federated settings. For large-scale realistic graphs, increased communication overhead

TABLE I
SUMMARY OF DISCUSSED APPROACHES

Approach	Challenge addressed	Approach categorization	Privacy-preserving mechanism			Client data	Downstream task
			Perturbation	Encryption	Obfuscation		
GCFL+	Heterogeneity	Clustering-based aggregation				Multiple graphs	Graph classification
FedCG	Heterogeneity	Clustering-based aggregation				Latent domains as nodes	Image classification
FGSSL	Heterogeneity	Global alignment -based calibration				A subgraph	Node classification
FedSpray	Heterogeneity	Global alignment -based calibration				A graph	Node classification
FedHGN	Heterogeneity	Global alignment -based calibration				A subgraph	Node classification
FedSage+	Missing cross-client information	Local neighbors augmentation				A subgraph	Node classification
FedNI	Missing cross-client information	Local neighbors augmentation	✓			A subgraph	Node classification
FedDEP	Missing cross-client information	Local neighbors augmentation	✓			A subgraph	Node classification
DP-FedRec	Missing cross-client information	Cross-client graph extension	✓	✓		A user-item graph	Rating prediction
FedGL	Missing cross-client information	Cross-client graph extension				A subgraph	Node classification
FedGNN	Missing cross-client information	Cross-client graph extension	✓	✓	✓	A user-item graph	Rating prediction
FedPerGNN	Missing cross-client information	Cross-client graph extension	✓	✓	✓	A user-item graph	Rating prediction
FKGE	Missing cross-client information	KG embedding -based alignment	✓			A KG	KG completion
FedE	Missing cross-client information	KG embedding -based alignment				A KG	KG completion
FedR	Missing cross-client information	KG embedding -based alignment		✓		A KG	KG completion

and failure-prone devices might introduce significant bottlenecks. Future development is needed to explore efficient sampling (e.g. neighbor sampling on-device), communication compression, and failure recovery mechanism.

- **Communication efficiency**: This is an intertwining issue along with scalability. Graph neural network federated learning involves exchanging large messages (node embeddings, adjacency blocks, gradient matrices, etc.). Reducing communication is critical, especially for large-scale graphs distributed across clients located geographically far from one another. Potential approaches include gradient quantization, sporadic update, or decentralized gossip to avoid a bottleneck server.
- **Decentralized protocols**: Most federated graph learning schemes assume a trusted coordinating server (and for some previously discussed, an additional server for graph expansion). Fully decentralized algorithms (e.g. based on consensus or graph-based aggregation) remain largely unexplored. Designing peer-to-peer graph neural networks training that tolerates network delays and heterogeneity is an open direction.
- **Heterogeneity**: Graph heterogeneity remains a deep challenge. While clustering or calibrating can alleviate some issues, a unified theory is lacking. Future works can further explore meta-learning or graph transfer learning to adapt global models to local subgraph peculiarities.
- **Security and robustness**: Adversarial attacks on federated graph learning remain largely unexplored. While privacy-preserving mechanisms have emerged, further robustness-enhancing techniques need future research. The reliance on one or more central servers also leaves rooms for corrupted servers that might collude with adversaries to infer sensitive data and graph structures. Further fortification strategies for graph-specific components remain an open research direction.

VIII. CONCLUSION

Federated graph learning has emerged as a powerful paradigm for building high-performing, privacy-aware graph models across distributed data silos owned by different organizations. This paper synthesizes representative methods across key dimensions including feature and structural heterogeneity, cross-client structural recovery, and privacy preservation. Despite rapid progress, many challenges remain open in terms of scalability, communication efficiency, heterogeneity, among others. As interest grows in domains such as healthcare, finance, and recommendation systems, federated graph learning is well positioned to yield real-world impact, but only with thoughtful, systems-aware design and rigorous theoretical grounding.

REFERENCES

[1] P. Voigt and A. von dem Bussche, *The EU General Data Protection Regulation (GDPR)*. Springer International Publishing, 2017. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-57959-7

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[3] X. Fu, B. Zhang, Y. Dong, C. Chen, and J. Li, "Federated graph machine learning: A survey of concepts, techniques, and applications," *SIGKDD Explor. Newsl.*, vol. 24, no. 2, p. 32–47, Dec. 2022. [Online]. Available: https://doi.org/10.1145/3575637.3575644

[4] R. Liu, P. Xing, Z. Deng, A. Li, C. Guan, and H. Yu, "Federated graph neural networks: Overview, techniques, and challenges," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 3, pp. 4279–4295, 2025.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[6] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[7] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 1024–1034.

[8] H. Zhang, T. Shen, F. Wu, M. Yin, H. Yang, and C. Wu, "Federated graph learning - A position paper," *CoRR*, vol. abs/2105.11099, 2021. [Online]. Available: https://arxiv.org/abs/2105.11099

[9] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, H. Qi, D. Ramage, R. Raskar, M. Raykova, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021. [Online]. Available: http://dx.doi.org/10.1561/2200000083

[10] H. Xie, J. Ma, L. Xiong, and C. Yang, "Federated graph classification over non-iid graphs," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 18 839–18 852.

[11] L. Ge, Y. Li, H. Li, L. Tian, and Z. Wang, "A review of privacy-preserving research on federated graph neural networks," *Neurocomputing*, vol. 600, p. 128166, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231224009378

[12] Y. Zhang, Y. Zhao, Z. Li, X. Cheng, Y. Wang, O. Kotevska, P. S. Yu, and T. Derr, "A survey on privacy in graph neural networks: Attacks, preservation, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 12, pp. 7497–7515, 2024.

[13] N. L. Olsen, B. Markussen, and L. L. Raket, "Simultaneous inference for misaligned multivariate functional data," *Journal of the Royal Statistical Society Series C: Applied Statistics*, vol. 67, no. 5, pp. 1147–1176, 03 2018. [Online]. Available: https://doi.org/10.1111/rssc.12276

[14] M. Stoer and F. Wagner, "A simple min-cut algorithm," *J. ACM*, vol. 44, no. 4, p. 585–591, Jul. 1997. [Online]. Available: https://doi.org/10.1145/263867.263872

[15] D. Caldarola, M. Mancini, F. Galasso, M. Ciccone, E. Rodola, and B. Caputo, "Cluster-driven graph federated learning over multiple domains," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021, pp. 2749–2758.

[16] W. Huang, G. Wan, M. Ye, and B. Du, "Federated graph semantic and structural learning," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, ser. IJCAI '23, 2023. [Online]. Available: https://doi.org/10.24963/ijcai.2023/426

[17] X. Fu, Z. Chen, B. Zhang, C. Chen, and J. Li, "Federated graph learning with structure proxy alignment," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 827–838. [Online]. Available: https://doi.org/10.1145/3637528.3671717

[18] X. Fu and I. King, "Fedhgn: a federated framework for heterogeneous graph neural networks," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, ser. IJCAI '23, 2023. [Online]. Available: https://doi.org/10.24963/ijcai.2023/412

[19] K. Zhang, C. Yang, X. Li, L. Sun, and S. M. Yiu, "Subgraph federated learning with missing neighbor generation," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 6671–6682.

[20] L. Peng, N. Wang, N. Dvornek, X. Zhu, and X. Li, "Fedni: Federated graph learning with network inpainting for population-based disease prediction," *IEEE Transactions on Medical Imaging*, vol. 42, no. 7, pp. 2032–2043, 2023.

[21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014.

[22] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks." in *ICLR*. OpenReview.net, 2018. [Online]. Available: http://dblp.uni-trier.de/db/conf/iclr/iclr2018.htmlMiyatoKKY18

[23] K. Zhang, L. Sun, B. Ding, S. M. Yiu, and C. Yang, "Deep efficient private neighbor generation for subgraph federated learning," in *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, 2024, pp. 806–814. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611978032.92

[24] Y. Qiu, C. Huang, J. Wang, Z. Huang, and J. Xiao, "A privacy-preserving subgraph-level federated graph neural network via differential privacy," in *Knowledge Science, Engineering and Management*, G. Memmi, B. Yang, L. Kong, T. Zhang, and M. Qiu, Eds. Cham: Springer International Publishing, 2022, pp. 165–177.

[25] C. Chen, Z. Xu, W. Hu, Z. Zheng, and J. Zhang, "Fedgl: Federated graph learning framework with global self-supervision," *Information Sciences*, vol. 657, p. 119976, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002002552301561X

[26] C. Wu, F. Wu, Y. Cao, Y. Huang, and X. Xie, "Fedgnn: Federated graph neural network for privacy-preserving recommendation," *CoRR*, vol. abs/2102.04925, 2021. [Online]. Available: https://arxiv.org/abs/2102.04925

[27] C. Wu, F. Wu, L. Lyu, T. Qi, Y. Huang, and X. Xie, "A federated graph neural network framework for privacy-preserving personalization," *Nature Communications*, vol. 13, no. 1, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1038/s41467-022-30714-9

[28] H. Peng, H. Li, Y. Song, V. Zheng, and J. Li, "Differentially private federated knowledge graphs embedding," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, ser. CIKM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1416–1425. [Online]. Available: https://doi.org/10.1145/3459637.3482252

[29] M. Chen, W. Zhang, Z. Yuan, Y. Jia, and H. Chen, "Fede: Embedding knowledge graphs in federated setting," in *Proceedings of the 10th International Joint Conference on Knowledge Graphs*, ser. IJCKG '21. New York, NY, USA: Association for Computing Machinery, 2022, p. 80–88. [Online]. Available: https://doi.org/10.1145/3502223.3502233

[30] K. Zhang, Y. Wang, H. Wang, L. Huang, C. Yang, X. Chen, and L. Sun, "Efficient federated learning on knowledge graphs via privacy-preserving relation embedding aggregation," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 613–621. [Online]. Available: https://aclanthology.org/2022.findings-emnlp.43/

[31] J. Yoon, J. Jordon, and M. van der Schaar, "PATE-GAN: Generating synthetic data with differential privacy guarantees," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=S1zk9iRqF7

[32] M. Kim, O. Günlü, and R. F. Schaefer, "Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 2650–2654.

[33] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation, Academia Press*, pp. 169–179, 1978.

[34] D. Morales, I. Agudo, and J. Lopez, "Private set intersection: A systematic literature review," *Computer Science Review*, vol. 49, p. 100567, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013723000345

[35] V. Kolesnikov, M. Rosulek, N. Trieu, and X. Wang, "Scalable private set union from symmetric-key techniques," in *Advances in Cryptology – ASIACRYPT 2019*, S. D. Galbraith and S. Moriai, Eds. Cham: Springer International Publishing, 2019, pp. 636–666.

[36] J. Mao, J. Liu, X. Tian, Y. Pan, E. Trucco, and H. Lin, "Toward integrating federated learning with split learning via spatio-temporal graph framework for brain disease prediction," *IEEE Transactions on Medical Imaging*, vol. 44, no. 3, pp. 1334–1346, 2025.

[37] C. Hausleitner, H. Mueller, A. Holzinger, and B. Pfeifer, "Collaborative weighting in federated graph neural networks for disease classification with the human-in-the-loop," *Scientific Reports*, vol. 14, no. 1, Sep. 2024. [Online]. Available: http://dx.doi.org/10.1038/s41598-024-72748-7

[38] G. Lou, Y. Liu, T. Zhang, and J. X. Zheng, "STFL: A temporal-spatial federated learning framework for graph neural networks," *CoRR*, vol. abs/2111.06750, 2021. [Online]. Available: https://arxiv.org/abs/2111.06750

[39] A. Thakur, P. Sharma, and D. A. Clifton, "Dynamic neural graphs based federated reptile for semi-supervised multi-tasking in healthcare applications," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 4, pp. 1761–1772, 2022.

[40] H. C. Bayram and I. Rekik, "A federated multigraph integration approach for connectional brain template learning," in *Multimodal Learning for Clinical Decision Support*, T. Syeda-Mahmood, X. Li, A. Madabhushi, H. Greenspan, Q. Li, R. Leahy, B. Dong, and H. Wang, Eds. Cham: Springer International Publishing, 2021, pp. 36–47.

[41] D. Manu, Y. Sheng, J. Yang, J. Deng, T. Geng, A. Li, C. Ding, W. Jiang, and L. Yang, "Fl-disco: Federated generative adversarial network for graph-based molecule drug discovery: Special session paper," in *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2021, pp. 1–7.

[42] W. Zhu, J. Luo, and A. D. White, "Federated learning of molecular properties with graph neural networks in a heterogeneous setting," *Patterns*, vol. 3, no. 6, Jun 2022. [Online]. Available: https://doi.org/10.1016/j.patter.2022.100521

[43] Z. Liu, L. Yang, Z. Fan, H. Peng, and P. S. Yu, "Federated social recommendation with graph neural network," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, Aug. 2022. [Online]. Available: https://doi.org/10.1145/3501815

[44] P. Mai and Y. Pang, "Vertical federated graph neural network for recommender system," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 23 516–23 535. [Online]. Available: https://proceedings.mlr.press/v202/mai23b.html

[45] Z. Yang, Z. Peng, Z. Wang, J. Qi, C. Chen, W. Pan, C. Wen, C. Wang, and X. Fan, "Federated graph learning for cross-domain recommendation," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=UBpPOqrBKE

[46] M. Jiang, T. Jung, R. Karl, and T. Zhao, "Federated dynamic graph neural networks with secure aggregation for video-based distributed surveillance," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, May 2022. [Online]. Available: https://doi.org/10.1145/3501808

[47] 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," *ArXiv*, vol. abs/1909.12946, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:203593220

[48] H. Du, M. Shen, R. Sun, J. Jia, L. Zhu, and Y. Zhai, "Malicious transaction identification in digital currency via federated graph deep learning," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2022, pp. 1–6.