A OUTLINE

A.1 FedPG Algorithm and Complexity Analysis

For a more comprehensive presentation, we provide the complete algorithm of FedPG in Algorithm 1 and Algorithm 2. Based on this, to provide a clear motivation behind our proposed FedPG and reveal its algorithm complexity advantages compared to prevalent FL and FGL studies, we first conduct a comprehensive review in Table 6, where Subgraph/Graph-FL denotes the evaluation scenario of these methods within their respective original paper. Notably, most of them are confined to a single FGL scenario, lacking descriptions of their adaptability to other scenarios. In our investigation, we reasonably extend these methods to encompass broader FGL scenarios. Regrettably, they all fail to demonstrate competitive performance in the new FGL scenario. In contrast, we present a detailed description in Appendix A.8 that naturally extends FedPG to Graph-FL. Experimental results presented in Sec. 4.2 and Appendix A.8 validate the satisfied generalization of FedPG. Moreover, Table 6 underscores that existing methods often present significant communication overhead. For instance, while FedSage+ and FedGTA achieve competitive performance, they necessitate the exchange of other privacy-sensitive information. Although other methods exhibit relatively lower communication overhead, they still lag behind FedPG in terms of accuracy and effectiveness, as depicted in Fig. 1. This review underscores the limitations of existing approaches in addressing the multi-level heterogeneity inherent in the FGL.

Subsequently, we provide a theoretical algorithm complexity analysis of the FedPG and these SOTA baselines, as illustrated in Table 7, where n, m, c, and f are the number of nodes, edges, classes, and feature dimensions, respectively. s is the number of selected augmented nodes and g is the number of generated neighbors. b and T are the batch size and training rounds, respectively. k and K correspond to the number of times we aggregate features and moments order, respectively. N is the number of participating clients in each training round. t represents the number of clients exchanging information with the current client. ω represents the model-wise weight alignment loss term, Q denotes the size of the query set used for CL, E stands for the number of models for ensemble learning, M and p indicate the dimension of the trainable matrix used to mask trainable weights and the prototypes. Besides, P_g represents pseudo-graph data stored on the server side.

For general collaborative strategies, we choose SGC [61] as the local model (k-step feature propagation), otherwise, we adopt the model architecture (L-layer) used in their original paper. For the k-layer SGC model with batch size b, the $\mathbf{X}^{(k)}$ is the k-layers propagated feature bounded by a space complexity of $O((b+k)f)$. The overhead for linear regression by multiplying $\mathbf{W}$ is $O(f^2)$. In the training stage, the above procedure is repeated to iteratively update the model weights. For the server performing FedAvg, it needs to receive the model weights and the number of samples participating in this round. Its space complexity and time complexity are bounded as $O(Nf^2)$ and $O(N)$. As discovered by previous studies [13, 78, 79], the dominating term is $O(kmf)$ or $O(Lmf)$ when the graph is large since feature learning can be accelerated by parallel computation.

The current mainstream trend in FL or FGL studies emphasizes the development of well-designed client-side updates to fit local data. For instance, FedProx introduces model weight alignment loss ($O(\omega f^2)$ time complexity). Similarly, approaches like MOON,

Algorithm 1 Prototype-guided FGL (FedPG-Client)

for each participating client $i = 1, ..., N_t$ in round t **do**
  for each local model training epoch $e = 1, ..., E$ **do**
    Obtain the local prediction for nodes in the training sets;
    if $t = 1$ **then**
      Update client-independent GNN model weights f and g only by cross-entropy loss;
    else
      Receive the personalized global prototype;
      Update client-independent GNN model weights f and g according to the Eq. (6);
    end if
    Execute inference based on the local GNN to obtain the node embedding and soft labels;
    Perform unlabelled node annotations based on soft labels;
    Obtain topology-aware prototypes $\mathcal{P}_{c,h}^{i}$ for each class c and hop within RF by Eq. (2);
    Upload client-independent topology-aware local prototypes;
  end for
end for

Algorithm 2 Prototype-guided FGL (FedPG-Server)

for each communication round $t = 1, ..., T$ **do**
  if $t \neq 1$ **then**
    Receive the uploaded local prototypes;
    for each label class $c = 1, ..., |C|$ **do**
      for each hop $h = 0, ..., \min(L)$ **do**
        Obtain positive/negative query sets for each label class and hop within RF;
        for each server model training epoch $e = 1, ..., E$ **do**
          Obtain the universal global prototypes through $\tilde{\mathcal{P}}_{c,h} = GPG\left(\ddot{\mathcal{P}}_{c,h}, \theta\right)$;
          Update server-side trainable generation module weights θ by Eq. (4);
        end for
      end for
    end for
    for each participating client $i = 1, \ldots, N$ in round t **do**
      Obtain the personalized global prototype by Eq. (5)
    end for
    Broadcast the personalized global prototypes;
  end if
end for

Table 6: A summary of recent FGL studies. Comprehensive DH refers to the simultaneous consideration of both features and topology heterogeneity. Rank represents the average ranking of prediction accuracy.

Methods	Model Heterogeneity	Comprehensive DH	Communication Efficiency	Subgraph-FL	Graph-FL
GCFL+ [64] (NeurIPS 2021)	✗	✗	(Model Weights)	✗ (Rank 8.0)	✓ (Rank 3.6)
FedStar [53] (AAAI 2023)	✗	✓	(Model Weights)	✗ (Rank 6.8)	✓ (Rank 2.0)
FedSage+ [77] (NeurIPS 2021)	✗	✗	(Model Weights + Node Embeddings...)	✓ (Rank 4.8)	✗ (Rank 7.6)
FGSSL [23] (IJCAI 2023)	✗	✓	(Model Weights)	✓ (Rank 6.2)	✗ (Rank 4.2)
Fed-PUB [4] (ICML 2023)	✗	✗	(Model Weights + Trainable Masks)	✓ (Rank 5.4)	✗ (Rank 6.8)
FedGTA [38] (VLDB 2023)	✗	✗	(Model Weights + Lightweight Statistics)	✓ (Rank 3.2)	✗ (Rank 6.4)
AdaFGL [37] (ICDE 2024)	✗	✓	(Model Weights)	✓ (Rank 2.4)	✗ (Rank 4.6)
FedPG (This Paper)	✓	✓	(Prototypes)	✓ (Rank 1.0)	✓ (Rank 1.0)

Table 7: Algorithm complexity analysis for existing prevalent FL and FGL studies.

Method	Client Mem.	Server Mem.	Inference Mem.	Client Time.	Server Time.	Inference Time
FedAvg	$O((b+k)f+f^2)$	$O(Nf^2)$	$O((b+k)f+f^2)$	$O(kmf+nf^2)$	$O(N)$	$O(kmf+nf^2)$
FedProx	$O((b+k)f+\omega f^2)$	$O(Nf^2)$	$O((b+k)f+f^2)$	$O(kmf+nf^2+f^2)$	$O(N)$	$O(kmf+nf^2)$
MOON	$O((b+k)f+Qf^2)$	$O(Nf^2)$	$O((b+k)f+f^2)$	$O(kmf+nf^2+Qnf)$	$O(N)$	$O(kmf+nf^2)$
GCFL+	$O((b+k)f+f^2)$	$O(TNf^2)$	$O((b+k)f+f^2)$	$O(kmf+nf^2)$	$O(N^2(\log(N)+T^2f^2))$	$O(kmf+nf^2)$
FedStar	$O(E((b+k)f+f^2))$	$O(Nf^2)$	$O(E((b+k)f+f^2))$	$O(E(kmf+nf^2))$	$O(N)$	$O(E(kmf+nf^2))$
FedSage+	$O(L((n+sg)f+f^2))$	$O(LtNf^2)$	$O(L(n+sg)f+Lf^2)$	$O(L((m+sg)f+(n+sg)f^2))$	$O(N)$	$O(L((m+sg)f+(n+sg)f^2))$
FGSSL	$O(Q((b+k)f+f^2))$	$O(Nf^2)$	$O(L(b+k)f+Lf^2)$	$O(Qkmf+Qnf^2)$	$O(N)$	$O(kmf+nf^2)$
Fed-PUB	$O(M((b+k)f+f^2)+M^2)$	$O(N(f^2+M)+P_g)$	$O(M(b+k)f+Mf^2)$	$O(Mkmf+Mnf^2)$	$O(N^2(\log(N)+M^2))$	$O(Mkmf+Mnf^2)$
FGGP	$O((n+sg)f+f^2+Qcp)$	$O(Ncp)$	$O((b+k)f+f^2)$	$O((m+sg)f+(n+sg)f^2+Qcp^2)$	$O(N^2(\log(N)+c^2p^2)+Ncp)$	$O(kmf+nf^2)$
FedGTA	$O((b+k)f+f^2+kKc)$	$O(Nf^2+NkKc)$	$O((b+k)f+f^2)$	$O(km(f+knc)+n(f^2+c))$	$O(N+NkKc)$	$O(kmf+nf^2)$
AdaFGL	$O(E((b+k)f+f^2))$	$O(Nf^2)$	$O(E((b+k)f+f^2))$	$O(E(kmf+nf^2))$	$O(N)$	$O(E(kmf+nf^2))$
FedPG (ours)	$O((b+k)f+f^2+kcp)$	$O(NQkcp+p^2)$	$O((b+k)f+f^2)$	$O(kmf+nf^2+kp)$	$O(Qkcp^2)$	$O(kmf+nf^2)$

FGSSL, FGGP, FedStar, and AdaFGL employ CL loss and ensemble learning for local updates, introducing additional computational overhead upon the graph learning. Specifically, for the contrastive learning in MOON, FGSSL, and FGGP, the additional computational cost depends on the size and semantics of the query sample set, resulting in complexities of $O(Qf^2)$, $Q((b+k)f+f^2)$, $O(Qcp^2)$ respectively. This will lead to unacceptable computational overhead as the scale of local data increases. As for ensemble learning approaches like FedStar and AdaFGL, which maintain multiple models locally to extract private data semantics from various perspectives, they can be bounded by $O(E((b+k)f+f^2))$. Furthermore, FedSage+, Fed-PUB, and FedGTA exchange additional information during communication to improve federated training. Despite their inherent similarities, these methods exhibit significantly different time-space complexities due to variations in their design. Specifically, FedSage+ involves client-side data sharing for local subgraph data augmentation, leading to a complexity of $O(L((n+sg)f+f^2))$. Fed-PUB maintains a global pseudo-graph on the server side and utilizes locally uploaded weights to update trainable mask matrices for personalized learning, introducing a complexity of $O(N(f^2+M)+P_g)$. In contrast, FedGTA is a lightweight method that utilizes topology-aware soft labels to encode local data, enabling personalized model aggregation on the server side to facilitate federated training. However, the additional encoding signal results in a complexity of $O(kKC)$.

Compared to the above methods, our proposed FedPG exhibits significant scalability advantages in terms of algorithm time-space complexity. Specifically, in addition to the local graph learning common to all methods, FedPG discards a large number of trainable neurons in model weights, reducing complexity from $O(f)$ to $O(p)$ based on semantic encoding extracted from embedding space, where exists $p \ll f$. Meanwhile, in most cases, both k and c are relatively small real values, resulting in negligible computational overhead. Despite involving trainable generation of universal global prototypes on the server side, our approach maintains user-friendly computational overhead, owing to lightweight neural architectures and highly compressed prototype representations in low-dimensional space. Additionally, the experimental results regarding running efficiency in Sec. 4.2 further substantiate our claims from a practical application standpoint.

A.2 FedPG Server-side Optimization Direction

In this paper, we have introduced topology-guided prototype CL in Sec. 3.2, enabling the trainable generation of universal global prototypes shared by all participating clients. Intuitively, we aim for these global prototypes to inherit private graph semantic context from each local prototype within RF and extract the real semantics of each class from a global perspective, exhibiting consistent intra-class semantics and clear inter-class boundaries. In our proposed CL loss, we contrast the anchor prototypes generated by the trainable module (i.e., universal global prototypes) with prototypes from each topology-guided query sample sharing the same class and prototypes from other classes with different semantics. With the CL process, we aim to minimize the distance between anchor prototypes and class semantics-aligned prototypes. Simultaneously,

we maximize the distance from anchor prototypes to others. This CL strategy ensures compact intra-class semantic consistency and clear inter-class decision boundaries in FGL from a topology-aware multi-granularity representation perspective. Naturally, based on the Eq. (4), the following optimization objectives can be derived:

$$\begin{aligned}\mathcal{L}_{c,h} &= -\log \frac{\sum_{i\in\mathcal{S}_{c,\Delta h}} \exp\left(\tilde{D}+\tilde{M}\right)}{\sum_{i\in\mathcal{S}_{c,\Delta h}} \exp\left(\tilde{D}+\tilde{M}\right)+\sum_{i\in\mathcal{S}_{c',\Delta h}} \exp\left(\tilde{M}\right)} \\ &= \log \frac{\sum_{i\in\mathcal{S}_{c,\Delta h}} \exp\left(\tilde{D}+\tilde{M}\right)+\sum_{i\in\mathcal{S}_{c',\Delta h}} \exp\left(\tilde{D}\right)}{\sum_{i\in\mathcal{S}_{c,\Delta h}} \exp\left(\tilde{D}+\tilde{M}\right)} \\ &= \log\left(1+\frac{\sum_{i\in\mathcal{S}_{c',\Delta h}} \exp\left(\tilde{D}\right)}{\sum_{i\in\mathcal{S}_{c,\Delta h}} \exp\left(\tilde{D}+\tilde{M}\right)}\right) \\ &= \underbrace{\log\left(\sum_{i\in\mathcal{S}_{c',\Delta h}} \exp\left(\tilde{D}\right)\right)}_{\textbf{Semantic Consistency}} \underbrace{-\log\left(\sum_{i\in\mathcal{S}_{c,\Delta h}} \exp\left(\tilde{D}+\tilde{M}\right)\right)}_{\textbf{Consistency + Separability}}.\end{aligned} \tag{7}$$

In Eq. (7), we reformulate the topology-guided CL loss for a detailed analysis of its optimization directions. Based on our analysis, we derive the following key insights of generated universal global prototypes regarding its underlying principles:

(1) Intra-class semantic consistency: With the CL process, minimizing Eq. (7) implies minimizing $\log\left(\sum_{i\in\mathcal{S}_{c',\Delta h}} \exp\left(\tilde{D}\right)\right)$. Since the logarithmic function with a base greater than 1 and $\tilde{D}(\cdot) \in [0,1]$, this encourages $\tilde{D}$ to approach 1, effectively maximizing the distance between anchor prototypes and prototypes of different classes. Similarly, maximizing the objective drives $\log\left(\sum_{i\in\mathcal{S}_{c,\Delta h}} \exp\left(\tilde{D}\right)\right)$ to approach 0, effectively minimizing the distance between anchor prototypes and prototypes of the same class. These objectives ensure the semantic accuracy of the generated anchor prototypes, manifested in intra-class semantic consistency.

(2) Inter-class semantic boundaries separability: While (1) guarantees accurate extraction of semantic information for each label class, the separation boundaries between anchor prototypes representing different classes may not be sufficiently clear, especially when generating prototypes for each hop. This is due to without distinguishing the semantic boundaries, graph homophily brings semantic space distortions, leading to the confusion of label class semantics. Hence, to ensure discriminative semantic spaces for prototypes, we introduce adaptive maximum discrepancy to increase the optimization difficulty of $\log\left(\sum_{i\in\mathcal{S}_{c,\Delta h}} \exp\left(\tilde{D}\right)\right)$. This allows the trainable module to focus more on aligning the semantic representations between anchor prototypes and corresponding query positive sample prototypes, thereby increasing the representation distance between anchor prototypes of different classes and achieving separable inter-class semantic boundaries.

A.3 Dataset Description

In Graph-FL, we conduct experiments on the molecules (NCI1) [56], protein (PROTEINS) [8], movie (IMDB-BINARY), and collaboration (COLLAB) network [68]. Regarding the transductive Subgraph-FL, we conduct experiments on the citation (Cora, Citeseer, PubMed, ogbn-arxiv) [21, 71], co-purchase (Computer, ogbn-products) [21, 50], and co-author (Physics) [50] network. As for the inductive setting, we conduct experiments on the image (Flickr) and social (Reddit) network [74]. The description of all datasets is listed below:

Cora, **CiteSeer**, and **PubMed** [71] are three citation network datasets, where nodes represent papers and edges represent citation relationships. The node features are word vectors, where each element indicates the presence or absence of each word in the paper.

Amazon Computers [50] are segments of the Amazon co-purchase graph, where nodes represent items and edges represent that two goods are frequently bought together. Given product reviews as bag-of-words node features.

Coauthor Physics [50] are co-authorship graphs based on the Microsoft Academic Graph [58], where nodes are authors, edges are co-author relationships, node features represent paper keywords, and labels indicate the research field.

ogbn-arxiv [21] are two citation graphs indexed by MAG [58]. Each paper involves averaging the embeddings of words in its title and abstract. The embeddings of individual words are computed by running the skip-gram model.

ogbn-products [21] is a co-purchasing network, where nodes represent products and edges represent that two products are frequently bought together. Node features are generated by extracting bag-of-words features from the product descriptions.

Flickr [74] dataset originates from the SNAP, where nodes represent images, and connected images from common properties. Node features are 500-dimensional bag-of-words representations.

Reddit [20] dataset collected from Reddit, where 50 large communities have been sampled to build a post-to-post graph, connecting posts if the same user comments on both. For features, off-the-shelf 300-dimensional GloVe vectors are used.

NCI1 [56] dataset originates from cheminformatics. Each graph represents a chemical compound, where nodes represent atoms and edges represent chemical bonds. The dataset is used in anticancer screenings, classifying chemicals as positive or negative for lung cancer. Each node has an input feature representing the corresponding atom type.

PROTEINS [8] is a dataset of proteins that comprises proteins categorized as either enzymes or non-enzymes. Nodes correspond to amino acids, and an edge connects two nodes if they are within 6 Angstroms of each other.

IMDB-BINARY [68] is a movie collaboration dataset containing the ego-networks of 1,000 actors/actresses who have participated in movies listed on IMDB. In each graph, nodes represent actors/actresses, and edges exist between them if they have appeared together in the same movie. These graphs focus on the Action and Romance genres.

COLLAB [68] is a scientific collaboration dataset consisting of ego networks of researchers. Each graph represents the network of a researcher, where the researcher and their collaborators are nodes, and edges denote collaboration between them. Researchers in the dataset are categorized into three fields. Each researcher's ego network is labeled with one of these three fields.

Notably, for IMDB-BINARY and COLLAB, the provided paper [68] only reveals the topology without corresponding node features. Thus, we derive features based on node degree. The motivation

Table 8: The statistical information of the Graph-FL experimental datasets.

Graph-FL	#Graphs	#Avg.Edges	#Features	#Classes	#Train/Test	#Task	Description
NCI1	4,110	32.30	37	2	90%/10%	Graph Classification	molecules network
PROTEINS	1,113	72.82	4	2	90%/10%	Graph Classification	protein network
IMDB-BINARY	1,000	96.53	degree	2	90%/10%	Graph Classification	movie network
COLLAB	5,000	2457.78	degree	3	90%/10%	Graph Classification	collaboration network

behind this approach is as follows: (1) Node degree serves as supplementary information, aiding the model in understanding the node's structural position and importance in the graph. (2) One-hot encoding of node degree creates a uniform and comparable feature space, facilitating the distinction of structural roles by GNNs. (3) Using one-hot encoding of degrees provides the model with a clear, standardized input format for information propagation and updating between nodes. (4) Corresponding node feature initialization strategies have been widely employed in previous studies [53, 64, 66].

A.4 Data Simulation

Building upon the above datasets, we employ a label Dirichlet and uniform-based strategy (i.e., Non-iid and iid) to simulate distributed graph storage [53, 64]. Meanwhile, we apply Louvain- and Metis-based strategies [7, 28] for distributed subgraph simulation [4, 37, 38, 77]. Notably, since the ogbn-papers100M dataset contains a large number of unlabeled nodes, we only perform Louvain split on it. This is because we can control the amount of labeled data contained by each client by assigning communities.

Louvain-based Simulation for Subgraph-FL. In this setting, we first apply the Louvain [7] algorithm to the global graph to obtain various communities of different sizes. Subsequently, adhering to the principle that each client has approximately equal numbers of nodes, we assign all the obtained communities to multiple client subgraphs. This strategy draws inspiration from previous FGL investigations [23, 77, 81] and addresses practical scenarios encountered in Subgraph-FL. Specifically, each subgraph should be gathered by distinct data collectors within a relatively consistent spatio-temporal scope. Given notable differences in collection methodologies, data quality, and well-known homophily assumption, it is anticipated that subgraphs from each data collection entity will manifest diverse community cluster distributions and substantial quantity discrepancies. In such circumstances, a reasonably trustworthy third party should perform uniform allocation without explicit access to query data to facilitate collaboration.

Metis-based Simulation for Subgraph-FL. In contrast to the aforementioned Louvain-based split, the Metis-based [28] strategy simulates a more general scenario. Specifically, drawing inspiration from distributed graph systems and prior FGL studies [4, 37, 38], we directly apply the Metis algorithm to the original global graph based on the desired number of simulated clients and allocate the results directly to clients. While fundamentally similar to Louvain in that both are based on topological discovery of potential community structures, Metis is inherently designed for distributed settings. Essentially, in practical applications, we no longer assume data collection occurs across multiple data collection entities within a relatively fixed spatio-temporal scope, nor do we require the presence of a relatively trusted fair third party.

Single-based Simulation for Graph-FL. We distribute graphs from a single dataset to multiple clients, inspired by the label Dirichlet and uniform distribution used in CV-based FL [32, 43, 83]. Intuitively, we treat each independent graph held by each client as an image, as independent graphs relatively lack interconnections in graph-based classification or regression tasks. In a real-world scenario based on graph-FL, we could consider a situation as follows: There has been a surge in startups focusing on AI4Science, aiming to apply artificial intelligence and computer science technologies to traditional scientific challenges like bioinformatics. For example, they leverage graph machine learning for graph-level prediction in drug design tasks. These startups often collaborate with medical institutions to expand their operations. However, due to the valuable nature of bioinformatics data, competitive concerns, and legal restrictions on data sharing among medical institutions, many companies aim to optimize their local models without directly sharing data to improve the practical performance of their business pipelines. Additionally, the label Dirichlet distribution simulates scenarios where data sparsity for certain drug-protein molecules leads to label distribution imbalances.

A.5 Compared Baselines

To alleviate the randomness and ensure a fair comparison, we repeat each experiment 10 times for unbiased performance. Unless otherwise stated, for Graph- and Subgraph-FL, we respectively adopt GIN and GAMLP as the backbone and employ the uniform- and Metis-based 10-client simulation. For the detailed extension of FedPG to the Graph-FL, please refer to Appendix A.8. Meanwhile, to evaluate the effectiveness of our proposed FedPG for FGL MH (detailed in Sec. 1), we consider Subgraph/Graph-FL with 5x client partitions. Then, we randomly and equally allocate the following 5 local backbones: GCN [29], GAT [55]/GIN [66], GraphSAGE [20], SGC [61], and GCNII [14]. The baseline details are listed below:

FedAvg [44] serves as a foundational method in FL, enabling decentralized model training across diverse devices. Initiated by a central server that distributes a global model, clients independently execute local updates through SGD. Subsequently, these updates are aggregated by the server via averaging to refine the global model.

FedProx [34] allows for variable amounts of work to be performed locally across devices, and relies on a proximal term to help stabilize the method. Theoretically, it offers convergence guarantees under conditions of data Non-iid and variable device workloads.

Scaffold [27] employs control variates to mitigate client-drift in FL. Demonstrating significant reductions in communication rounds, Scaffold is resilient to data heterogeneity and client sampling.

MOON [33] is a model-contrastive FL framework that enhances local training by leveraging model representation similarities through contrastive learning at the model level.

FedDC [17] is a novel FL algorithm that corrects local drift through lightweight modifications. Each client tracks the deviation between local and global model parameters using an auxiliary variable, enhancing parameter-level consistency.
FedDistill [26] introduces federated distillation and augmentation to optimize distributed model training. the former minimizes communication payloads compared to traditional FL, especially with large models. the latter counters Non-iid data issues by each device training a generative model to approximate an IID dataset locally.
FedGen [85] proposes a data-free knowledge distillation approach for FL to address DH. It involves the server learning a lightweight generator that assembles user information without requiring a proxy dataset. The generated knowledge then guides local training as an inductive bias, enhancing the quality of the aggregated model.
FedKD [60] introduces a communication-efficient FL method utilizing adaptive distillation. Clients reciprocally train a teacher and a student model locally, sharing only the student model to minimize bandwidth. Distillation intensity is modulated by prediction quality. Additionally, they employ a dynamic gradient approximation using singular value decomposition to further cut communication costs.
FedProto [54] is the first federated prototype learning framework for FL heterogeneity. Instead of exchanging gradients, clients and the server share abstract class prototypes. FedProto aggregates local prototypes and distributes global ones back to clients to regularize local model training, aiming to align local prototypes with global standards while minimizing local classification errors.
FedNH [15] addresses class imbalance by enhancing both personalization and generalization of local models. FedNH distributes class prototypes uniformly in the latent space, infusing class semantics to prevent prototype collapse and enhance model performance. This dual approach improves local models, boosting the generalization of the global model and thus refining personalized models.
FPL [24] aims to address domain shift in FL by constructing cluster and unbiased prototypes. These prototypes encapsulate domain knowledge and offer a fair convergence target, enhancing generalizability. FPL aligns sample embeddings with same-class cluster prototypes and enforces consistency regularization with unbiased prototypes to maintain local instance alignment.
FedTGP [75] unlikes conventional methods that aggregate prototypes via weighted averaging, FedTGP uses adaptive contrastive Learning to train global prototypes on the server, enhancing prototype separability and preserving semantic integrity.
FGGP [57] is designed to enhance the generalization of global models across diverse domain-specific graphs. FGGP separates the global model into feature extraction and classification levels, linked by semantic center prototypes that capture domain information and improve class discrimination. This approach utilizes contrastive learning to refine these prototypes, thereby boosting the overall predictive performance and feature extractor generalization.
GCFL+ [64] dynamically clusters local systems using GNN gradients to reduce structural and feature heterogeneity. Addressing the issue of fluctuating gradients, they enhance GCFL with a gradient sequence-based clustering mechanism using dynamic time warping, thereby improving clustering quality and theoretical robustness.
FedStar [53] shares structural embeddings across clients using an independent structure encoder. This design allows FedStar to capture domain-invariant structural information while enabling personalized feature learning, thereby avoiding feature misalignment and enhancing inter-graph learning efficacy.
FedSage+ [77] integrates node features, link structures, and labels using a GraphSage model and FedAvg across local subgraphs. FedSage+ extends this by training a generator to address missing links, enhancing model robustness and completeness.
FGSSL [23] handles local client distortion in FL by focusing on node-level semantics and graph-level structures. They enhance node discrimination by aligning local nodes with their global counterparts of the same class and distancing them from different classes. Additionally, FGSSL transforms adjacency relationships into similarity distributions, using the global model to distill relational knowledge into local models, preserving both structure and discriminability.
Fed-PUB [4] is a novel framework for personalized subgraph FL that enhances local GNNs interdependently rather than forming a single global model. Fed-PUB computes similarities between local GNNs using functional embeddings derived from random graph inputs, facilitating weighted averaging for server-side aggregation. Additionally, it employs a personalized sparse mask at each client to selectively update subgraph-relevant parameters.
FedGTA [38] innovatively merges large-scale graph learning with FGL. Clients encode topology and node attributes, compute local smoothing confidence and mixed moments of neighbor features, and then upload these to the server. The server uses this data to perform personalized model aggregation, utilizing local smoothing confidence as weights for effective integration.
AdaFGL [37] introduces a two-step personalized approach that first aggregates multi-client models into a federated knowledge extractor during the final round at the server. Subsequently, each client undertakes personalized training utilizing the local subgraph and this federated extractor.
GCN [29] introduces a novel approach to graphs that is based on a first-order approximation of spectral convolutions on graphs. This approach learns hidden layer representations that encode both local graph structure and features of nodes.
GAT [55] utilizes attention mechanisms to quantify the importance of neighbors for message aggregation. This strategy enables implicitly specifying different weights to different nodes in a neighborhood, without depending on the graph structure upfront.
GIN [66] construct a straightforward architecture that is demonstrably the most expressive within the GNN class and matches the power of the Weisfeiler-Lehman graph isomorphism test
GraphSAGE [20] leverages neighbor node attribute information to efficiently generate representations. This method introduces a general inductive framework that leverages node feature information to generate node embeddings for previously unseen data.
SGC [61] simplifies GCN by removing non-linearities and collapsing weight matrices between consecutive layers. Theoretical analysis show that the simplified model corresponds to a fixed low-pass filter followed by a linear classifier.
GCNII [14] incorporates initial residual and identity mapping. Theoretical and empirical evidence is presented to demonstrate how these techniques alleviate the over-smoothing problem.
GAMLP [79] is designed to capture the inherent correlations between different scales of graph knowledge to break the limitations of the enormous size and high sparsity level of graphs hinder their applications under industrial scenarios.

Table 9: Graph-FL test accuracy (%). The best result is bold. The second result is underlined.

Datasets →	NCI1					PROTEINS				
Simulation →	Non-iid (ACC)			iid		Non-iid (ACC)			iid	
Method (↓)	0.3	0.5	1	ACC	AUC	0.3	0.5	1	ACC	AUC
FedAvg	68.7±5.4	70.6±4.0	72.8±3.4	74.5±2.2	73.8±1.7	67.4±4.5	69.5±3.9	72.5±2.8	73.4±2.5	73.9±2.6
FedProx	69.4±3.9	70.8±3.5	72.0±2.8	72.8±2.5	73.1±2.6	66.9±4.7	69.9±3.8	72.1±2.6	73.9±2.9	73.5±2.7
FedDC	68.5±4.8	70.1±4.6	72.4±3.5	73.6±2.8	73.5±2.4	66.5±5.2	69.2±4.0	72.9±3.0	74.6±2.6	75.0±2.4
FedProto	67.4±3.8	69.2±3.5	72.2±3.2	73.8±1.9	74.2±1.7	67.1±3.4	68.6±2.9	71.8±2.4	73.1±1.9	72.8±2.1
PFL	69.0±3.0	69.8±3.2	72.0±2.9	73.2±2.6	73.5±2.2	67.5±3.5	68.2±3.4	71.4±2.8	73.5±2.0	73.1±2.2
FedTGP	66.0±5.2	68.7±4.1	72.5±2.5	73.5±2.0	74.0±2.3	66.9±4.3	68.3±3.6	71.5±2.7	72.8±2.4	73.0±2.5
GCFL+	74.3±2.0	75.4±2.3	76.1±1.8	77.3±1.6	77.6±1.8	71.5±3.4	73.2±3.6	75.5±2.4	76.8±2.8	76.5±3.3
FedStar	71.5±4.9	74.2±3.8	76.8±2.6	77.9±1.5	78.2±1.3	68.6±4.9	71.8±4.2	75.9±3.0	77.5±3.2	77.2±3.5
FGGP	73.6±3.7	75.0±2.5	76.5±2.4	77.6±2.1	77.1±2.2	69.3±4.5	70.2±3.9	73.4±3.3	74.3±2.6	74.0±2.7
FedPG	**77.4±2.6**	**78.2±2.3**	**79.3±2.0**	**79.8±1.7**	**79.6±1.6**	**76.5±3.4**	**77.4±3.0**	**78.6±2.5**	**79.2±2.2**	**79.4±2.0**

Table 10: Graph-FL test accuracy (%). The best result is bold. The second result is underlined.

Datasets →	IMDB-BINARY					COLLAB				
Simulation →	Non-iid (ACC)			iid		Non-iid (ACC)			iid	
Method (↓)	0.3	0.5	1	ACC	AUC	0.3	0.5	1	ACC	AUC
FedAvg	72.4±4.2	74.8±3.4	77.2±3.0	78.6±2.7	78.9±2.6	74.1±2.9	75.6±2.4	77.2±1.8	77.8±1.6	78.0±1.9
FedProx	73.1±4.6	75.1±3.7	77.6±2.8	78.2±3.0	78.5±3.4	74.2±3.6	74.9±3.2	77.5±1.7	78.1±1.4	78.5±1.5
FedDC	72.2±5.9	74.5±5.2	76.8±4.2	77.9±3.8	78.3±3.5	73.4±4.6	75.2±3.5	77.0±2.3	77.5±2.4	77.9±2.2
FedProto	71.9±3.5	73.2±3.1	75.6±2.4	76.3±2.2	76.7±2.3	72.2±2.6	73.0±1.9	74.2±1.5	74.5±1.3	74.2±1.5
PFL	72.2±4.0	72.8±3.6	75.9±2.3	76.1±2.5	76.9±2.6	72.0±3.0	73.1±2.5	74.4±1.8	74.9±1.7	75.0±1.5
FedTGP	72.4±3.8	73.0±3.3	75.2±3.0	76.6±2.4	76.5±2.8	71.7±2.8	72.8±2.2	75.0±1.4	75.3±1.6	75.5±1.9
GCFL+	76.2±5.4	77.9±4.2	79.5±3.7	80.4±3.5	79.8±3.9	73.5±3.5	74.5±2.7	77.6±1.8	78.2±1.5	78.0±1.8
FedStar	75.0±6.3	77.2±5.6	80.3±4.5	81.2±3.9	80.9±3.8	73.9±4.9	75.0±3.8	78.5±3.0	79.4±2.8	79.9±2.7
FGGP	71.6±4.3	73.0±3.8	76.9±2.5	77.4±2.4	77.5±2.5	72.9±3.6	74.7±3.1	76.1±2.2	76.5±1.9	76.9±2.1
FedPG	**78.0±3.7**	**79.2±3.2**	**80.9±3.0**	**81.5±2.6**	**82.2±2.9**	**78.0±2.8**	**78.9±2.2**	**80.0±1.6**	**80.4±1.5**	**80.1±1.6**

A.6 Experiment Environment

The experiments are conducted on the machine with Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz, and NVIDIA A100 80GB PCIe and CUDA 12.2. The operating system is Ubuntu 18.04.6 with 216GB memory. As for software versions, we use Python 3.8.10, Pytorch 1.10.0, CUDA 11.7.0, and the Optuna version we use is 3.6.1.

A.7 Hyperparameter Settings

The hyperparameters in the baselines are set according to the original paper if available. Otherwise, we perform a hyperparameter search via the Optuna [2]. For our proposed FedPG, we explore $\epsilon \in [0, 1]$ and $\triangle s \in [0, 1]$ for the adaptive maximum discrepancy and hop-based sampling, where $\triangle s$ is the ratio of additional sampling from other hops based on the number of current query samples. The personalized fusion factor α, similarity threshold λ, and loss factor μ are explored within the ranges of 0 to 1.

A.8 Experiments for Graph-FL

Extend FedPG to Graph-FL. Compared to Subgraph-FL with structured relationships, defining effective neighborhoods is challenging in Graph-FL due to the absence of inherent connections between independent graphs. To address this, our approach utilizes appropriate representation distance metric functions and decaying thresholds to identify potentially similar independent graphs as a generalized neighborhood. Introducing a decaying threshold allows the method to uncover multi-hop neighborhoods, beyond 1-hop connections. This strategy is inspired by the graph homophily assumption from complex network research [3, 41, 45], which suggests that similar samples are more likely to have relational links. Building on this, we posit that independently represented graphs sharing similarities should exhibit underlying correlations. This belief arises from the effective representations of GNNs in graph classification, offering insights into structural nuances and node feature interactions. By emphasizing the identification and utilization of these similarities, our approach aims to improve neighborhood definition effectiveness in Graph-FL scenarios. To this end, in Graph-FL, the natural extension of FedPG can be achieved by adding a generalized neighborhood construction step after obtaining the encoded representations of each independent graph locally.

End-to-end Comparison in Graph-FL. To answer **Q2**, we present experimental results in Table 9 and Table 10 under various data simulation settings. In our implementation, 0.3, 0.5, and 1 represent 0.3-, 0.5-, and 1-Dirichlet Non-iid settings. Conversely, iid denotes a distributed graph storage simulation achieved by uniformly sampling label distributions. Based on this, we employ ACC and AUC

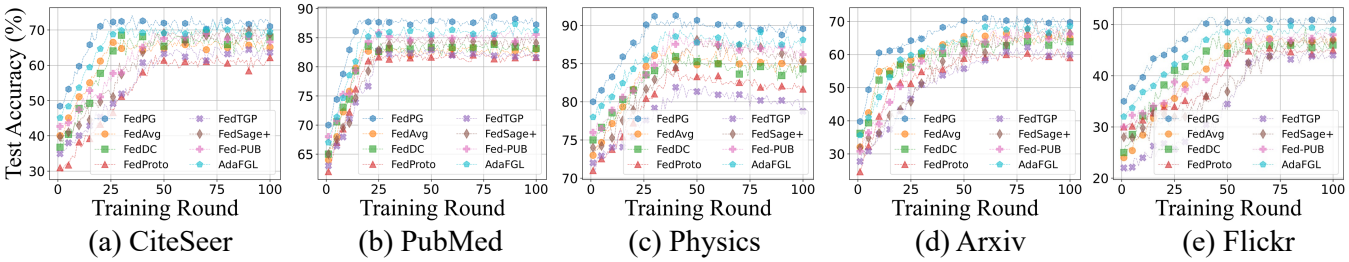

Figure 6: Convergence curves with Louvain split and GCNII backbone.

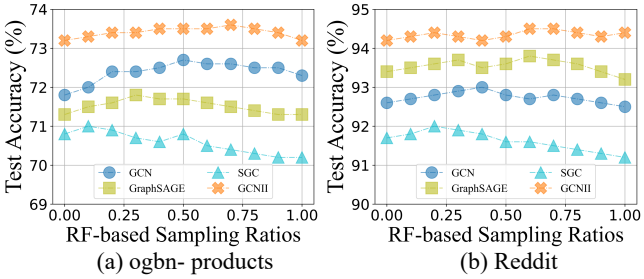

Figure 7: Performance with RF prompts.

for comprehensive assessment. Experimental results demonstrate that FedPG significantly outperforms SOTA baselines across all scenarios. Compared to other FPL methods, FedPG's explicit consideration of topology-driven optimal prototype generation positively impacts federated collaborative optimization. Additionally, our in-depth analysis yields the following insights:

(1) Non-iid data settings pose collaborative optimization challenges for all FL approaches, hindering the acquisition of valuable global consensus for guiding local model updates. However, FPL methods mitigate this issue to some extent by directly extracting global class semantic knowledge without aggregating model weights. (2) For FPL methods, the intrinsic semantic content of sample features is crucial, directly affecting the accuracy of extracting global semantic consensus to guide local updates. This aspect is confirmed in experimental results: FPL methods perform significantly better on NCI1 and PROTEINS datasets with genuine semantic content compared to IMBD-BINARY and COLLAB datasets where node degree is used as a feature. While local graph learning models based on node degree can encode structural insights to some extent within embeddings, they still lack genuine semantic knowledge compared to other methods. Despite challenges posed by the absence of genuine semantics, our proposed FedPG still stands out and achieves satisfactory predictive performance.

A.9 Experiments for Comprehensive Evaluation

Training Efficiency with Convergence Curves. In our proposed FedPG, we aim to derive optimal prototypes for federated collaborative training as follows: (1) Each client generates multi-granularity representations for prototypes of each class in a topology-aware manner to fully extract local graph semantic contexts. (2) The server customizes personalized global prototypes for each client through topology-guided CL and personalized mechanisms for broadcasting. The essence of these processes is to fully consider the graph-specific topological context in graph-based FPL to maximize training efficiency, reflected in faster and more stable convergence curves and higher predictive performance during federated training.

To validate our claims, we provide comprehensive experimental results in Fig. 6. In PubMed, we observe that all methods demonstrate rapid convergence and ensure stable performance. However, on datasets like ogbn-arxiv and Flickr, achieving optimal predictions requires more communication rounds and is accompanied by unstable optimization. Although this difficulty varies across different datasets due to differences in data quantity and quality, FedPG consistently demonstrates SOTA performance from start to finish, regardless of dataset intricacies. For example, in the Physics dataset, FedPG nearly reaches converged performance by the 25th epoch and maintains stability throughout subsequent training.

Performance under RF-based Query Sets Sampling. As highlighted in Sec. 3.2, RF-prompted Context plays a pivotal role in our proposed FedPG framework. Specifically, it first enhances the ability to extract graph semantic information by providing topology-aware multi-granularity representations for local prototypes of each label class. Subsequently, on the server side, FedPG conducts CL for each label class within each hop, where the query set naturally comprises prototypes of different label classes within the current hop. However, to enrich FedPG's perception of topological context and introduce abundant semantic samples for performance enhancement, we sample prototypes of the same class from different hops to supplement the query set. To verify the effectiveness of the above strategy, we provide experiments in Fig. 7 based on the ablation study in Table 5. In our experiments, we select four commonly used local backbone GNN models in FGL to explore the underlying mechanisms of RF-based sampling. Our findings are as follows:

(1) In all backbones, both insufficient and excessive sampling adversely affect the final performance. This can be attributed to inadequate sampling failing to provide a diverse range of query samples, thus limiting the representation capacity of the generated universal global prototypes. However, an excessive number of samples rigorously tests the quality of locally generated prototypes, potentially resulting in misleading representations. (2) The quality of local prototypes fundamentally depends on the local backbone's reasoning ability. This implies that stronger local backbones (more trainable weights and reasonable architectures) are more likely to generate high-quality local prototypes, thereby ensuring sample quality and benefiting more from supplementary samples. This is confirmed by the performance curve of GCNII in Fig. 7. Notably, GraphSAGE is specifically designed for inductive scenarios, hence exhibiting slightly better performance on Reddit compared to GCN, while slightly underperforming on the transductive ogbn-products.

REFERENCES

[1] 2024. FedPG Technical Report. In *https://github.com/xkLi-Allen/FedPG*.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD*.

[3] Org. Cambridge. Ebooks. Online. Book. Author@Ac. [n.d.]. *Networks, Crowds, and Markets*. Networks, Crowds, and Markets.

[4] Jinheon Baek, Wonyong Jeong, Jiongdao Jin, Jaehong Yoon, and Sung Ju Hwang. 2023. Personalized Subgraph Federated Learning. (2023).

[5] Vicente Balmaseda, María Coronado, and Gonzalo de Cadenas-Santiagoc. 2023. Predicting Systemic Risk in Financial Systems Using Deep Graph Learning. *Intelligent Systems with Applications* (2023), 200240.

[6] Dongmin Bang, Sangsoo Lim, Sangseon Lee, and Sun Kim. 2023. Biomedical knowledge graph learning for drug repurposing by extending guilt-by-association to multiple layers. *Nature Communications* 14, 1 (2023), 3570.

[7] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (2008), P10008.

[8] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. 2005. Protein function prediction via graph kernels. *Bioinformatics* 21, suppl_1 (2005), i47–i56.

[9] Lei Cai, Jundong Li, Jie Wang, and Shuiwang Ji. 2021. Line graph neural networks for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[10] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *International Conference on Learning Representations, ICLR*.

[11] Chuan Chen, Ziyue Xu, Weibo Hu, Zibin Zheng, and Jie Zhang. 2024. Fedgl: Federated graph learning framework with global self-supervision. *Information Sciences* 657 (2024), 119976.

[12] Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. 2023. NAGphormer: A tokenized graph transformer for node classification in large graphs. In *International Conference on Learning Representations, ICLR*.

[13] Ming Chen, Zhewei Wei, Bolin Ding, Yaliang Li, Ye Yuan, Xiaoyong Du, and Ji-Rong Wen. 2020. Scalable graph neural networks via bidirectional propagation. *Advances in Neural Information Processing Systems, NeurIPS* (2020).

[14] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. 2020. Simple and deep graph convolutional networks. In *International Conference on Machine Learning, ICML*.

[15] Yutong Dai, Zeyuan Chen, Junnan Li, Shelby Heinecke, Lichao Sun, and Ran Xu. 2023. Tackling data heterogeneity in federated learning with class prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI*.

[16] Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Ben Chamberlain, Michael Bronstein, and Federico Monti. 2020. Sign: Scalable inception graph neural networks. *arXiv preprint arXiv:2004.11198* (2020).

[17] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. 2022. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*.

[18] Zihao Gao, Huifang Ma, Xiaohui Zhang, Yike Wang, and Zheyu Wu. 2023. Similarity measures-based graph co-contrastive learning for drug–disease association prediction. *Bioinformatics* 39, 6 (2023), btad357.

[19] Kun Guo, Yutong Fang, Qingqing Huang, Yuting Liang, Ziyao Zhang, Wenyu He, Liu Yang, Kai Chen, Ximeng Liu, and Wenzhong Guo. 2023. Globally consistent federated graph autoencoder for non-IID graphs. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*.

[20] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems, NeurIPS* (2017).

[21] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in Neural Information Processing Systems, NeurIPS* (2020).

[22] Youpeng Hu, Xunkai Li, Yujie Wang, Yixuan Wu, Yining Zhao, Chenggang Yan, Jian Yin, and Yue Gao. 2021. Adaptive hypergraph auto-encoder for relational data clustering. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[23] Wenke Huang, Guancheng Wan, Mang Ye, and Bo Du. 2023. Federated graph semantic and structural learning. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*.

[24] Wenke Huang, Mang Ye, Zekun Shi, He Li, and Bo Du. 2023. Rethinking federated learning with domain shift: A prototype view. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*.

[25] Woochang Hyun, Jaehong Lee, and Bongwon Suh. 2023. Anti-Money Laundering in Cryptocurrency via Multi-Relational Graph Neural Network. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 118–130.

[26] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2018. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479* (2018).

[27] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning, ICML*.

[28] George Karypis and Vipin Kumar. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing* 20, 1 (1998), 359–392.

[29] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations, ICLR*.

[30] J. Klicpera, A. Bojchevski, and S Günnemann. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *International Conference on Learning Representations, ICLR*.

[31] Guohao Li, Chenxin Xiong, Guocheng Qian, Ali Thabet, and Bernard Ghanem. 2023. DeeperGCN: Training Deeper GCNs with Generalized Aggregation Functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[32] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated learning on non-iid data silos: An experimental study. In *International Conference on Data Engineering, ICDE*.

[33] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*.

[34] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems, MLSys* (2020).

[35] Xunkai Li, Ronghui Guo, Jianwen Chen, Youpeng Hu, Meixia Qu, and Bin Jiang. 2022. Effective Hybrid Graph and Hypergraph Convolution Network for Collaborative Filtering. *Neural Comput. Appl.* 35, 3 (sep 2022), 2633–2646.

[36] Xunkai Li, Jingyuan Ma, Zhengyu Wu, Daohan Su, Wentao Zhang, Rong-Hua Li, and Guoren Wang. 2024. Rethinking Node-wise Propagation for Large-scale Graph Learning. In *Proceedings of the ACM Web Conference, WWW*.

[37] Xunkai Li, Zhengyu Wu, Wentao Zhang, Henan Sun, Rong-Hua Li, and Guoren Wang. 2024. AdaFGL: A New Paradigm for Federated Node Classification with Topology Heterogeneity. *arXiv preprint arXiv:2401.11750* (2024).

[38] Xunkai Li, Zhengyu Wu, Wentao Zhang, Yinlin Zhu, Rong-Hua Li, and Guoren Wang. 2023. FedGTA: Topology-Aware Averaging for Federated Graph Learning. *Proceedings of the VLDB Endowment* (2023).

[39] Xinting Liao, Weiming Liu, Chaochao Chen, Pengyang Zhou, Huabin Zhu, Yanchao Tan, Jun Wang, and Yue Qi. 2023. HyperFed: hyperbolic prototypes exploration with consistent aggregation for non-IID data in federated learning. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*.

[40] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. 2024. One for all: Towards training one graph model for all classification tasks. In *International Conference on Learning Representations, ICLR*.

[41] M., E., J., and Newman. 2003. Mixing patterns in networks. *Physical Review E* 67, 2 (2003), 26126–26126.

[42] Peihua Mai and Yan Pang. 2023. Vertical Federated Graph Neural Network for Recommender System. *arXiv preprint arXiv:2303.05786* (2023).

[43] Koji Matsuda, Yuya Sasaki, Chuan Xiao, and Makoto Onizuka. 2022. An Empirical Study of Personalized Federated Learning. *arXiv preprint arXiv:2206.13190* (2022).

[44] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. *Artificial Intelligence and Statistics* (2017).

[45] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual Review of SocioLoGy* 27, 1 (2001), 415–444.

[46] Zhirui Pan, Guangzhong Wang, Zhaoning Li, Lifeng Chen, Yang Bian, and Zhongyuan Lai. 2022. 2SFGL: A Simple And Robust Protocol For Graph-Based Fraud Detection. In *IEEE International Conference on Cloud Computing TechnoLoGy and Science, CloudCom*. IEEE, 194–201.

[47] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations, ICLR*.

[48] Yuxin Qiu. 2023. Default Risk Assessment of Internet Financial Enterprises Based on Graph Neural Network. In *IEEE Information TechnoLoGy, Networking, Electronic and Automation Control Conference*, Vol. 6. IEEE, 592–596.

[49] Zongshuai Qu, Tao Yao, Xinghui Liu, and Gang Wang. 2023. A Graph Convolutional Network Based on Univariate Neurodegeneration Biomarker for Alzheimer's Disease Diagnosis. *IEEE Journal of Translational Engineering in Health and Medicine* (2023).

[50] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868* (2018).

[51] Chuxiong Sun, Hongming Gu, and Jie Hu. 2021. Scalable and adaptive graph neural networks with self-label-enhanced training. *arXiv preprint arXiv:2104.09376* (2021).

[52] Qiaoyu Tan, Xin Zhang, Ninghao Liu, Daochen Zha, Li Li, Rui Chen, Soo-Hyun Choi, and Xia Hu. 2023. Bring your own view: Graph neural networks for link prediction with personalized subgraph selection. In *Proceedings of the ACM International Conference on Web Search and Data Mining, WSDM.*

[53] Yue Tan, Yixin Liu, Guodong Long, Jing Jiang, Qinghua Lu, and Chengqi Zhang. 2023. Federated learning on non-iid graphs via structural knowledge sharing. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI.*

[54] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. 2022. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI.*

[55] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations, ICLR.*

[56] Nikil Wale, Ian A Watson, and George Karypis. 2008. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems* 14 (2008), 347–375.

[57] Guancheng Wan, Wenke Huang, and Mang Ye. 2024. Federated Graph Learning under Domain Shift with Generalizable Prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI.*

[58] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies* 1, 1 (2020), 396–413.

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

[60] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2022. Communication-efficient federated learning via knowledge distillation. *Nature communications* 13, 1 (2022), 2032.

[61] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International Conference on Machine Learning, ICML.*

[62] Xiaotong Wu, Jiaquan Gao, Muhammad Bilal, Fei Dai, Xiaolong Xu, Lianyong Qi, and Wanchun Dou. 2023. Federated learning-based private medical knowledge graph for epidemic surveillance in internet of things. *Expert Systems* (2023), e13372.

[63] Zhanghao Wu, Paras Jain, Matthew Wright, Azalia Mirhoseini, Joseph E Gonzalez, and Ion Stoica. 2021. Representing long-range context for graph neural networks with global attention. *Advances in Neural Information Processing Systems, NeurIPS* (2021).

[64] Han Xie, Jing Ma, Li Xiong, and Carl Yang. 2021. Federated graph classification over non-iid graphs. *Advances in Neural Information Processing Systems, NeurIPS* (2021).

[65] Han Xie, Li Xiong, and Carl Yang. 2023. Federated node classification over graphs with latent link-type heterogeneity. In *Proceedings of the ACM Web Conference, WWW.*

[66] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks? (2019).

[67] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning, ICML.*

[68] Pinar Yanardag and SVN Vishwanathan. 2015. Deep graph kernels. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD.*

[69] Liangwei Yang, Shengjie Wang, Yunzhe Tao, Jiankai Sun, Xiaolong Liu, Philip S Yu, and Taiqing Wang. 2023. DGRec: Graph Neural Network for Recommendation with Diversified Embedding Generation. In *Proceedings of the ACM International Conference on Web Search and Data Mining, WSDM.*

[70] Mingqi Yang, Yanming Shen, Rui Li, Heng Qi, Qiang Zhang, and Baocai Yin. 2022. A new perspective on the effects of spectrum in graph neural networks. In *International Conference on Machine Learning, ICML.*

[71] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In *International Conference on Machine Learning, ICML.*

[72] Yuhang Yao, Weizhao Jin, Srivatsan Ravi, and Carlee Joe-Wong. 2023. FedGCN: Convergence-communication tradeoffs in federated training of graph convolutional networks. *Advances in neural information processing systems* 36 (2023).

[73] Yuyu Yin, Youhuizi Li, Honghao Gao, Tingting Liang, and Qianqian Pan. 2022. FGC: GCN based federated learning approach for trust industrial service recommendation. *IEEE Transactions on Industrial Informatics* (2022).

[74] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. 2020. Graphsaint: Graph sampling based inductive learning method. In *International conference on learning representations, ICLR.*

[75] Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. 2024. FedTGP: Trainable Global Prototypes with Adaptive-Margin-Enhanced Contrastive Learning for Data and Model Heterogeneity in Federated Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI.*

[76] Ke Zhang, Lichao Sun, Bolin Ding, Siu Ming Yiu, and Carl Yang. 2024. Deep efficient private neighbor generation for subgraph federated learning. In *Proceedings of SIAM International Conference on Data Mining, SDM.*

[77] Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. 2021. Subgraph federated learning with missing neighbor generation. *Advances in Neural Information Processing Systems, NeurIPS* (2021).

[78] Wentao Zhang, Yu Shen, Zheyu Lin, Yang Li, Xiaosen Li, Wen Ouyang, Yangyu Tao, Zhi Yang, and Bin Cui. 2022. PaSca: A Graph Neural Architecture Search System under the Scalable Paradigm. In *Proceedings of the ACM Web Conference, WWW.*

[79] Wentao Zhang, Ziqi Yin, Zeang Sheng, Yang Li, Wen Ouyang, Xiaosen Li, Yangyu Tao, Zhi Yang, and Bin Cui. 2022. Graph Attention Multi-Layer Perceptron. *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD* (2022).

[80] Zhen Zhang, Jiajun Bu, Martin Ester, Jianfeng Zhang, Chengwei Yao, Zhi Yu, and Can Wang. 2019. Hierarchical graph pooling with structure learning. *arXiv preprint arXiv:1911.05954* (2019).

[81] Zaixi Zhang, Qingyong Hu, Yang Yu, Weibo Gao, and Qi Liu. 2024. FedGT: Federated Node Classification with Scalable Graph Transformer. *arXiv preprint arXiv:2401.15203* (2024).

[82] Geng Zhao, Yi Huang, and Chi Hsu Tsai. 2022. Fedgsl: Federated graph structure learning for local subgraph augmentation. In *IEEE International Conference on Big Data.* IEEE.

[83] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. 2021. Federated learning on non-IID data: A survey. *Neurocomputing* 465 (2021), 371–390.

[84] Yinlin Zhu, Xunkai Li, Zhengyu Wu, Di Wu, Miao Hu, and Rong-Hua Li. 2024. FedTAD: Topology-aware Data-free Knowledge Distillation for Subgraph Federated Learning. *arXiv preprint arXiv:2404.14061* (2024).

[85] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. 2021. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning, ICML.* PMLR.