FedGOG: Federated Graph Out-of-Distribution Generalization with Diffusion Data Exploration and Latent Embedding Decorrelation

Pengyang Zhou[1], Chaochao Chen[1], Weiming Liu[1], Xinting Liao[1], Wenkai Shen[2], Jiahe Xu[1], Zhihui Fu[3], Jun Wang[3], Wu Wen[1], Xiaolin Zheng[1]*

[1]Zhejiang University
[2]Northwestern Polytechnical University
[3]OPPO Research Institute

{zhoupy,zjuccc,21831010,xintingliao}@zju.edu.cn, shenwenkai@mail.nwpu.edu.cn, jiahexu@zju.edu.cn, luca@oppo.com, junwang.lu@gmail.com, wuwen@intl.zju.edu.cn, xlzheng@zju.edu.cn

Abstract

Federated graph learning (FGL) has emerged as a promising approach to enable collaborative training of graph models while preserving data privacy. However, current FGL methods overlook the out-of-distribution (OOD) shifts that occur in real-world scenarios. The distribution shifts between training and testing datasets in each client impact the FGL performance. To address this issue, we propose federated graph OOD generalization framework `FedGOG`, which includes two modules, i.e., diffusion data exploration (DDE) and latent embedding decorrelation (LED). In DDE, all clients jointly train score models to accurately estimate the global graph data distribution and sufficiently explore sample space using score-based graph diffusion with conditional generation. In LED, each client models a global invariant GNN and a personalized spurious GNN. LED aims to decorrelate spuriousness from invariant relationships by minimizing the mutual information between two categories of latent embeddings from different GNN models. Extensive experiments on six benchmark datasets demonstrate the superiority of `FedGOG`.

1 Introduction

Graph structured data is widely studied in various domains such as medical diagnosis (Kim et al. 2023), drug discovery (Askr et al. 2023) and social networks (Liu et al. 2024b). In the real world, graph data is often distributed across multiple sources, making centralized training challenging due to privacy concerns (Voigt and Von dem Bussche 2017). Federated graph learning (FGL) (Fu et al. 2022; Liu et al. 2024a) has emerged as a promising approach to enable collaborative learning without leaking data privacy.

However, current FGL methods overlook the out-of-distribution (OOD) shifts (Gui et al. 2022) in the practical deployment. In the literature of graph OOD generalization, it is typically assumed that a graph includes class-invariant information that determines the graph property. Meanwhile, the graph also contains spuriousness influenced by environments, which are causally irrelevant to the class labels. As shown in Fig. 1, the spuriousness varies between training and testing datasets in each client, leading to OOD shifts that occur in both features and structures. These shifts reflect

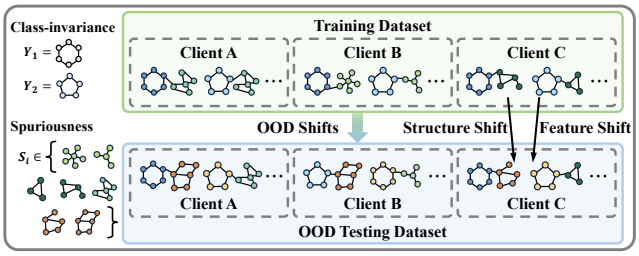


Figure 1: An illustration of the out-of-distribution shifts in the federated graph learning.

spurious correlations between graphs and environments, resulting in unstable predictions and degraded FGL performance. Existing federated learning methods on graph classification mainly focus on handling training graphs with non-independent and identical distributions (non-IID) through strategies such as client clustering (Xie et al. 2021) or structural knowledge sharing (Tan et al. 2023). How to tackle the OOD generalization problem in FGL remains a significant but unresolved challenge.

Although recent efforts attempt to improve the graph OOD generalization capability in centralized learning, they are not directly applicable to the FGL scenarios. *Firstly*, several methods adopt the graph data augmentation strategies to enrich the data distribution during training, e.g., applying intervention to diversify the training environments (Wu et al. 2022b; Sui et al. 2024; Liu et al. 2022; Yu, Liang, and He 2023), or using mixup operation for IN-distribution graph data (Lu et al. 2024; Jia et al. 2024). However, the available graph data in each client is typically limited and biased, making it challenging to effectively explore the OOD graph sample space and generalize to new and unseen environments. *Secondly*, a series of studies focus on extracting label-related invariant information from the graph data, which involves invariant subgraph extraction (Li et al. 2022; Chen et al. 2022, 2024a; Gui et al. 2023), structural information bottleneck (Yang et al. 2023) or reweighting strategies (Fan et al. 2023; Chen et al. 2024b). Nevertheless, the heterogeneity of graph data across different clients complicates the extraction of stable and invariant information.

Different from existing work, we highlight two unre-

*Corresponding author.

solved challenges in the FGL OOD generalization problem. **CH1**: *How to explore the OOD sample space with limited and biased client graph data?* The insufficient client graph data results in a constrained IN-distribution sample space, making it difficult to capture diverse OOD shifts. To enhance the generalization capability of the GNN model, it is essential to explore a broader OOD sample space. This requires leveraging the graph distribution across all clients and extending exploration beyond the original local training data. **CH2**: *How to extract invariant relationships across heterogeneous client environments?* Due to the heterogeneous data distribution among clients, each client extracts different and inconsistent relationships between the input graph and the class labels. Additionally, the explored graph samples outside the original training space introduce new distributions, further increasing the difficulty of eliminating spurious information unrelated to class labels in FGL scenarios.

To fill this gap, we propose a federated graph OOD generalization framework, `FedGOG`. To address **CH1**, we devise **diffusion data exploration (DDE)**. Specifically, all clients collaboratively train score models that estimate the distribution of graph data. Through local training and global aggregation of the score models, DDE obtains a comprehensive global graph distribution. DDE then generates OOD graph samples with the knowledge of the global distribution. This explores OOD data sample space beyond the limited and biased training distribution in each client. To address **CH2**, we propose **latent embedding decorrelation (LED)**. Each client maintains a global invariant GNN for predicting class labels and a personalized GNN for inferring spurious information. LED performs decorrelation by minimizing the mutual information between the latent embeddings produced by the two GNNs. This decorrelation process is used for both the original training graph data and the newly generated samples, eliminating the impacts of the spurious correlations and enhancing the extraction of invariant relationships.

We summarize our contributions as follows: (1) We are the first to investigate the OOD problem in FGL, providing a formal definition and scenario analysis. (2) We propose a framework `FedGOG` consisting of DDE and LED. DDE performs data augmentation using graph diffusion to explore the OOD sample space, and LED decorrelates spurious information to extract invariant relationships, thus enhancing the generalization of FGL. (3) We conduct experiments on six datasets and prove the effectiveness of `FedGOG`.

2 Related Work

2.1 Federated Graph Learning

FGL is a distributed learning paradigm in which multiple parties collaboratively train graph models while keeping their individual graph data private. Existing FGL work attempts to deal with three aspects of graph tasks, including node-level, subgraph-level and graph-level learning. Node-level research mainly applies to on-device recommendation scenarios that involve only first-order graphs, where each client is a user with their own purchase information (Wu et al. 2022a; Mao et al. 2024; Luo et al. 2024). Subgraph-level studies consider the scenarios that each client holds a portion of the overall graph, addressing the issue of cross-client missing information (Zhang et al. 2021; Liu et al. 2023; Baek et al. 2023). In this work, we focus on the graph-level task where each client owns a set of distinct graph samples, e.g., molecular graphs. Current efforts mainly consider the non-IID issue (Liao et al. 2023b,a) across client graph datasets. GCFL (Xie et al. 2021) dynamically clusters clients based on the gradients of GNNs and performs the aggregation process in each cluster to alleviate the influence of client heterogeneity. FedStar (Tan et al. 2023) designs GNNs in a feature-structure decoupled manner and shares the structure encoder across clients. However, they overlook the OOD shifts between the training and testing datasets in real-world deployment, which degrades the performance of GNN and results in unstable predictions.

2.2 Graph Out-of-Distribution Generalization

The OOD generalization capability of GNNs has attracted growing attention. Existing approaches can be broadly categorized into two groups. Firstly, several studies focus on data augmentation techniques. These methods typically involve intervening or recombining training environments to generate new samples (Wu et al. 2022b; Sui et al. 2024; Liu et al. 2022; Yu, Liang, and He 2023). Some approaches also employ mixup strategies, either in the sample space or representation space, to create hybrid combinations of data (Lu et al. 2024; Jia et al. 2024). Secondly, a series of studies aim to extract label-related invariant information from graph data. From the perspectives of invariant learning and stable learning, several works try to identify subgraphs that are informative substructures of the entire graph and use these subgraphs to predict class labels (Li et al. 2022; Chen et al. 2022, 2024a). Another line of research seeks to eliminate spurious correlations in variable representations through reweighting strategies (Fan et al. 2023; Chen et al. 2024b). Recent attempts adopt structural graph information bottleneck to capture latent invariant relationships between data samples and class labels (Yang et al. 2023). However, the above two categories of centralized graph OOD methods cannot be directly applied to FGL. The augmentation will be hindered due to modeling limited and biased client data, meanwhile, capturing invariance among heterogeneous clients is still non-trivial.

3 Method

3.1 Problem Statement

A graph is denoted as $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{A}) \in \mathcal{G}$, where $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ is the node feature matrix and $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the adjacency matrix, with n being the number of nodes and d the dimension of features. Based on (Fan et al. 2023), each graph sample is composed of class-invariant information and environment-spurious information. Class-invariant information is associated with the class label y_c, which represents inherent properties of the graph, such as the chemical properties of molecular graphs. Environment-spurious information is not related to graph class, instead, their spurious labels are determined by various environments, e.g., the scaffold structure or the size of the graph.

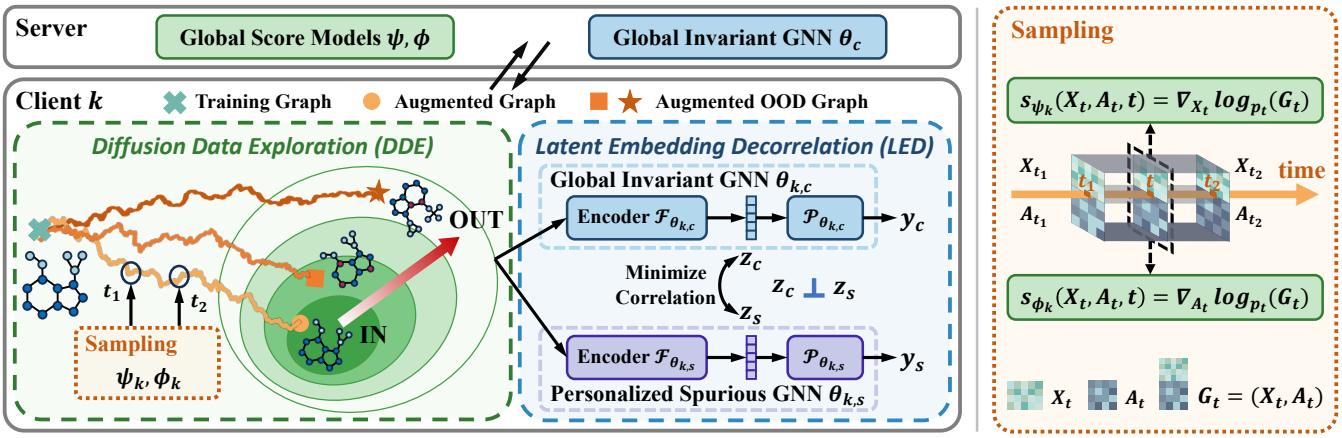


Figure 2: The left part depicts the main framework of FedGOG, the right part elaborates on the details of sampling in diffusion data exploration. IN indicates the IN-distribution graph space, OUT indicates the OOD shifts graph space.

In a federated learning system designed for graph-level tasks, we consider a setup involving K clients and a central server. Each client k possesses a training dataset $\mathcal{D}_k$ which consists of N_k graph samples. The data distribution among clients is heterogeneous, i.e., $p_k(\boldsymbol{G}, y_c) \neq p_j(\boldsymbol{G}, y_c)$ for different clients k and j. During the deployment of FedGOG, it suffers from tackling the testing dataset with OOD shifts, i.e., $p_{\text{te}}(\boldsymbol{G}, y_c) \neq p_k(\boldsymbol{G}, y_c)\ \forall k \in [K]$. The distribution shift is primarily reflected among spurious environments (Krueger et al. 2021), $p_k(y_s) \neq p_{\text{te}}(y_s)$, where the sets of the training environments $\mathcal{E}_{\text{tr}}$ and the testing environments $\mathcal{E}_{\text{te}}$ differ from each other. We aim to jointly train a global invariant GNN model with parameter $\boldsymbol{\theta}_c$ through the collaboration among clients to infer class labels on the testing dataset. The overall objective is formulated as:

$$\min_{\boldsymbol{\theta}_c} \sup_{e \in \mathcal{E}_{\text{te}}} \mathbb{E}^e_{(\boldsymbol{G}, y_c) \sim \mathcal{D}_{\text{te}}} [\mathcal{L}(\boldsymbol{\theta}_c; \boldsymbol{G}, y_c)], \tag{1}$$

where $\mathcal{L}$ is the loss function. The overall objective is to accurately predict class labels on seen graph spurious environments as well as effectively generalize to unseen graph environments during the testing period.

3.2 Framework Overview

The overall framework of FedGOG is depicted in Fig. 2, illustrating the interactions between the server and the k-th client. FedGOG consists of two modules, i.e., diffusion data exploration (DDE) and latent embedding decorrelation (LED). Firstly, in DDE, clients locally train two score models s_ψ and s_ϕ to estimate the distribution of node features $\boldsymbol{X}$ and graph structure $\boldsymbol{A}$, respectively. These score models are then aggregated in the server following the federated paradigm. The process iterates until convergence to accurately capture the global graph distribution across all clients. Then each client employs the diffusion sampling process with score models to generate new samples, utilizing controllable OOD conditions to explore samples that differ from the training distribution. Secondly, in LED, both original graph and newly generated graph samples of each client will be represented in two embedding space using the global invariant encoder $\mathcal{F}_{\boldsymbol{\theta}_{k,c}}$ and the personalized spurious encoder $\mathcal{F}_{\boldsymbol{\theta}_{k,s}}$. We minimize the mutual information of these embeddings to mitigate the correlations between the global invariance and the environmental spuriousness. Thus FedGOG can infer the class labels based on invariant graph representation, bringing more stable and accurate predictions. The clients then upload the parameters of the invariant GNN, ensuring consistent global extraction of invariant information. This training process iterates until the performance of FedGOG converges.

3.3 Diffusion Data Exploration

Motivation. To enhance the graph OOD generalization capability, existing efforts often adopt data augmentation techniques. However, conventional augmentation methods will fail in FGL settings, since the available graph samples are limited and biased in each client. We utilize a controllable score-based diffusion process(Song et al. 2021; Jo, Lee, and Hwang 2022) to capture the global graph distribution across all clients, enabling the generation of augmented OOD data samples based on this global knowledge. Incorporating these OOD samples into client local datasets effectively extends the exploration beyond the existing In-distribution training sample space. We describe the details below.

Graph Diffusion Process. The generative approach using diffusion models involves two main processes: a forward diffusion process and a reverse denoising process. Formally, the forward diffusion process can be described as the sequence of random variables $\{\boldsymbol{G}_t = (\boldsymbol{X}_t, \boldsymbol{A}_t)\}_{t \in [0,T]}$ over a fixed time interval $[0, T]$, with $\boldsymbol{G}_0$ initially distributed according to client local distribution p_k. The dynamics of the forward process are governed by the following Itô stochastic differential equations (SDE):

$$\mathrm{d}\boldsymbol{G}_t = \boldsymbol{f}(\boldsymbol{G}_t, t)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}, \quad \boldsymbol{G}_0 \sim p_k, \tag{2}$$

where $\boldsymbol{f}(\cdot, t) : \mathcal{G} \to \mathcal{G}$ represents the linear drift coefficient, $g(t) : \mathbb{R} \to \mathbb{R}$ is the diffusion coefficient, and $\boldsymbol{w}$ is the standard Wiener process. The reverse of the diffusion process is

described by the following reverse-time SDE:

$$\mathrm{d}\boldsymbol{G}_t = \left[\boldsymbol{f}(\boldsymbol{G}_t, t) - g(t)^2 \nabla_{\boldsymbol{G}_t} \log p_{k,t}(\boldsymbol{G}_t)\right] \mathrm{d}t + g(t) \mathrm{d}\bar{\boldsymbol{w}}, \tag{3}$$

where $\bar{\boldsymbol{w}}$ is the reverse time Wiener processes and $p_{k,t}$ denotes the marginal distribution at time t for client k. However, directly computing graph diffusion using high-dimensional score model, i.e., $\nabla_{\boldsymbol{G}_t} \log p_{k,t}(\boldsymbol{G}_t) \in \mathbb{R}^{n \times d} \times \mathbb{R}^{n \times n}$, requires overwhelming computation costs. Motivated by (Jo, Lee, and Hwang 2022), we decompose Eq. (3) into an equivalent system of SDEs:

$$\begin{cases} \mathrm{d}\boldsymbol{X}_t = \left[\boldsymbol{f}_{\boldsymbol{X}}(\boldsymbol{X}_t, t) - g_{\boldsymbol{X}}(t)^2 \nabla_{\boldsymbol{X}_t} \log p_{k,t}(\boldsymbol{G}_t)\right] \mathrm{d}t + g_{\boldsymbol{X}}(t) \mathrm{d}\bar{\boldsymbol{w}}_{\boldsymbol{X}}, \\ \mathrm{d}\boldsymbol{A}_t = \left[\boldsymbol{f}_{\boldsymbol{A}}(\boldsymbol{A}_t, t) - g_{\boldsymbol{A}}(t)^2 \nabla_{\boldsymbol{A}_t} \log p_{k,t}(\boldsymbol{G}_t)\right] \mathrm{d}t + g_{\boldsymbol{A}}(t) \mathrm{d}\bar{\boldsymbol{w}}_{\boldsymbol{A}}, \end{cases} \tag{4}$$

with the separation of drift coefficient $\boldsymbol{f} = (\boldsymbol{f}_{\boldsymbol{X}}, \boldsymbol{f}_{\boldsymbol{A}})$ and diffusion coefficient $g = (g_{\boldsymbol{X}}, g_{\boldsymbol{A}})$. The joint log-density $\nabla_{\boldsymbol{X}_t} \log p_{k,t}(\boldsymbol{G}_t) \in \mathbb{R}^{n \times d}$ and $\nabla_{\boldsymbol{A}_t} \log p_{k,t}(\boldsymbol{G}_t) \in \mathbb{R}^{n \times n}$ are commonly referred to as the partial score functions. This separation brings two simpler graph diffusion processes, mitigating the computation overhead.

Joint Training of Score Models. Each client k attempts to train the time-dependent score models $s_{\boldsymbol{\psi}}(\boldsymbol{G}, t)$ and $s_{\boldsymbol{\phi}}(\boldsymbol{G}, t)$ using their own graph data to predict the partial score functions. However, since the graph data from each client has heterogeneous distributions, we use federated aggregation to combine the score models learned by individual clients. This approach seeks to obtain global score models that capture a more generalized distribution, reflecting a diverse range of environmental conditions spuriousness.

Federated learning has the capability to integrate global information from local data through model aggregation (Bonawitz et al. 2016; McMahan et al. 2017). The aggregated global score models obtain more comprehensive global information. During each communication round, clients independently train their local score models on their own graph datasets using the denoising score matching approach (Vincent 2011; Song and Ermon 2019):

$$\begin{aligned} &\min_{\boldsymbol{\psi}_k} \mathbb{E}_t \left\{ \mathbb{E}_* \left\| s_{\boldsymbol{\psi}_k}(\boldsymbol{G}_t, t) - \nabla_{\boldsymbol{X}_t} \log p_{k,0t}(\boldsymbol{X}_t | \boldsymbol{X}_0) \right\|_2^2 \right\}, \\ &\min_{\boldsymbol{\phi}_k} \mathbb{E}_t \left\{ \mathbb{E}_* \left\| s_{\boldsymbol{\phi}_k}(\boldsymbol{G}_t, t) - \nabla_{\boldsymbol{A}_t} \log p_{k,0t}(\boldsymbol{A}_t | \boldsymbol{A}_0) \right\|_2^2 \right\}. \end{aligned} \tag{5}$$

Here, t is uniformly sampled from $[0, T]$, and the expectation $\mathbb{E}_* = \mathbb{E}_{\boldsymbol{G}_0} \mathbb{E}_{\boldsymbol{G}_t | \boldsymbol{G}_0}$ is taken over the samples $\boldsymbol{G}_0 \sim p_k$ and $\boldsymbol{G}_t \sim p_{k,0t}(\boldsymbol{G}_t | \boldsymbol{G}_0)$ with $p_{k,0t}$ representing the transition distribution from $p_{k,0}$ to $p_{k,t}$. The expectations can be efficiently estimated using the Monte Carlo method (Rubinstein and Kroese 2016) with the samples $(t, \boldsymbol{G}_0, \boldsymbol{G}_t)$. Then, each client uploads their trained score models to the central server for aggregating global score models, i.e.,

$$\{\boldsymbol{\psi}, \boldsymbol{\phi}\} = \sum_{k=1}^{K} w_k \{\boldsymbol{\psi}_k, \boldsymbol{\phi}_k\}, \tag{6}$$

where $w_k = \frac{|\mathcal{D}_k|}{\sum_{k=1}^{K} |\mathcal{D}_k|}, \forall k \in [K]$. The aggregated score models are then distributed back to each client for subsequent training rounds. This collaborative approach enables the server to capture a comprehensive global graph distribution and further allows clients to leverage the global score models for improved sample space exploration. Notably, the aggregated score models carry only knowledge of the global distribution. It ensures the privacy of `FedGOG`, i.e., clients cannot access specific data details or infer the original graph data of others through the score models.

Controllable OOD Exploration. By leveraging the global score models, each client can diversify their training datasets by generating new samples through the graph diffusion sampling process. However, this process still tends to collapse into the space of existing training samples, limiting the effectiveness of data augmentation. To address this limitation, we propose incorporating controllable OOD conditions into the sampling process. This strategy obtains the generated OOD samples, thus broadening the range of sample space available to GNN models.

Inspired by (Lee, Jo, and Hwang 2023), we consider the conditional distribution $p_t(\boldsymbol{G}_t | \boldsymbol{\zeta} = \mu_1)$, where p_t is the global distribution among all the clients, $\boldsymbol{\zeta}$ denotes the degree of OOD shifts and $\mu_1 \in [0, 1)$ is a hyperparameter. We follow the conditional reverse-time SDE for sampling:

$$\mathrm{d}\boldsymbol{G}_t = \left[\boldsymbol{f}(\boldsymbol{G}_t, t) - g(t)^2 \nabla_{\boldsymbol{G}_t} \log p_t(\boldsymbol{G}_t | \boldsymbol{\zeta} = \mu_1)\right] \mathrm{d}t + g(t) \mathrm{d}\bar{\boldsymbol{w}}. \tag{7}$$

Here, the diffusion term is guided by the conditional score $\nabla_{\boldsymbol{G}_t} \log p_t(\boldsymbol{G}_t | \boldsymbol{\zeta} = \mu_1)$, which can further be computed as $\nabla_{\boldsymbol{G}_t} \log p_t(\boldsymbol{G}_t) + \nabla_{\boldsymbol{G}_t} \log p_t(\boldsymbol{\zeta} = \mu_1 | \boldsymbol{G}_t)$. Given that OOD samples are in the low-density probability regions, the distribution follows $p_t(\boldsymbol{\zeta} = \mu_1 | \boldsymbol{G}_t) \propto p_t(\boldsymbol{G}_t)^{-\mu_1}$. Thus, Eq. (7) can be reformulated as:

$$\begin{cases} \mathrm{d}\boldsymbol{X}_t = \left[\boldsymbol{f}_{\boldsymbol{X}}(\boldsymbol{X}_t, t) - (1 - \mu_1) g_{\boldsymbol{X}}(t)^2 s_{\boldsymbol{\psi}}(\boldsymbol{G}_t, t)\right] \mathrm{d}t + g_{\boldsymbol{X}}(t) \mathrm{d}\bar{\boldsymbol{w}}_{\boldsymbol{X}}, \\ \mathrm{d}\boldsymbol{A}_t = \left[\boldsymbol{f}_{\boldsymbol{A}}(\boldsymbol{A}_t, t) - (1 - \mu_1) g_{\boldsymbol{A}}(t)^2 s_{\boldsymbol{\phi}}(\boldsymbol{G}_t, t)\right] \mathrm{d}t + g_{\boldsymbol{A}}(t) \mathrm{d}\bar{\boldsymbol{w}}_{\boldsymbol{A}}. \end{cases} \tag{8}$$

Setting $\mu_1 = 0$ aligns the sampling process with the global distribution among clients, while increasing μ_1 progressively moves samples away from the training distribution.

To enhance the efficiency, we avoid starting the sampling process from random noise. Instead, each client randomly selects existing graph data from the training dataset and iterates the forward diffusion process for τ steps, following the reverse diffusion process with controllable OOD conditions formulated in Eq. (8). Here, τ is significantly smaller than the total diffusion steps T, reducing the computation burdens. The newly generated samples share the same class labels as the selected training samples. This practical implementation not only explores a wider but less unbiased OOD space for generalization, but also reduces the required number of sampling steps and improves efficiency.

3.4 Latent Embedding Decorrelation

Motivation. Graph OOD research typically considers invariant and stable learning principles that assign class labels to specific substructures within a graph. However, the distribution heterogeneity among clients prevents the consistent extraction of the class-invariant information. To mitigate it, `FedGOG` introduces a global invariant GNN for extracting embeddings related to graph class labels and a personalized GNN for capturing spurious information. By minimizing the mutual information between the representations extracted by

these two GNNs, we aim to decorrelate global invariant relationships from local spurious ones.

Spuriousness Decorrelation. Each client k trains global invariant GNN $\boldsymbol{\theta}_{k,c}$ and personalized spurious GNN $\boldsymbol{\theta}_{k,s}$ using their local graph datasets and newly generated augmented graphs with controllable OOD graph diffusion process. Each GNN consists of an embedding encoder $\mathcal{F}$ and a label predictor $\mathcal{P}$. For a batch of graphs $\boldsymbol{G}$, the encoders obtain $\boldsymbol{z}_c = \mathcal{F}_{\boldsymbol{\theta}_{k,c}}(\boldsymbol{G})$ and $\boldsymbol{z}_s = \mathcal{F}_{\boldsymbol{\theta}_{k,s}}(\boldsymbol{G})$. The predictors then attempt to infer the class labels $\hat{\boldsymbol{y}}_c = \mathcal{P}_{\boldsymbol{\theta}_{k,c}}(\boldsymbol{z}_c)$ and spurious labels $\hat{\boldsymbol{y}}_s = \mathcal{P}_{\boldsymbol{\theta}_{k,s}}(\boldsymbol{z}_s)$. The newly generated samples are assigned to a new class of spuriousness, since their distributions are different from the existing in-distribution training graph data. Typical cross-entropy loss is employed to train the GNNs:

$$\mathcal{L}_{\text{pred}} = \mathcal{L}_{\text{ce}}(\hat{\boldsymbol{y}}_c, \boldsymbol{y}_c) + \mathcal{L}_{\text{ce}}(\hat{\boldsymbol{y}}_s, \boldsymbol{y}_s). \tag{9}$$

From these two GNNs, we not only learn class-invariant information related to the graph properties but also identify spuriousness caused by the environments. However, learning the two tasks suffers from interference and entanglement, i.e., capturing the information related to the class labels will be inevitably impacted by the spuriousness. This hinders the effective use of invariant information for accurate prediction. Thus, we aim to conduct decorrelation at the embedding level, extracting invariance that determines the class labels while removing the influence of spuriousness. The mutual information between $\boldsymbol{z}_c$ and $\boldsymbol{z}_s$ is defined as:

$$I(\boldsymbol{z}_c; \boldsymbol{z}_s) = \mathbb{E}_{p(\boldsymbol{z}_c, \boldsymbol{z}_s)} \left[\log \frac{p(\boldsymbol{z}_c, \boldsymbol{z}_s)}{p(\boldsymbol{z}_c)p(\boldsymbol{z}_s)} \right], \tag{10}$$

which can also be represented as the KL divergence $D_{KL}\left(p\left(\boldsymbol{z}_c, \boldsymbol{z}_s\right) || p(\boldsymbol{z}_c)p(\boldsymbol{z}_s)\right)$ between the joint and the product of the marginal distributions. However, the probability densities of the latent embeddings are unknown, making direct computation of their mutual information intractable. Following (Belghazi et al. 2018), we adopt the Donsker-Varadhan dual representation of the KL divergence:

$$D_{KL}\left(p(\boldsymbol{z}_c) || p(\boldsymbol{z}_s)\right) = \sup_{\xi: \Omega \to \mathbb{R}} \mathbb{E}_{p(\boldsymbol{z}_c)}[\xi] - \log\left(\mathbb{E}_{p(\boldsymbol{z}_s)}[e^{\xi}]\right), \tag{11}$$

where the supremum is taken over all functions ξ in the function space Ω. Here, we use an MLP to estimate ξ, and the mutual information is calculated as the objective:

$$\mathcal{L}_{\text{MI}} = \max_{\xi} \left[\frac{1}{B} \sum_{i=1}^{B} \xi(\boldsymbol{z}_{c,i}, \boldsymbol{z}_{s,i}) - \log\left(\frac{1}{B} \sum_{i=1}^{B} e^{\xi(\boldsymbol{z}_{c,i}, \tilde{\boldsymbol{z}}_{s,i})} \right) \right], \tag{12}$$

where B is the batch size and $\tilde{\boldsymbol{z}}_{s,i}$ is obtained by randomly shuffling $\boldsymbol{z}_s$ within the batch. The full derivation is provided in Appendix B. In practice, ξ can be easily optimized using standard training procedures to maximize the objective function. With the help of estimating the mutual information between the invariant embeddings $\boldsymbol{z}_c$ and the spurious embeddings $\boldsymbol{z}_s$, we try to perform decorrelation by minimizing the mutual information. The overall local training process is then described by:

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \mu_2 \mathcal{L}_{\text{MI}}, \tag{13}$$

where μ_2 is the hyperparameter that controls the weight of decorrelation. Decorrelating the invariant and spurious embeddings facilitates the invariant GNN to focus more on aspects relevant to the class labels while mitigating the influence of spuriousness.

Federated Invariance Aggregation. After the local training process, each client uploads the parameters of their updated invariant GNN, while keeping the spurious GNN locally. The global invariant GNN parameters are aggregated via $\boldsymbol{\theta}_c = \sum_{k=1}^{K} w_k \boldsymbol{\theta}_{k,c}$. Due to variations in data distribution across clients, the locally retained spurious GNNs are capable of capturing spuriousness relevant to each environment. Concurrently, the aggregated invariant GNN across all clients effectively extracts invariance that has been decorrelated from spurious information, thus tackling the distribution heterogeneity across clients.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on six datasets following GOOD benchmark (Gui et al. 2022) and DrugOOD (Ji et al. 2023). These include two synthetic datasets Motif and CMNIST, where OOD shifts occur in structure on Motif, and features on CMNIST. Additionally, we evaluate four real-world graph datasets LBAPcore, HIV, Twitter and SST2 to investigate more intricate and challenging OOD scenarios, where shifts exist in both features and structure. We simulate the heterogeneous client training distribution using the train set in the original dataset, and the original OOD test set shared among all clients. In this study, we explore two scenarios of heterogeneity among client training distributions, i.e., class heterogeneity and spuriousness heterogeneity. In the class heterogeneity scenario, the distribution of class labels y_c follows a Dirichlet distribution, denoted as $p_{j,k} \sim \text{Dir}(\alpha)$, which describes the allocation of class j to client k. The spuriousness heterogeneity scenario also employs a Dirichlet distribution for the spurious labels y_s.

Baselines and Implementation Details. To evaluate the performance of `FedGOG`, we compare it with three categories of baselines: (1) standard FL methods: **FedAvg** (McMahan et al. 2017), **FedIIR** (Guo et al. 2023), (2) centralized graph OOD generalization methods adapted to the FL scenario: **Fed-DIR** (Wu et al. 2022b), **Fed-GIL** (Li et al. 2022), **Fed-LECI** (Gui et al. 2023), **Fed-AIA** (Sui et al. 2024), and (3) graph-level FGL methods: **GCFL** (Xie et al. 2021), **FedStar** (Tan et al. 2023). For all methods, we use the standard three-layer GINs (Xu et al. 2019) as the base graph neural networks. The embedding dimension is 128. All the training process uses an Adam optimizer with the learning rate set to 0.001. We conduct experiments with local epochs 5 and batch size 128 until convergence.

4.2 Performance Comparison

Class Heterogeneity Result. Tab. 1 shows the mean and standard deviation from five random seeds. We observe that: (1) **Client heterogeneity:** As client heterogeneity increases from $\alpha = 5.0$ to $\alpha = 0.1$, all methods suffer from

Datasets	Motif			CMNIST			HIV		
Methods \ α	0.1	0.5	5.0	0.1	0.5	5.0	0.1	0.5	5.0
FedAvg	44.80±0.49	51.71±0.50	55.15±1.50	20.40±0.46	31.07±0.90	31.06±0.53	61.32±0.71	61.36±1.05	63.59±0.99
FedIIR	47.63±0.52	52.30±1.15	53.04±1.69	21.62±0.39	30.85±0.47	30.65±0.92	63.05±0.47	62.10±1.25	64.54±0.35
Fed-DIR	52.33±5.70	52.53±4.60	56.18±2.62	19.08±1.84	22.95±0.81	19.62±2.24	55.22±1.57	59.65±3.30	55.89±1.93
Fed-GIL	50.86±5.25	52.13±3.35	57.50±2.00	25.82±1.07	31.12±2.44	27.29±2.20	51.42±2.46	51.63±0.67	52.15±1.54
Fed-LECI	49.65±1.69	62.93±0.86	62.74±0.74	35.11±1.51	45.10±3.48	48.24±5.41	56.71±1.15	59.67±0.14	62.74±0.92
Fed-AIA	53.06±2.00	64.25±0.57	68.44±2.88	19.28±2.57	31.81±0.65	34.86±0.06	57.62±2.85	57.81±3.06	60.91±0.64
GCFL	47.83±1.82	54.09±1.04	53.28±0.80	20.64±0.29	29.93±0.81	31.18±0.17	61.33±0.36	61.84±1.23	65.35±1.02
FedStar	53.09±0.41	64.35±1.59	71.46±1.31	15.66±0.28	22.50±0.36	23.07±1.07	56.97±0.65	57.58±0.45	59.06±0.89
FedGOG	**59.65±1.04**	**69.57±1.14**	**73.69±1.35**	**40.74±0.45**	**53.52±0.26**	**55.88±0.11**	**64.26±0.80**	**65.14±1.05**	**67.38±0.68**

Datasets	LBAPcore			Twitter			SST2		
Methods \ α	0.1	0.5	5.0	0.1	0.5	5.0	0.1	0.5	5.0
FedAvg	67.08±0.12	66.36±0.26	68.98±0.17	52.48±0.32	53.60±0.29	57.14±0.58	58.91±1.76	78.61±0.49	77.05±0.79
FedIIR	67.75±0.39	66.32±0.54	69.40±0.07	54.66±0.43	53.35±0.37	55.57±0.41	59.75±1.18	79.01±0.32	78.52±0.46
Fed-DIR	64.67±0.89	67.04±0.59	68.02±1.23	47.05±1.01	44.51±0.46	51.72±0.19	59.72±1.59	77.06±0.52	78.45±0.61
Fed-GIL	66.86±1.21	67.97±0.75	69.85±1.01	50.93±2.22	47.11±0.51	55.71±0.62	58.95±1.66	79.11±0.37	78.98±0.67
Fed-LECI	65.01±1.26	67.71±0.26	69.60±0.30	48.88±1.14	49.59±0.50	56.02±0.56	59.91±1.42	77.95±0.13	80.35±0.46
Fed-AIA	67.18±1.12	67.59±0.34	68.69±0.13	49.15±0.13	47.73±0.76	54.43±1.12	51.59±0.72	65.35±0.26	76.56±0.58
GCFL	67.24±0.53	66.37±0.21	68.96±0.28	53.57±0.39	52.55±0.46	54.53±0.28	59.98±2.17	78.21±0.36	78.24±0.36
FedStar	55.94±0.25	65.19±0.72	69.63±0.14	42.40±0.44	45.90±0.33	51.64±0.39	54.94±0.62	71.75±0.13	77.66±0.48
FedGOG	**70.66±0.39**	**71.75±0.15**	**72.88±0.22**	**57.98±0.53**	**58.76±0.61**	**59.89±0.30**	**64.77±1.72**	**79.45±1.09**	**81.13±0.96**

Table 1: Comparison of OOD shifts generalization in class heterogeneity scenarios.

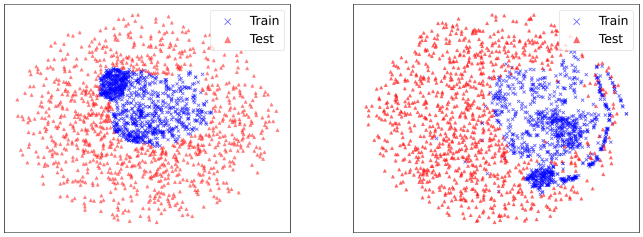
Datasets	Motif		CMNIST	
Methods \ α	0.1	5.0	0.1	5.0
FedAvg	42.57±0.75	58.38±2.34	26.69±1.41	30.50±0.73
FedIIR	42.90±0.53	59.78±2.03	25.62±0.76	31.85±1.71
Fed-DIR	46.96±2.33	57.50±8.16	14.38±0.27	24.68±3.49
Fed-GIL	44.63±0.75	49.84±2.84	17.73±1.31	28.41±2.56
Fed-LECI	48.13±0.59	65.58±1.55	48.46±3.12	49.19±4.53
Fed-AIA	50.67±0.67	61.87±4.14	28.21±0.72	34.62±2.00
GCFL	45.75±0.81	54.84±1.12	28.49±1.68	32.26±1.17
FedStar	64.20±3.27	69.42±3.93	21.69±2.01	22.91±0.41
FedGOG	**69.85±1.06**	**71.41±2.01**	**54.52±1.86**	**55.97±3.34**

Dataset	LBAPcore		SST2	
Methods \ α	0.1	5.0	0.1	5.0
FedAvg	67.36±0.07	69.30±0.23	80.08±0.19	80.13±0.49
FedIIR	67.07±0.41	69.12±0.04	79.55±0.12	79.89±0.25
Fed-DIR	66.56±1.44	67.77±0.93	78.64±0.33	79.95±1.17
Fed-GIL	69.70±0.62	69.87±0.49	82.01±0.38	81.32±0.09
Fed-LECI	68.00±0.06	69.70±0.17	81.69±0.26	81.52±0.15
Fed-AIA	68.93±0.24	70.06±0.24	74.57±0.90	76.84±0.56
GCFL	67.35±0.24	69.11±0.17	78.94±0.68	79.54±0.06
FedStar	67.80±0.11	69.50±0.06	74.28±0.42	78.67±0.23
FedGOG	**71.61±0.19**	**72.22±0.33**	**84.29±1.08**	**85.71±0.51**

Table 2: Comparison of OOD shifts generalization in spuriousness heterogeneity scenarios.

a decline in performance, indicating that greater distribution heterogeneity among clients makes it more challenging to extract invariant information, which emphasizes the importance of decorrelation strategies. (2) **Compare with standard FL methods:** The federated OOD generalization method FedIIR, applied previously in image classification, does not significantly outperform standard FedAvg in the FGL OOD scenario. The structural complexity of graph data prevents satisfactory results from simply constraining the gradient of the classifier to leverage implicit invariant relationships. (3) **Compare with centralized graph OOD methods:** We observe that these methods only show improvements in the simple Motif dataset, with limited or even worse performance in other graph datasets. For instance, the GIL method, which predicts class label by extracting invariant subgraphs, shows improvements in the more balanced LBAPcore $\alpha = 5.0$ scenario, but performs worse than FedAvg in LBAPcore $\alpha = 0.1$. This indicates the challenge of handling heterogeneous client data distributions in FGL OOD problem, which complicates reliable augmentation and consistent invariance extraction. LECI significantly improves CMNIST by leveraging the DANN (Ganin et al. 2016) concept from image domain adversarial training, which is effective due to the color shift in node features and minimal structural changes in CMNIST. However, LECI does not show consistent enhancements in other real-world graph datasets in FGL scenarios that exhibit both feature and structural shifts. (4) **Compare with federated graph learning methods:** We find neither GCFL nor FedStar can consistently perform well when confronted with OOD shifts in testing datasets. The only exception is the significant performance of FedStar in Motif, a synthetic dataset with clear structural patterns. This improvement is largely due to the specially designed structural embedding and additional structure encoder within the FedStar framework. However, FedStar does not maintain this advantage in more complex graph datasets, indicating limitations in its approach to handling diverse OOD challenges. (5) **Overall performance of FedGOG:** FedGOG consistently outperforms all other methods across various scenarios. In the particularly challenging CMNIST $\alpha = 0.1$ scenario, FedGOG achieves a significant 5.63% increase in accuracy. Notably, the model performance variance of FedGOG is practically controllable, demonstrating the stability and effectiveness in handling FGL OOD generalization problem.

Spuriousness Heterogeneity Result. We also conduct experiments under spuriousness heterogeneity scenarios in Tab. 2, where clients have different distributions of environmental spuriousness. This simulates situations where different clients may have collected their graph datasets with varying spurious correlations. We observe that in more heterogeneous scenarios $\alpha = 0.1$, all methods exhibit performance



(a) FedAvg (b) LECI

(c) FedStar (d) FedGOG

Figure 3: T-SNE visualization on CMNIST ($\alpha = 0.5$).

Datasets	HIV		CMNIST	
Methods \ α	0.1	5.0	0.1	5.0
FedAvg	61.32±0.71	63.59±0.99	20.40±0.46	31.06±0.53
FedGOG-w/o-DDE	63.98±0.67	66.27±0.52	37.41±1.03	52.17±0.21
FedGOG-w/o-LED	62.59±1.12	64.02±0.76	29.44±1.24	42.27±0.88
FedGOG	**64.26±0.80**	**67.38±0.68**	**40.74±0.45**	**55.88±0.11**

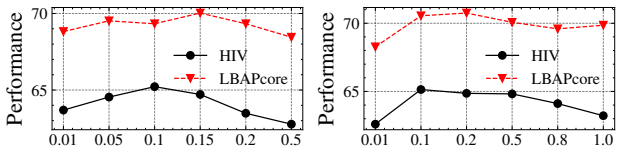
Table 3: Ablation study.

decreases compared to more uniform scenarios $\alpha = 5.0$. FedGOG demonstrates superior performances when compared to all baseline models in these experiments.

4.3 In-depth Analysis

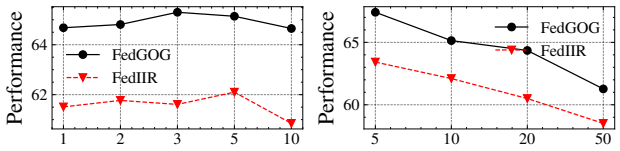
Visualization. We employ t-SNE to visualize training and testing data. The results in Fig. 3 depict a clear separation between training and testing data representations for both FedAvg and FedStar, indicating that they overlook the issue of graph OOD shifts. LECI shows slight improvements with partial overlap between the training and testing data, suggesting limited generalization capability. In contrast, FedGOG significantly increases the overlap between the training and testing datasets, which demonstrates the effectiveness of DDE to explore the space of OOD samples, filling the gap of the limited and biased client training distribution. Furthermore, using LED to reduce spurious correlations enhances the ability of FedGOG to extract invariant relationships, further boosting its generalization capability.

Ablation Study. To evaluate the effectiveness of each module in FedGOG, we devise the following two versions, i.e., FedGOG-w/o-DDE and FedGOG-w/o-LED. FedGOG-w/o-DDE highlights the benefits of exploring the OOD sample space beyond the training distribution, while FedGOG-w/o-LED demonstrates the impact of decorrelating spuriousness. The results in Tab. 3 show that each variation achieves significant improvements over the vanilla FedAvg. Notably, FedGOG-w/o-LED shows more performance drop, indicating that we should pay more attention to the decorrelation process of spuriousness. These findings suggest that the two modules are closely related and can function together to improve the graph OOD generalization capability.



(a) Weight of exploration μ_1 (b) Weight of decorrelation μ_2

Figure 4: Effect of hyperparameters.



(a) Local epochs E (b) Number of clients K

Figure 5: Effect of FGL settings on HIV ($\alpha = 0.5$).

Hyperparameters Sensitivity. To investigate the sensitivity of various hyperparameters, we conduct experiments on the HIV and LBAPcore under class heterogeneity scenario with $\alpha = 0.5$. We tune the weight of OOD exploration $\mu_1 = \{0.01, 0.05, 0.1, 0.2, 0.3, 0.5\}$, the weight of decorrelation $\mu_2 = \{0.01, 0.1, 0.2, 0.5, 0.8, 1.0\}$, local epochs $E = \{1, 2, 3, 5, 10\}$ and number of clients $K = \{5, 10, 20, 50\}$ in Fig. 4 and 5. We observe that: (1) Increasing μ_1 enhances OOD exploration, but a large μ_1 may cause samples to deviate far from the original distribution, failing to reflect inherent properties of the graph data. (2) Adjusting μ_2 also forms a bell curve, where initially increasing μ_2 reduces spuriousness, but excessive decorrelation may hinder learning from class labels. (3) Varying the number of local epochs slightly affects results, while increasing the number of clients significantly reduces performance. Notably, FedGOG consistently outperforms the best baseline across all settings, which verifies the effectiveness of enhancing data diversity through controllable graph diffusion and extracting invariant relationships by decorrelating spuriousness.

5 Conclusion

In this paper, we propose a federated graph OOD generalization framework FedGOG which contains diffusion data exploration (DDE) module and latent embedding decorrelation (LED) module. DDE utilizes graph diffusion model with OOD control to generate new samples, thereby extending the exploration of the OOD graph sample space. LED further decorrelates the embeddings produced by the global invariant GNN and the personalized spurious GNN, which helps extract invariant relationships and eliminates environmental spuriousness. Extensive experiments on both class heterogeneity and spuriousness heterogeneity scenarios demonstrate the effectiveness of FedGOG.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No.62172362).

References

Askr, H.; Elgeldawi, E.; Aboul Ella, H.; Elshaier, Y. A.; Gomaa, M. M.; and Hassanien, A. E. 2023. Deep learning in drug discovery: an integrative review and future challenges. *Artificial Intelligence Review*, 56(7): 5975–6037.

Baek, J.; Jeong, W.; Jin, J.; Yoon, J.; and Hwang, S. J. 2023. Personalized subgraph federated learning. In *International Conference on Machine Learning*, 1396–1415. PMLR.

Belghazi, M. I.; Baratin, A.; Rajeshwar, S.; Ozair, S.; Bengio, Y.; Courville, A.; and Hjelm, D. 2018. Mutual Information Neural Estimation. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 531–540. PMLR.

Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H. B.; Patel, S.; Ramage, D.; Segal, A.; and Seth, K. 2016. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*.

Chen, Y.; Bian, Y.; Zhou, K.; Xie, B.; Han, B.; and Cheng, J. 2024a. Does invariant graph learning via environment augmentation learn invariance? *Advances in Neural Information Processing Systems*, 36.

Chen, Y.; Zhang, Y.; Bian, Y.; Yang, H.; Kaili, M.; Xie, B.; Liu, T.; Han, B.; and Cheng, J. 2022. Learning causally invariant representations for out-of-distribution generalization on graphs. *Advances in Neural Information Processing Systems*, 35: 22131–22148.

Chen, Z.; Xiao, T.; Kuang, K.; Lv, Z.; Zhang, M.; Yang, J.; Lu, C.; Yang, H.; and Wu, F. 2024b. Learning to reweight for generalizable graph neural network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 8320–8328.

Fan, S.; Wang, X.; Shi, C.; Cui, P.; and Wang, B. 2023. Generalizing graph neural networks on out-of-distribution graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Fu, X.; Zhang, B.; Dong, Y.; Chen, C.; and Li, J. 2022. Federated graph machine learning: A survey of concepts, techniques, and applications. *ACM SIGKDD Explorations Newsletter*, 24(2): 32–47.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; March, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59): 1–35.

Gui, S.; Li, X.; Wang, L.; and Ji, S. 2022. Good: A graph out-of-distribution benchmark. *Advances in Neural Information Processing Systems*, 35: 2059–2073.

Gui, S.; Liu, M.; Li, X.; Luo, Y.; and Ji, S. 2023. Joint Learning of Label and Environment Causal Independence for Graph Out-of-Distribution Generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Guo, Y.; Guo, K.; Cao, X.; Wu, T.; and Chang, Y. 2023. Out-of-distribution generalization of federated learning via implicit invariant relationships. In *International Conference on Machine Learning*, 11905–11933. PMLR.

Ji, Y.; Zhang, L.; Wu, J.; Wu, B.; Li, L.; Huang, L.-K.; Xu, T.; Rong, Y.; Ren, J.; Xue, D.; et al. 2023. Drugood: Out-of-distribution dataset curator and benchmark for ai-aided drug discovery–a focus on affinity prediction problems with noise annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 8023–8031.

Jia, T.; Li, H.; Yang, C.; Tao, T.; and Shi, C. 2024. Graph invariant learning with subgraph co-mixup for out-of-distribution generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 8562–8570.

Jo, J.; Lee, S.; and Hwang, S. J. 2022. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International conference on machine learning*, 10362–10383. PMLR.

Kim, S.; Lee, N.; Lee, J.; Hyun, D.; and Park, C. 2023. Heterogeneous graph learning for multi-modal medical data analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 5141–5150.

Krueger, D.; Caballero, E.; Jacobsen, J.-H.; Zhang, A.; Binas, J.; Zhang, D.; Le Priol, R.; and Courville, A. 2021. Out-of-distribution generalization via risk extrapolation (rex). In *International conference on machine learning*, 5815–5826. PMLR.

Lee, S.; Jo, J.; and Hwang, S. J. 2023. Exploring chemical space with score-based out-of-distribution generation. In *International Conference on Machine Learning*, 18872–18892. PMLR.

Li, H.; Zhang, Z.; Wang, X.; and Zhu, W. 2022. Learning invariant graph representations for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 35: 11828–11841.

Liao, X.; Chen, C.; Liu, W.; Zhou, P.; Zhu, H.; Shen, S.; Wang, W.; Hu, M.; Tan, Y.; and Zheng, X. 2023a. Joint local relational augmentation and global nash equilibrium for federated learning with non-iid data. In *Proceedings of the 31st ACM International Conference on Multimedia*, 1536–1545.

Liao, X.; Liu, W.; Chen, C.; Zhou, P.; Zhu, H.; Tan, Y.; Wang, J.; and Qi, Y. 2023b. HyperFed: Hyperbolic Prototypes Exploration with Consistent Aggregation for Non-IID Data in Federated Learning. In Elkind, E., ed., *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 3957–3965. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Liu, G.; Zhao, T.; Xu, J.; Luo, T.; and Jiang, M. 2022. Graph rationalization with environment-based augmentations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1069–1078.

Liu, R.; Xing, P.; Deng, Z.; Li, A.; Guan, C.; and Yu, H. 2024a. Federated graph neural networks: Overview, techniques, and challenges. *IEEE Transactions on Neural Networks and Learning Systems*.

Liu, W.; Chen, C.; Liao, X.; Hu, M.; Tan, Y.; Wang, F.; Zheng, X.; and Ong, Y. S. 2024b. Learning Accurate and Bidirectional Transformation via Dynamic Embedding Transportation for Cross-Domain Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 8815–8823.

Liu, W.; Chen, C.; Liao, X.; Hu, M.; Yin, J.; Tan, Y.; and Zheng, L. 2023. Federated Probabilistic Preference Distribution Modelling with Compactness Co-Clustering for Privacy-Preserving Multi-Domain Recommendation. In *IJCAI*, 2206–2214.

Lu, B.; Zhao, Z.; Gan, X.; Liang, S.; Fu, L.; Wang, X.; and Zhou, C. 2024. Graph out-of-distribution generalization with controllable data augmentation. *IEEE Transactions on Knowledge and Data Engineering*.

Luo, S.; Xiao, Y.; Zhang, X.; Liu, Y.; Ding, W.; and Song, L. 2024. Perfedrec++: Enhancing personalized federated recommendation with self-supervised pre-training. *ACM Transactions on Intelligent Systems and Technology*, 15(5): 1–24.

Mao, X.; Liu, Y.; Qi, L.; Duan, L.; Xu, X.; Zhang, X.; Dou, W.; Beheshti, A.; and Zhou, X. 2024. Cluster-driven Personalized Federated Recommendation with Interest-aware Graph Convolution Network for Multimedia. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 5614–5622.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Rubinstein, R. Y.; and Kroese, D. P. 2016. *Simulation and the Monte Carlo method*. John Wiley & Sons.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

Sui, Y.; Wu, Q.; Wu, J.; Cui, Q.; Li, L.; Zhou, J.; Wang, X.; and He, X. 2024. Unleashing the power of graph data augmentation on covariate distribution shift. *Advances in Neural Information Processing Systems*, 36.

Tan, Y.; Liu, Y.; Long, G.; Jiang, J.; Lu, Q.; and Zhang, C. 2023. Federated Learning on Non-IID Graphs via Structural Knowledge Sharing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8): 9953–9961.

Vincent, P. 2011. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7): 1661–1674.

Voigt, P.; and Von dem Bussche, A. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676): 10–5555.

Wu, C.; Wu, F.; Lyu, L.; Qi, T.; Huang, Y.; and Xie, X. 2022a. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications*, 13(1): 3091.

Wu, Y.; Wang, X.; Zhang, A.; He, X.; and Chua, T.-S. 2022b. Discovering Invariant Rationales for Graph Neural Networks. In *International Conference on Learning Representations*.

Xie, H.; Ma, J.; Xiong, L.; and Yang, C. 2021. Federated Graph Classification over Non-IID Graphs. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 18839–18852. Curran Associates, Inc.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Yang, L.; Zheng, J.; Wang, H.; Liu, Z.; Huang, Z.; Hong, S.; Zhang, W.; and Cui, B. 2023. Individual and structural graph information bottlenecks for out-of-distribution generalization. *IEEE Transactions on Knowledge and Data Engineering*.

Yu, J.; Liang, J.; and He, R. 2023. Mind the label shift of augmentation-based graph ood generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11620–11630.

Zhang, K.; Yang, C.; Li, X.; Sun, L.; and Yiu, S. M. 2021. Subgraph federated learning with missing neighbor generation. *Advances in Neural Information Processing Systems*, 34: 6671–6682.