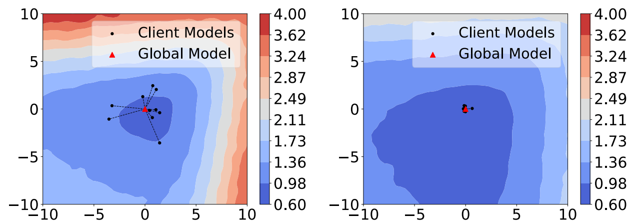
Prior Refinement is Better: Diffusion-Driven Graph Harmonization for Federated Graph Learning

Shuman Zhuang[1], Zhihao Wu[2], Wei Huang[1], Luojun Lin[1], Jia-Li Yin[1*], Lele Fu[3*], Hong-Ning Dai[4]

[1]College of Computer and Data Science, Fuzhou University, Fuzhou, China
[2]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[3]School of Systems Science and Engineering, Sun Yat-sen University, Guangzhou, China
[4]Department of Computer Science, Hong Kong Baptist University, Hong Kong, China
shumanzhuang@163.com, zhihaowu1999@gmail.com, {huangweifujian, linluojun2009}@126.com,
fulle@mail2.sysu.edu.cn, jlyin@fzu.edu.cn, henrydai@comp.hkbu.edu.hk

Abstract

Federated Graph Learning (FGL) has emerged as a compelling paradigm for collaboratively training a global model while preserving the privacy of multi-source graphs. Nonetheless, FGL faces a critical challenge of data heterogeneity, where semantic and structural discrepancies across clients significantly degrade its performance. Although existing methods attempt to calibrate client-specific graph distributions during federated training, they inevitably fall short in aligning the optimization behaviors across clients due to dynamic parameter updates, thereby inducing a bottleneck in generalization improvement. To tackle this challenge, we propose a solution from a new perspective of prior refinement, which seeks to proactively harmonize client graph distributions before the federated training. In particular, we propose a Federated Graph Harmonization (FedGH) framework that exploits the generative strengths of graph diffusion models to perform prior refinement of local graphs. In a nutshell, FedGH designs a conditional diffusion mechanism on each client that synthesizes pseudo-graphs encapsulating both feature and structural priors, thereby facilitating explicit correction of inter-client distributional bias. On the server side, we employ the graph contrastive learning between various client-specific pseudo-graphs to incorporate the global information, subsequently guiding local data reconstruction. Importantly, model-agnostic FedGH can be seamlessly deployed as a plug-and-play module to be easily integrated with existing FGL architectures. Extensive experiments demonstrate that FedGH consistently outperforms state-of-the-art FGL baselines.



(a) FedAvg (b) The proposed FedGH

Figure 1: Loss landscapes w.r.t. parameters on Cora using ten clients. The red triangle denotes the global model initialized at the origin, while the black dots represent the relative positions of client models in the parameter space.

Introduction

Graph learning is increasingly recognized for its ability to extract insights from pervasive relational data found in modern systems (Zhuang et al. 2025a; Tu et al. 2025; Wang et al. 2026). Driven by the rapid development of advanced techniques like Graph Neural Networks (GNNs) (Wu, Zhang, and Fan 2023; Wang et al. 2024; Zhuang et al. 2024), this field has witnessed breakthroughs in many applications (Wu et al. 2020, 2023; Lu et al. 2025b), including recommender systems (Jin et al. 2023), fraud detection (Liu et al. 2024; Cai et al. 2025), and bioinformatics (Lu et al. 2025a; Wu et al. 2025). However, the practical deployment of existing workflows is still hindered by many challenges (Zhuang et al. 2025b). Among these, a critical bottleneck is the reliance on centralized training, which is fundamentally incompatible with the growing demand for privacy protection and the distributed nature of graph data (Wan et al. 2025; Huang et al. 2025). Federated Learning (FL) (McMahan et al. 2017; Qi et al. 2023; Fu et al. 2025b) presents a viable distributed paradigm to overcome these barriers. Inspired by this, Federated Graph Learning (FGL) has recently gained attention as a promising framework (Wan, Huang, and Ye 2024; Fu et al. 2025c), designed to handle the unique challenges of learning from decentralized and privacy-sensitive graph data.

While data heterogeneity is a notorious challenge in general FL, it becomes exacerbated in the context of FGL due to the complex nature of graph data. Specifically, FGL heterogeneity is twofold: 1) semantic heterogeneity involving disparities in feature and label distributions, and 2) topological heterogeneity reflecting variations in local structures. Recent advances seek to mitigate such heterogeneity by calibrating distributional discrepancies across clients during federated training. A prominent line of studies improves local

*Corresponding Authors.

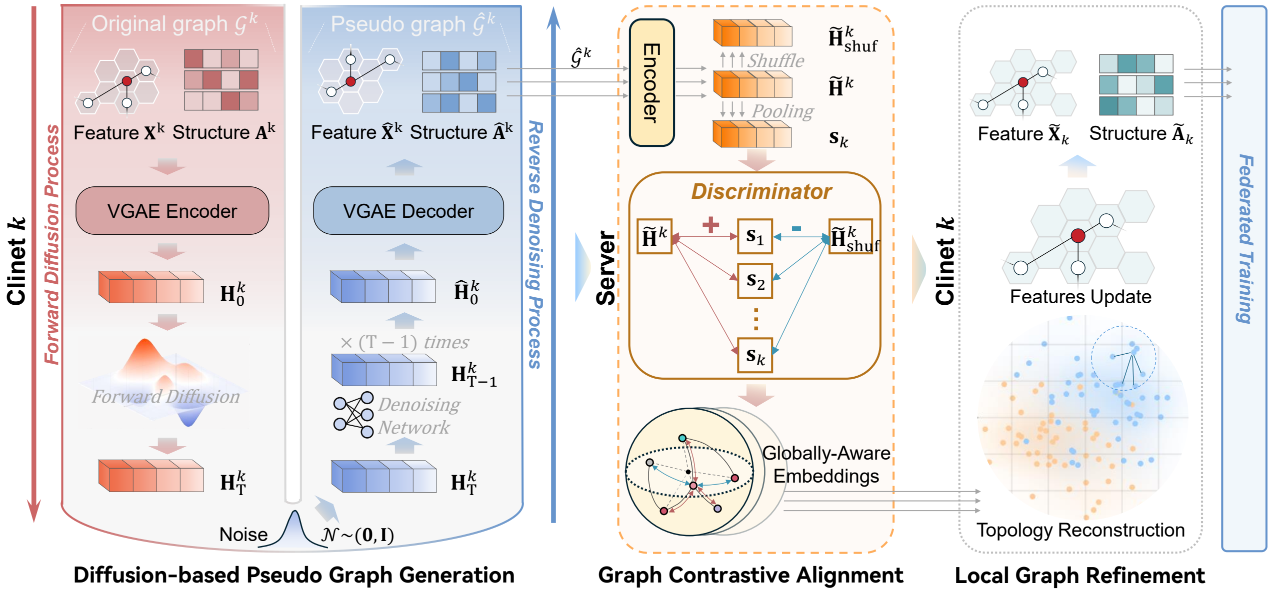


Figure 2: Overview of the proposed FedGH. Each client generates pseudo-graphs using the generative capacity of graph diffusion and uploads them to the server, where graph contrastive learning is performed to obtain globally-aware embeddings. These embeddings are then transmitted back to each client to guide the refinement of local graph data.

model adaptability or develops a more generalizable global model by leveraging knowledge distillation (Huang et al. 2023; Zhu et al. 2024), meta-learning (Wang et al. 2022), and Mixup-based data blending (Zhang et al. 2023) to guide client-specific optimization. In parallel, another line of endeavors focuses on improving the aggregation process itself to extract invariant knowledge, using re-weighted aggregation (Baek et al. 2023), structure-semantics decoupling (Tan et al. 2023), and causal disentanglement (Fu et al. 2025a).

The intrinsic efficacy of these existing studies is to calibrate the graph distributions across multiple clients during federated training. Nonetheless, this practice poses two open challenges. On the one hand, dynamic training cannot guarantee consistent global information extraction, which in turn affects the generalization ability and convergence efficiency of the global model. On the other hand, the complex federated training design limits the model scalability and makes it difficult to collaborate with existing federated learning (FL) algorithms. This raises a natural question: *could prior refinement of local graphs before federated training offer a more effective and flexible solution?* To explore this hypothesis, we visualize the loss landscapes of FedAvg and the proposed FedGH, along with the positions of global and local models in the optimization space, as shown in Figure 1. Local models converging within flatter regions of the loss surface generally indicate better generalization capacity of the resulting global model. From visualization, it can be observed that client models under FedGH exhibit well-aligned optimization directions and produce a more generalizable global model, although no architectural modifications were made and the only intervention was graph harmonization before federated training. This empirical evidence motivates us to pursue a more principled alternative: *conducting prior refinement at the local graph level before training, rather than directly adapting to heterogeneity during training.*

Based on the above analysis, a prior refinement is introduced before federated training to proactively mitigate data heterogeneity. However, two key challenges remain for prior refinement in the context of federated learning: (1) How to sufficiently extract data distributions from the local client without compromising privacy? To calibrate local graphs in advance, it is a prerequisite to extract their data distributions, including node features and topological structure. Therefore, it is a necessity to develop a privacy-preserving graph distribution extraction method. (2) How to harmonize the distribution gaps between clients using the extracted information? Once client-specific distributions of features and topologies are obtained, a critical challenge lies in aligning these heterogeneous patterns and leveraging the aligned results to refine local data before the federated training.

To tackle these challenges, we propose a Federated Graph Harmonization framework (FedGH), which performs prior refinement on local data by leveraging the generative capacity of graph diffusion. (1) To extract feature and topology distributions without exposing raw data, we leverage the inherent generative nature of graph diffusion and design a conditional diffusion process, built upon a pre-trained variational graph autoencoder, to generate pseudo-graphs locally on each client. By adaptively guiding the reverse denoising process from pure Gaussian noise, the conditional diffusion synthesizes pseudo-graphs that capture client-specific semantic and structural priors, while inherently avoiding direct exposure of the original data. (2) To integrate complementary information in multiple clients, we aim to learn the globally-aware representations from the generated pseudo-graphs. To achieve this goal, a graph contrastive learning module is employed on the server to collaboratively train these pseudo-graphs and obtain globally-aware latent embeddings. Finally, each client then refines and reconstructs its data under the guidance of these enriched embeddings prior to federated training. The refined graph data can be directly used in downstream federated learning without architectural modifications, making FedGH a plug-and-play preprocessor that can be seamlessly integrated into existing FL pipelines. Figure 2 overviews the working flow of FedGH. Our contributions are summarized as follows:

- We conduct empirical analysis and induce a novel perspective of performing prior refinement before federated training to proactively calibrate client graph distribution.
- We propose FedGH, a federated graph harmonization framework that explores the potential of diffusion models to alleviate cross-client heterogeneity.
- Extensive experiments on diverse graph benchmarks demonstrate the superior performance of FedGH over state-of-the-art FGL baselines.

Related Work

Federated Graph Learning

Federated Graph Learning (FGL) is an effective approach to addressing privacy concerns in distributed graph-based training, enabling collaborative learning without data sharing. Current methods focus on calibrating graph distributions across clients during federated training. A key line of research enhances local models or develops more generalizable global models. GraphFL (Wang et al. 2022) introduces a meta-learning-inspired framework for non-IID and partially labeled graphs. FedEgo (Zhang et al. 2023) addresses privacy and heterogeneity using ego-graphs in decentralized settings, while FGSSL (Huang et al. 2023) aligns node semantics and graph structure through contrastive learning and structural distillation. FedNCN (Liu et al. 2025) solves missing cross-subgraph links by leveraging clustering prior knowledge to restore connections. Another research direction improves federated aggregation processes to extract invariant information. FED-PUB (Baek et al. 2023) personalizes federated aggregation by computing inter-client functional similarity, while FedATH (Fu et al. 2025a) decomposes local graphs into causal and biased subgraphs, aggregating only the causal branches. However, correcting graph distributions during federated training remains challenging due to the inherent heterogeneity of graph data.

Diffusion Model

Diffusion models (DMs) (Ho, Jain, and Abbeel 2020; Song et al. 2021) have recently garnered significant attention as a powerful class of generative models, demonstrating exceptional performance in data generation across various domains. These models work by gradually transforming data into noise via a Markov process, and then learning to reverse this process to generate new samples that resemble the original data distribution. In the context of federated learning with image data, several works have leveraged the generative capabilities of diffusion models to enhance federated training. For instance, FedDiff (Li et al. 2024) proposes a multi-modal diffusion framework for federated learning in remote sensing. FedDISC (Yang et al. 2024) uses diffusion models to generate synthetic datasets that align with client distributions, while FedBiP (Chen et al. 2025) personalizes latent diffusion models to generate images that reflect client-specific distributions. Despite these advancements, relatively few studies have explored the use of diffusion models in Federated Graph Learning (FGL). Recent work FedGOG (Zhou et al. 2025) specifically focuses on out-of-distribution (OOD) generalization in federated graph learning through diffusion-driven global data exploration.

Preliminaries

Federated Graph Learning. Consider a federated system with K clients, where the k-th client holds a private graph $\mathcal{G}^k = (\mathcal{V}^k, \mathcal{E}^k)$, with $\mathcal{V}^k$ and $\mathcal{E}^k$ representing the node and edge sets, respectively. The feature matrix is denoted by $\mathbf{X}^k = \{\mathbf{x}_1^k, \ldots, \mathbf{x}_{|\mathcal{V}^k|}^k\}$, where each feature vector $\mathbf{x}_i^k \in \mathbb{R}^m$. The local topology for $\mathcal{G}^k$ is recorded by the adjacency matrix $\mathbf{A}^k \in \{0,1\}^{|\mathcal{V}^k| \times |\mathcal{V}^k|}$, where $\mathbf{A}_{ij}^k = 1$ denotes the existence of an edge $e_{ij} \in \mathcal{E}^k$ between nodes v_i^k and v_j^k.

In each communication round t, a central server broadcasts the global model to activated clients. Each client performs local training on its private graph $\mathcal{G}^k$ and updates its local model parameters $\mathbf{g}^k$. Then, local GNN parameters are transmitted to the server, which aggregates them into a new global model. A classical aggregation strategy of FGL follows FedAvg (McMahan et al. 2017), where the global model update is given by:

$$\bar{\mathbf{g}} = \sum_{k=1}^{K} \frac{|\mathcal{V}^k|}{\sum_{j=1}^{K} |\mathcal{V}^j|} \mathbf{g}^k. \tag{1}$$

Diffusion Probabilistic Model. As parameterized Markov chains, Diffusion Probabilistic Models (DPMs) are trained through variational inference. DPMs are often applied to generate samples that match the data distribution. Specifically, they describe the data generation process via a forward and reverse diffusion mechanism, as illustrated by the Denoising Diffusion Probabilistic Model (DDPM) (Ho, Jain, and Abbeel 2020).

In the forward process, the original data $\mathbf{x}_0$ is gradually corrupted by noise over T timesteps, and is ultimately transitioned into a Gaussian distribution $\mathbf{x}_t$, where we can sample $\mathbf{x}_t$ at any arbitrary timestep t in closed form:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{2}$$

where $\mathbf{I}$ denotes an identity covariance matrix, and $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$ represents the cumulative noise up to timestep t. The noise variance schedule, controlled by β_i, dictates the noise added at each timestep. Thus, to accelerate the forward process, the cumulative noise can be approximated by sampling Gaussian noise only once:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3}$$

where ϵ is a standard Gaussian noise.

The reverse process seeks to recover the original data $\mathbf{x}_0$ from the noisy samples $\mathbf{x}_t$ by predicting the added noise ϵ, a learnable denoise network $\epsilon_\theta(\cdot)$ is used to iteratively denoise, which is formulated as:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}, \tag{4}$$

where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ is sampled from a normal Gaussian distribution and σ_t is derived from the noise schedule. The training objective involves minimizing the mean squared error between the predicted noise and the actual noise, ensuring that the model can effectively denoise the noisy samples and recover the distribution of the original data.

Proposed Method

Diffusion-based Pseudo Graph Generation

Based on preliminary analysis, we shift the focus to the prior refinement of the data source before federated training. A promising solution is to extract both the semantic and topological distributions of each local graph while preserving privacy, which are uploaded to the server and refined to incorporate the global information via contrastive learning. Fortunately, the graph diffusion naturally satisfies this need, as it generates new data that reflects the original data distribution without exposing sensitive information. Motivated by this, the proposed FedGH generates pseudo-graphs via a diffusion process. Specifically, each client first pre-trains a variational graph autoencoder (VGAE) (Kipf and Welling 2016) to encode the local graph. During the diffusion process, the representation encoded by the well-trained autoencoder serves as input. After passing through a conditional diffusion, the obtained latent representation is decoded by the VGAE to generate the desired pseudo-graph.

To model the normality patterns of graph-structured data, we employ a VGAE to learn a low-dimensional latent representation $\mathbf{H}^k$ from each local graph $\mathcal{G}^k = (\mathbf{X}^k, \mathbf{A}^k)$ on client k. The encoding process is given by the client-specific variational posterior distribution:

$$q^k(\mathbf{H}^k \mid \mathbf{A}^k, \mathbf{X}^k) = \mathcal{N}(\mathbf{H}^k \mid \boldsymbol{\mu}^k, \boldsymbol{\sigma}^k), \tag{5}$$

where $\boldsymbol{\mu}^k = \mathrm{GCN}_{\boldsymbol{\mu}^k}(\mathbf{A}^k, \mathbf{X}^k)$ is the matrix of mean vectors, and $\boldsymbol{\sigma}^k = \mathrm{GCN}_{\boldsymbol{\sigma}^k}(\mathbf{A}^k, \mathbf{X}^k)$ denotes the matrix of standard deviations. The encoder maps each graph to latent variables through the reparameterization trick, where $\mathbf{H}^k = \boldsymbol{\mu}^k + \boldsymbol{\sigma}^k \odot \boldsymbol{\eta}$ with $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Once the latent variables $\mathbf{H}^k$ are obtained, the decoder attempts to reconstruct both the node features $\hat{\mathbf{X}}^k$ using a standard MLP decoder and the graph structure $\hat{\mathbf{A}}^k$ via a dot product:

$$\hat{\mathbf{X}}^k = \mathrm{MLP}(\mathbf{H}^k), \quad \hat{\mathbf{A}}^k = \sigma(\mathbf{H}^k [\mathbf{H}^k]^\top), \tag{6}$$

where $\sigma(\cdot)$ is the sigmoid function. The variational lower bound optimizes the object of VGAE is written as:

$$\begin{aligned} \mathcal{L}^k_{\mathrm{VGAE}} = \mathbb{E}_{q(\mathbf{H}^k|\mathbf{A}^k,\mathbf{X}^k)} \left[\log p_\phi(\hat{\mathcal{G}}^k \mid \mathbf{H}^k) \right] \\ -\gamma \mathrm{KL}\left(q(\mathbf{H}^k \mid \mathbf{A}^k, \mathbf{X}^k) \,\|\, p(\mathbf{H}^k) \right), \end{aligned} \tag{7}$$

where γ balances the reconstruction loss and the KL divergence, with p as the reconstruction probability distribution and q as the approximate posterior distribution.

Through the pre-trained VGAE model, we can obtain its latent representation $\mathbf{H}^k$ from the local graph $\mathcal{G}^k = \{\mathbf{X}^k, \mathbf{A}^k\}$. The captured latent representation $\mathbf{H}^k$ reflects the normal distribution of the original data and is used as input to the diffusion model. We set up a conditional diffusion process to control the stepwise denoising procedure. First, in the forward process, Gaussian noise is sampled over T timesteps to progressively corrupt the latent representation:

$$\mathbf{h}^k_t = \sqrt{\bar{\alpha}_t} \mathbf{h}^k_0 + \sqrt{1-\bar{\alpha}_t} \epsilon^k_t, \quad \epsilon^k_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{8}$$

where $\mathbf{h}^k_0$ is initialized by $\mathbf{H}^k$ and $\bar{\alpha}_t = \prod_{i=1}^t (1-\beta_i)$ where β_i controls the noise added at each timestep. This formulation progressively destroys the latent representation $\mathbf{h}^k_t$ with increasing noise over multiple steps.

To balance the recovery of the original data distribution and the protection of local privacy, we introduce a conditional denoising process:

$$\mathbf{h}^k_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{h}^k_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{h}^k_t, \mathbf{r}^k, t) \right) + \sigma_t \mathbf{z}^k \tag{9}$$

where $\mathbf{r}^k$ denotes a client-aware noise regulator designed to modulate the denoising trajectory. Unlike traditional denoising, which injects fixed and indiscriminate noise, $\mathbf{r}^k$ serves as an adaptive control signal that incorporates client-specific structural and semantic priors while avoiding direct leakage of raw data. This design enables the model to dynamically balance the trade-off between distributional fidelity and privacy preservation. To construct $\mathbf{r}^k$, we first perturb initial latent embeddings with Gaussian noise $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and then apply a nonlinear transformation, yielding $\mathbf{r}^k = \mathcal{F}^k_\omega(\mathbf{h}^k_0 + \boldsymbol{\eta})$, where $\mathcal{F}_\omega(\cdot)$ is a multi-layer perceptron. This operation enriches the input with controllable stochasticity while abstracting over heterogeneous client priors. This transformation endows the latent diffusion process with client-aware variability, allowing flexible adaptation to diverse data distributions while preserving privacy.

Finally, the training objective is to learn a reverse trajectory that progressively denoises the perturbed representations, formulated as:

$$\mathcal{L}^k_{\mathrm{diff}} = \mathbb{E}_{t, \mathbf{h}^k_0, \epsilon^k} \left\| \epsilon^k - \epsilon^k_\theta \left(\sqrt{\bar{\alpha}_t} \mathbf{h}^k_0 + \sqrt{1-\bar{\alpha}_t} \epsilon, \mathbf{r}^k, t \right) \right\|^2. \tag{10}$$

After obtaining the latent representation $\hat{\mathbf{h}}^k_0$ via the conditional diffusion process, i.e., $\mathbf{h}^k_t \to \mathbf{h}^k_{t-1} \to \cdots \to \hat{\mathbf{h}}^k_0$, we generate the final pseudo-graph for each client by decoding it with the pre-trained decoder, as defined in Eq. (6): $\hat{\mathbf{X}}^k = \mathrm{MLP}(\hat{\mathbf{H}}^k)$, $\hat{\mathbf{A}}^k = \sigma(\hat{\mathbf{H}}^k [\hat{\mathbf{H}}^k]^\top)$, where $\hat{\mathbf{H}}^k = \{[\hat{\mathbf{h}}^k_0]_i\}_{i=1}^{|\mathcal{V}^k|}$.

Graph Alignment via Pseudo-Graph Contrast

Given the generated pseudo-graphs that retain the underlying distribution of local data, our objective is to learn globally-aware latent representations on the server by aggregating information in these pseudo-graphs and then redistributing them back. To achieve this goal, we adopt a graph contrastive learning framework that collaboratively trains on the collected pseudo-graphs. Formally, for a given client k, the pseudo-graph $\hat{\mathcal{G}}^k = \{\hat{\mathbf{X}}^k, \hat{\mathbf{A}}^k\}$ is first encoded via the local GNN encoder $f^k_\Theta(\cdot)$ to produce node-level embeddings: $\tilde{\mathbf{H}}^k = f^k_\Theta(\hat{\mathbf{X}}^k, \hat{\mathbf{A}}^k)$. They are then aggregated into a global summary vector via an average pooling function:

$$\mathbf{s}_k = \sigma\left(\frac{1}{|\mathcal{V}^k|}\sum_{i=1}^{|\mathcal{V}^k|}\tilde{\mathbf{h}}_i^k\right), \tag{11}$$

where $\sigma(\cdot)$ is a sigmoid activation, and $\tilde{\mathbf{h}}_i^k$ is the embedding of node v_i in $\hat{\mathcal{G}}^k$. Thus, to obtain globally aligned embeddings for each client, we maximize the mutual information (MI) between a client's local pseudo-representation and the global pseudo-representations from peer clients. A discriminator $\mathcal{D}_\psi$ is employed to distinguish positive and negative node-graph pairs. We randomly shuffle the node embeddings to produce $\tilde{\mathbf{H}}^k_{\text{shuf}}$ to construct negative samples. The MI is then maximized using a binary cross-entropy objective:

$$\begin{aligned}\mathcal{L}^k_{\text{con}} = -\frac{1}{2|\mathcal{V}^k|}\sum_{i=1}^{|\mathcal{V}^k|}\Bigg(\frac{1}{K}\sum_{j=1}^{K}\Big(\log\mathcal{D}_\psi(\tilde{\mathbf{h}}_i^k, \mathbf{s}_j) \\ + \log\Big(1-\mathcal{D}_\psi([\tilde{\mathbf{h}}^k_{\text{shuf}}]_i, \mathbf{s}_j)\Big)\Big)\Bigg).\end{aligned} \tag{12}$$

By minimizing this loss, we encourage each client's local pseudo-representation to align with global summaries from all clients, thereby yielding globally-aware embeddings. The server then distributes these client-specific representations back to each client to refine the original graph.

Local Graph Refinement

Upon receiving globally-aware node representations $\tilde{\mathbf{H}}^k$, each client refines its local graph $\mathcal{G}^k = (\mathbf{X}^k, \mathbf{A}^k)$ by utilizing these enriched embeddings to guide both structural and feature enhancement in a principled manner. This procedure leverages the global information encoded in $\tilde{\mathbf{H}}^k$ as supervised signals to rectify and align local representations across clients, ensuring better consistency before federated training. First of all, we compute the pairwise similarity:

$$\mathbf{S}^k_{ij} = \text{sim}(\tilde{\mathbf{h}}_i^k, \tilde{\mathbf{h}}_j^k), \tag{13}$$

where $\text{sim}(\cdot,\cdot)$ denotes a similarity function measuring semantic proximity between nodes. These similarity scores serve as soft supervision to guide the reconstruction of the local graph topology. We introduce a graph structure adjustment operation $\mathcal{O}_\tau(\cdot)$ that updates the local edge set based on the similarity matrix:

$$\mathcal{E}^k_{\text{refined}} = \mathcal{O}_\tau(\tilde{\mathcal{G}}^k, \mathbf{S}^k) = (\mathcal{V}^k, \tilde{\mathcal{E}}^k \cup \mathcal{E}^k_{\text{aug}}), \tag{14}$$

where $\mathcal{E}^k_{\text{aug}} = \{(v_i, v_j) \notin \tilde{\mathcal{E}}^k \mid \mathbf{S}^k_{ij} > \tau\}$ denotes a similarity-aware augmentation set determined by a threshold τ. Alternatively, operation $\mathcal{O}_\tau(\cdot)$ can be defined as a Top-k policy by selecting the k most similar neighbors for each node. In parallel, the node features are also updated by incorporating global information through a fusion operator:

$$\tilde{\mathbf{X}}^k = \mathcal{T}(\mathbf{X}^k, \tilde{\mathbf{H}}^k), \tag{15}$$

Algorithm 1: FedGH

Input: Local graph data $\mathcal{G}^k = (\mathbf{X}^k, \mathbf{A}^k)$, number of clients K, diffusion steps T, diffusion training epochs E_{diff}, contrastive learning epochs E_{con}.
Output: Refined graph data $\tilde{\mathcal{G}}^k = (\tilde{\mathbf{X}}^k, \tilde{\mathbf{A}}^k)$.

<u>Client Side:</u>
Initialize the network parameters
Pre-train the VGAE by minimizing Eq. (7)
Freeze the parameters of the pre-trained model
for epoch $= 1$ to E_{diff} **do**
  Obtain $\mathbf{H}^k$ from the pre-trained encoder
  Apply forward diffusion to $\mathbf{H}^k$ via Eq. (8)
  Generate $\hat{\mathbf{H}}^k$ via reverse diffusion Eq. (9)
  Decode $\hat{\mathcal{G}}^k$ from $\hat{\mathbf{H}}^k$ via pre-trained decoder
  Update parameters by minimizing Eq. (10)
end for
<u>Server Side:</u>
for $k = 1$ to K **do**
  for epoch $= 1$ to E_{con} **do**
    Receive $\{\hat{\mathcal{G}}^i\}_{i=1}^K$ from clients
    Compute $\{\tilde{\mathbf{H}}^i\}_{i=1}^K$ and $\{\tilde{\mathbf{H}}^i_{\text{shuf}}\}_{i=1}^K$ via GNN
    Compute global summary $\{\mathbf{s}_k\}_{i=1}^K$ via Eq. (11)
    Update parameters by minimizing Eq. (12)
  end for
end for
<u>Client Side:</u>
Receive globally-aware embeddings $\tilde{\mathbf{H}}^k$
Refine the graph based on $\tilde{\mathbf{H}}^k$ via Eqs. (14) and (15)
return $\tilde{\mathcal{G}}^k = (\tilde{\mathbf{X}}^k, \tilde{\mathbf{A}}^k)$

where $\mathcal{T}$ is a fusion operator. Here, we adopt the concatenation, i.e., $\tilde{\mathbf{X}}^k = [\mathbf{X}^k \,||\, \tilde{\mathbf{H}}^k]$. The resulting refined graph $\tilde{\mathcal{G}}^k = (\tilde{\mathbf{X}}^k, \tilde{\mathbf{A}}^k)$ serves as the updated input for each client prior to federated training, leading to improved distribution alignment under non-IID conditions. Specifically, the refined graph $\tilde{\mathcal{G}}^k$ can be integrated with various existing FL methods, with FedAvg being employed in this case.

Complexity and Privacy Analysis

Complexity Analysis. Recall that the node and edge numbers for client k are $|\mathcal{V}^k|$ and $|\mathcal{E}^k|$, respectively. The input feature dimension is m, and the latent embedding dimension is d. There are three main stages prior to federated training:

- **Diffusion-based Pseudo Graph Generation:** We employ an L-layer GIN encoder for VGAE pretraining, with cost $\mathcal{O}(L(|\mathcal{E}^k|d + |\mathcal{V}^k|d^2))$. The conditional diffusion process adds $\mathcal{O}(T|\mathcal{V}^k|d^2)$ due to denoising steps, while adjacency reconstruction via decoding requires $\mathcal{O}(|\mathcal{E}^k|d)$ by applying a sampling strategy in practice.
- **Graph Contrastive Alignment:** Encoding the pseudo-graph costs $\mathcal{O}(L(|\mathcal{E}^k|d + |\mathcal{V}^k|d^2))$ and contrastive learning via a bilinear discriminator over K client summaries costs $\mathcal{O}(K|\mathcal{V}^k|d^2)$.
- **Local Graph Refinement:** The topology reconstruction costs $\mathcal{O}(|\mathcal{V}^k|^2 d)$, and the final update process of node

Dataset	Cora			PubMed			Photo		
Method	10 Clients	15 Clients	20 Clients	10 Clients	15 Clients	20 Clients	10 Clients	15 Clients	20 Clients
FedAvg	73.6 (72.0)	69.5 (66.2)	62.4 (58.8)	82.4 (80.2)	81.5 (79.0)	80.8 (78.9)	87.2 (84.9)	86.0 (83.1)	84.4 (81.4)
FedProx	74.3 (72.9)	70.0 (66.7)	63.2 (59.6)	82.4 (80.2)	81.6 (79.1)	80.8 (78.9)	87.1 (84.8)	86.5 (83.7)	84.4 (81.4)
MOON	74.2 (71.9)	70.5 (68.0)	61.4 (58.0)	82.5 (80.4)	81.6 (79.1)	80.2 (78.1)	86.5 (83.6)	85.2 (82.0)	81.5 (79.4)
FedOPT	74.6 (67.7)	71.3 (63.6)	63.7 (59.8)	81.5 (78.1)	80.5 (75.3)	80.9 (76.5)	87.7 (84.9)	86.1 (83.0)	84.2 (80.3)
FedProto	74.6 (73.0)	70.0 (67.0)	63.4 (59.4)	82.7 (80.4)	81.5 (79.0)	80.8 (78.9)	87.3 (84.9)	87.9 (85.1)	84.8 (81.7)
FedSage+	74.0 (72.0)	67.4 (62.1)	65.0 (59.9)	82.4 (82.2)	78.2 (78.0)	78.7 (76.2)	88.5 (86.1)	86.0 (82.0)	84.4 (82.9)
FGSSL	74.5 (72.9)	72.2 (69.3)	65.9 (61.9)	82.4 (79.9)	81.9 (79.3)	81.0 (78.8)	88.6 (86.0)	86.2 (82.6)	84.3 (80.2)
FedPUB	75.4 (73.1)	72.4 (66.8)	66.4 (60.0)	82.7 (80.4)	79.1 (75.4)	79.6 (79.6)	88.8 (85.9)	87.2 (83.5)	84.2 (79.3)
FedTAD	74.3 (71.6)	72.2 (70.4)	63.7 (60.3)	82.7 (80.3)	82.0 (78.8)	81.2 (78.6)	87.8 (85.2)	86.6 (83.5)	84.9 (81.2)
FedATH	77.2 (74.1)	73.7 (72.2)	68.3 (65.4)	83.8 (82.1)	82.6 (80.7)	83.0 (81.4)	88.5 (86.5)	86.4 (85.3)	85.5 (82.7)
FedGH	**78.0 (74.9)**	**75.8 (72.4)**	**70.7 (66.0)**	**85.2 (83.9)**	**84.7 (82.9)**	**84.0 (83.1)**	**90.0 (88.2)**	**88.1 (86.2)**	**86.9 (84.5)**
Dataset	**WikiCS**			**Roman-empire**			**ogbn-arxiv**		
Method	10 Clients	15 Clients	20 Clients	10 Clients	15 Clients	20 Clients	10 Clients	15 Clients	20 Clients
FedAvg	69.6 (63.7)	66.3 (61.7)	67.5 (62.4)	34.4 (30.3)	33.8 (29.7)	32.2 (27.8)	35.8 (26.1)	34.7 (26.2)	33.6 (25.1)
FedProx	69.5 (63.6)	66.2 (61.5)	67.6 (62.3)	34.3 (30.2)	33.6 (29.5)	32.0 (27.6)	35.7 (26.1)	34.4 (25.9)	33.6 (25.1)
MOON	70.0 (64.0)	67.2 (62.2)	66.5 (60.9)	34.0 (29.9)	33.8 (29.6)	33.0 (28.6)	34.9 (25.8)	33.7 (24.7)	33.9 (25.6)
FedOPT	69.4 (60.5)	68.1 (60.4)	66.5 (56.9)	34.4 (30.5)	33.4 (29.6)	32.1 (28.1)	36.4 (24.1)	34.4 (24.2)	35.8 (26.4)
FedProto	70.0 (64.0)	67.0 (62.0)	68.2 (62.2)	34.9 (30.0)	33.6 (28.8)	32.3 (27.3)	35.8 (26.1)	34.6 (26.0)	33.6 (25.0)
FedSage+	71.7 (61.5)	69.3 (62.2)	70.7 (61.2)	41.6 (31.9)	39.4 (29.4)	39.1 (29.6)	41.0 (27.5)	36.6 (28.8)	37.2 (30.2)
FGSSL	70.6 (64.8)	69.0 (64.1)	68.6 (62.9)	37.0 (33.6)	36.9 (33.1)	35.2 (31.5)	39.2 (30.4)	36.6 (27.9)	35.2 (26.6)
FedPUB	72.3 (63.8)	69.8 (59.2)	69.9 (59.7)	37.3 (30.2)	36.1 (29.2)	35.5 (28.7)	39.0 (29.6)	36.9 (24.8)	36.4 (24.4)
FedTAD	71.6 (65.3)	69.3 (63.7)	69.0 (63.1)	39.0 (35.1)	37.9 (33.8)	37.3 (33.2)	37.6 (27.2)	36.3 (26.7)	34.6 (25.1)
FedATH	75.8 (71.3)	73.1 (68.6)	72.5 (67.6)	43.8 (38.1)	41.5 (35.8)	40.3 (35.5)	41.6 (34.0)	37.3 (30.6)	**38.9 (32.0)**
FedGH	**77.1 (72.8)**	**76.8 (72.2)**	**76.4 (71.8)**	**44.3 (39.0)**	**42.2 (36.1)**	**41.0 (35.8)**	**42.6 (35.4)**	**38.8 (32.1)**	38.1 (31.6)

Table 1: Performance comparison (ACC% and (F1%)) of SOTA methods, where the best results are highlighted in **bold** and the second-best results are highlighted in underline.

features via concatenation requires $\mathcal{O}(|\mathcal{V}^k|(m+d))$.

Combining the above components, the overall complexity for each client is: $\mathcal{O}\big(L|\mathcal{E}^k|d+(L+T+K)|\mathcal{V}^k|d^2+|\mathcal{V}^k|(d+m)+|\mathcal{V}^k|^2 d\big)$. In practice, the terms involving L, T, and K are typically small constants, and the computation is dominated by the topology reconstruction. This cost is incurred once by each client during the prior refinement stage, making it a manageable trade-off for effective graph refinement. The procedure of FedGH is described in Algorithm 1.

Privacy Analysis. Inspired by (Patel, Zhang, and Wang 2025), we provide a theoretical analysis to address privacy concerns, proving FedGH guarantees privacy preservation:

Theorem 1 (Privacy Guarantee of FedGH) *Given the latent representation $[\mathbf{h}_0^k]_i \in \mathbb{R}^d$ of node i for client k with $\|[\mathbf{h}_0^k]_i\|_2 \le C$, and the forward diffusion process with t timesteps, the mechanism satisfies (ϵ, δ)-local differential privacy (LDP) where:*

$$\epsilon = \frac{2\bar{\alpha}_t C^2}{1-\bar{\alpha}_t} + C\sqrt{\frac{8\bar{\alpha}_t \log(1/\delta)}{1-\bar{\alpha}_t}}, \quad \delta \in (0,1),$$

where $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$ is the cumulative noise factor, β_i controls the noise added at each timestep, σ_t is derived from the noise schedule, and C is the latent norm bound.

The proof is deferred to Appendix A. The privacy guarantee in Theorem 1 addresses the critical privacy risks in the proposed FedGH by ensuring that the generation of pseudo-graphs does not reveal sensitive node-level information.

Experiments

In this section, we evaluate the effectiveness of FedGH by answering the following research questions:

- **RQ1:** Does FedGH outperform existing FGL baselines?
- **RQ2:** How well does FedGH mitigate heterogeneity?
- **RQ3:** How does FedGH ensure privacy preservation?
- **RQ4:** How effective is FedGH as a plug-in module?
- **RQ5:** How does each component or hyperparameter affect the performance of FedGH?

Experimental Settings

Dataset: To assess the performance of FedGH, we conduct experiments on six real-world datasets: two citation networks, Cora and PubMed; one co-purchase network–Photo; one Wiki-page network–WikiCS; one article-syntax network–Roman-empire; and one large-scale citation network–obgn-arXiv. *Methods:* We compare FedGH with ten representative methods to assess its effectiveness, including: 1) conventional methods: FedAvg (McMahan et al. 2017), FedProx (Li et al. 2020), MOON (Li, He, and Song 2021), FedOPT (Reddi et al. 2021), and FedProto (Tan et al.

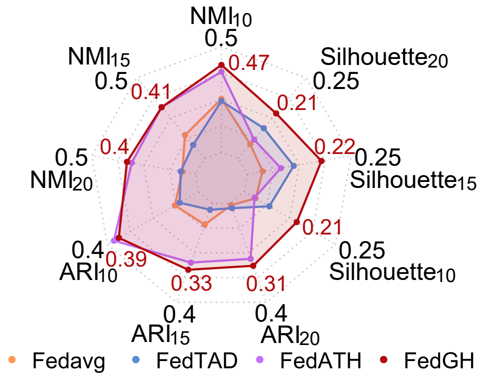
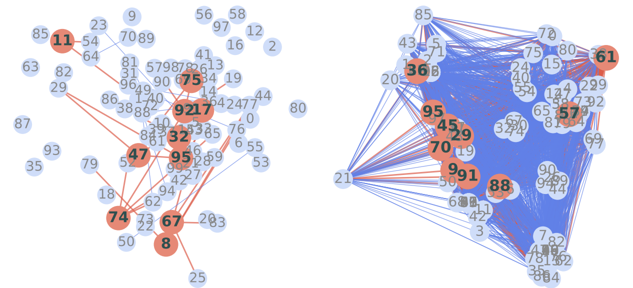


Figure 3: Consistency evaluation of the learned node embeddings via clustering-based metrics. Subscripts on each metric indicate the number of clients involved in the evaluation.



(a) Original Graph (b) Pseudo Graph

Figure 4: The visualization of the original graph and the corresponding pseudo-graph generated by diffusion from the first subgraph among the 10 clients in the Cora dataset.

2022); 2) graph-oriented methods: FedSage+ (Zhang et al. 2021), FGSSL (Huang et al. 2023), FedPUB (Baek et al. 2023), FedTAD (Zhu et al. 2024), and FedATH (Fu et al. 2025a). Please refer to Appendix B for detailed settings.

Performance Comparison (RQ1)

Table 1 presents the performance comparison of the proposed FedGH against representative baselines, evaluating performance in terms of accuracy and F1-score across multiple client configurations (10, 15, and 20 clients) to simulate varying degrees of data distribution. It can be observed that graph-oriented methods, which are specifically designed to handle graph data, generally outperform traditional federated learning methods. This is attributed to their ability to leverage graph-specific structures and semantic information, leading to better alignment of local models and data representation. As shown in Table 1, FedGH consistently outperforms all other methods on most datasets and client configurations. The superior performance highlights that data-level refinement prior to federated training effectively calibrates client-specific optimization trajectories. By addressing graph data inconsistencies before the training process, FedGH enables more efficient aggregation of client models, leading to improved generalization across various datasets.

Effectiveness in Tackling Heterogeneity (RQ2)

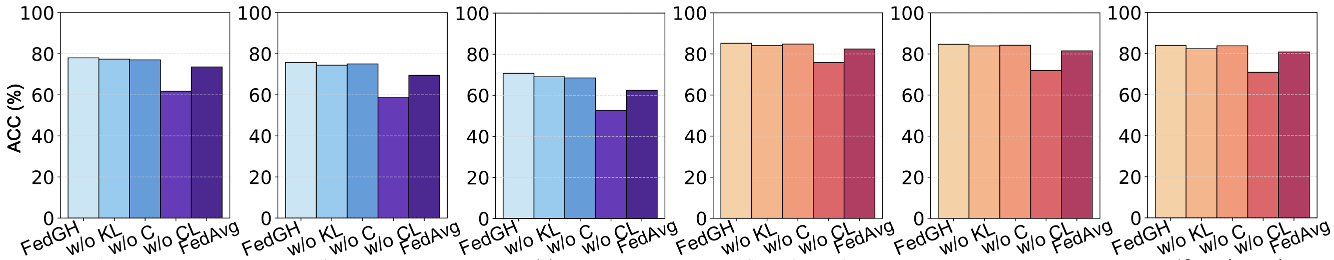
We evaluate the effectiveness of different methods in mitigating data heterogeneity by measuring the semantic consistency of node embeddings across clients via clustering-based metrics. For each approach, we merge client-specific embeddings pairwise, apply K-Means to perform clustering, and average the computed metrics over all client pairs. The resulting clusters are compared with ground-truth labels using NMI, ARI, and Silhouette scores, where higher values indicate better alignment with the underlying semantic structure. As illustrated in Figure 3, FedGH consistently outperforms both classical (e.g., FedAvg) and SOTA FGL methods (e.g., FedTAD, FedATH) across all metrics and client configurations in the Cora dataset. These results indicate that FedGH produces embeddings with stronger semantic alignment across clients, suggesting improved representation quality in heterogeneous graph distributions.

Analysis of Privacy Concerns (RQ3)

To assess the privacy-preserving capability of the proposed FedGH, we investigate whether the pseudo-graphs generated via graph diffusion expose sensitive information from the original data. FedGH generates these pseudo-graphs through a conditional reverse diffusion process initialized with pure Gaussian noise, which inherently mitigates the risk of privacy leakage. To empirically validate this capability, we visualize the first subgraph of the Cora dataset by applying t-SNE (Van der Maaten L 2008), comparing the original graph with the corresponding pseudo-graph produced by diffusion. For clarity, we display only nodes with IDs from 0 to 99 and highlight the top 10 nodes with the highest degree in red. Since t-SNE projects high-dimensional node features into two-dimensional space, the resulting node positions and connectivity patterns reflect both semantic and structural characteristics. As shown in Figure 4, the node locations, edge connections, and hub nodes substantially differ between the original and synthesized graphs. These discrepancies provide strong empirical evidence that the diffusion process effectively conceals sensitive structural and feature-level information. Hence, FedGH offers privacy protection without relying on encryption techniques, supporting its suitability for privacy-aware federated graph learning.

Improving Existing FGL Methods (RQ4)

To examine the plug-and-play capability of FedGH as a data-level prior refinement strategy, we investigate its integration with an SOTA FGL model, FedTAD (Zhu et al. 2024). Specifically, we refine each client's local graph using FedGH and then train FedTAD on these harmonized graphs. As shown in Table 2, the performance consistently improves across three datasets and varying numbers of clients, in both classification accuracy and F1-score. Notably, the largest gains are observed under more heterogeneous scenarios with 20 clients, such as a 3.5% increase in ACC and a 3.6% boost in F1-score on Cora. These results demonstrate that FedGH

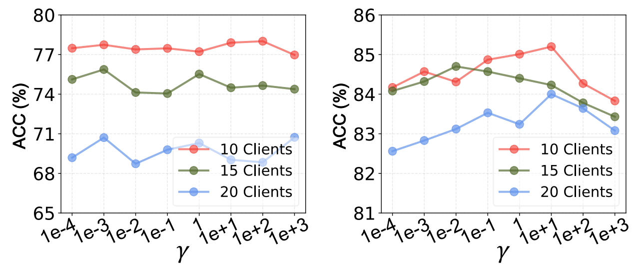


(a) Cora$_{10}$ (b) Cora$_{15}$ (c) Cora$_{20}$ (d) PubMed$_{10}$ (e) PubMed$_{15}$ (f) PubMed$_{20}$

Figure 5: Accuracy comparison between FedGH and its variants across different numbers of clients on the Cora and PubMed datasets, where subscripts in dataset names indicate the number of clients.

Metric	Dataset	Method	10 Clients	15 Clients	20 Clients
ACC	Cora	FedTAD	74.29	72.23	63.74
		+ FedGH	75.89 (↑1.6)	72.79 (↑0.6)	67.28 (↑3.5)
	PubMed	FedTAD	82.72	82.03	81.19
		+ FedGH	84.53 (↑1.8)	84.14 (↑2.1)	83.73 (↑2.5)
	Photo	FedTAD	87.76	86.60	84.94
		+ FedGH	87.92 (↑0.2)	86.92 (↑0.3)	86.29 (↑1.4)
F1-score	Cora	FedTAD	71.57	70.38	60.34
		+ FedGH	72.20 (↑0.6)	70.90 (↑0.5)	63.90 (↑3.6)
	PubMed	FedTAD	80.26	78.84	78.57
		+ FedGH	80.47 (↑0.2)	78.98 (↑0.1)	79.01 (↑0.4)
	Photo	FedTAD	85.24	83.45	81.20
		+ FedGH	85.31 (↑0.1)	83.74 (↑0.3)	81.63 (↑0.4)

Table 2: ACC% and F1-score% of FedTAD with and without the FedGH refinement under different client settings.



(a) Cora (b) PubMed

Figure 6: Parameter sensitivity on Cora and PubMed, evaluated across 10, 15, and 20 clients.

serves as a general and effective pre-processing module that integrates seamlessly with existing FGL frameworks, offering consistent benefits without requiring any changes to model architecture or training procedures.

Component and Parameter Analysis (RQ5)

Ablation Study. To validate the effectiveness of each component in FedGH, we conduct an ablation study to assess its impact. Specifically, we evaluate four variants by removing specific components: **w/o KL** (removal of the KL divergence term in the VGAE loss, equivalent to using a basic autoencoder), **w/o C** (removal of the conditional vector $\mathbf{r}^k$ in the diffusion model), **w/o CL** (no contrastive learning, using pseudo-graphs directly for training), and **FedAvg** (no prior refinement, equivalent to the FedAvg baseline). Note that we cannot remove the diffusion model alone, as using the graph encoded and decoded solely by the VGAE model and then uploading it to the server leads to data leakage, since the graph still retains the original data. As shown in Figure 5, the results clearly demonstrate that each component contributes positively to improving the model's performance. Notably, the significant performance drop in w/o CL is likely due to the disruption of the original data by the diffusion model. However, the fact that performance does not collapse entirely suggests that the model still captures valuable distributional information. Overall, FedGH consistently outperforms all variants, demonstrating its ability to effectively mitigate data heterogeneity.

Parameter Sensitivity. We analyze the sensitivity of FedGH to the hyperparameter γ, which balances the KL divergence term in the VGAE loss (Eq. (7)). Figure 6 reports the classification accuracy on Cora and PubMed across varying γ values and different numbers of clients. FedGH yields consistently stable performance across all configurations, with only marginal fluctuations in accuracy as γ changes. When γ is too small, the latent space may become under-constrained, while excessively large values may suppress meaningful representations. Nonetheless, performance remains competitive across a wide range of values, indicating that FedGH does not require extensive hyperparameter tuning and is relatively insensitive to the choice of γ.

Conclusion

In this paper, we present an empirical analysis that reveals a novel perspective of prior refinement before federated training, as opposed to methods that correct client-specific graph distributions during training. Inspired by these findings, we propose a Federated Graph Harmonization framework, FedGH, which explores the generative power of diffusion models to extract valuable distributional information for prior refinement. Specifically, a conditional denoising process is designed to extract pseudo-graphs, with graph contrastive learning then employed to align client-specific distributions and generate globally-aware embeddings. These embeddings subsequently guide the refinement of local data. Extensive experiments demonstrate the superior performance of the proposed FedGH.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China under Grant Nos. 62202104, and U2441239.

References

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

Cai, J.; Zhang, Y.; Liu, F.; and Ng, S.-K. 2025. Leveraging Diffusion Model as Pseudo-Anomalous Graph Generator for Graph-Level Anomaly Detection. In *Proceedings of the International Conference on Machine Learning*.

Chen, H.; Li, H.; Zhang, Y.; Bi, J.; Zhang, G.; Zhang, Y.; Torr, P.; Gu, J.; Krompass, D.; and Tresp, V. 2025. Fedbip: Heterogeneous one-shot federated learning with personalized latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 30440–30450.

Fu, L.; Deng, B.; Huang, S.; Liao, T.; Pan, S.; and Chen, C. 2025a. Less is More: Federated Graph Learning with Alleviating Topology Heterogeneity from A Causal Perspective. In *Proceedings of the International Conference on Machine Learning*.

Fu, L.; Li, Y.; Huang, S.; Chen, C.; Zhang, C.; and Zheng, Z. 2025b. Parameter-oriented contrastive schema and multi-level knowledge distillation for heterogeneous federated learning. *Information Fusion*, 121: 103123.

Fu, L.; Liao, T.; Huang, S.; Deng, B.; Pan, S.; Chen, C.; et al. 2025c. Unsupervised Federated Graph Learning. In *Advances in Neural Information Processing Systems*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 6840–6851.

Huang, S.; Fu, L.; Liao, T.; Deng, B.; Zhang, C.; and Chen, C. 2025. FedBG: Proactively Mitigating Bias in Cross-Domain Graph Federated Learning Using Background Data. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 5408–5416.

Huang, W.; Wan, G.; Ye, M.; and Du, B. 2023. Federated graph semantic and structural learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Jin, D.; Wang, L.; Zheng, Y.; Song, G.; Jiang, F.; Li, X.; Lin, W.; and Pan, S. 2023. Dual intent enhanced graph neural network for session-based new item recommendation. In *Proceedings of the ACM web conference*, 684–693.

Kipf, T. N.; and Welling, M. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Li, D.; Xie, W.; Wang, Z.; Lu, Y.; Li, Y.; and Fang, L. 2024. Feddiff: Diffusion model driven federated learning for multi-modal and multi-clients. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(10): 10353–10367.

Li, Q.; He, B.; and Song, D. 2021. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10713–10722.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2: 429–450.

Liu, J.; Han, R.; Tu, W.; Wang, H.; Wu, J.; and Cheng, J. 2025. Federated Node-Level Clustering Network with Cross-Subgraph Link Mending. In *Proceedings of the International Conference on Machine Learning*.

Liu, Y.; Li, S.; Zheng, Y.; Chen, Q.; Zhang, C.; and Pan, S. 2024. Arc: A generalist graph anomaly detector with in-context learning. In *Advances in Neural Information Processing Systems*, volume 37, 50772–50804.

Lu, J.; Wu, Z.; Yu, J.; Bu, J.; and Wang, H. 2025a. Multi-omics analysis for cancer subtype inference via unrolling graph smoothness priors. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 5842–5850.

Lu, Z.; Yu, Y.; Ma, L.; Nie, F.; and Wang, R. 2025b. Capturing Individuality and Commonality Between Anchor Graphs for Multi-View Clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 5860–5868.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics*, 1273–1282.

Patel, K. K.; Zhang, W.; and Wang, L. 2025. Personalized Federated Training of Diffusion Models with Privacy Guarantees. *arXiv preprint arXiv:2504.00952*.

Qi, Z.; Meng, L.; Chen, Z.; Hu, H.; Lin, H.; and Meng, X. 2023. Cross-silo prototypical calibration for federated learning with non-iid data. In *Proceedings of the 31st ACM international conference on multimedia*, 3099–3107.

Reddi, S.; Charles, Z.; Zaheer, M.; Garrett, Z.; Rush, K.; Konečnỳ, J.; Kumar, S.; and McMahan, H. B. 2021. Adaptive federated optimization. In *Proceedings of the International Conference on Learning Representations*.

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 *Proceedings of the International Conference on Learning Representations*.

Tan, Y.; Liu, Y.; Long, G.; Jiang, J.; Lu, Q.; and Zhang, C. 2023. Federated learning on non-iid graphs via structural knowledge sharing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 9953–9961.

Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; and Zhang, C. 2022. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8432–8440.

Tu, W.; Zhou, S.; Liu, X.; Cai, Z.; Zhao, Y.; Liu, Y.; and He, K. 2025. Wage: Weight-sharing attribute-missing graph autoencoder. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Van der Maaten L, H. G. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*.

Wan, G.; Huang, W.; and Ye, M. 2024. Federated Graph Learning under Domain Shift with Generalizable Prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 15429–15437.

Wan, G.; Shang, X.; Wu, Y.; Zhang, G.; Bi, J.; Zheng, L.; Lin, X.; Liu, Y.; Ma, Y.; Huang, W.; et al. 2025. HYPERION: Fine-Grained Hypersphere Alignment for Robust Federated Graph Learning. In *Advances in Neural Information Processing Systems*.

Wang, B.; Li, A.; Pang, M.; Li, H.; and Chen, Y. 2022. Graphfl: A federated learning framework for semi-supervised node classification on graphs. In *IEEE International Conference on Data Mining*, 498–507.

Wang, L.; He, D.; Zhang, H.; Liu, Y.; Wang, W.; Pan, S.; Jin, D.; and Chua, T.-S. 2024. Goodat: Towards test-time graph out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 15537–15545.

Wang, M.; Liu, F.; Jiao, L.; Li, S.; Li, L.; Chen, P.; Liu, X.; and Ma, W. 2026. VCGPrompt: Visual Concept Graph-Aware Prompt Learning for Vision-Language Models. *Pattern Recognition*, 170: 112012.

Wu, Z.; Lin, X.; Lin, Z.; Chen, Z.; Bai, Y.; and Wang, S. 2023. Interpretable Graph Convolutional Network for Multi-View Semi-Supervised Learning. *IEEE Transactions on Multimedia*, 25: 8593–8606.

Wu, Z.; Lu, J.; Yu, J.; Zhou, S.; Pi, Y.; and Wang, H. 2025. Divide and Conquer: Coordinating Multiplex Mixture of Graph Learners to Handle Multi-Omics Analysis. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24.

Wu, Z.; Zhang, Z.; and Fan, J. 2023. Graph Convolutional Kernel Machine versus Graph Convolutional Networks. In *Advances in Neural Information Processing Systems*, volume 36, 19650–19672.

Yang, M.; Su, S.; Li, B.; and Xue, X. 2024. Exploring one-shot semi-supervised federated learning with pre-trained diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16325–16333.

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

Zhang, T.; Mai, C.; Chang, Y.; Chen, C.; Shu, L.; and Zheng, Z. 2023. Fedego: privacy-preserving personalized federated graph learning with ego-graphs. *ACM Transactions on Knowledge Discovery from Data*, 18(2): 1–27.

Zhou, P.; Chen, C.; Liu, W.; Liao, X.; Shen, W.; Xu, J.; Fu, Z.; Wang, J.; Wen, W.; and Zheng, X. 2025. FedGOG: Federated Graph Out-of-Distribution Generalization with Diffusion Data Exploration and Latent Embedding Decorrelation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 22965–22973.

Zhu, Y.; Li, X.; Wu, Z.; Wu, D.; Hu, M.; and Li, R.-H. 2024. Fedtad: Topology-aware data-free knowledge distillation for subgraph federated learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Zhuang, D.; Dai, J.; Li, X.; Wu, X.; Sun, Y.; and Ren, Z. 2025a. Robust graph contrastive learning for incomplete multi-view clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 7282–7290.

Zhuang, S.; Huang, S.; Huang, W.; Chen, Y.; Wu, Z.; and Liu, X. 2024. Enhancing Multi-view Graph Neural Network with Cross-view Confluent Message Passing. In *Proceedings of the ACM International Conference on Multimedia*, 10065–10074.

Zhuang, S.; Wu, Z.; Chen, Z.; Dai, H.-N.; and Liu, X. 2025b. Refine then Classify: Robust Graph Neural Networks with Reliable Neighborhood Contrastive Refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 13473–13482.