
OASIS: One-Shot Federated Graph Learning via Wasserstein Assisted Knowledge Integration

Guancheng Wan^{1†}, Jiaru Qian^{1†}, Wenke Huang^{1†}, Qilin Xu¹, Xianda Guo¹,
Boheng Li², Guibin Zhang³, Bo Du¹, Mang Ye^{1*}

¹School of Computer Science, Wuhan University ²NTU ³NUS
{guanchengwan, yemang}@whu.edu.cn

Abstract

Federated Graph Learning (FGL) offers a promising framework for collaboratively training Graph Neural Networks (GNNs) while preserving data privacy. In resource-constrained environments, One-shot Federated Learning (OFL) emerges as an effective solution by limiting communication to a single round. Current OFL approaches employing generative models have attracted considerable attention; however, they face unresolved challenges: these methods are primarily designed for traditional image data and fail to capture the fine-grained structural information of local graph data. Consequently, they struggle to integrate the intricate correlations necessary and transfer subtle structural insights from each client to the global model. To address these issues, we introduce OASIS, an innovative one-shot FGL framework. In OASIS, we propose a Synergy Graph Synthesizer designed to generate informative synthetic graphs and introduce a Topological Codebook to construct a structural latent space. Moreover, we propose the Wasserstein-Enhanced Semantic Affinity Distillation (WESAD) to incorporate rich inter-class relationships and the Wasserstein-Driven Structural Relation Distillation (WDSRD) to facilitate the effective transfer of structural knowledge from the Topological Codebook. Extensive experiments on real-world tasks demonstrate the superior performance and generalization capability of OASIS, with an average improvement of 15.81% over the baseline. The code is available for anonymous access at <https://github.com/JiaruQian/OASIS>.

1 Introduction

Federated Learning (FL) [38, 19, 20] enables decentralized model training, allowing collaboration across clients while preserving privacy. Many real-world applications involve non-Euclidean data structures, such as *graphs*, where entities are interconnected through complex relationships. These graph-structured data are common in various domains, including biological networks [69], urban mobility systems [68], and online social platforms [47]. To learn from such structures from multiple participants, Graph Neural Networks (GNNs) [25, 58] have been integrated with FL, leading to Federated Graph Learning (FGL) [9, 31]. This approach combines both paradigms, ensuring privacy while enabling efficient learning on distributed graph data.

Although most research in FGL has focused on personalized learning, where each client has a tailored model [3, 73, 30], there is growing demand for a global model that can generalize across diverse graph data from multiple clients [52]. This is especially important in scenarios with limited data or where consistency between clients is necessary, such as in medical data analysis across

[†] Equal Contribution.

^{*} Corresponding Author.

hospitals or regional traffic network predictions [57, 7]. However, training a global model in traditional FGL requires multiple rounds of communication, which can be burdensome for edge devices with limited bandwidth and computational power [13, 2]. To address these challenges, *One-shot Federated Learning (OFL)* has been proposed, restricting communication to a single round [12, 34], thus reducing communication costs and potentially enhancing privacy [62]. However, existing OFL methods are primarily designed for traditional data types like images and do not tackle the unique challenges of graph-structured data. We define the task of **Generalizable One-Shot Federated Graph Learning** as developing a global model capable of generalizing well for graphs distributed across multiple clients using only one round of communication. A key research question is:

How to design a generalizable OFL framework, specifically tailored for graph-structured data?

Several OFL approaches have proposed using public datasets to train a global model. However, this approach may be impractical because of the limited availability of suitable datasets or due to stringent data sensitivity constraints [46, 36]. Consequently, generative methodologies have emerged as compelling alternatives. For instance, DENSE [66] employs a strategy in which each client trains a generator to produce synthetic data that reflects its distribution. In contrast, FedDEO [61] utilizes diffusion models as generators, leveraging client-trained descriptors to guide the server in training a global model. Nevertheless, these methods are primarily designed for image data and do not capture the fine-grained structural information inherent in nodes, particularly substructure variations such as connectivity patterns and neighbor distributions, as illustrated in Figure 2. The direct application of these generators fails to encapsulate the spatial intricacies of graphs, thereby limiting their effectiveness. This limitation raises a critical question: **I) How can client generators better capture the fine-grained structural knowledge of local graphs?**

After training the local generator, several methods have leveraged knowledge distillation (KD) on the server to train the global model [16, 41]. For example, DENSE first ensembles local models and then distills the resulting ensemble, while FedCVAE-KD [15] employs KD from local decoders to train the server decoder. However, these distillation methods depend excessively on class-related semantic signals when aggregating client knowledge, and the class space may not be sufficiently expressive to represent the diverse local graph structures of nodes. Consequently, they fail to capture the intricate correlations and struggle to transfer the clients’ structural awareness to the final global model. This observation raises an important follow-up question: **II) How can we effectively transfer subtle structural knowledge during server distillation?**

To address the aforementioned challenges, this paper presents the first systematic study on the problem of generalizable one-shot FGL. We introduce the **OASIS** framework: **O**ne-Shot Federated Gr**A**ph Learning via Wasserstein As**S**isted Knowledge Integration. To address problem **I**), we introduce the Synergy Graph Synthesizer, which is designed to generate informative synthetic graphs. This approach is further refined by incorporating alignment constraints based on the Fused Gromov-Wasserstein distance, which effectively harmonizes the reconstruction of both features and structures. More importantly, inspired by [55, 64], we construct a novel structural latent space tailored for graph data. This enables the learning of a structure-aware tokenizer that encodes each node along with its substructure as a discrete code, thereby capturing its spatial characteristics. By utilizing the established *Topological Codebook*, we can precisely characterize subtle structural variations, thereby enabling the meticulous integration of fine-grained structural knowledge from client graphs. On the server side, to address issue **II**), we first propose the Wasserstein-Enhanced Semantic Affinity Distillation, which models the rich inter-class relationships by distilling logits from the teacher local model to the

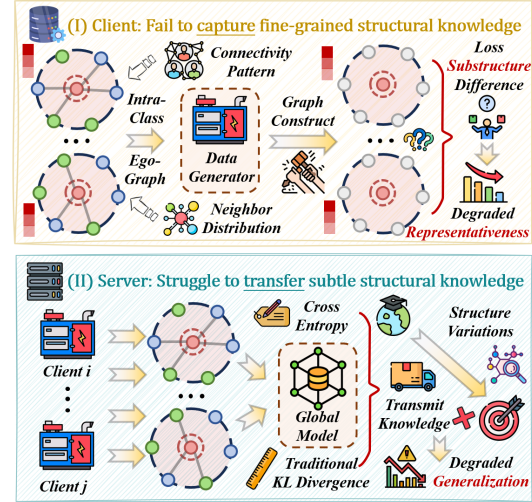


Figure 1: **Problem Illustration.** For current one-shot FGL scenarios: **I)** at the client stage, existing generative methods fail to **capture the fine-grained structural knowledge** of local graphs; **II)** at the server stage, conventional KD techniques are unable to effectively **transfer the subtle structural characteristics** from diverse clients, thereby hindering the global model generalization.

student global model, while learning more general structural differences. At the intra-class level, we introduce the Wasserstein-Driven Structural Relation Distillation, which dynamically measures the distance between samples and the discrete codes derived from the *Topological Codebook*, thereby facilitating the sophisticated transfer of subtle structural knowledge. Our principal contributions are summarized as follows:

- ❶ **Problem Identification.** We identify the key challenges in generalizable one-shot FGL: how to effectively capture fine-grained structural knowledge of local graphs, and then how to transfer this knowledge during distillation to enable global model generalization.
- ❷ **Practical Solution.** We propose a novel approach that integrates the Synergy Graph Synthesizer with Topological Codebook capturing subtle structural variations, then we introduce hierarchical Wasserstein-based distillation to skillfully transfer structural knowledge on server.
- ❸ **Experimental Validation.** We examine OASIS through extensive experiments on various graph datasets, demonstrating superior generalization ability for one-shot FGL.

2 Preliminaries

2.1 Notations

Graph Neural Networks. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of N nodes, and $\mathcal{E}$ represents the edges. Each node v_i is associated with an F -dimensional feature vector x_i , and these vectors collectively form the feature matrix $\mathbf{X} = \{x_1, x_2, \dots, x_N\}^\top$. The structure of $\mathcal{G}$ is encoded in the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, where the entries are defined such that $\mathbf{A}(i, j) = 1$ if nodes i and j are connected, and $\mathbf{A}(i, j) = 0$ otherwise. Graph Neural Networks (GNNs) iteratively build node representations by aggregating features from neighboring nodes and then applying an update function. Specifically, the representation h_i^{l+1} of node v_i at layer $l + 1$ is computed as:

$$h_i^{l+1} = \text{Update}\left(h_i^l, \text{Aggregate}\left(\{h_j^l : v_j \in \mathcal{N}(v_i)\}\right)\right), \quad (1)$$

where h_i^l is the current representation at layer l , $\mathcal{N}(v_i)$ denotes the set of neighbors of node v_i , $\text{Aggregate}(\cdot)$ combines the neighbor features, and $\text{Update}(\cdot, \cdot)$ updates the node’s representation. Initially, $h_i^0 = x_i$.

Problem Formulation. In a One-shot Federated Graph Learning framework, a central server coordinates K clients (collectively denoted as $\mathcal{C}$, where each client is indexed by k). Each client k holds its own graph $\mathcal{G}^k = (\mathcal{V}^k, \mathcal{E}^k)$ with the corresponding adjacency matrix $\mathbf{A}^k$. For every node $v_i \in \mathcal{V}^k$, there exists an associated feature vector x_i^k and a label y_i^k , if it is available. Within this setup, each client trains a model $\mathcal{F}_{\theta^k}$ parameterized by θ^k and then transmits them to the server. The objective is to minimize:

$$\min_{\theta} \sum_{k=1}^K \frac{N^k}{\mathbb{N}} L^k(\phi). \quad (2)$$

This is computed as the weighted sum of the K local nodes N^k , where $\mathbb{N}$ denotes the total number of samples across all clients. The local objective $L^k(\cdot)$ is typically defined as the expected error over all nodes from the local graph $\mathcal{G}^k$, with the global model parameterized by ϕ . Unlike conventional FGL, one-shot FGL restricts the exchange to a *single communication round*.

2.2 Motivation

This paper systematically explores the challenge of maximizing the generalization ability of the global model in one-shot FGL. Traditional OFL approaches that rely on generative models such as Variational Autoencoders or Generative Adversarial Networks—originally designed for image-based data—fail to deliver competitive performance on graphs, as demonstrated in Sec. 4. The primary issue is their inability to effectively capture the spatial and relational dynamics between nodes. These models emphasize coarse, class-based semantic signals while overlooking the underlying substructural variations among nodes. To better illustrate this, we present the t-SNE visualization of the learned graph representation space after training the teacher GNN and the Topological Codebook. (Detailed methods are illustrated in Sec. 3.3.) We select three categories with their representative nodes and corresponding two-hop substructures. Results are shown in Figure 2, where nodes within

the same class can exhibit significantly different connectivity patterns and neighbor distributions. Consequently, the key objective in the local training stage of one-shot FGL is to *identify a knowledge*

repository that optimally encapsulates both local semantics and fine-grained structural insights. We formally define this pursuit as follows:

$$\mathcal{R}_\varphi^* = \arg \min_{\varphi} \mathbb{E}_{\mathcal{G}^* \sim \mathcal{R}_\varphi(\mathcal{G}^* | \mathbf{X}, \mathbf{A}), (\mathbf{X}, \mathbf{A}) \sim \mathcal{G}} \mathbb{Q}(\mathcal{G}^*; \mathcal{G}), \quad (3)$$

Here, $\mathcal{R}_\varphi^*$ denotes the optimal knowledge repository obtained by minimizing the expected distance $\mathbb{Q}$ between the generated graph $\mathcal{G}^*$ and the original graph $\mathcal{G}$. In this formulation, $\mathcal{R}_\varphi$ is a parameterized function that produces $\mathcal{G}^*$ given the node features $\mathbf{X}$ and the adjacency matrix $\mathbf{A}$. After obtaining the powerful knowledge repository $\mathcal{R}_\varphi^*$, our goal is to *integrate these granular insights into the global model $\mathcal{F}_\phi$ during the server distillation phase without erosion of knowledge*:

$$\mathcal{F}_\phi^* = \arg \min_{\phi} \sum_{k=1}^K \mathbb{E}_{\hat{\mathcal{G}}^k \sim \mathcal{R}_\varphi^k} \mathbb{T}(\mathcal{F}_\phi(\hat{\mathcal{G}}^k); \mathcal{F}_{\theta^k}(\hat{\mathcal{G}}^k)). \quad (4)$$

For each client k , $\mathcal{R}_\varphi^k$ represents a specialized knowledge repository. The global model $\mathcal{F}_\phi$ is trained to align its outputs with those of the corresponding client models, while $\mathbb{T}$ quantifies the knowledge difference between them. Minimizing this expected transfer loss ensures that the ultimate global model $\mathcal{F}_\phi^*$ effectively assimilates the granular information encapsulated within the knowledge repositories. Based on these discussions, we present the principle for designing the ideal Generalizable one-shot FGL pipeline:

Generalizable One-shot FGL Design Principle: ***Communication Efficiency:** Achieve robust generalization via a single, streamlined communication round, thereby reducing overhead without compromising performance. **Knowledge Extraction:** Precisely capture both local semantic signals and fine-grained structural variations from graphs. **Knowledge Integration:** Seamlessly integrate these granular insights into the global model, ensuring minimal knowledge loss during the server distillation.*

In following sections, we will elaborate on how OASIS adheres to these principles, encapsulates fine-grained knowledge and then effectively distills it to the global model.

3 Methodology

3.1 Framework Overview

In this section, we present an overview of OASIS. On the client side, we perform Local Graph Knowledge Extraction, where a synergy graph synthesizer and a topological codebook are constructed to capture fine-grained structural knowledge. After the one-shot communication, we employ Wasserstein-Enhanced Semantic Affinity Distillation and Wasserstein-Driven Structural Relation Distillation to transfer intricate topological knowledge during server distillation. The framework illustration is provided in Figure 3.

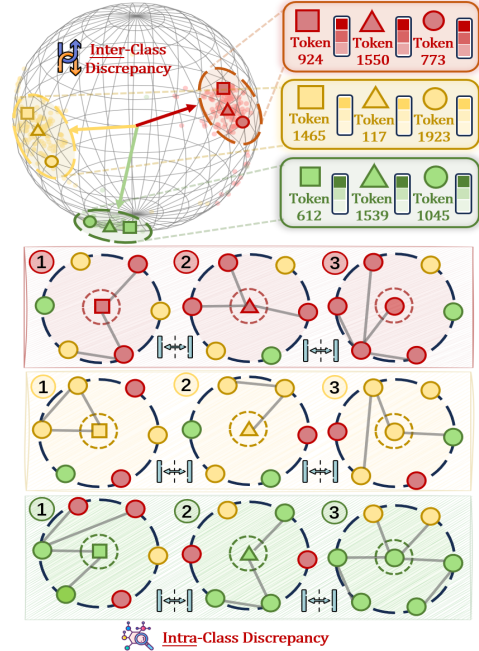


Figure 2: **Motivation.** In the t-SNE visualization, nodes from different categories (colored in red, yellow and green) are scattered, representing *inter-class discrepancy*. However, nodes within the same category are also mapped to **different** topological codebook tokens (We visualized three for each color), indicating the *intra-class discrepancy* in terms of connectivity patterns and neighbor distributions. Please refer to Sec. 2.2 for details.

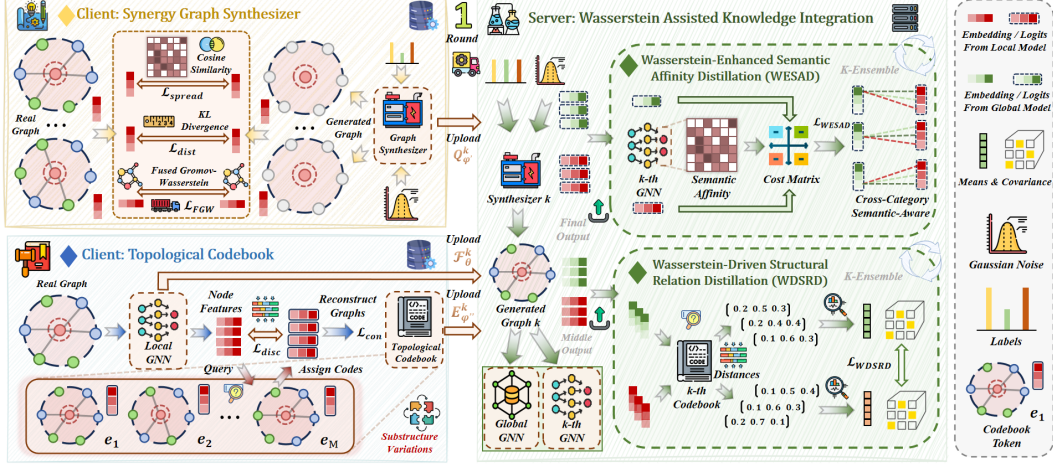


Figure 3: Architecture illustration of OASIS. (a) *The left part* shows the training process of Synergy Graph Synthesizer and Topological Codebook on the client side. (b) *The right part* presents the knowledge integration part on the server side. The communication round is limited to one, with codebooks and parameters of the local GNN and Synthesizer uploaded. The far right section displays the legend. Zoom in for details.

3.2 Local Graph Knowledge Extraction

Synergy Graph Synthesizer. To construct high-fidelity graphs, we introduce a novel Synergy Graph Synthesizer $\mathcal{Q}_{\varphi'}$ as part of the knowledge repository $\mathcal{R}_{\varphi}$ (omitting k for brevity), which reconstructs both node features and graph topology in a structured manner. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, for each node $v_i \in \mathcal{V}$, a synthetic feature vector is generated by mapping Gaussian noise ϵ (sampled from a standard Gaussian distribution) and the corresponding label y_i through the generator: $\hat{x}_i = \mathcal{Q}_{\varphi'}(\epsilon; y_i)$, where $\mathcal{Q}_{\varphi'}$ is parameterized by φ' . This process constructs the synthetic feature matrix $\hat{\mathbf{X}} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N]^\top \in \mathbb{R}^{N \times F}$. Next, an activated similarity matrix is computed as $\mathbf{H} = \sigma(\hat{\mathbf{X}}\hat{\mathbf{X}}^\top)$, where σ denotes the sigmoid function. By applying a K -Nearest Neighbors strategy to each row of $\mathbf{H}$, we obtain the synthetic adjacency matrix $\hat{\mathbf{A}}$:

$$\hat{\mathbf{A}}(i, j) = \begin{cases} 1, & \text{if } j \in \text{TopK}(\mathbf{H}(i)), \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Thus, the synthesized graph is denoted as $\hat{\mathcal{G}} = (\hat{\mathbf{X}}, \hat{\mathbf{A}})$ while preserving the original labels $y = [y_1, y_2, \dots, y_N]^\top$. We adopt a transductive setting, where some nodes in the local data are unlabeled. Therefore, we utilize only the labeled nodes in the local training set to train the synthesizer. Meanwhile, to ensure that the synthesized graph faithfully reflects both the distribution of node features and intrinsic topological patterns of the original graph, we design a two-fold consistency mechanism to compute the synthesizing objective $\mathcal{L}_{\text{syn}}$. Details can be found in Appendix B.

Topological Codebook. While the Synergy Graph Synthesizer effectively aligns global feature distributions and preserves large-scale structural patterns, it inevitably smooths out nuanced distinctions among fine-grained substructures. To address this limitation, we introduce a discrete structural latent space—termed the *Topological Codebook*—to explicitly model subtle neighborhood variations. First, each node $v_i \in \mathcal{V}$ is mapped to a continuous embedding $\mathbf{h}_i \in \mathbb{R}^D$ via an encoder. We then construct a learnable codebook $\mathbf{E}_{\varphi''} = [\mathbf{e}_1, \dots, \mathbf{e}_M] \in \mathbb{R}^{M \times D}$, where each code vector $\mathbf{e}_j$ represents a distinct local substructure pattern. For each node v_i , the discrete token corresponding to its local substructure is determined by quantizing its continuous embedding to the nearest code:

$$z_i = \arg \min_{j \in \{1, \dots, M\}} \|\mathbf{h}_i - \mathbf{e}_j\|_2^2. \quad (6)$$

This quantization process maps continuous embeddings into discrete tokens $\{z_1, \dots, z_N\}$, where each node is assigned a token corresponding to its most similar local substructure. Using these tokens, we derive the quantized representations $\mathbf{Q} = \{\mathbf{e}_{z_1}, \dots, \mathbf{e}_{z_N}\}$ for all nodes. These discrete representations enable the model to capture subtle variations in node neighborhoods more effectively.

Concurrent Optimization. To learn the Topological Codebook, we adapt the core principles of Vector Quantized-Variational Autoencoders (VQ-VAE) for graph data. The model optimizes the

following components: the reconstruction loss $\mathcal{L}_{\text{syn}}$ and two additional losses: the consistency loss and the discretization loss. The consistency loss ensures that the selected code $\mathbf{e}_{z_i}$ preserves original node attributes and topology information. Specifically, we design two decoders $p_\gamma, p_{\gamma'}$ to separately map the quantized representations to the original dimension: $\mathbf{e}'_{z_i} = p_\gamma(\mathbf{e}_{z_i})$, $\mathbf{Q}' = p_{\gamma'}(\mathbf{Q})$. Then, we introduce our consistency loss:

$$\mathcal{L}_{\text{con}} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{e}'_{z_i} - x_i\|_2^2 + \|\mathbf{A} - \sigma(\mathbf{Q}' \cdot \mathbf{Q}'^\top)\|_2^2, \quad (7)$$

where σ denotes the sigmoid function. Meanwhile, the discretization loss consists of three key components: the codebook term, the commitment term, and the orthogonal term. The codebook term ensures that the selected code $\mathbf{e}_{z_i}$ is aligned with the encoder output $\mathbf{h}_i$, maintaining the coherence between continuous and discrete representations. The commitment term encourages the encoder output to stay close to the chosen code, preventing excessive fluctuations between code vectors that could destabilize the learning process. By stabilizing the encoding process, the commitment term ensures that each node's representation remains consistent with its assigned code. The orthogonal term promotes diversity among codebook vectors by encouraging them to be independent and avoiding convergence in the same direction:

$$\mathcal{L}_{\text{disc}} = \frac{1}{N} \sum_{i=1}^N \|\text{sg}[\mathbf{h}_i] - \mathbf{e}_{z_i}\|_2^2 + \frac{\eta}{N} \sum_{i=1}^N \|\text{sg}[\mathbf{e}_{z_i}] - \mathbf{h}_i\|_2^2 + \lambda_o \left(\frac{1}{M^2} \sum_{i,j}^M \left(\frac{\mathbf{e}_i^\top \mathbf{e}_j}{\|\mathbf{e}_i\| \|\mathbf{e}_j\|} \right)^2 - \frac{1}{M} \right), \quad (8)$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator, and η, λ_o are hyperparameters controlling the strength of each term. The overall optimization objective combines these losses with a negative log-likelihood term $\mathcal{L}_{\text{NLL}}(y_i, \hat{y}_{\mathbf{e}_{z_i}})$ to maintain node-label consistency with GNN $\mathcal{F}_\theta$ predictions $\hat{y}_{\mathbf{e}_{z_i}}$:

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{syn}} + \mathcal{L}_{\text{NLL}}(y_i, \hat{y}_{\mathbf{e}_{z_i}}) + \mathcal{L}_{\text{con}} + \lambda_c \mathcal{L}_{\text{disc}}, \quad (9)$$

where λ_c is a balancing hyperparameter. By jointly optimizing these terms, we obtain a Topological Codebook $\mathbf{E}_{\varphi'}$ that captures a rich set of discrete tokens, each representing a unique local substructure. This latent structural space empowers the model to integrate localized topological knowledge with global graph structures. The concurrent optimization of the Synthesizer $\mathcal{Q}_{\varphi'}$ and the Topological Codebook leads to the creation of a refined knowledge repository $\mathcal{R}_\varphi^* = \{\mathcal{Q}_{\varphi'}^*, \mathbf{E}_{\varphi'}^*\}$, which effectively synthesizes both micro- and macro-level graph information. In the subsequent section, we explore how this extracted knowledge can be transferred to construct a generalizable global model.

3.3 Server Graph Knowledge Integration

Wasserstein-Enhanced Semantic Affinity Distillation. After local training, each client uploads its respective knowledge base $\mathcal{R}_\varphi^k = \{\mathcal{Q}_{\varphi'}^k, \mathbf{E}_{\varphi'}^k\}$ and local GNN $\mathcal{F}_\theta^k$ to the central server. At the server, for each local model k , the local GNN model $\mathcal{F}_\theta^k$ serves as the teacher, while the global GNN model $\mathcal{F}_\phi$ acts as the student. Specifically, a proxy graph $\hat{\mathcal{G}}^k$ is generated via $\mathcal{Q}_{\varphi'}^k(\epsilon; \hat{y}_{\text{uni}})$, where $\hat{y}_{\text{uni}}$ denotes a class-balanced distribution. This proxy graph retains general information from the client data distribution, thereby providing an effective signal for subsequent knowledge transfer.

A central challenge in this knowledge transfer process is minimizing the loss incurred between the teacher and student models. Traditional distillation techniques based on Kullback-Leibler divergence perform only category-to-category comparisons, failing to capture nuanced cross-category semantic affinities. To address this limitation, we propose the *Wasserstein-Enhanced Semantic Affinity Distillation (WESAD)* method, which employs the discrete Wasserstein distance to achieve a comprehensive alignment between the probability distributions of the teacher and student models. For the k -th teacher GNN model, let p_T^k denote the class probability distribution produced by $\mathcal{F}_\theta^k$ and p_S the distribution produced by $\mathcal{F}_\phi$. These distributions are computed via the softmax function σ with temperature τ . To measure their discrepancy, we define the discrete Wasserstein distance loss D_{WESAD}^k as follows:

$$D_{\text{WESAD}}^k(p_T^k, p_S) = \min_{q_{ab}} \sum_{a,b} c_{ab}^k q_{ab} + \eta q_{ab} \log q_{ab}, \quad (10)$$

where q_{ab} represents the mass transferred from the teacher's category C_a to the student's category C_b , subject to the constraints:

$$q_{ab} \geq 0, \quad \sum_b q_{ab} = p_{T,a}^k, \quad \sum_a q_{ab} = p_{S,b}^k. \quad (11)$$

Here, η is a hyperparameter controlling the entropy regularization term. A key component of this formulation is the cost matrix c_{ab}^k , which encapsulates the semantic dissimilarity between categories. We convert the semantic affinity, denoted as $\text{SA}^k(C_a, C_b)$, into a distance metric by defining

$$\text{SA}^k(C_a, C_b) = W^k[C_a][:] \otimes W^k[C_b][:], \quad (12)$$

where $\otimes$ denotes element-wise multiplication, and W^k is the weight matrix of the teacher’s projection head after l_2 normalization. The term $\text{SA}^k(C_a, C_b)$ reflects the intrinsic semantic affinity between categories in the local GNN teacher $\mathcal{F}_{\theta^k}$. We then compute the cost matrix c_{ab}^k as

$$c_{ab}^k = 1 - \kappa \text{SA}^k(C_a, C_b), \quad (13)$$

where κ controls the cost. When two categories are semantically similar, a higher $\text{SA}^k(C_a, C_b)$ results in a lower c_{ab}^k , thereby reducing the transportation cost between them; conversely, semantically dissimilar categories incur a higher cost. This design ensures that probability mass is effectively reallocated between semantically proximate categories during knowledge transfer, facilitating fine-grained semantic alignment. By minimizing $D_{\text{WESAD}}^k(p_T^k, p_S)$, the global student model $\mathcal{F}_\phi$ is not only aligned with the output distribution of each local teacher model $\mathcal{F}_{\theta^k}$ but also benefits from the incorporation of inter-class semantic correlations.

Wasserstein-Driven Structural Relation Distillation. Building upon the semantic knowledge transfer described before, we now seek to integrate the fine-grained latent structural knowledge. Unlike semantic alignment, which primarily focuses on matching class-level distributions, structural distillation emphasizes the nuanced topological patterns underlying node interactions. We introduce a *Wasserstein-Driven Structural Relation Distillation (WDSRD)* scheme that harmonizes the structural articulations by leveraging the topological codebook $\mathbf{E}_{\varphi''}^k = [\mathbf{e}_1^k, \dots, \mathbf{e}_M^k]$, where each $\mathbf{e}_m^k \in \mathbb{R}^D$ is a learnable structural token that represents a prototypical substructure. This codebook unifies latent structural information for both local and global models. Concretely, for each generated proxy graph $\hat{\mathcal{G}}^k$, we compute two sets of node-level representations:

$$\hat{\mathbf{H}}_{\text{local}}^k = \mathcal{F}_{\theta^k}(\hat{\mathcal{G}}^k) \in \mathbb{R}^{\hat{N}^k \times D}, \quad \hat{\mathbf{H}}_{\text{global}}^k = \mathcal{F}_\phi(\hat{\mathcal{G}}^k) \in \mathbb{R}^{\hat{N}^k \times D}, \quad (14)$$

where $\hat{N}^k$ denotes the number of nodes in $\hat{\mathcal{G}}^k$ and D is the feature dimension. For each node v_i , let $h_i^k \in \mathbb{R}^D$ be its representation from the local model and $h_i \in \mathbb{R}^D$ from the global model. We then compare each $\hat{h}_i^k$ (or $\hat{h}_i$) against all M codes in $\mathbf{E}_{\varphi''}^k$ to derive a soft assignment distribution:

$$\mathcal{B}_i^k = \sigma(\text{Dist}(\hat{h}_i^k, \mathbf{E}_{\varphi''}^k)/\tau), \quad \tilde{\mathcal{B}}_i^k = \sigma(\text{Dist}(\hat{h}_i, \mathbf{E}_{\varphi''}^k)/\tau), \quad (15)$$

where $\text{Dist} : \mathbb{R}^D \times \mathbb{R}^{M \times D} \rightarrow \mathbb{R}^M$ is a distance-based comparison function (e.g., Euclidean distance) that assigns each node v_i to every code $\mathbf{e}_m$ with a probability reflecting their similarity. We then obtain structural code assignments $\mathcal{B}_i^k$ from global GNN and $\tilde{\mathcal{B}}_i^k$ from local GNN over the M structural codes from k -th topological codebook $\mathbf{E}_{\varphi''}^k$. We first define the mean as the simple average of the assignment vectors across all nodes. Specifically, the local and global means are given by:

$$\mu_{\text{local}}^k = \frac{1}{\hat{N}^k} \sum_{i=1}^{\hat{N}^k} \mathcal{B}_i^k, \quad \mu_{\text{global}}^k = \frac{1}{\hat{N}^k} \sum_{i=1}^{\hat{N}^k} \tilde{\mathcal{B}}_i^k. \quad (16)$$

To characterize the dispersion of the assignments, we compute the covariance matrices as:

$$\begin{aligned} \Sigma_{\text{local}}^k &= \frac{1}{\hat{N}^k} \sum_{i=1}^{\hat{N}^k} (\mathcal{B}_i^k - \mu_{\text{local}}^k) (\mathcal{B}_i^k - \mu_{\text{local}}^k)^\top, \\ \Sigma_{\text{global}}^k &= \frac{1}{\hat{N}^k} \sum_{i=1}^{\hat{N}^k} (\tilde{\mathcal{B}}_i^k - \mu_{\text{global}}^k) (\tilde{\mathcal{B}}_i^k - \mu_{\text{global}}^k)^\top. \end{aligned} \quad (17)$$

Thus, the local and global assignments for the agent graph are approximated by the Gaussian distributions: $\mathcal{N}(\mu_{\text{local}}^k, \Sigma_{\text{local}}^k), \mathcal{N}(\mu_{\text{global}}^k, \Sigma_{\text{global}}^k)$. Inspired by the Wasserstein distance, we employ the closed-form Wasserstein distance between two Gaussian distributions. For two Gaussians, $\mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$, the Wasserstein distance is decomposed into a term measuring the difference in means and a term capturing the difference in covariances:

$$\begin{aligned} D_{\text{WD}}(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)) &= \|\mu_1 - \mu_2\|_2 \\ &+ \text{tr}\left(\Sigma_1 + \Sigma_2 - 2\left(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2}\right)^{1/2}\right). \end{aligned} \quad (18)$$

Table 1: **Comparison with the state-of-the-art methods** on eight real-world datasets. We report node classification accuracies (%) for downstream task performance. Green arrows $\uparrow$ denote advancements in accuracy metrics than FedAvg while red arrows $\downarrow$ indicate regressions. OOM means out-of-memory error. The best and second results are highlighted with **bold** and underline, respectively.

Category	Methods	Cora	CiteSeer	PubMed	Amz-Photo	Coauthor-CS	Actor	Roman-Empire	Ogbn-Arxiv
FL	FedAvg [ASTAT17]	30.61	32.88	<u>57.91</u>	23.12	22.50	<u>24.40</u>	18.49	<u>14.58</u>
	FedProx [MLSys20]	30.98 $\uparrow$ 0.37	<u>35.73</u> $\uparrow$ 2.85	50.56 $\downarrow$ 7.35	24.16 $\uparrow$ 1.04	21.44 $\downarrow$ 1.06	23.75 $\downarrow$ 0.65	15.46 $\downarrow$ 3.03	13.99 $\downarrow$ 0.59
	FedNova [NeurIPS20]	14.21 $\downarrow$ 16.40	18.58 $\downarrow$ 17.30	33.48 $\downarrow$ 24.43	6.15 $\downarrow$ 16.97	18.83 $\downarrow$ 3.67	20.04 $\downarrow$ 4.36	6.49 $\downarrow$ 12.00	1.17 $\downarrow$ 13.41
	FedRCL [CVPR24]	17.60 $\downarrow$ 13.01	12.73 $\downarrow$ 20.15	28.12 $\downarrow$ 29.79	4.92 $\downarrow$ 18.20	14.75 $\downarrow$ 7.75	10.72 $\downarrow$ 13.68	11.47 $\downarrow$ 7.02	2.56 $\downarrow$ 12.02
FGL	FedPub [ICML23]	30.52 $\downarrow$ 0.09	34.91 $\uparrow$ 2.03	41.22 $\downarrow$ 16.69	21.91 $\downarrow$ 1.21	<u>26.75</u> $\uparrow$ 4.25	22.17 $\downarrow$ 2.23	13.71 $\downarrow$ 4.78	10.02 $\downarrow$ 4.56
	FGSSL [IJCAI23]	30.23 $\downarrow$ 0.38	21.95 $\downarrow$ 10.93	39.68 $\downarrow$ 18.23	13.06 $\downarrow$ 10.06	22.44 $\downarrow$ 0.06	22.19 $\downarrow$ 2.21	14.62 $\downarrow$ 1.87	9.24 $\downarrow$ 5.34
	FedGTA [VLDB24]	14.02 $\downarrow$ 16.59	17.75 $\downarrow$ 15.13	31.45 $\downarrow$ 26.46	4.10 $\downarrow$ 19.02	10.80 $\downarrow$ 11.70	19.25 $\downarrow$ 5.15	4.55 $\downarrow$ 13.94	1.15 $\downarrow$ 13.43
	FedTAD [IJCAI24]	30.43 $\downarrow$ 0.18	33.86 $\uparrow$ 0.98	39.32 $\downarrow$ 18.59	22.01 $\downarrow$ 1.11	14.09 $\downarrow$ 8.41	23.58 $\downarrow$ 0.82	14.40 $\downarrow$ 4.09	OOM
OFL	DENSE [NeurIPS22]	12.92 $\downarrow$ 17.71	7.87 $\downarrow$ 25.01	20.84 $\downarrow$ 37.07	4.93 $\downarrow$ 18.19	3.96 $\downarrow$ 18.54	10.72 $\downarrow$ 13.68	3.90 $\downarrow$ 14.59	0.33 $\downarrow$ 14.25
	FedCVAE [ICLR23]	30.89 $\uparrow$ 0.28	34.76 $\uparrow$ 1.88	52.01 $\downarrow$ 5.90	<u>31.62</u> $\uparrow$ 8.50	14.60 $\downarrow$ 7.90	19.38 $\downarrow$ 5.02	<u>28.99</u> $\uparrow$ 10.50	13.71 $\downarrow$ 0.87
	FedSD2C [NeurIPS24]	17.78 $\downarrow$ 12.83	29.96 $\downarrow$ 2.92	26.12 $\downarrow$ 31.79	8.73 $\downarrow$ 14.39	3.88 $\downarrow$ 18.62	18.83 $\downarrow$ 5.57	3.92 $\downarrow$ 14.57	0.76 $\downarrow$ 13.82
	FENS [NeurIPS24]	<u>31.43</u> $\uparrow$ 0.82	20.97 $\downarrow$ 11.91	49.07 $\downarrow$ 18.84	25.30 $\uparrow$ 2.18	22.54 $\uparrow$ 0.04	24.30 $\downarrow$ 0.10	13.96 $\downarrow$ 4.53	13.09 $\downarrow$ 1.49
OFGL	OASIS	49.59 $\uparrow$ 18.98	45.69 $\uparrow$ 12.81	60.99 $\uparrow$ 3.08	63.73 $\uparrow$ 40.61	60.44 $\uparrow$ 37.94	25.42 $\uparrow$ 1.02	30.07 $\uparrow$ 11.58	15.05 $\uparrow$ 0.47

Accordingly, the D_{WDSRD} between the local and global assignments for the entire agent graph is

$$D_{\text{WDSRD}}^k = \left\| \mu_{\text{local}}^k - \mu_{\text{global}}^k \right\|_2 + \text{tr} \left(\Sigma_{\text{local}}^k + \Sigma_{\text{global}}^k - 2 \left(\left(\Sigma_{\text{local}}^k \right)^{\frac{1}{2}} \Sigma_{\text{global}}^k \left(\Sigma_{\text{local}}^k \right)^{\frac{1}{2}} \right)^{\frac{1}{2}} \right), \quad (19)$$

where the first term quantifies the discrepancy between the local and global means, while the second term reflects the differences in the distribution shapes as captured by the covariances. Minimizing $\mathcal{L}_{\text{WDSRD}}$ encourages the local and global GNNs to progressively align their output distributions of structural assignments, thereby achieving an effective distillation of fine-grained topological knowledge.

In our final server optimization, we integrate both the distillation losses into a single objective with addition of NLL loss on the proxy graph:

$$L_{\text{server}} = \sum_{k=1}^K \frac{\hat{N}^k}{\hat{N}} (\mathcal{L}_{\text{NLL}} + \lambda_k D_{\text{WESAD}}^k + \lambda_s \tau^2 D_{\text{WDSRD}}^k), \quad (20)$$

where $\hat{N}$ denotes the overall size of all synthesized graphs and λ_k, λ_s balance each term. This unified approach leverages the strengths of both methods: WESAD captures and aligns inter-class semantic affinities to transfer rich semantic signals, while WDSRD preserves the fine-grained topological structures inherent in local graphs. Minimizing this combined loss ensures that the global model assimilates both semantic and structural knowledge from the local teachers, resulting in a more generalizable global model.

4 Experiment

In this section, we comprehensively evaluate OASIS through four axes: **Q1** (Superiority), **Q2** (Effectiveness), **Q3** (Sensitivity), **Q4** (Privacy). The answers of **Q1-Q2** are illustrated in Sec. 4.2- Sec. 4.3 and the analyses of **Q3-Q4** can be found in Appendix K and Appendix L.

4.1 Experimental Setup

Datasets. To effectively evaluate the performance of our approach, we employed eight benchmark graph datasets of various scales and distributions, including Cora [37], CiteSeer [10], PubMed [5], Amazon-Photo, Coauthor-CS [45], Actor [42], Roman-empire [43] and Ogbn-Arxiv [17]. Detailed descriptions and splits for these datasets can be found in Appendix D.

Counterparts. We compare OASIS against several traditional FL methods: (1) **FedAvg** [ASTAT17] [38], (2) **FedProx** [MLSys20] [27], (3) **FedNova** [NeurIPS20] [54], (4) **FedRCL** [CVPR24] [44]; four popular FGL approaches: (5) **FedPub** [ICML23] [4], (6) **FGSSL** [IJCAI23] [18]; (7) **FedTAD** [IJCAI24] [73], (8) **FedGTA** [VLDB24] [30]; four One-shot FL methods: (9) **DENSE** [NeurIPS22] [66], (10) **FedCVAE** [ICLR23] [15], (11) **FedSD2C** [NeurIPS24] [67], (12) **FENS** [NeurIPS24] [1]. Detailed descriptions can be found in Appendix E.

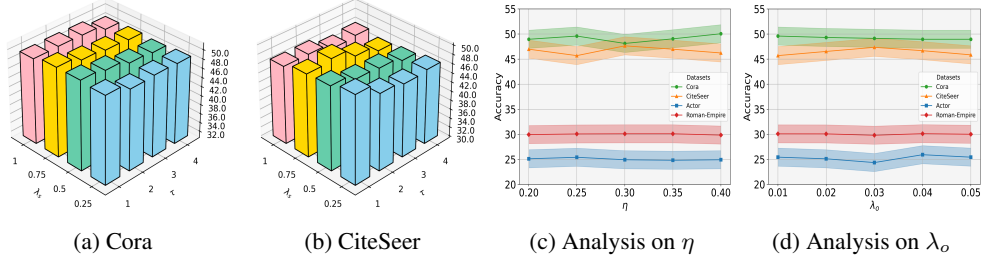


Figure 4: **Sensitivity Study** of hyperparameters. In (a) and (b), we vary τ and λ_s on Cora and CiteSeer respectively. In (c) and (d), we conduct sensitive study of η and λ_o on four datasets. Please refer to Appendix K for further analysis.

Implementation Details. We adopt a two-layer GCN as the backbone, with a hidden layer size of 128. We set $K = 10$ clients and draw $p_k \sim \text{Dir}(\alpha)$ from a Dirichlet distribution [40] and assign a fraction p_k^c of class c to client k . Specifically, α is set as 0.05 to simulate a highly non-IID senario. The codebook size is set in the range $\{2^6, 2^7, 2^8\}$. More implementation details and experiments on various client numbers can be found in Appendix F and Appendix G.

4.2 Superiority

To address **Q1**, we analyze the superior performance of **OASIS**. We demonstrate the node classification performance with various real-world graph datasets and summarize the generalized test accuracy in Tab. 1. From the table, several key observations can be made (**Obs.**): **Obs. ①** **OASIS** consistently outperforms other counterparts, with an average improvement of 15.81% over the baseline. By capturing fine-grained structural information through our synthesizer and codebook, the global model is able to acquire more intricate knowledge during distillation. **Obs. ②** Traditional FL and FGL methods heavily depend on gradual local model updates over multiple rounds, and some are tailored for personalized optimization, such as FedPub and FedGTA. Therefore, clients fail to efficiently share generalized information within one communication round, leading to the poor model performance. **Obs. ③** One-shot FL approaches such as FedCVAE and FedSD2C perform well in some small graphs. However, they excessively rely on class-related semantic signals and overlook the structural information, thus struggling with large-scale graphs.

4.3 Effectiveness

To address **Q2**, we conduct an ablation study on the key components in the Server Graph Knowledge Integration part: **Semantic Affinity Distillation** (WESAD) and the **Topological Codebook** (WESRD). The experimental results on Cora and Amazon-Photo are shown in Figure 5. From the bar chart, we can observe that both components improve the model performance significantly. WESAD employs the Wasserstein distance to align the probability distribution between the teacher (local GNN) and the student (global GNN), highlighting the incorporation of inter-class semantic correlations. Meanwhile, WDSRD integrates the fine-grained structural knowledge from the topological codebook. When both WESAD and WDSRD are combined, the performance reaches its peak, with both semantic and structural knowledge effectively distilled to the well-generalizable student global model.

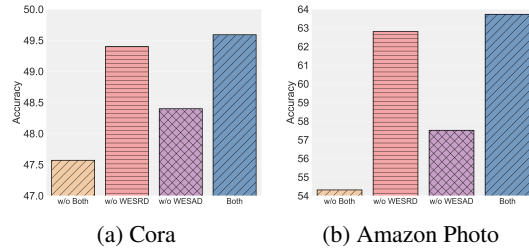


Figure 5: **Ablation Study** of the Semantic Affinity Distillation (WESAD) and Topological Codebook (WESRD) on Cora and Amazon-Photo datasets. For an in-depth analysis, please refer to Sec. 4.3.

5 Conclusion

In this paper, we introduce **OASIS** to address two key challenges in existing One-shot Federated Learning approaches with generative models: weak awareness of fine-grained structural knowledge and poor distillation capability from the topological aspect. We first establish a Synergy Graph Synthesizer to capture complex structural knowledge and then construct a structural latent space by

introducing the Topological Codebook. On the server side, we propose Wasserstein-Enhanced Semantic Affinity Distillation to model inter-class relationships and build Wasserstein-Driven Structural Relation Distillation to precisely transfer intricate topological knowledge from the codebook to the global model. Extensive experiments on diverse datasets demonstrate the effectiveness of OASIS.

Acknowledgement

This work is supported by National Natural Science Foundation of China under Grant (62361166629, 62225113, 623B2080), the Major Project of Science and Technology Innovation of Hubei Province (2024BCA003, 2025BEA002), and the Innovative Research Group Project of Hubei Province under Grants 2024AFA017. The supercomputing system at the Supercomputing Center of Wuhan University supported the numerical calculations in this paper.

References

- [1] Y. Allouah, A. Dhasade, R. Guerraoui, N. Gupta, A.-M. Kermarrec, R. Pinot, R. Pires, and R. Sharma. Revisiting ensembling in one-shot federated learning. *NeurIPS*, 37:68500–68527, 2025.
- [2] O. R. A. Almanifi, C.-O. Chow, M.-L. Tham, J. H. Chuah, and J. Kanesan. Communication and computation efficiency in federated learning: A survey. 22:100742.
- [3] J. Baek, W. Jeong, J. Jin, J. Yoon, and S. J. Hwang. Personalized subgraph federated learning. In *International conference on machine learning*, pages 1396–1415. PMLR, 2023.
- [4] J. Baek, W. Jeong, J. Jin, J. Yoon, and S. J. Hwang. Personalized subgraph federated learning. In *ICML*, pages 1396–1415. PMLR, 2023.
- [5] K. Canese and S. Weis. Pubmed: the bibliographic database. *The NCBI handbook*, 2(1), 2013.
- [6] S. Cheng, J. Wu, Y. Xiao, Y. Liu, and Y. Liu. Fedgems: Federated learning of larger server models via selective knowledge fusion, 2021.
- [7] K. Ding, X. Feng, and H. Yu. To be global or personalized: Generalized federated learning with cooperative adaptation for data heterogeneity. 301:112317.
- [8] C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.
- [9] X. Fu, B. Zhang, Y. Dong, C. Chen, and J. Li. Federated graph machine learning: A survey of concepts, techniques, and applications. *arXiv preprint arXiv:2207.11812*, 2022.
- [10] C. L. Giles, K. D. Bollacker, and S. Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, pages 89–98, 1998.
- [11] J. Gou, B. Yu, S. J. Maybank, and D. Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.
- [12] N. Guha, A. Talwalkar, and V. Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.
- [13] F. Hanzely and P. Richtárik. Federated learning of a mixture of global and local models.
- [14] C. He, K. Balasubramanian, E. Ceyani, C. Yang, H. Xie, L. Sun, L. He, L. Yang, P. S. Yu, Y. Rong, et al. Fedgraphnn: A federated learning system and benchmark for graph neural networks. In *ICLR*, 2021.
- [15] C. E. Heinbaugh, E. Luz-Ricca, and H. Shao. Data-free one-shot federated learning under very high statistical heterogeneity. In *ICLR*, 2023.
- [16] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

- [17] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.
- [18] W. Huang, G. Wan, M. Ye, and B. Du. Federated graph semantic and structural learning. 2023.
- [19] W. Huang, M. Ye, and B. Du. Learn from others and be yourself in heterogeneous federated learning. In *CVPR*, 2022.
- [20] W. Huang, M. Ye, Z. Shi, G. Wan, H. Li, B. Du, and Q. Yang. A federated learning for generalization, robustness, fairness: A survey and benchmark. *arXiv*, 2023.
- [21] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data, 2023.
- [22] Z. Kang, H. Pan, S. C. Hoi, and Z. Xu. Robust graph learning from noisy data. *IEEE transactions on cybernetics*, 50(5):1833–1843, 2019.
- [23] J. Kim, J. Kong, and J. Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *ICML*, pages 5530–5540. PMLR, 2021.
- [24] T. N. Kipf and M. Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.
- [25] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [26] D. Li, H. Qian, Q. Li, Z. Tan, Z. Gan, J. Wang, and X. Li. Fedrgl: Robust federated graph learning for label noise. *arXiv preprint arXiv:2411.18905*, 2024.
- [27] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.
- [28] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2020.
- [29] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.
- [30] X. Li, Z. Wu, W. Zhang, Y. Zhu, R.-H. Li, and G. Wang. Fedgta: Topology-aware averaging for federated graph learning. *Proc. VLDB Endow.*, 17(1):41–50, Sept. 2023.
- [31] X. Li, Y. Zhu, B. Pang, G. Yan, Y. Yan, Z. Li, Z. Wu, W. Zhang, R.-H. Li, and G. Wang. OpenFGL: A Comprehensive Benchmarks for Federated Graph Learning, Aug. 2024. *arXiv:2408.16288* [cs].
- [32] P. Liashchynskiy and P. Liashchynskiy. Grid search, random search, genetic algorithm: a big comparison for nas. *arXiv preprint arXiv:1912.06059*, 2019.
- [33] T. Lin, L. Kong, S. U. Stich, and M. Jaggi. Ensemble distillation for robust model fusion in federated learning. In *NeurIPS*, pages 2351–2363, 2020.
- [34] X. Liu, Z. Tang, X. Li, Y. Song, S. Ji, Z. Liu, B. Han, L. Jiang, and J. Li. One-shot federated learning methods: A practical guide.
- [35] Z. Lu, J. Ma, Z. Wu, B. Zhou, and X. Zhu. A noise-resistant graph neural network by semi-supervised contrastive learning. *Information Sciences*, 658:120001, 2024.
- [36] J. Luo, X. Wu, Y. Luo, A. Huang, Y. Huang, Y. Liu, and Q. Yang. Real-world image datasets for federated learning.
- [37] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3:127–163, 2000.
- [38] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282, 2017.

- [39] S. Mehta and A. Aneja. Securing data privacy in machine learning: The fedavg of federated learning approach. In *2024 4th Asian Conference on Innovation in Technology (ASIANCON)*, pages 1–5. IEEE, 2024.
- [40] T. Minka. Estimating a dirichlet distribution, 2000.
- [41] A. Mora, I. Tenison, P. Bellavista, and I. Rish. Knowledge distillation for federated learning: a practical guide. *arXiv preprint arXiv:2211.04742*, 2022.
- [42] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.
- [43] O. Platonov, D. Kuznedelev, M. Diskin, A. Babenko, and L. Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? *arXiv preprint arXiv:2302.11640*, 2023.
- [44] S. Seo, J. Kim, G. Kim, and B. Han. Relaxed contrastive learning for federated learning. In *CVPR*, pages 12279–12288, 2024.
- [45] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.
- [46] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen, and S. Bakas. Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data. 10(1):12598.
- [47] P. Shirzadian, B. Antony, A. G. Gattani, N. Tasnina, and L. S. Heath. A time evolving online social network generation algorithm. 13(1):2395.
- [48] R. Song, D. Liu, D. Z. Chen, A. Festag, C. Trinitis, M. Schulz, and A. Knoll. Federated learning via decentralized dataset distillation in resource-constrained edge environments, 2023.
- [49] J. Tan, Z. Dou, Y. Zhu, P. Guo, K. Fang, and J.-R. Wen. Small models, big insights: Leveraging slim proxy models to decide when and what to retrieve for llms. *arXiv preprint arXiv:2402.12052*, 2024.
- [50] Z. Tan, G. Wan, W. Huang, and M. Ye. Fedssp: Federated graph learning with spectral knowledge and personalized preference, 2024.
- [51] Z. Tang, Y. Zhang, P. Dong, Y. ming Cheung, A. C. Zhou, B. Han, and X. Chu. Fusefl: One-shot federated learning through the lens of causality with progressive model fusion, 2024.
- [52] G. Wan, W. Huang, and M. Ye. Federated graph learning under domain shift with generalizable prototypes. In *AAAI*, 2024.
- [53] G. Wan, Y. Tian, W. Huang, N. V. Chawla, and M. Ye. S3gcl: Spectral, swift, spatial graph contrastive learning. In *ICML*, 2024.
- [54] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *NeurIPS*, pages 7611–7623, 2020.
- [55] X. Wang, S. Takaki, J. Yamagishi, S. King, and K. Tokuda. A vector quantized variational autoencoder (vq-vae) autoregressive neural f_0 model for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:157–170, 2019.
- [56] C. Wu, F. Wu, Y. Cao, Y. Huang, and X. Xie. Fedgcn: Federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925*, 2021.
- [57] C. Wu, F. Wu, L. Lyu, T. Qi, Y. Huang, and X. Xie. A federated graph neural network framework for privacy-preserving personalization. 13(1):3091.
- [58] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE TNNLS*, pages 4–24, 2020.

- [59] H. Xie, J. Ma, L. Xiong, and C. Yang. Federated graph classification over non-iid graphs. *NeurIPS*, 34:18839–18852, 2021.
- [60] L. Yang, Y. Tian, M. Xu, Z. Liu, S. Hong, W. Qu, W. Zhang, B. Cui, M. Zhang, and J. Leskovec. Vqgraph: Rethinking graph representation space for bridging gnns and mlps. *arXiv preprint arXiv:2308.02117*, 2023.
- [61] M. Yang, S. Su, B. Li, and X. Xue. Feddeo: Description-enhanced one-shot federated learning with diffusion models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6666–6675, 2024.
- [62] X. Yao, C. Huang, and L. Sun. Two-stream federated learning: Reduce the communication costs. In *2018 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4.
- [63] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen. Graph contrastive learning with augmentations. *NeurIPS*, 33:5812–5823, 2020.
- [64] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.
- [65] H. Zhang, T. Shen, F. Wu, M. Yin, H. Yang, and C. Wu. Federated graph learning – a position paper, 2021.
- [66] J. Zhang, C. Chen, B. Li, L. Lyu, S. Wu, S. Ding, C. Shen, and C. Wu. Dense: Data-free one-shot federated learning. *NeurIPS*, 35:21414–21428, 2022.
- [67] J. Zhang, S. Liu, and X. Wang. One-shot federated learning via synthetic distiller-distillate communication. *arXiv preprint arXiv:2412.05186*, 2024.
- [68] Q. Zhang, Z. Ma, P. Zhang, and E. Jenelius. Mobility knowledge graph: Review and its application in public transport.
- [69] X.-M. Zhang, L. Liang, L. Liu, and M.-J. Tang. Graph neural networks and their current applications in bioinformatics. 12.
- [70] Y. Zhou, G. Pu, X. Ma, X. Li, and D. Wu. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999*, 2020.
- [71] Y. Zhou, G. Pu, X. Ma, X. Li, and D. Wu. Distilled one-shot federated learning, 2021.
- [72] Y. Zhu, X. Li, M. Hu, and D. Wu. Federated continual graph learning. *arXiv preprint arXiv:2411.18919*, 2024.
- [73] Y. Zhu, X. Li, Z. Wu, D. Wu, M. Hu, and R.-H. Li. Fedtad: Topology-aware data-free knowledge distillation for subgraph federated learning. *arXiv preprint arXiv:2404.14061*, 2024.

A Notations

We present a comprehensive review of the commonly used notations and their definitions in Tab. 2.

Table 2: Notation and Definitions

Notation	Definition
$\mathcal{G}$	Graph data.
$\mathcal{V}$	The node set of $\mathcal{G}$.
$\mathcal{E}$	The edge set of $\mathcal{G}$.
$\mathbf{X}$	The feature matrix of $\mathcal{G}$.
$\mathbf{A}$	The adjacency matrix of $\mathcal{G}$.
F	The dimension of the node feature.
D	The dimension of hidden embeddings and codebook tokens.
K	The number of clients.
$\hat{\mathcal{G}}^k$	The generated graph for client k .
$\hat{\mathbf{X}}$	The synthetic feature matrix.
$\mathbf{H}$	The activated similarity matrix.
$\hat{\mathbf{A}}$	The synthetic adjacency matrix.
v_i	Node i in $\mathcal{V}$.
h_i^l	The representation of v_i at the l -th layer of GNN.
$\mathcal{N}(v_i)$	The set of neighbours of node v_i .
$\mathcal{F}_{\theta^k}$	The local model of client k .
θ^k	The parameters of the local model $\mathcal{F}_{\theta^k}$ of client k .
$\mathcal{F}_{\phi}$	The global model.
$\mathcal{F}_{\phi}^*$	The ultimate global model.
ϕ	The parameters of the global model.
$\mathcal{R}_{\varphi}^*$	The optimal knowledge repository.
$\mathcal{R}_{\varphi}^k$	The specialized knowledge repository of client k .
ϵ	The Gaussian noise.
$\mathbf{E}_{\varphi''}^k$	The topological codebook of client k .
M	The number of tokens in the codebook.
z_i	The discrete token index in the codebook.
$\mathbf{Q}$	The quantized representations.
$\mathcal{Q}_{\varphi'}^k$	The Synergy Graph Synthesizer of client k .
$\hat{y}_{\text{uni}}$	The class-balanced distribution.
p_T^k	The class probability distribution produced by $\mathcal{F}_{\theta^k}$.
p_S	The class probability distribution produced by $\mathcal{F}_{\phi}$.
τ	The temperature.
C_a	The teacher’s category.
C_b	The student’s category.
c_{ab}^k	The k -th cost matrix.
q_{ab}	The mass transferred from C_a to C_b .
W^k	The weight matrix of the projection head of client k ’s teacher GNN after l_2 normalization.
$\mathcal{B}_i^k$	The structural code assignments from the teacher GNN.
$\tilde{\mathcal{B}}_i^k$	The structural code assignments from the global GNN.
$\hat{N}^k$	The size of the synthesized graph of client k .
$\hat{N}$	The overall size of all synthesized graphs.
μ_{local}^k	The local means of the assignment $\mathcal{B}_i^k$.
μ_{global}^k	The global means of the assignment $\mathcal{B}_i^k$.
Σ_{local}^k	The covariance matrix of $\mathcal{B}_i^k$.
Σ_{global}^k	The covariance matrix of $\mathcal{B}_i^k$.

B Alignment of the Synergy Graph Synthesizer

To ensure that the synthesized graph faithfully reflects both the statistical distribution of node features and the intrinsic topological patterns of the original graph, we design a two-fold consistency mechanism. First, to align the distributions of synthetic and original features, we introduce a distribution convergence term:

$$\mathcal{L}_{\text{dist}} = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \sum_{f=1}^F s_i(f) \log \frac{s_i(f)}{\hat{s}_i(f)}, \quad (21)$$

where $s_i = \text{softmax}(x_i)$ and $\hat{s}_i = \text{softmax}(\hat{x}_i)$ represent the normalized representations of the original and synthetic features, respectively. Second, to enhance local feature consistency and mitigate mode collapse, we incorporate a feature dispersion constraint:

$$\mathcal{L}_{\text{spread}} = 1 - \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \frac{\hat{x}_i \cdot x_i}{\|\hat{x}_i\|_2 \cdot \|x_i\|_2}. \quad (22)$$

Moreover, to simultaneously reconcile structural and feature discrepancies, we employ a fused Gromov-Wasserstein loss that quantifies the differences between the original and synthetic graphs within an optimal transport framework:

$$\begin{aligned} \mathcal{L}_{\text{FGW}} = \min_{\Gamma(\mu, \hat{\mu})} \sum_{i,j,u,v} & \left(a(\mathbf{A}(i,j) - \hat{\mathbf{A}}(u,v))^2 \right. \\ & \left. + (1-a)\|\mathbf{X}(i) - \hat{\mathbf{X}}(u)\|_2^2 \right) \Gamma_{i,u} \Gamma_{j,v}, \end{aligned} \quad (23)$$

where $a \in [0, 1]$ regulates the relative importance of structural and feature fidelity, and the transport plan Γ is determined under the assumption of uniform marginal distributions μ and $\hat{\mu}$. These components are integrated into a unified synthesizing objective:

$$\mathcal{L}_{\text{syn}} = \mathcal{L}_{\text{spread}} + \lambda_d \mathcal{L}_{\text{dist}} + \lambda_f \mathcal{L}_{\text{FGW}}, \quad (24)$$

where hyperparameters λ_d and λ_f control the relative contributions of each term. By leveraging the Synergy Graph Synthesizer, each client generates a refined synthetic graph. However, this approach relies on a continuous latent space and employs techniques such as KNN-based adjacency construction, which primarily capture global statistical trends and ensure overall structural consistency. Consequently, these methods tend to aggregate partial connectivity patterns, inadvertently smoothing fine-grained details and diminishing subtle local nuances.

C Related Work

Federated Graph Learning. Federated Graph Learning (FGL) extends Federated Learning (FL) to graph-structured data, enabling decentralized training while preventing the exposure of raw graph data, thus enhancing privacy protection [14, 20]. Existing FGL methods can be categorized into intra-graph FGL and inter-graph FGL [65]. Inter-graph FGL approaches such as GCFL+ [59] and FedGNN [56] consider settings where clients possess disjoint graphs, such as molecular graphs or independent social networks, and focus on training separate graph models for each client. In contrast, intra-graph FGL assumes that each client holds a subgraph of a globally connected graph, with methods such as FedSSP [50] and FGGP [52] aiming to aggregate local updates while maintaining connectivity. However, these approaches struggle to capture fine-grained structural variations and cross-client dependencies. Moreover, their dependence on iterative communication leads to high costs. To address these issues, we propose a one-shot FGL approach that removes iterative communication while preserving fine-grained structural knowledge through a novel hierarchical knowledge distillation framework, thereby improving model generalization.

One-shot Federated Learning. One-shot Federated Learning (OFL) is a paradigm that reduces communication overhead by limiting the number of communication rounds to one, making it particularly advantageous for resource-constrained and privacy-sensitive environments [12, 34]. Unlike traditional federated learning methods that require multiple iterative updates [38, 28], OFL eliminates prolonged client-server interactions, significantly reducing latency and computational overhead. Several recent methods seek to enhance OFL through generative modeling techniques, such as FedDEO [61] and Dense [66], or ensemble-based strategies, such as FuseFL [51] and FENS [1]. Additionally, knowledge distillation techniques have been introduced to facilitate more effective cross-client knowledge

transfer in OFL, as seen in FedDF [33] and FedGEMS [6]. However, existing OFL methods fail to effectively capture local semantics and fine-grained structural variations in federated graph learning. Consequently, we introduce a Synergy Graph Synthesizer to align global features while preserving large-scale structures and a Topological Codebook to model neighborhood variations, improving graph representation in one-shot FGL.

Knowledge Distillation. Knowledge Distillation (KD) is widely utilized for model compression and knowledge transfer [16], allowing smaller models to perform comparably to larger models while reducing computational demands. In federated learning, methods such as FD-FAug [21, 60] extend KD techniques to address non-independent and identically distributed (non-IID) data. Similarly, KD has been leveraged in OFL, preserving knowledge transfer, where clients distill local knowledge into a compact representation for global aggregation. Existing OFL distillation methods can be classified into data-based and model-based distillation [34]. Data-based approaches, including DOSFL [71], FedD3 [48], and FedSD2C [67], utilize synthetic data to transfer knowledge across clients. Conversely, model-based distillation methods transfer knowledge through latent feature compressions, such as Dense [66] and FedCVAE [15]. However, existing methods fail to preserve intricate graph structures, while traditional KD techniques weaken inter-class semantic affinities, limiting their effectiveness in heterogeneous FL. Thus, this study introduces a Hierarchical Wasserstein-based Distillation framework that aligns semantic affinities and preserves structural dependencies, facilitating efficient and privacy-preserving knowledge transfer.

D Dataset Details.

To assess the effectiveness of , we conduct experiments on eight real-world graph datasets: Cora, CiteSeer, PubMed, Amazon-Photo, CoAuthor-CS, Actor, Roman-Empire, and Ogbn-Arxiv. Each dataset is split into training, validation, and test sets in a fixed 20%/40%/40% ratio. The key statistics of these datasets are summarized in Tab. 3. A detailed description is provided below:

- **Cora, CiteSeer, and PubMed.** These three citation network datasets are standard benchmarks in graph-based machine learning, especially for tasks like node classification and link prediction. In these datasets, nodes correspond to academic papers, while edges represent citation links. Each node is assigned a class label, and its feature vector is constructed from textual information such as words in the title or abstract. These datasets exhibit sparsity and high dimensionality, making them well-suited for evaluating the effectiveness and scalability of graph neural networks (GNNs).
- **Amazon-Photo.** This dataset is built from the Amazon product catalog, where nodes represent product images and edges indicate co-purchase relationships. Each photo is categorized into a specific class, and node features are derived from image metadata. Amazon-Photo serves as a benchmark for testing graph-based learning models in visual domains.
- **CoAuthor-CS.** This dataset represents a co-authorship network in the field of computer science, where nodes correspond to research papers, and edges denote co-authorship relations. Each paper is associated with a topic category, and features are extracted from the paper’s title and abstract. This dataset is commonly used to evaluate node classification and community detection algorithms.
- **Actor.** The Actor dataset is a heterophilic graph where nodes represent actors, and edges indicate their co-occurrence on the same Wikipedia page. Node features are derived from textual descriptions, and classification is performed based on predefined actor categories. This dataset presents unique challenges due to its structural differences from traditional citation networks.
- **Roman-empire.** The Roman-empire dataset captures historical relationships in an ancient setting, where nodes correspond to different entities, and edges represent interactions between them. The dataset is particularly useful for studying graph-based algorithms in non-traditional network structures, offering a distinct perspective on real-world graph learning.
- **Ogbn-Arxiv.** This large-scale citation network is constructed from arXiv papers, where nodes represent papers and edges capture citation links. Each paper belongs to a specific subject category, including physics, computer science, and mathematics. Node features are extracted from paper abstracts. Ogbn-Arxiv is widely used for benchmarking GNNs due to its diversity and scale.

E Counterpart Details.

This section provides a comprehensive overview of the baseline approaches employed in our study.

Table 3: **Statistics** of datasets used in experiments.

Dataset	#Nodes	#Edges	#Classes	#Features
Cora	2,708	5,278	7	1,433
Citeseer	3,327	4,552	6	3,703
Pubmed	19,717	44,324	3	500
Amz-Photo	7,650	287,326	8	745
Coauthor-CS	18,333	327,576	15	6,805
Actor	7600	30,019	5	932
Roman-empire	22,622	65,854	18	300
Obgn-Arxiv	169,343	1,166,243	40	128

- **FedAvg** [ASTAT17]. A foundational algorithm in Federated Learning, FedAvg operates by allowing clients to independently train models on their local datasets and subsequently transmit their model updates to a central server. The server performs a weighted aggregation of these updates to refine the global model, which is then redistributed to the clients for further local training. By transmitting only model parameters instead of raw data, FedAvg reduces communication costs and enhances privacy. However, it struggles with performance degradation in scenarios where client data distributions are highly non-IID [29, 39].
- **FedProx** [MLSys20]. As an enhancement of FedAvg, FedProx is specifically designed to address the challenges posed by statistical heterogeneity in federated learning. It introduces an additional regularization term that constrains local updates, preventing excessive divergence from the global model. This proximal term mitigates the impact of local data distribution shifts, leading to more stable convergence. By ensuring consistency in updates across clients, FedProx demonstrates improved robustness in non-IID settings.
- **FedNova** [NeurIPS20]. FedNova refines the FedAvg framework by introducing normalization to local updates before aggregation. Unlike standard averaging methods, FedNova ensures that each client’s contribution to the global model is proportional to the amount of data it possesses. This approach addresses the issue of unequal client influence, leading to more balanced and efficient convergence. FedNova is particularly beneficial in federated environments where data distributions are skewed across clients.
- **FedRCL** [CVPR24]. FedRCL incorporates contrastive learning to improve federated learning performance under data heterogeneity [63, 53]. It examines inconsistencies in gradient updates across clients and attributes them to variations in feature distributions. To counteract this, FedRCL employs a contrastive regularization strategy that penalizes overly similar samples within a class, ensuring diverse and transferable feature representations. This approach enhances collaborative learning among clients and leads to notable performance improvements.
- **FedPub** [ICML23]. Unlike traditional FL methods that focus on training a single global model, FedPub adopts a personalized approach by facilitating the interaction of local Graph Neural Networks (GNNs). It employs functional embeddings to quantify similarity between client models, enabling an adaptive weighted aggregation at the server. Furthermore, a sparse mask mechanism allows clients to selectively update subgraph-relevant parameters, improving both privacy preservation and learning efficiency in heterogeneous graph scenarios.
- **FGSSL** [IJCAI23]. FGSSL addresses local client distortion caused by both node-level semantics and graph-level structures. It improves discrimination by contrasting nodes from different classes, aligning local nodes with their global counterparts of the same class while pushing them away from different classes. To handle structural information, it transforms adjacency relationships into similarity distributions and distills relational knowledge from the global model into local models. This approach preserves both structural integrity and discriminability, achieving superior performance on multiple graph datasets.
- **FedTAD** [IJCAI24]. Designed to address subgraph heterogeneity in federated learning, FedTAD decomposes variations in local graphs into differences in label distributions and structural homophily. By analyzing these discrepancies, it prevents misleading model aggregation, which can occur when local models contribute inconsistently. The framework enhances knowledge transfer through topology-aware knowledge distillation, improving both reliability and aggregation efficiency in FL settings.
- **FedGTA** [VLDB24]. FedGTA is tailored for large-scale graph federated learning, tackling issues of slow convergence and suboptimal scalability. Unlike prior methods that focus on either optimization

strategies or complex local models, FedGTA integrates topology-aware local smoothing with mixed neighbor feature aggregation to improve learning efficiency [72]. By leveraging graph structures in aggregation, it enhances scalability and performance in federated graph learning.

- **DENSE** [NeurIPS22]. A framework designed to overcome limitations of conventional one-shot FL, DENSE eliminates the need for additional auxiliary datasets or model information by employing a two-stage learning process. It first synthesizes data representations and then applies model distillation to refine the global model. This approach ensures that a federated model can be effectively trained in a single round of communication while accommodating heterogeneous client architectures.
- **FedCVAE** [ICLR23]. A data-free one-shot FL method, FedCVAE utilizes a Conditional Variational Autoencoder (CVAE) [8, 24, 23] to improve generalization under statistical heterogeneity. The approach reframes local learning objectives, allowing effective global aggregation despite distribution disparities. An extended variant, FedCVAE-KD, incorporates knowledge distillation [11] to consolidate local decoders into a unified global model. FedCVAE outperforms traditional baselines, particularly in highly heterogeneous settings.
- **FedSD2C** [NeurIPS24]. FedSD2C is a novel one-shot FL framework that mitigates performance degradation caused by data heterogeneity. It leverages a distillation-based strategy to synthesize informative data representations directly from local distributions, bypassing the inconsistency issues present in conventional model aggregation. By sharing distilled representations instead of raw model updates, FedSD2C enhances knowledge transfer and ensures greater consistency across federated clients.
- **FENS** [NeurIPS24]. FENS is a novel approach to OFL that aims to bridge the accuracy gap between standard federated learning (FL) and OFL while maintaining high communication efficiency. FENS employs a two-phase learning process: first, clients train local models and send them to the server, as in OFL; second, clients collaboratively train a lightweight prediction aggregator using FL. Extensive experiments demonstrate that FENS achieves performance close to FL while preserving the efficiency of OFL.

F Implementation Details.

The experiments are conducted using NVIDIA GeForce RTX 3090 GPUs as the hardware platform, coupled with Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz. The deep learning framework employed was Pytorch, version 2.3.1, alongside CUDA version 12.1. We adopt a two-layer GCN as the backbone, with a hidden layer size of 128. Moreover, we utilize 3 hidden linear layers and a projection head as the synergic graph synthesizer which concats random noise and one-hot label as the input and generates pseudo features as the output. We set $K = 10$ clients and draw $p_k \sim \text{Dir}(\alpha)$ from a Dirichlet distribution [40] and assign a fraction p_k^c of class c to client k . As for optimization of the graph synthesizer, Adaptive Moment Estimation (Adam) was chosen, featuring a learning rate of $5e - 3$ and a weight decay of $4e - 4$. The codebook size is set in the range $\{2^6, 2^7, 2^8\}$, with the same optimizer and learning parameter. At the local training phase, we set the training epoch T_S of the synthesizer to 100 and epoch T_C of the teacher GNN and the codebook to 50. λ_d and λ_f are determined through a grid search [32] within $\{0.01, 0.05, 0.1, 0.5\}$ and $\{0.1, 0.2, 0.5, 1\}$ respectively. η, λ_o are set as 0.25, 0.01 and λ_c is set to 1. To make sure that $\mathcal{L}_{\text{FGW}}$ is on the same scale as other loss functions for Amz-Photo and Ogbn-Arxiv datasets, we set their λ_f scales to $1e - 5$ and $1e - 7$, respectively. We set a in $\mathcal{L}_{\text{FGW}}$ as 0.5 to balance the feature part and the structure part. The communication round is limited to **one**. At the server side, we determine the global distilling epoch in the range $\{10, 20, 30\}$ and adopt Adam as the optimizer for the global model with a learning rate of $1e - 2$ and a weight decay of $4e - 4$. The synthesized graphs generated by the synthesizer of each client have the same scale $\hat{N}^k$ as the corresponding local subgraph. For the large graph Ogbn-Arxiv, we set T_C to 1 and $\hat{N}^k$ to one-tenth of the local graph. Moreover, λ_k, κ is set to 0.01, 1 and λ_s is determined in range $\{1, 5, 10\}$. The distillation temperature τ is set to 3 for all datasets.

G Ablation Study on Different Numbers of Clients.

In this section, we vary the number of clients in $\{5, 10, 20\}$ and conduct the node classification task on CiteSeer, Amazon-Photo. Experimental Results are shown in Tab. 4. From the table, we can observe that our OASIS outperforms all counterparts with different numbers of clients, demonstrating the stability of OASIS across various data distributions and subgraph scales. As we simulate a highly

Table 4: **Comparison with the state-of-the-art methods with different numbers of clients.** We report node classification accuracies (%) for downstream task performance. Green arrows $\uparrow$ denote advancements in accuracy metrics than FedAvg while red arrows $\downarrow$ indicate regressions. OOM means out-of-memory error. The best and second results are highlighted with **bold** and underline, respectively.

Datasets ($\rightarrow$)		CiteSeer			Amazon-Photo		
Category	Methods ($\downarrow$)	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients
FL	FedAvg [ASTAT17]	35.43	32.88	38.13	49.93	23.12	21.79
	FedProx [MLSys20]	40.32 $\uparrow$ 4.89	35.73 $\uparrow$ 2.85	39.39 $\uparrow$ 1.26	46.54 $\downarrow$ 3.39	24.16 $\uparrow$ 1.04	23.58 $\uparrow$ 1.79
	FedNova [NeurIPS20]	18.43 $\downarrow$ 17.00	18.58 $\downarrow$ 14.30	16.17 $\downarrow$ 21.96	9.69 $\downarrow$ 40.24	6.15 $\downarrow$ 16.97	8.57 $\downarrow$ 13.22
	FedRCL [CVPR24]	15.88 $\downarrow$ 19.55	12.73 $\downarrow$ 20.15	7.05 $\downarrow$ 31.08	21.80 $\downarrow$ 28.13	4.92 $\downarrow$ 18.20	10.69 $\downarrow$ 11.10
FGL	FedPub [ICML23]	20.07 $\downarrow$ 15.36	34.91 $\uparrow$ 2.03	32.27 $\downarrow$ 5.86	42.00 $\downarrow$ 7.93	21.91 $\downarrow$ 1.21	21.76 $\downarrow$ 0.03
	FGSSL [IJCAI23]	20.82 $\downarrow$ 14.61	21.95 $\downarrow$ 10.93	34.79 $\downarrow$ 3.34	41.64 $\downarrow$ 8.29	13.06 $\downarrow$ 10.06	22.28 $\uparrow$ 0.49
	FedGTA [VLDB24]	16.93 $\downarrow$ 18.50	17.75 $\downarrow$ 15.13	15.73 $\downarrow$ 22.40	5.39 $\downarrow$ 44.54	4.10 $\downarrow$ 19.02	6.17 $\downarrow$ 15.62
	FedTAD [IJCAI24]	19.25 $\downarrow$ 16.18	33.86 $\uparrow$ 0.98	28.34 $\downarrow$ 9.79	25.29 $\downarrow$ 24.64	22.01 $\downarrow$ 1.11	21.57 $\downarrow$ 0.22
OFL	DENSE [NeurIPS22]	7.64 $\downarrow$ 27.79	7.87 $\downarrow$ 25.01	7.06 $\downarrow$ 31.07	4.96 $\downarrow$ 44.97	4.93 $\downarrow$ 18.19	4.90 $\downarrow$ 16.89
	FedCVAE [ICLR23]	40.33 $\uparrow$ 4.90	34.76 $\uparrow$ 1.88	18.10 $\downarrow$ 20.03	52.39 $\uparrow$ 2.54	31.62 $\uparrow$ 8.50	24.87 $\uparrow$ 3.08
	FedSD2C [NeurIPS24]	20.97 $\downarrow$ 14.46	29.96 $\downarrow$ 2.92	23.66 $\downarrow$ 14.47	11.75 $\downarrow$ 38.18	8.73 $\downarrow$ 14.39	23.38 $\uparrow$ 1.59
	FENS [NeurIPS24]	21.50 $\downarrow$ 13.93	20.97 $\downarrow$ 11.91	17.95 $\downarrow$ 20.18	22.26 $\downarrow$ 27.67	25.30 $\uparrow$ 2.18	25.53 $\uparrow$ 3.74
OFGL	OASIS	40.45 $\uparrow$ 5.02	45.69 $\uparrow$ 12.81	47.92 $\uparrow$ 9.79	71.18 $\uparrow$ 21.25	63.73 $\uparrow$ 40.61	54.11 $\uparrow$ 32.82

non-IID scenario using a Dirichlet distribution with a small concentration parameter $\alpha = 0.05$, increasing the number of clients may lead to situations where some clients possess very limited data or even no training samples at all, which does not fully align with real-world settings. Nevertheless, as shown in Tab. 4, **OASIS** consistently maintains a stable and effective performance trend, even under the presence of a larger number of clients.

H Performance of OASIS under moderate or mild heterogeneity

To demonstrate the performance of **OASIS** under moderate or mild heterogeneity (or even i.i.d. data), we have conducted additional experiments with varying levels of data heterogeneity. Specifically, we evaluate the performance of **OASIS** under different values of the Dirichlet distribution parameter $\alpha \in \{1, 10, 100, 1000, 10000\}$ to simulate a range of heterogeneity scenarios. These experiments are performed on the Cora dataset, and we adopt FedAvg, FedPub, FedCVAE and FedGCN(1-hop & 2-hop) for comparison. Notably, when $\alpha=10000$, the data distribution approximates an i.i.d. setting.

Table 5: Performance of **OASIS** under milder heterogeneity (or even i.i.d.) scenarios.

α	1	10	100	1000	10000
FedAvg	49.50	56.23	57.93	57.84	45.11
FedPub	39.87	44.96	60.13	52.70	58.55
FedCVAE	60.45	56.50	59.58	55.27	56.63
FedGCN (1-hop)	52.06	41.39	50.60	51.97	56.91
FedGCN (2-hop)	58.66	52.11	55.82	57.56	61.76
OASIS	63.15	75.64	76.35	69.11	68.44

As shown Tab. 5, **OASIS** consistently outperforms the existing methods across various heterogeneity levels, including strong, moderate, mild, and even i.i.d. conditions. The advantage of **OASIS** remains robust even as the heterogeneity decreases, demonstrating that its performance is not diminished by a shift towards more homogeneous data. We will ensure that this additional experimental result is included in the revision.

The reason we choose to focus on strong heterogeneity in our experiments is that, in real-world scenarios, non-uniform data distribution is more prevalent and presents a greater challenge. We aim to showcase that **OASIS** not only performs well in typical i.i.d. and mild heterogeneity settings but also excels in more challenging scenarios with high heterogeneity.

I Performance of OASIS on other prominent GNN models

To demonstrate the performance of **OASIS** on other GNN models, we conducted experiments on the Cora and CiteSeer datasets using GAT and GraphSAGE respectively. We compare **OASIS** against five baselines. Results are shown in Tab. 6.

Table 6: The performance of OASIS on GAT and GraphSAGE.

Methods	Cora-GAT	Cora-GraphSAGE	CiteSeer-GAT	CiteSeer-GraphSAGE
FedAvg	30.98	29.88	30.49	37.23
FedProx	31.71	29.97	27.94	39.40
FGSSL	29.79	27.45	20.52	20.22
FedCVAE	29.61	25.94	22.05	23.52
FENS	29.70	29.88	21.05	21.50
OASIS	42.62	39.96	38.73	44.87

From the table, we can observe that our **OASIS** consistently outperforms other baselines in both GAT and GraphSAGE backbones, demonstrating the capability of **OASIS** to generalize to other prominent GNN models.

J Mathematical Analysis of OASIS

Here we provide a thorough analysis on the mathematical bound on information retention of the synthesizer and the impact of global distillation.

J.1 Mathematical Bound on Information Retention

Let $\mathcal{G}_l = (\mathcal{V}_l, \mathcal{E}_l, X_l)$ be the local graph, where $\mathcal{V}_l$ is the node set, $\mathcal{E}_l$ the edge set, and $X_l \in \mathbb{R}^{|\mathcal{V}_l| \times d}$ the node feature matrix. The synthesized graph is $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s, X_s)$, with $X_s \in \mathbb{R}^{|\mathcal{V}_s| \times d}$ and adjacency matrix A_s . The goal is to ensure that $\mathcal{G}_s$ preserves the critical information in $\mathcal{G}_l$, including node feature distributions and topological structures.

To quantify the retention of critical information, we derive a bound on the divergence between the local graph $\mathcal{G}_l \sim \mathcal{D}_l$ and the synthesized graph $\mathcal{G}_s \sim \mathcal{D}_s$, using the Synthesizer’s loss as a proxy. We measure the divergence between their distributions using a combined metric that accounts for both feature and topological differences.

Step 1: Feature Distribution Divergence. The KL divergence loss $L_{\text{dist}} = D_{\text{KL}}(P_{X_s} \| P_{X_l})$ directly measures the feature distribution mismatch. By Pinsker’s inequality, the total variation distance is bounded by:

$$\delta_{\text{TV}}(P_{X_s}, P_{X_l}) \leq \sqrt{\frac{1}{2} D_{\text{KL}}(P_{X_s} \| P_{X_l})} = \sqrt{\frac{1}{2} L_{\text{dist}}}.$$

Assuming $L_{\text{dist}} \leq \epsilon_d$, the feature distributions are close in total variation:

$$\delta_{\text{TV}}(P_{X_s}, P_{X_l}) \leq \sqrt{\frac{\epsilon_d}{2}}.$$

Step 2: Feature and Topological Alignment via FGW. The Fused Gromov-Wasserstein loss L_{fgw} bounds the combined feature and topological mismatch. The FGW distance can be decomposed as:

$$L_{\text{fgw}} = \text{FGW}(\mathcal{G}_l, \mathcal{G}_s) = \alpha W_2^2(P_{X_l}, P_{X_s}) + (1 - \alpha) \text{GW}(A_l, A_s),$$

where W_2^2 is the squared 2-Wasserstein distance between feature distributions, and GW is the Gromov-Wasserstein distance between structures. Assuming $L_{\text{fgw}} \leq \epsilon_f$, we have:

$$\alpha W_2^2(P_{X_l}, P_{X_s}) + (1 - \alpha) \text{GW}(A_l, A_s) \leq \epsilon_f.$$

Since both terms are non-negative, we get:

$$W_2(P_{X_l}, P_{X_s}) \leq \sqrt{\frac{\epsilon_f}{\alpha}}, \quad \text{GW}(A_l, A_s) \leq \frac{\epsilon_f}{1-\alpha}.$$

The Wasserstein distance further bounds the feature distribution divergence, and the Gromov-Wasserstein distance bounds the structural discrepancy.

Step 3: Feature Diversity. The cosine similarity loss L_{spread} ensures that the synthesized features do not collapse. We assume $L_{\text{spread}} \geq -\eta$, where $\eta > 0$ is a constant reflecting sufficient feature spread. This prevents degenerate solutions where $X_s^i \approx X_s^j$ for all i, j .

Step 4: Combined Bound. To derive a bound on the overall divergence between $\mathcal{D}_l$ and $\mathcal{D}_s$, we consider a joint metric that combines feature and topological differences. Define a graph divergence metric:

$$\Delta(\mathcal{G}_l, \mathcal{G}_s) = W_2(P_{X_l}, P_{X_s}) + \text{GW}(A_l, A_s).$$

From the FGW loss:

$$\Delta(\mathcal{G}_l, \mathcal{G}_s) \leq W_2(P_{X_l}, P_{X_s}) + \text{GW}(A_l, A_s) \leq \sqrt{\frac{\epsilon_f}{\alpha}} + \frac{\epsilon_f}{1-\alpha} = \epsilon_f \left(\sqrt{\frac{1}{\alpha}} + \frac{1}{1-\alpha} \right).$$

The KL divergence provides an additional constraint on features. Combining with Pinsker's inequality, the total variation distance on features is:

$$\delta_{\text{TV}}(P_{X_s}, P_{X_l}) \leq \sqrt{\frac{\epsilon_d}{2}}.$$

Since Wasserstein and total variation distances are related (e.g., via transport inequalities in bounded spaces), we focus on the FGW-based bound for simplicity, as it captures both features and topology.

Theorem 1. Assuming the Synthesizer's loss is bounded as $L_{\text{syn}} \leq \epsilon$, with $L_{\text{dist}} \leq \epsilon_d$, $L_{\text{fgw}} \leq \epsilon_f$, and $L_{\text{spread}} \geq -\eta$, the divergence between the local and synthesized graph distributions is bounded as:

$$\Delta(\mathcal{D}_l, \mathcal{D}_s) \leq \epsilon_f \left(\sqrt{\frac{1}{\alpha}} + \frac{1}{1-\alpha} \right) + \sqrt{\frac{\epsilon_d}{2}},$$

where $\Delta(\mathcal{D}_l, \mathcal{D}_s) = \mathbb{E}_{\mathcal{G}_l \sim \mathcal{D}_l, \mathcal{G}_s \sim \mathcal{D}_s} [W_2(P_{X_l}, P_{X_s}) + \text{GW}(A_l, A_s)]$.

Proof. The total synthesis loss is:

$$L_{\text{syn}} = L_{\text{spread}} + \lambda_d L_{\text{dist}} + \lambda_f L_{\text{fgw}} \leq \epsilon.$$

Since $L_{\text{spread}} \geq -\eta$, we have:

$$\lambda_d L_{\text{dist}} + \lambda_f L_{\text{fgw}} \leq \epsilon + \eta.$$

Assume $L_{\text{dist}} \leq \epsilon_d$, $L_{\text{fgw}} \leq \epsilon_f$, with $\lambda_d \epsilon_d + \lambda_f \epsilon_f \leq \epsilon + \eta$. The feature divergence is bounded via Pinsker's inequality:

$$\delta_{\text{TV}}(P_{X_s}, P_{X_l}) \leq \sqrt{\frac{\epsilon_d}{2}}.$$

The FGW loss bounds the combined feature and topological divergence:

$$\alpha W_2^2(P_{X_l}, P_{X_s}) + (1-\alpha) \text{GW}(A_l, A_s) \leq \epsilon_f.$$

Thus:

$$W_2(P_{X_l}, P_{X_s}) \leq \sqrt{\frac{\epsilon_f}{\alpha}}, \quad \text{GW}(A_l, A_s) \leq \frac{\epsilon_f}{1-\alpha}.$$

The total divergence is:

$$\Delta(\mathcal{D}_l, \mathcal{D}_s) \leq \mathbb{E}[W_2(P_{X_l}, P_{X_s}) + \text{GW}(A_l, A_s)] \leq \sqrt{\frac{\epsilon_f}{\alpha}} + \frac{\epsilon_f}{1-\alpha}.$$

Incorporating the feature distribution bound from L_{dist} , we add the total variation term for completeness, yielding:

$$\Delta(\mathcal{D}_l, \mathcal{D}_s) \leq \epsilon_f \left(\sqrt{\frac{1}{\alpha}} + \frac{1}{1-\alpha} \right) + \sqrt{\frac{\epsilon_d}{2}}.$$

This bound quantifies the retention of critical information: a small ϵ_f and ϵ_d ensure that the synthesized graph’s feature and topological distributions are close to those of the local graph.

The divergence between the local and synthesized graph distributions is bounded by:

$$\Delta(\mathcal{D}_l, \mathcal{D}_s) \leq \epsilon_f \left(\sqrt{\frac{1}{\alpha}} + \frac{1}{1-\alpha} \right) + \sqrt{\frac{\epsilon_d}{2}},$$

where $\Delta(\mathcal{D}_l, \mathcal{D}_s)$ combines feature and topological differences. This bound rigorously quantifies the retention of critical information, ensuring that the synthesized graph effectively captures the local graph’s semantic and structural properties when ϵ_d and ϵ_f are small.

J.2 Impact on Global Model Generalization

The retention of critical information and the distillation mechanisms directly influence the generalization performance of the global model. Our OASIS employs two distillation mechanisms to transfer local knowledge to the global model:

- **WESAD**: Transfers inter-class semantic relationships.
- **WDSRD**: Transfers fine-grained topological structures.

Generalization Bound. To quantify the impact on generalization, we derive a bound on the global model’s expected error using the Rademacher complexity framework, adapted for federated graph learning.

Let $\mathcal{D}_l$ denote the local data distribution for client l , and $\mathcal{D}_s$ the distribution of the synthesized data. The global model $f_g \in \mathcal{F}$ (a hypothesis class of graph neural networks) is trained on synthesized data to minimize the empirical risk:

$$\hat{R}_s(f_g) = \mathbb{E}_{\mathcal{D}_s}[\ell(f_g(\mathcal{G}_s), y)],$$

where ℓ is the loss function (e.g., cross-entropy), and y is the label. The true risk is:

$$R(f_g) = \mathbb{E}_{\mathcal{D}_l}[\ell(f_g(\mathcal{G}_l), y)].$$

The generalization gap is $R(f_g) - \hat{R}_s(f_g)$. Assuming the synthesized graph retains critical information (i.e., $W_1(\mathcal{D}_l, \mathcal{D}_s) \leq \epsilon$), we bound the generalization error using the Wasserstein distance and Rademacher complexity.

Theorem 2. For a hypothesis class $\mathcal{F}$ with Rademacher complexity $\mathcal{R}_n(\mathcal{F})$ over n samples, and assuming the loss function ℓ is L -Lipschitz, the expected generalization error of the global model is bounded as:

$$\mathbb{E}[R(f_g)] \leq \hat{R}_s(f_g) + 2L\mathcal{R}_n(\mathcal{F}) + L\epsilon + C\sqrt{\frac{\log(1/\delta)}{n}},$$

with probability at least $1 - \delta$, where $\epsilon = W_1(\mathcal{D}_l, \mathcal{D}_s)$ is the Wasserstein distance between local and synthesized distributions, and C is a constant.

The term $L\epsilon$ quantifies the impact of information retention: if ϵ is small (i.e., the synthesized graph closely matches the local graph), the generalization error is tightly bounded.

Proof. By the Wasserstein distance property, the difference in expected loss is bounded:

$$|\mathbb{E}_{\mathcal{D}_l}[\ell(f_g(\mathcal{G}_l), y)] - \mathbb{E}_{\mathcal{D}_s}[\ell(f_g(\mathcal{G}_s), y)]| \leq LW_1(\mathcal{D}_l, \mathcal{D}_s) = L\epsilon.$$

Using standard generalization bounds for empirical risk minimization:

$$\mathbb{E}[R(f_g)] \leq \hat{R}_s(f_g) + 2\mathcal{R}_n(\mathcal{F}) + C\sqrt{\frac{\log(1/\delta)}{n}}.$$

Combining these, we obtain:

$$\mathbb{E}[R(f_g)] \leq \hat{R}_s(f_g) + 2L\mathcal{R}_n(\mathcal{F}) + L\epsilon + C\sqrt{\frac{\log(1/\delta)}{n}}.$$

K Sensitivity

To address **Q3**, we conduct analyses on hyperparameters of **OASIS**. Specifically, we compare the model performance under different values of temperature τ and λ_c . We vary τ and λ_c in range $[1, 4]$ and $[0.25, 1]$ with 1 and 0.25 as the step size respectively. Moreover, as for the Topology Codebook, we vary η and λ_o in range $[0.2, 0.4]$ and $[0.01, 0.05]$ with 0.05 and 0.01 as the step size. Results shown in Figure 4 demonstrate that the performance of **OASIS** stays consolidated under different hyperparameter values, proving the robustness of **OASIS**.

L Privacy Security

L.1 How to prevent malicious attackers from stealing communication data?

Privacy security plays a crucial role in FGL systems. In Sec. 3.2, we propose a novel Synergy Graph Synthesizer $\mathcal{Q}_{\phi'}$ to generate powerful synthesized graphs with labels and Gaussian noise ϵ . For simplicity, a standard normal distribution is typically utilized, where the center Ω is set to 0. However, we introduce a shift ς to the center Ω at the client side and communicate ς to the server either offline or through encryption methods [15, 70]. We consider the worst-case scenario where an eavesdropping attacker intercepts all the parameters. However, without knowledge of the specific shift ς , the attacker can only utilize the original center $\hat{\Omega}$, which is distant from the true center ($\Omega + \varsigma$). Alternatively, the attacker might attempt to overlap the center using a wide uniform distribution.

To simulate these situations, we first set the original noise distribution center $\Omega = 0$ with shift ς varies in $\{30, 60, 90\}$ while the attacker still takes $\hat{\Omega} = 0$. We conduct all experiments here on the node classification task with the knowledge distillation part excluded to explicitly demonstrate the influence of the generated data. Results are shown in Figure 6a. From the bar chart, we can observe that the model performance sharply declined by 60%, proving that the data generated with center $\hat{\Omega}$ are totally different from local data. Moreover, to simulate the overlapping attempt, we set $\Omega + \varsigma = 0$ and the overlap range in $\{\mathcal{U}(-100, 100), \mathcal{U}(-200, 200), \mathcal{U}(-300, 300)\}$ and conduct the experiment in the same setting. Results are shown in Figure 6b, with the same phenomenon observed. Therefore, we prove the security of data privacy of our **OASIS**.

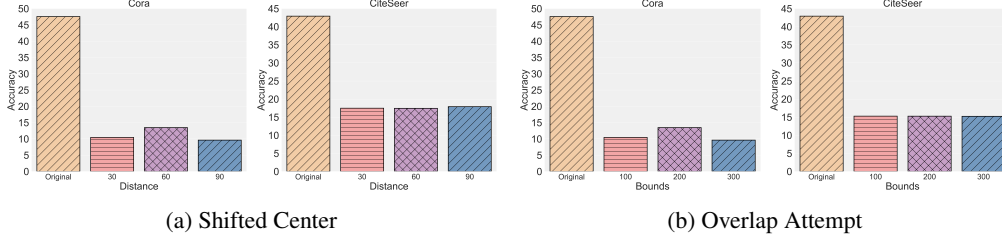


Figure 6: **Privacy Study** of OASIS. **Original** here means the server generates the graphs with the same distribution center as clients. For an in-depth analysis, please refer to Appendix L.1.

L.2 How to prevent a curious server from inspecting client privacy?

Although we design an encrypted shift ς for the distribution center Ω , a curious server can still utilize the correct distribution to synthesize data. However, the true local data information remains protected and is not exposed to the server, due to the following reasons:

(1) Discrepancy in Label Distributions. In practical FGL scenarios, the label distribution across clients is typically highly non-identical. In our experiments, we simulate an extremely non-IID setting by sampling labels from a Dirichlet distribution with concentration parameter $\alpha = 0.05$. Importantly, clients do not transmit their actual label distribution to the server. Instead, the data synthesis is conducted under the assumption of a uniform label distribution $\hat{y}_{\text{uni}}$ (Sec. 3.3). Consequently, the overall label distribution of the synthesized graph on the server deviates significantly from that of any individual client, thereby mitigating the risk of direct data leakage.

(2) Structural Dissimilarity of Nodes. Even for synthesized nodes that share the same label as some local data, their structural context (e.g., neighborhood connectivity) is distinct. As described in Equation (5), each node in the synthesized graph is connected to its five most similar nodes based on feature similarity. This fixed K-nearest-neighbor (KNN) construction introduces structural differences compared to the true graph topology of local data. Although our alignment module (Appendix B) encourages the preservation of structural semantics via $\mathcal{L}_{\text{FGW}}$, it does not enforce strict local topological isomorphism. Therefore, structural privacy is preserved to a considerable degree.

(3) Feature Perturbation via Two-Fold Consistency Mechanism. The synthesizer is trained with a two-fold consistency objective $\mathcal{L}_{\text{syn}}$ involving both feature and structure alignments (Equation (24)). This objective steers the optimization away from directly replicating raw node features. As a result, even nodes with the same labels as in local data will exhibit distinct feature representations in the synthesized graph.

In summary, privacy preservation in our framework is inherently balanced with the learning objectives of the Synergy Graph Synthesizer. Through the structural regularization from the KNN construction and the two-fold consistency mechanism, we ensure that the synthesized data avoids leaking sensitive client information, while still capturing high-level latent knowledge required for effective global model learning. This design satisfies the fundamental goals of federated graph learning.

M Discussion on Limitations.

While OASIS achieves notable success in efficiently capturing fine-grained structural knowledge of local graphs and effectively transfer the knowledge during server distillation in the one-shot scenario, it still has inherent limitations as a synthesizer-based method [49]. In particular, the presence of noise in local data [22] can impair the ability of the synthesizer to effectively extract and learn local-specific patterns, which may in turn impede the distillation module. Improving the robustness of the Synergy Graph Synthesizer against such noise interference [35, 26] remains a promising direction for future work.

N Discussion on Broader Impacts

Our proposed OASIS framework contributes to the broader field of FGL by enabling more efficient and privacy-preserving collaboration across decentralized graph datasets, especially under stringent communication constraints. By introducing domain-specific generative strategies and novel structural distillation techniques, OASIS opens new possibilities for applying OFL to graph-structured data such

as social networks, biomedical graphs, and knowledge graphs, where data are inherently sensitive and dispersed. This advancement can benefit applications involving privacy-critical domains like healthcare, finance, and cybersecurity, empowering institutions to jointly learn high-quality models without exposing private graph data. As FL technologies become more widely adopted, continued attention to the ethical implications of synthetic data generation and the interpretability of structural knowledge transfer will be essential to fostering responsible AI deployment.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The main claims made in the abstract and introduction accurately reflect the contributions and scope of this paper.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

We discuss the limitations in Appendix M.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We fully disclose all the information needed to reproduce the main experimental results in this paper and our code. We are convinced that the obtained results can be reproduced.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is accessible in this paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Details are included in Appendix F.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Statistical significance of the experiments is considered and included in Sec. 4.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide sufficient information on the computer resources.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in the paper strictly adheres to the NeurIPS Code of Ethics, ensuring that all aspects of the work are in compliance with the guidelines provided.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Broader Impacts are included in Appendix N.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: This paper does not use existing assets.

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.