HYPERION: Fine-Grained Hypersphere Alignment for Robust Federated Graph Learning

Guancheng Wan^{1†}, Xiaoran Shang^{1†}, Yuxin Wu^{4†},
Guibin Zhang^{2}, Jinhe Bi^{3}, Liangtao Zheng^{5} ,
Xin Lin^{5}, Yue Liu^{2}, Yanbiao Ma^{4}, Wenke Huang^{1*}, Bo Du^{1}
^{1}Wuhan University ^{2}NUS ^{3}Ludwig-Maximilians-Universität München
^{4}Renmin University of China ^{5}UCSD
guanchengwan@whu.edu.cn

Abstract

Robust Federated Graph Learning (FGL) provides an effective decentralized framework for training Graph Neural Networks (GNNs) in noisy-label environments. However, the subtlety of noise during training presents formidable obstacles for developing robust FGL systems. Previous robust FL approaches neither adequately constrain edge-mediated error propagation nor account for intra-class topological differences. At the client level, we innovatively demonstrate that hyperspherical embedding can effectively capture graph structures in a fine-grained manner. Correspondingly, our method effectively addresses the aforementioned issues through fine-grained hypersphere alignment. Moreover, we uncover undetected noise arising from localized perspective constraints and propose the geometric-aware hyperspherical purification module at the server level. Combining both level strategies, we present our robust FGL framework, HYPERION, which operates all components within a unified hyperspherical space. HYPERION demonstrates remarkable robustness across multiple datasets, for instance, achieving a $29.7\% \uparrow$ F1-macro score with 50%-pair noise on Cora. The code is available at: https://github.com/GuanchengWan/HYPERION.

1 Introduction

Federated Learning (FL) [20, 34] has recently emerged as a key area in decentralized machine learning. FL enables multiple clients to collaboratively train a shared global model while preserving data privacy [65, 20]. To leverage graph-structured data from diverse participants, Graph Neural Networks (GNNs) [22, 13, 31] have been integrated into FL, giving rise to Federated Graph Learning (FGL) [59, 49, 16, 45]. FGL combines two paradigms, effectively ensuring privacy[17, 19, 50] while enabling efficient distributed graph learning through neural message-passing mechanisms, which propagate node features and hidden representations in graph data.

As shown in Figure 1, although FGL offers numerous benefits [12, 2], it also introduces new vulnerabilities. Prior studies demonstrate that even minor structural or semantic perturbations can lead to misclassification in GNNs [7, 57, 68, 72]. These subtle differences may obscure critical information that defines node relationships and class boundaries. In FGL, coarse-grained representations of nodes within the same class can forcibly smooth out local topological differences, impeding the effective filtering of subtle noise and hindering the accurate capture of real semantic information and the underlying graph structure. Hence, we pose the following question: **I)** ***How can we learn class representations that are robust to noise while capturing subtle structural differences between***

† Equal Contribution.
* Corresponding Author.

similar nodes? Such noise is not only inherently difficult to detect but also pervasive in graph data. Studies show that existing datasets can easily contain over 30% label errors [23, 41]. Recent FL methods address label noise via label correction [47, 60] and self-supervised learning [10, 58, 8], but these methods do not explicitly model the complex topological characteristics of graph data. Therefore, when dealing with graph data exhibiting complex topological structures, these approaches typically aggregate neighbor features indiscriminately, mixing noise with valid signals and degrading both alignment level and generalization performance. This leads to the question: **II)** ***How can we adaptively identify and select high-confidence, stable nodes in each client's noisy graph data?***

Nevertheless, it is extremely difficult to completely remove noisy nodes solely relying on the client. The federated framework's privacy constraints limit each client's view to its local subgraph, preventing a global perspective. As a result, client models learn only local semantics and limited topological context, causing certain abnormal nodes to appear "normal" locally and evade detection. The limitation of this local perspective indirectly damages the generalization ability of the global model. Therefore, we ask: **III)** ***How can we robustly refine semantic and topological knowledge during global aggregation and transfer it efficiently to the global model?***

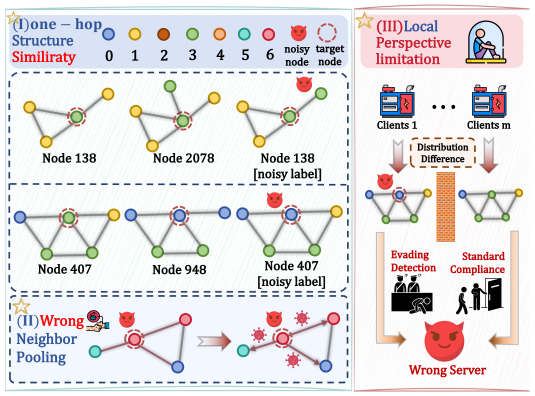


Figure 1: **Problem Illustration**. We describe the challenges FGL encounters under noisy labels: **I)** The coarse-grained representation method leads to ineffective differentiation between similar nodes, resulting in the coupling of noise and valid information. **II)** The edges between nodes in a graph facilitate the propagation of noise. **III)** The restricted view of individual clients leads to missed detection of certain noise.

To holistically address these challenges, we propose <u>`HYPERION`</u>: a **Hyperspherical-Embedding-Centric** Framework for Robust Federated Graph Learning, where all components operate on a unified hyperspherical space. To address issue **I)**, we introduce **Topological Prototypes <u>Hyperspherical</u> Learning (TP-HSL)** to fully capture the rich topological differences between nodes of the same class. Our method projects training node samples onto a hyperspherical embedding space. On the one hand, it maximizes the minimum spherical angle between different class prototype clusters, actively amplifying inter-class differences and enhancing the discriminability of decision boundaries. On the other hand, it minimizes the average spherical angle between nodes of the same class and their prototype centers to ensure tight intra-class clustering and strengthen structural correlations. Compared to conventional one-class-one-prototype approaches [46, 18, 51], TP-HSL provides finer structural modeling capabilities and heightened topological sensitivity. To address problem **II)**, we propose **<u>Hyperspherical</u> Consistency Noise Calibration (HS-CNC)**, which constructs a perturbed view of the graph. We retain only high-confidence nodes that consistently map to the same prototype cluster across views and filter out potential noisy nodes with unstable mappings. This process explicitly exposes the potential perturbation-sensitive areas through the "amplification of noise shifts" effect, ensuring information purity while effectively strengthening the correlation between nodes and the true topological structure. To solve issue **III)**, we propose **Geometric-Aware <u>Hyperspherical</u> Purification (GA-HSP)**. Driven by the Wasserstein distance, we distill, refine, and aggregate local prototype knowledge into robust global prototypes, constructing a well-defined global hypersphere. Then, by leveraging the covariance structure between the client hyperspheres and the global hypersphere, we apply the Mahalanobis distance to assess each node's outlier risk and eliminate drifting nodes biased by noise. To summarize, we make the following key contributions:

❶ ***Problem Identification.*** We study a challenging problem: overcoming the label noise in FGL. Our focus is on mitigating the negative influence of the label noise while overcoming several key limitations of existing solutions, including reliance on coarse-grained representations, neglect of the graph data's topological structure, and the absence of re-correction from a global perspective.

❷ ***Practical Solution.*** We introduce `HYPERION`, a novel and effective methodology that disentangles complex topological structures and mitigates malicious noise in FGL through hyperspherical representation. With the help of several technical innovations, `HYPERION` significantly enhances the model's ability to distinguish subtle structural differences while maintaining strong robustness.

❸ ***Experimental Validation.*** We conducted extensive experiments on five mainstream datasets under different noise types and ratios. The results demonstrate that our approach outperforms the state-of-the-art methods in multiple FGL environments. For instance, under the 50%-pair noise setting on the Cora dataset, our method achieves an impressive F1-macro score of 51.15%, outperforming the second-best method of 39.41% by a significant margin.

2 Preliminaries

Notations. Following the typical FGL framework, M participants (indexed by m) collaboratively train a shared global model using their private graph data. Participant m holds a graph $\mathcal{G}_m = (\mathcal{V}_m, \mathcal{A}_m, \mathcal{X}_m)$, where $\mathcal{V}_m = \{v_i\}_{i=1}^{N_m}$ is the node set containing $|\mathcal{V}_m| = N_m$ nodes, $\mathcal{A}_m \in \{0,1\}^{N_m \times N_m}$ is the adjacency matrix with $A_{ij} = 1$ if there is an edge between nodes v_i and v_j (and 0 otherwise), and $\mathcal{X}_m = \{x_i\}_{i=1}^{N_m}, x_i \in \mathbb{R}^d$ is the node feature set of dimension d. Moreover, $\mathcal{Y}_m \in \{0,1\}^{N_m \times C}$ is the label matrix, where each label $y_i \in \{0,1\}^C$ is a one-hot vector over C classes. See Appendix A for detailed notation.

Problem Formulation. We focus on the semi-supervised node classification problem. Only a small set of nodes $\mathcal{V}_m^L$ is labeled for training, denoted as $\mathcal{V}_m^L = \mathcal{V}_m \setminus \mathcal{V}_m^U$, where N_m^L is the number of labeled nodes. The remaining nodes are unlabeled and denoted as $\mathcal{V}_m^U$. Given $\mathcal{X}_m$ and $\mathcal{A}_m$, the goal of node classification is to train a classifier $f_{\theta_m} : (\mathcal{X}_m, \mathcal{A}_m) \to \hat{\mathcal{Y}}_m^L$, where the model parameters are optimized by minimizing the following objective:

$$\min_{\theta_m} \mathcal{L}(f_{\theta_m}(\mathcal{X}_m, \mathcal{A}_m), \mathcal{Y}_m^L), \tag{1}$$

where $\mathcal{L}$ is a loss function that measures the discrepancy between predictions and ground-truth labels. In this way, according to the Empirical Risk Minimization (ERM) principle, the well-trained classifier f_{θ_m} can generalize effectively to unseen nodes $\mathcal{V}_m^U$.

However, in real-world scenarios, the available labels $\mathcal{Y}_m^L$ may be corrupted, which degrades the generalization ability of the m-th client's classifier f_{θ_m}. We denote these noisy labels as $\mathcal{Y}_m^N = \{\tilde{y_1}, \ldots, \tilde{y_l}\}$, where $\mathcal{Y}_m^L$ represents their corresponding ground-truth labels. To realistically model label noise in multi-source data, we consider two common types of label noise, defined as follows:

Uniform noise [44]: This noise model assumes that the true label has a probability $\in (0,1)$ of being uniformly flipped to any of the other classes with equal probability. Formally, for all $j \neq i$,

$$p(y_m^N = j \mid y_m^L = i) = \frac{\epsilon}{d-1}. \tag{2}$$

Pair noise [63]: This noise model assumes that the true label can only be flipped to a specific paired class with a fixed probability ϵ, while remaining unchanged with probability $1 - \epsilon$.

The optimization objective is to learn a generalizable global model through the federated learning process that performs well under noisy conditions while maintaining strong robustness.

3 Methodology

3.1 Framework Overview

Inspired by our observations in Sec. 1 that FGL is sensitive to label noise, we propose `HYPERION` to finely enhance the model's ability to capture subtle structural differences among similar nodes and thereby improve robustness to noise. `HYPERION` comprises three key components: **I)** on each client, we extract local graph knowledge in a hyperspherical embedding space, where multiple class-specific prototype clusters capture fine-grained structural patterns; **II)** we select nodes whose embeddings remain consistently stable relative to their prototype clusters under perturbed views to ensure reliability; **III)** after the aggregation, we employ Wasserstein-driven prototype distillation and Mahalanobis-guided node purification at the server to refine and transfer complex structural knowledge. The detailed description of `HYPERION` is illustrated in Figure 2.

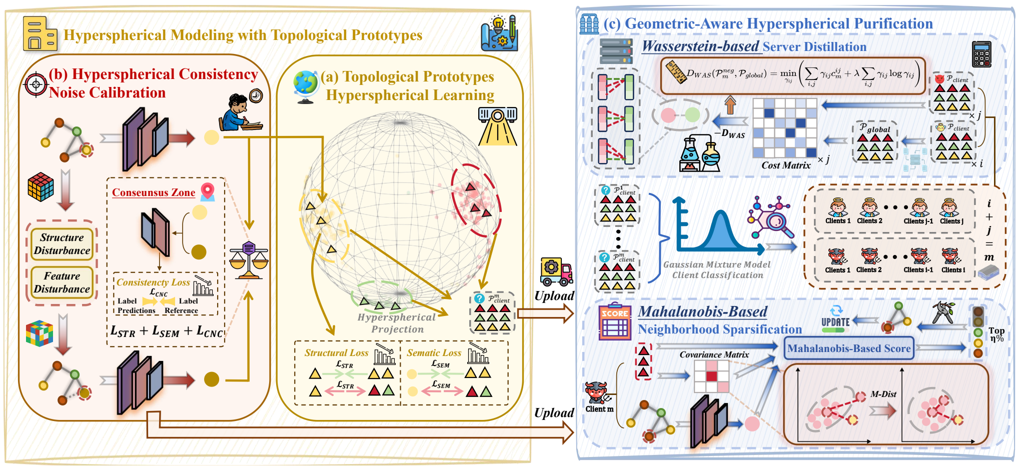

Hyperspherical Modeling with Topological Prototypes
(b) Hyperspherical Consistency Noise Calibration
(a) Topological Prototypes Hyperspherical Learning
Conseunsus Zone
Structure Disturbance
Feature Disturbance
Consistency Loss
Label Predictions
Label Reference
Hyperspherical Projection
Structural Loss
Sematic Loss
Upload
(c) Geometric-Aware Hyperspherical Purification
Wasserstein-based Server Distillation
Cost Matrix
Gaussian Mixture Model Client Classification
Mahalanobis-Based Neighborhood Sparsification
Covariance Matrix
Mahalanobis-Based Score
M-Dist
Client m
UPDATE
Top η%

Figure 2: **Architecture illustration** of HYPERION. HYPERION comprises Topological Prototypes Hyperspherical Learning (TP-HSL), Hyperspherical Consistency Noise Calibration (HS-CNC) and Geometric-Aware Hyperspherical Purification (GA-HSP). Best viewed in color and zoom in for details.

3.2 Topological Prototypes Hyperspherical Learning (TP-HSL)

Motivation. Existing methods suffer from noise amplification due to their reliance on coarse Euclidean representations that fail to capture multiple structural variations among intra-class nodes. We address this limitation by introducing prototype clusters on the hypersphere, which reveal and mitigate noise propagation through fine-grained topological modeling.

Hyperspherical Modeling with Topological Prototypes. Each client projects its node features $\mathcal{X}_{m_i}$ and adjacency $\mathcal{A}_{m_i}$ into a unit hypersphere via an independent feature projector θ_m^p:

$$z'_{m_i} = \theta_m^p(\mathcal{X}_{m_i}, \mathcal{A}_{m_i}), z_{m_i} = \frac{z'_{m_i}}{\left\|z'_{m_i}\right\|_2}. \tag{3}$$

The hyperspherical embedding representation z_{m_i} can be modeled using the von Mises–Fisher (vMF) distribution [35, 55]. The vmF distribution is well-suited for accurately measuring the angular differences between embeddings, especially in high-dimensional spaces. In graph data, node embeddings are typically high-dimensional, and the vmF distribution, through its concentration parameter, effectively quantifies the similarity and dissimilarity between node embeddings [55, 21]. This geometric property makes the model more sensitive to meaningful semantic relationships while being less susceptible to noise and feature magnitude variations. Compared to traditional Euclidean spaces, the hyperspherical projection offers superior noise robustness by encoding semantic similarity in angular relationships rather than absolute positions, which is particularly crucial for handling label noise and structural variations in decentralized graph data. This approach allows the model to better capture fine-grained topological differences while maintaining strong generalization across clients.

However, prototype-based modeling with a single center suffers from inherently limited expressiveness [32], which fails to capture all the underlying topologies adequately. To solve the issue, we allocate a prototype cluster of shape $[M, C, K]$, defined as $\mathcal{P} = \{\mathbf{p}_m^{c,k} | c \in [C], k \in [K], m \in [M]\}$, which partitions the entire hyperspherical space into multiple topology-aware subspaces, each centered around a different prototype:

$$viewsp(z_{m_i}; \mathbf{w}_m^c, \mathcal{P}_m^c; \kappa) = \sum_{k=1}^{K} w_m^{c,k} Z_D(\kappa) \exp(\kappa \mathbf{p}_m^{c,k} \cdot z_{m_i}), \tag{4}$$

where $\mathbf{p}_m^{c,k}$ denotes the centroid prototype in the k-th topological subspace for class c on the m-th client, $\omega_m^{c,k}$ denotes the corresponding prototype weight, and κ is the concentration parameter. This approach ensures that the node embeddings within each class are better aligned with the corresponding prototypes, thereby improving both intra-class cohesion and inter-class distinction. By doing so, we can finely capture the structural differences within each class. For the input vector z_i, we further compute its prediction probability as follows:

$$p(y = y_i \mid z_{m_i}; \{\mathbf{w}_m^c, \mathcal{P}_m^c\}_{c=1}^C) = \frac{\sum_{k=1}^K w_m^{y_i,k} \exp(\kappa \mathbf{p}_m^{y_i,k} \cdot z_{m_i})}{\sum_{c=1}^C \sum_{k=1}^K w_m^{c,k} \exp(\kappa \mathbf{p}_m^{c,k} \cdot z_{m_i})}. \tag{5}$$

Topological Prototypes Regularization Learning. To strengthen inter-class separability and intra-class cohesion, we introduce several regularization terms based on the aforementioned class prediction probabilities. First, we maximize the minimal spherical angle among different class prototypes to optimize the distribution of inter-class prototypes further. Specifically, we calculate the cosine similarity matrices to measure the similarity between prototypes in Equation (6):

$$\mathbf{S}_m[i,j] = \exp\left(\frac{(\mathbf{p}_m^{c_i,k_i})^\top \mathbf{p}_m^{c_j,k_j}}{\|\mathbf{p}_m^{c_i,k_i}\|_2 \cdot \|\mathbf{p}_m^{c_j,k_j}\|_2}\right), \quad \forall i \in [1,C], j \in [1,K], \tag{6}$$

where $\mathbf{S}_m[i,j]$ quantifies the pairwise similarity between prototype i and prototype j within the hyperspherical space. As shown in Equation (7), we apply binary masks Γ_m^{pos} and Γ_m^{neg} to further categorize intra- and inter-class similarity:

$$\mathbf{S}_m^{\text{pos}} = \mathbf{S}_m \odot \Gamma_m^{\text{pos}}, \quad \mathbf{S}_m^{\text{neg}} = \mathbf{S}_m \odot \Gamma_m^{\text{neg}}, \tag{7}$$

where Γ_m^{pos} activates entries corresponding to prototype pairs from the same class (excluding self-pairs), and Γ_m^{neg} identifies prototypes pairs from different classes. The symbol $\odot$ represents the Hadamard product. Finally, we arrive at the following regularization term:

$$\mathcal{L}_{STR} = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{\sum\limits_{j=1}^{N} \mathbf{S}_m^{\text{pos}}[i,j]}{\sum\limits_{j=1}^{N} \mathbf{S}_m^{\text{pos}}[i,j] + \sum\limits_{j=1}^{N} \mathbf{S}_m^{\text{neg}}[i,j] + \varepsilon}\right). \tag{8}$$

Here, ε is a small smoothing factor used to prevent division by zero. This regularization term serves two key purposes: on the one hand, it boosts inter-class separability, thereby sharpening the decision boundaries; on the other hand, it ensures intra-class similarity, reinforcing the class semantic features.

To prevent global shifts in the semantic space formed by multiple prototypes, it is also essential to regulate the spherical angular relationship between embeddings and their associated prototypes. To this end, we encourage the minimization of the average spherical angle between node embeddings and their corresponding prototype cluster centers. This reinforces semantic consistency within each class. This regularization term can be modeled as:

$$\mathcal{L}_{\text{SEM}} = -\frac{1}{N}\sum_{i=1}^{N} log \frac{\sum_{k=1}^{K} w_m^{y_i,k} \exp(\kappa \mathbf{p}_m^{y_i,k} \cdot z_{m_i})}{\sum_{c=1}^{C}\sum_{k=1}^{K} w_m^{c,k} \exp(\kappa \mathbf{p}_m^{c,k} \cdot z_{m_i})}. \tag{9}$$

3.3 Hyperspherical Consistency Noise Calibration (HS-CNC)

Motivation. Due to the unique neighborhood diffusion mechanism in graph data, noisy labels tend to propagate along the edges. Therefore, it is crucial to design a noise node filtering mechanism that considers both structural and semantic aspects.

Hyperspherical Robust Node Selection. To effectively filter out potential noisy nodes whose mapping trajectories exhibit significant fluctuations, we assess the stability of nodes across different augmented graph views. Specifically, inspired by previous works [71, 27], we introduce data augmentation techniques: edge dropping and feature masking. These techniques randomly drop edges and certain feature values in the graph $\mathcal{G}_m(\mathcal{X}_m, \mathcal{A}_m)$:

$$\tilde{\mathcal{A}}_m = \mathcal{A}_m \odot \Gamma_m^{\mathcal{A}}, \tilde{\mathcal{X}}_m = \mathcal{X}_m \odot \Gamma_m^{\mathcal{X}}, \tag{10}$$

where $\Gamma_m^{\mathcal{A}} \in \{0,1\}^{N_m * N_m}$ is the randomly generated edge mask matrix, and $\Gamma_m^{\mathcal{X}} \in \{0,1\}^{N_m * d}$ is the randomly generated feature mask matrix.

By applying the two mask matrices, the augmented graph $\tilde{\mathcal{G}}_m(\tilde{\mathcal{X}}_m, \tilde{\mathcal{A}}_m)$ contains perturbed structural and semantic information. We assess the stability of each node by calculating the consistency of its representation across different views. Nodes that exhibit high consistency across both views are considered "clean nodes" because they maintain stable and consistent semantic representations under different perturbation conditions. We train using the subset of stable nodes $\mathcal{X}'_m$, which fundamentally ensures that learned representations are grounded in meaningful:

$$L_{\text{CNC}} = \sum_{v_m^i \in \mathcal{V}_m} \mathbb{1}\left(f_m(\mathcal{G}_m, v_m^i) = f_m(\tilde{\mathcal{G}}_m, v_m^i)\right) \cdot L\left(f_m(\mathcal{G}_m, v_m^i), \mathcal{Y}_m(v_m^i)\right), \tag{11}$$

where $\mathbb{1}$ is an indicator function that outputs 1 if the predicted results for node v_m^i in both graphs are consistent, and 0 otherwise. The loss function L computes the discrepancy between the node's predicted value and its true label and then calculates the gradient to update the parameters.

This design naturally connects to two theoretical perspectives. From the *information bottleneck* viewpoint, random edge dropping acts as compression: by pruning away connections, the model must retain relationships that consistently survive perturbations, which more likely reflect the task-relevant core structure [5, 15]. From the *generalization under noise* viewpoint, these augmentations inject structured noise that preserves global characteristics but prevents overfitting to fragile details. Consequently, the model is nudged toward flatter minima in the loss landscape, leading to improved robustness and generalization. Such principles align with findings in graph contrastive learning [62, 53]. As an example, `HYPERION` algorithm is shown in Algorithm 1.

Algorithm 1 `HYPERION` Framework

```
Communication rounds $T$, participant scale $M$, $m$-th client private model $\theta_m$, $m$-th client local data $\mathcal{G}_m$, $m$-th client prototype cluster $\mathcal{P}_m$ and loss weight $\alpha, \beta$
The final global model $\theta_{global}$
for $t = 1, 2, \cdots, T$ do
    Client Side: for $m = 1$ to $M$ in parallel do
        $\mathcal{L}_{CNC} \leftarrow$ HypersphericalNoiseCalibration($\mathcal{G}_m, \mathcal{P}_m$)by Equation (11) // Select robust nodes and train with them
        $\mathbf{S}_m \leftarrow$ CalculateSimilarity($\mathcal{P}_m$)by Equation (6) // Calculate prototypes similarity metrix
        $\mathcal{L}_{STR} \leftarrow$ StructLoss($\mathbf{S}_m$)by Equation (8) // Calculate loss with inter- and intra-class prototypes
        $\mathcal{L}_{SEM} \leftarrow$ SemanticLoss($\mathcal{P}_m$)by Equation (9) // Calculate loss with embedding vector and prototypes
        $\theta_m^{t+1} \leftarrow$ LocalUpdating($\theta_m^t, \mathcal{L}_{CNC} + \alpha\mathcal{L}_{STR} + \beta\mathcal{L}_{SEM}$) // Backward propagation
    Server Side:
    $\theta^{pos}, \theta^{neg} \leftarrow$ GMM($\mathcal{P}$) // Client classification by prototypes
    $\mathcal{G}_m^{neg\prime}$ = NeighborhoodSparsification($\theta_m^{neg}, \mathcal{G}_m$), $\forall m$ by Equation (18) // pruning with Mahalanobis distance
    $\theta_{global}, \mathcal{P}_{global} \leftarrow$ Aggregate($\theta_m^{pos}, \mathcal{P}^{pos}$), $\forall m$ // Clean clients hyperspherical aggregation
    $\mathcal{P}'_{global} \leftarrow$ ServerDistillation($\mathcal{P}_m^{neg}, \mathcal{P}_{global}$), $\forall m$ by Equation (13) // Wassertein distance Server Distillation
    $\theta_m \leftarrow \theta_{global}, \forall m$ // Distribute parameters to clients
return $\theta_{global}$
```

3.4 Geometric-Aware Hyperspherical Purification (GA-HSP)

Motivation. It is exceptionally challenging to entirely remove noisy nodes relying only on the client-side. Due to the non-IID distribution, each client has a limited perspecitve, meaning that some noisy nodes are detected as normal locally, but are considered anomalous when viewed globally. To address this issue, we propose GA-HSP, which performs knowledge purification on the server side.

Prototype-based Client Classification. Existing research suggests that, in practical scenarios, the data noise ratio at each client may vary to some extent [60]. To effectively identify noisy clients, we design an unsupervised detection method that fully exploits the global distribution characteristics of local prototypes on the client side, distinguishing between benign and malicious clients.

After training on each client, the local prototype clusters $\mathcal{P}$ are uploaded to the server, where the server computes the Gaussian Mixture Model (GMM) of the prototype clusters $\mathcal{P}$ from all M clients.

$$p(\mathcal{P}) = \sum_{l=1}^{2} \pi_l \cdot \mathcal{N}(\mathcal{P} \mid \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l), \tag{12}$$

where $\pi_l, \boldsymbol{\mu}_l$ and $\boldsymbol{\Sigma}_l$ represent the weight, mean, and covariance matrix of the l-th Gaussian component, respectively. The client set ϕ is partitioned into two subsets: clean clients ϕ_{pos} and malicious clients ϕ_{neg}. To ensure the reliability of the global knowledge, we apply GMM to re-aggregate the prototypes from the clean clients ϕ_{pos}. For each class, we aim to aggregate the prototypes into K clusters and take the center of each cluster as the global prototype for that class: $\mathcal{P}_{global} = \{p^{c,k} | c \in [C], k \in [K]\}$. Through this unsupervised approach, we can not only effectively filter out malicious clients, but also integrate the local information from all clean clients, thereby constructing a more precise global prototype cluster $\mathcal{P}_{global}$.

Wasserstein-based Server Distillation. To further filter out erroneous information in the prototypes, inspired by recent work on negative distillation [33], we adopt a "negative distillation" strategy: each malicious client prototype cluster $\mathcal{P}_m^{neg}$ serves as the teacher, while the global prototype cluster $\mathcal{P}_{global}$ functions as the student. The core of this distillation is to constrain the angular distance between the teacher and the student in the hyperspherical space, thereby suppressing and correcting anomalous representations in the global prototype cluster.

Traditional distillation methods fail to capture cross-dimensional similarities, leading to the underutilization of dimensional information. To address this issue, we propose a Wasserstein distance-driven prototype negative distillation method, which deploys the discrete Wasserstein distance to comprehensively measure the distributional differences between the teacher and student models. For the m-th teacher prototype cluster $\mathcal{P}_m^{neg}$, we define the discrete Wasserstein distance D_{WAS} as follows:

$$D_{WAS}(\mathcal{P}_m^{neg}, \mathcal{P}_{global}) = \min_{\gamma_{ij}} \left(\sum_{i,j} \gamma_{ij} c_m^{ij} + \lambda \sum_{i,j} \gamma_{ij} \log \gamma_{ij} \right), \tag{13}$$

where γ_{ij} represents the mass transferred from the teacher prototype cluster's dimension q_i to the student prototype cluster's dimension q_j, subject to the constraints:

$$\sum_j \gamma_{ij} = \mathcal{P}_{m,i}^{neg}, \sum_i \gamma_{ij} = \mathcal{P}_{global,j}, \gamma_{ij} \geq 0, \tag{14}$$

where λ is a hyperparameter controlling the entropy regularization term. A key component of this formulation is the cost matrix c^m, which encapsulates the dissimilarity between prototype dimensions:

$$c_m^{ij} = 1 - \frac{\mathcal{P}_{m,i}^{neg} \cdot \mathcal{P}_{global,j}}{\|\mathcal{P}_{m,i}^{neg}\| \|\mathcal{P}_{global,j}\|}. \tag{15}$$

The higher the similarity between prototype dimensions, the lower the transfer cost. Conversely, when there is a significant difference in the direction of the dimensions, the cost increases considerably. By minimizing D_{WAS}, the probability mass is effectively reallocated between proximate dimensions in the feature space, thereby naturally reinforcing the correlation of benign features between global prototypes, while effectively diminishing the impact of anomalous features.

Mahalanobis-Based Neighborhood Sparsification. Building on the negative distillation from malicious clients, we attempt to incorporate data pruning to purify them. Our out-of-distribution (OOD) detection method is based on Mahalanobis distance [32, 42], which is equivalent to the Euclidean distance scaled by the eigenvalues in the feature space. By introducing the inverse of the covariance matrix, the Mahalanobis distance can automatically adjust the importance of each dimension, avoiding certain dimensions dominating the distance calculation due to scale differences. In geometry, Mahalanobis distance transforms the data space into a standardized space, making distance calculations more equitable. We identify and remove noise nodes that deviate from the normal distribution by calculating the Mahalanobis distance between node embeddings and the global prototype distribution (as shown in Equation (18)). This method effectively filters out noise that is not detected by clients due to local perspective limitations. Inspired by the work [1], we simultaneously measure both inter-class and intra-class distances of prototype clusters to comprehensively assess the outlier risk of nodes. The inter-class prototype distance is defined as:

$$d_{\text{inter}}(z_{m_i}) = \frac{1}{|C| - 1} \sum_{c \neq y_i} \left[\frac{1}{\min_k \left((z_{m_i} - p_m^{c,k})^\top \Lambda_c^{-1} (z_{m_i} - p_m^{c,k}) \right) + \varepsilon} \right], \tag{16}$$

Table 1: **Comparison with the state-of-the-art methods** on five selected real-world datasets. For each dataset, we report accuracy (%) and F1-macro (%) (with red/green markers indicating regression/improvement over FedAvg). The noise type is set to **50%-uniform** (upper) and **50%-pair** (lower). The best and second-best results are highlighted with **bold** and underline, respectively. Additional experimental results on more settings can be found in Appendix D.

Category	Methods	Cora		CiteSeer		PubMed		Physics		Amazon_ratings	
	Metrics	ACC	F1-macro	ACC	F1-macro	ACC	F1-macro	ACC	F1-macro	ACC	F1-macro
FL	FedAvg [ASTAT17]	$32.91_{\uparrow 0.00}$	$30.47_{\uparrow 0.00}$	$28.15_{\uparrow 0.00}$	$26.98_{\uparrow 0.00}$	$59.39_{\uparrow 0.00}$	$57.19_{\uparrow 0.00}$	$70.69_{\uparrow 0.00}$	$39.21_{\uparrow 0.00}$	$33.65_{\uparrow 0.00}$	$\underline{25.07}_{\uparrow 0.00}$
	FedProx [MLSys20]	$34.64_{\uparrow 01.73}$	$33.12_{\uparrow 02.65}$	$26.67_{\downarrow 01.48}$	$25.84_{\downarrow 01.14}$	$61.31_{\uparrow 01.92}$	$58.77_{\uparrow 01.58}$	$\underline{72.53}_{\uparrow 01.84}$	$\mathbf{57.80}_{\uparrow 18.59}$	$35.86_{\uparrow 02.21}$	$24.94_{\downarrow 00.13}$
	FedNova [NeurIPS20]	$37.48_{\uparrow 04.57}$	$35.33_{\uparrow 04.86}$	$31.56_{\uparrow 03.41}$	$30.46_{\uparrow 03.48}$	$63.49_{\uparrow 04.10}$	$61.61_{\uparrow 04.42}$	$57.37_{\downarrow 13.32}$	$26.80_{\downarrow 12.41}$	$36.61_{\uparrow 02.96}$	$24.18_{\downarrow 00.89}$
	MOON [CVPR21]	$33.64_{\uparrow 00.73}$	$31.86_{\uparrow 01.39}$	$27.26_{\downarrow 00.89}$	$26.78_{\downarrow 00.20}$	$70.14_{\uparrow 10.75}$	$58.89_{\uparrow 01.70}$	$59.06_{\downarrow 11.63}$	$13.46_{\downarrow 25.75}$	$\underline{38.49}_{\uparrow 04.84}$	$20.67_{\downarrow 04.40}$
FGL	FGSSL [IJCAI23]	$36.65_{\uparrow 03.74}$	$34.43_{\uparrow 03.96}$	$32.74_{\uparrow 04.59}$	$32.21_{\uparrow 05.23}$	$65.92_{\uparrow 06.53}$	$71.59_{\uparrow 14.40}$	$50.76_{\downarrow 19.93}$	$13.78_{\downarrow 25.43}$	$38.43_{\uparrow 04.78}$	$20.49_{\downarrow 04.58}$
	FedGTA [VLDB24]	$31.54_{\downarrow 01.37}$	$29.84_{\downarrow 00.63}$	$28.44_{\uparrow 00.29}$	$27.40_{\uparrow 00.42}$	$57.41_{\downarrow 01.98}$	$56.63_{\downarrow 00.56}$	$60.48_{\downarrow 10.21}$	$25.93_{\downarrow 13.28}$	$33.71_{\uparrow 00.06}$	$24.46_{\downarrow 00.61}$
	FedTAD [IJCAI24]	$31.26_{\downarrow 01.65}$	$29.53_{\downarrow 00.94}$	$28.59_{\uparrow 00.44}$	$26.56_{\downarrow 00.42}$	$58.81_{\downarrow 00.58}$	$56.56_{\downarrow 00.63}$	$57.12_{\downarrow 13.57}$	$28.24_{\downarrow 10.97}$	$32.12_{\downarrow 01.53}$	$24.60_{\downarrow 00.47}$
Robust FL	FedNoRo [IJCAI23]	$32.27_{\downarrow 00.64}$	$30.31_{\downarrow 00.16}$	$28.44_{\downarrow 00.29}$	$27.09_{\uparrow 00.11}$	$60.25_{\downarrow 00.86}$	$56.85_{\downarrow 00.34}$	$70.74_{\uparrow 00.05}$	$39.59_{\uparrow 00.38}$	$33.78_{\uparrow 00.13}$	$24.16_{\downarrow 00.91}$
	FedNed [AAAI24]	$32.82_{\downarrow 00.09}$	$29.83_{\downarrow 00.64}$	$30.52_{\uparrow 02.37}$	$28.87_{\uparrow 01.89}$	$57.41_{\downarrow 01.98}$	$55.78_{\downarrow 01.41}$	$64.48_{\downarrow 06.21}$	$32.09_{\downarrow 07.12}$	$34.00_{\uparrow 00.35}$	$\mathbf{25.23}_{\uparrow 00.16}$
	FedCorr [CVPR22]	$34.37_{\uparrow 01.46}$	$28.44_{\downarrow 02.03}$	$22.20_{\downarrow 05.95}$	$24.08_{\downarrow 02.90}$	$57.36_{\downarrow 02.03}$	$56.12_{\downarrow 00.69}$	$60.73_{\downarrow 09.96}$	$25.12_{\downarrow 14.09}$	$35.22_{\uparrow 01.57}$	$24.02_{\downarrow 01.05}$
Robust GL	CRGNN [NN24]	$44.61_{\uparrow 11.70}$	$\underline{39.41}_{\uparrow 08.94}$	$39.55_{\uparrow 11.40}$	$36.21_{\uparrow 09.23}$	$\underline{73.66}_{\uparrow 14.27}$	$\underline{72.33}_{\uparrow 15.14}$	$55.45_{\downarrow 15.24}$	$22.73_{\downarrow 16.48}$	$36.41_{\uparrow 02.76}$	$10.68_{\downarrow 14.39}$
	RTGNN [WWW23]	$\underline{47.81}_{\uparrow 14.90}$	$38.72_{\uparrow 08.25}$	$\underline{41.93}_{\uparrow 13.78}$	$\underline{36.33}_{\uparrow 09.35}$	$67.33_{\uparrow 07.94}$	$44.09_{\downarrow 13.10}$	$66.29_{\downarrow 04.40}$	$29.05_{\downarrow 10.16}$	$36.65_{\uparrow 03.00}$	$21.52_{\downarrow 03.55}$
	CLNode [WSDM23]	$35.10_{\uparrow 02.19}$	$33.13_{\uparrow 02.66}$	$30.37_{\uparrow 02.22}$	$30.40_{\uparrow 03.42}$	$54.73_{\downarrow 04.66}$	$52.14_{\downarrow 05.05}$	$66.29_{\downarrow 04.40}$	$29.61_{\downarrow 09.60}$	$31.35_{\downarrow 02.30}$	$24.43_{\downarrow 00.64}$
Robust FGL	`HYPERION`	$\mathbf{53.56}_{\uparrow 20.65}$	$\mathbf{51.15}_{\uparrow 20.68}$	$\mathbf{47.11}_{\uparrow 18.96}$	$\mathbf{40.25}_{\uparrow 13.27}$	$\mathbf{74.85}_{\uparrow 15.46}$	$\mathbf{73.74}_{\uparrow 16.55}$	$\mathbf{75.21}_{\uparrow 04.52}$	$\underline{45.71}_{\uparrow 06.50}$	$\mathbf{38.96}_{\uparrow 05.31}$	$22.77_{\downarrow 02.30}$

Category	Methods	Cora		CiteSeer		PubMed		Physics		Amazon_ratings	
	Metrics	ACC	F1-macro	ACC	F1-macro	ACC	F1-macro	ACC	F1-macro	ACC	F1-macro
FL	FedAvg [ASTAT17]	$33.27_{\uparrow 00.00}$	$31.97_{\uparrow 00.00}$	$30.67_{\uparrow 00.00}$	$30.23_{\uparrow 00.00}$	$49.75_{\uparrow 00.00}$	$49.39_{\uparrow 00.00}$	$49.93_{\uparrow 00.00}$	$33.81_{\uparrow 00.00}$	$34.94_{\uparrow 00.00}$	$21.69_{\uparrow 00.00}$
	FedProx [MLSys20]	$37.02_{\uparrow 03.75}$	$34.76_{\uparrow 02.79}$	$33.04_{\uparrow 02.37}$	$32.42_{\uparrow 02.19}$	$50.10_{\uparrow 00.35}$	$48.76_{\downarrow 00.63}$	$\underline{53.54}_{\uparrow 03.61}$	$\underline{41.90}_{\uparrow 08.09}$	$31.76_{\downarrow 03.18}$	$20.99_{\downarrow 00.70}$
	FedNova [NeurIPS20]	$36.11_{\uparrow 02.84}$	$34.30_{\uparrow 02.33}$	$32.44_{\uparrow 01.77}$	$32.21_{\uparrow 01.98}$	$53.16_{\uparrow 03.41}$	$51.93_{\uparrow 02.54}$	$46.76_{\downarrow 03.17}$	$31.77_{\downarrow 02.04}$	$35.57_{\uparrow 00.63}$	$21.20_{\downarrow 00.49}$
	MOON [CVPR21]	$32.63_{\downarrow 00.64}$	$31.47_{\downarrow 00.50}$	$29.48_{\downarrow 01.19}$	$29.32_{\uparrow 00.91}$	$51.90_{\uparrow 02.15}$	$50.60_{\downarrow 01.21}$	$50.80_{\uparrow 00.87}$	$35.27_{\uparrow 01.46}$	$35.37_{\uparrow 00.43}$	$\underline{22.17}_{\uparrow 00.48}$
FGL	FGSSL [IJCAI23]	$36.29_{\uparrow 03.02}$	$34.30_{\uparrow 02.33}$	$32.59_{\uparrow 01.92}$	$32.30_{\uparrow 02.07}$	$56.65_{\uparrow 06.90}$	$55.75_{\uparrow 06.36}$	$38.35_{\downarrow 11.58}$	$19.18_{\downarrow 14.63}$	$37.57_{\uparrow 02.63}$	$19.62_{\downarrow 02.07}$
	FedGTA [VLDB24]	$32.54_{\downarrow 00.73}$	$30.82_{\downarrow 01.15}$	$28.59_{\downarrow 02.08}$	$28.82_{\downarrow 01.41}$	$50.00_{\uparrow 00.25}$	$50.82_{\uparrow 01.43}$	$49.15_{\downarrow 00.78}$	$31.57_{\downarrow 02.24}$	$34.73_{\downarrow 00.21}$	$21.24_{\downarrow 00.45}$
	FedTAD [IJCAI24]	$31.44_{\downarrow 01.83}$	$30.36_{\downarrow 01.61}$	$31.70_{\uparrow 01.03}$	$29.41_{\downarrow 00.82}$	$51.80_{\uparrow 02.05}$	$51.43_{\uparrow 02.04}$	$44.75_{\downarrow 05.18}$	$33.44_{\downarrow 00.37}$	$35.73_{\uparrow 00.79}$	$21.08_{\downarrow 00.61}$
Robust FL	FedNoRo [IJCAI23]	$33.18_{\downarrow 00.09}$	$32.01_{\uparrow 00.04}$	$30.81_{\uparrow 00.14}$	$30.88_{\uparrow 00.65}$	$49.09_{\downarrow 00.66}$	$49.54_{\uparrow 00.15}$	$49.93_{\uparrow 00.00}$	$33.81_{\uparrow 00.00}$	$35.57_{\uparrow 00.63}$	$\mathbf{22.41}_{\uparrow 00.72}$
	FedNed [AAAI24]	$35.74_{\uparrow 02.47}$	$33.20_{\uparrow 01.23}$	$32.00_{\uparrow 01.33}$	$31.87_{\uparrow 01.64}$	$46.38_{\downarrow 03.37}$	$44.94_{\downarrow 04.45}$	$53.38_{\uparrow 03.45}$	$37.86_{\uparrow 04.05}$	$32.82_{\downarrow 02.12}$	$21.37_{\downarrow 00.32}$
	FedCorr [CVPR22]	$33.91_{\uparrow 00.64}$	$28.59_{\uparrow 03.38}$	$28.00_{\downarrow 02.67}$	$29.52_{\uparrow 00.71}$	$51.52_{\uparrow 01.77}$	$20.39_{\downarrow 29.00}$	$50.73_{\uparrow 00.80}$	$13.46_{\downarrow 20.35}$	$\underline{37.96}_{\uparrow 03.02}$	$17.27_{\downarrow 04.42}$
Robust GL	CRGNN [NN24]	$\underline{37.29}_{\uparrow 04.02}$	$\underline{35.87}_{\uparrow 03.90}$	$\underline{37.48}_{\uparrow 06.81}$	$\underline{34.87}_{\uparrow 04.64}$	$\underline{63.59}_{\uparrow 13.84}$	$\underline{58.58}_{\uparrow 09.19}$	$47.08_{\uparrow 02.85}$	$41.74_{\uparrow 07.93}$	$35.35_{\uparrow 00.41}$	$21.40_{\downarrow 00.29}$
	RTGNN [WWW23]	$32.91_{\downarrow 00.36}$	$28.21_{\downarrow 03.76}$	$36.44_{\uparrow 05.77}$	$32.83_{\uparrow 02.60}$	$46.89_{\downarrow 02.86}$	$47.82_{\downarrow 01.57}$	$29.36_{\downarrow 20.57}$	$14.64_{\downarrow 19.17}$	$35.88_{\uparrow 00.94}$	$21.34_{\downarrow 00.35}$
	CLNode [WSDM23]	$35.47_{\uparrow 02.20}$	$33.41_{\uparrow 01.44}$	$31.41_{\uparrow 00.74}$	$32.10_{\uparrow 01.87}$	$49.60_{\downarrow 00.15}$	$48.98_{\downarrow 00.41}$	$52.72_{\uparrow 02.79}$	$37.96_{\uparrow 04.15}$	$34.12_{\downarrow 00.82}$	$21.96_{\uparrow 00.27}$
Robust FGL	`HYPERION`	$\mathbf{41.50}_{\uparrow 08.23}$	$\mathbf{36.16}_{\uparrow 04.19}$	$\mathbf{43.11}_{\uparrow 12.44}$	$\mathbf{41.11}_{\uparrow 10.88}$	$\mathbf{74.85}_{\uparrow 25.10}$	$\mathbf{74.04}_{\uparrow 24.65}$	$\mathbf{70.10}_{\uparrow 20.17}$	$\mathbf{49.85}_{\uparrow 16.04}$	$\mathbf{39.65}_{\uparrow 04.71}$	$21.72_{\uparrow 00.03}$

where Λ_c^{-1} represents the inverse of the covariance matrix of all prototypes in class c, and ε is used to avoid division by zero errors. The intra-class prototype distance is calculated as:

$$d_{\text{intra}}(z_{m_i}) = \frac{1}{|P_{y_i}| - 1} \sum_{k=1}^{|P_{y_i}|-1} \left[\frac{1}{(z_{m_i} - p_m^{y_i,k})^\top \Lambda_{y_i}^{-1} (z_{m_i} - p_m^{y_i,k}) + \varepsilon} \right]. \tag{17}$$

Finally, we calculate the comprehensive outlier score and rank all nodes based on their outlier risks, pruning the top $\eta\%$ of the high-risk samples in each training round:

$$\text{Score}(z_{m_i}) = \frac{d_{\text{inter}}(z_{m_i})}{d_{\text{inter}}(z_{m_i}) + d_{\text{intra}}(z_{m_i})} \quad \longrightarrow \quad \mathcal{V}_m^{t+1} = \left\{ z_{m_i} \in \mathcal{V}_m^t \mid \text{rank}(z_{m_i}) > \lfloor \eta * |\mathcal{V}_m^t| \rfloor \right\}, \tag{18}$$

where $\mathcal{V}_m^t$ represents the sample nodes in the t round, and η denotes the pruning ratio. Further discussion and limitations can be found in Appendix E and Appendix F.

4 Experiment

In this section, we comprehensively evaluate `HYPERION` through four key axes: **Q1** (Superiority), **Q2** (Resilience), **Q3** (Effectiveness), and **Q4** (Sensitivity).

4.1 Experimental Setup

Datasets. To effectively evaluate the performance of our approach, we utilize five benchmark graph datasets of various scales and distributions with different characteristics, including Cora [36], CiteSeer [11], PubMed [3], Physics[43], and Amazon_ratings. These datasets represent a wide range of domains and are commonly used in graph-based machine learning tasks. Detailed descriptions and dataset splits for these datasets can be found in Appendix C.1. Furthermore, the implementation details and parameter settings can be found in Appendix C.3.

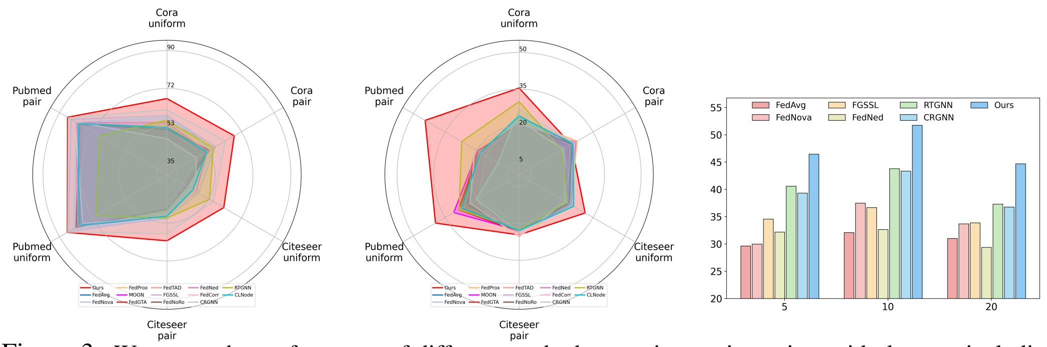


Figure 3: We report the performance of different methods at various noise ratios, with datasets including Cora, Citeseer, and Pubmed, and noise types of uniform and pair. The red color represents the performance of `HYPERION`. *(First)*: 20% mild noise. *(Second)*: 70% severe noise. *(Third)*: We compare `HYPERION` with several FL and FGL methods on the Cora dataset under uniform noise, as the number of clients ranges from 5 to 20.

Counterparts. We compare `HYPERION` against several traditional FL methods: (1) **FedAvg** [ASTAT17] [37], (2) **FedProx** [MLSys20] [25], (3) **FedNova** [NeurIPS20] [54], (4) **MOON** [CVPR 21] [24]; four popular FGL approaches: (5) **FGSSL** [IJCAI23] [16]; (6) **FedTAD** [IJCAI24] [70], (7) **FedGTA** [VLDB24] [28]; three Robust FL methods: (8) **FedNoRo** [IJCAI23] [58], (9) **FedNed** [AAAI24] [33], (10) **FedCorr** [CVPR22] [60];three Robust GraghLearning methods: (11) **CRGNN** [NN24] [27], (12) **RTGNN** [WWW23] [40], (13) **CLNode** [WSDM23] [56]. Detailed descriptions of all the baselines can be found in Appendix C.2.

4.2 Superiority

To answer **Q1**, we conducted systematic evaluations in a variety of noise environments, including two typical noise types (uniform noise and pair noise) and three noise intensity levels (0.2 for low noise ratio, 0.5 for medium noise ratio, and 0.7 for high noise ratio). The perfmance results are presented in Tab. 1, and Figure 3. Several observations from these experiments are summarized (**Obs.**):

Obs. ❶ Existing approaches exhibit suboptimal performance in FGL scenarios with noisy labels. For instance, in the uniform noise mode with a 50% noise ratio, most previous methods achieve accuracy rates below 40% on the Cora dataset. Notably, under a 70% noise ratio, the performance of most of these methods deteriorates significantly, worsening with average accuracy rates consistently dropping below 25%. Moreover, the existing robust methods do not demonstrate substantial improvements in noisy FGL scenarios, with the performance of most approaches being comparable to that of traditional FL and FGL methods.

Obs. ❷ `HYPERION` demonstrates remarkable robustness across various noise scales. Under moderate noise conditions (0.5), `HYPERION` shows a clear and significant advantage. As shown in Tab. 1 (upper), `HYPERION` consistently outperforms both FL and FGL baselines across various datasets and noise types. In the Citeseer-uniform setting, it achieves an accuracy of 47.11%, surpassing the best baseline, RTGNN (41.93%), by 5.18 percentage points. Additionally, as shown in Figure 3 (Second), `HYPERION` consistently outperforms both FL and FGL baselines across various noise scales. It is evident that, under a noise scale of 0.2, `HYPERION` achieves varying degrees of performance improvement over all baselines. In high-noise environments, `HYPERION` also demonstrates an average performance gain of 8.1% to 10.1% compared to the baselines, including RTGNN.

4.3 Resilience

To address **Q2**, we evaluate the performance of each method on the Cora dataset under the 0.5 uniform noise setting, across different client scales. Figure 3 (Third) illustrates that `HYPERION` consistently achieves robust performance gains across varying client numbers (5-20), outperforming FedAvg by at least 13.69% while maintaining a minimum 6% advantage over the top-performing baseline method. This demonstrates that `HYPERION` effectively identifies noise and maintains stable performance, even under challenging conditions with varying client populations.

4.4 Effectiveness

To address **Q3**, we conduct an ablation study on the key components of our method, both at the client-side and server-side, under a noise scale of 0.5. Tab. 2 reports the performance of `HYPERION` and its variants by removing specific components from the TP-HSL and HS-CNC modules-namely, the structural loss, robust node selection, and semantic loss. Tab. 3 presents the results for `HYPERION` and its variants derived from the GA-HSP module, where we ablate server-side distillation (SD), node pruning (NP), and client classification with server distillation (CC+SD). Individually, both TP-HSL and HS-CNC contribute significantly to improving model accuracy. Moreover, GA-HSP demonstrates substantial effectiveness in integrating global reliable information, thereby reinforcing the robustness of our design in mitigating label noise in FGL settings.

Table 2: **Ablation study** of **TP-HSL** and **HS-CNC** on the client-side of `HYPERION`. All results are reported under 0.5 noise ratio and 10-client scale.

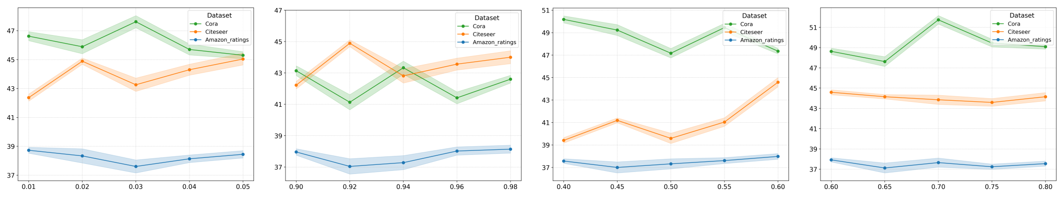
Client	Cora		Citeseer	
	uniform	pair	uniform	pair
w/o L_{STR}	50.27	36.93	42.37	42.07
w/o L_{CNC}	50.82	35.28	43.56	38.37
w/o L_{SEM}	50.09	36.20	40.29	41.04
`HYPERION`	**56.31**	**41.50**	**47.11**	**43.11**

Table 3: **Ablation study** of **GA-HSP** on the server-side of `HYPERION`. w/o CC+SD means SD depends on CC, so without CC, SD is also removed.

Server	Cora		Citeseer	
	uniform	pair	uniform	pair
w/o SD	49.18	37.11	39.85	38.67
w/o NP	37.29	34.83	30.07	31.26
w/o CC+SD	44.66	36.40	36.07	35.70
`HYPERION`	**56.31**	**41.50**	**47.11**	**43.11**

4.5 Sensitivity

To address **Q4**, we perform sensitivity analyses on hyperparameters of `HYPERION`. Specifically, we examine the model's performance under varying values of λ, η, α, and β, as illustrated in Figure 4, where these hyperparameters are fixed at different scales and values. We systematically vary the hyperparameters λ and η within the ranges $[0.01, 0.05]$ and $[0.90, 0.98]$, respectively, to evaluate the stability of GA-HSP under different settings. For TP-HSL, we vary α and β with in the ranges $[0.4, 0.6]$ and $[0.6, 0.8]$, using a step size of 0.05. The results indicate that the choice of λ and η has a minimal impact on the performance of `HYPERION`. However, when α is within the range of $[0.55, 0.6]$ and β is within the range of $[0.7, 0.75]$, `HYPERION` achieves the best performance across all datasets.



(a) GA-HSP Parameter λ (b) GA-HSP Parameter η (c) TP-HSL Parameter α (d) TP-HSL Parameter β

Figure 4: **Analysis on hyper-parameter in `HYPERION`**. Node classification results under varying values of λ, η, α, and β. The colors green, blue, and yellow correspond to performance on the Cora, Amazon_ratings, and Citeseer datasets, respectively. All experiments are conducted using 50%-uniform noise.

We investigate the impact of the hyperparameter K on the performance and efficiency of `HYPERION`. Specifically, we vary $K \in \{1, 2, 3, 4\}$ in the Citeseer-pair setting and observe the corresponding changes in performance. As shown in Figure 5, setting $K = 1$ results in under-learning of hypersphere, leading to consistently lower performance. In contrast, increasing K to 3 or 4 yields marginal performance improvements. In all experiments, we set $K = 3$.

Figure 5: Performance across varying K values under different noise ratios.

K	**20%**		**50%**		**70%**	
	ACC	F1-M	ACC	F1-M	ACC	F1-M
1	59.41	52.13	38.81	36.78	20.48	20.53
2	59.56	52.24	39.70	37.92	21.33	21.39
3	**61.93**	**56.58**	**43.11**	**41.11**	24.59	24.48
4	61.04	55.96	38.07	36.02	**24.65**	**25.02**

5 Conclusion

In this work, we propose an innovative exploration of robust FGL in noisy environment. To achieve this goal, we project nodes onto hyperspherical embedding space, thereby introducing a novel framework, `HYPERION`. For robust representation, TP-HSL is employed to project nodes onto hyperspherical space, effectively addressing the coupling problem associated with complex topologies and malicious noise. Leveraging hyperspherical representations, we also introduce HS-CNC, which filters out potential noisy nodes by considering both structural and semantic factors. Specifically, for effective global collaboration, we further design GA-HSP to facilitate knowledge purification. By integrating these three strategies, `HYPERION` outperforms various state-of-the-art methods in node classification tasks across different noisy scenarios.

Acknowledgement

This work is supported by National Natural Science Foundation of China under Grant (62225113, 623B2080), 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.

A Notations

We present a comprehensive review of the commonly used notations and their definitions in Tab. 4.

Notation	Definition
$\mathcal{G}_m$	Graph data for the m-th client .
N_m	The number of nodes for the m-th client.
$\mathcal{V}_m$	The node set of $\mathcal{G}_m$.
$\mathcal{X}_m$	The feature matrix of $\mathcal{G}_m$.
$\mathcal{A}_m$	The adjacency matrix of $\mathcal{G}_m$.
$\mathcal{Y}_m$	The one-hot label matrix of $\mathcal{G}_m$.
$\mathcal{X}_{m_i}$	The feature of node i in $\mathcal{G}_m$.
$\mathcal{A}_{m_i}$	The edges of node i in $\mathcal{G}_m$.
M	The number of clients.
d	The dimension of the node feature.
C	The number of node classes.
K	The number of prototypes in each class per client.
$\mathcal{V}_m^L$	The labeled node set of $\mathcal{G}_m$.
$\mathcal{V}_m^U$	The unlabeled node set of $\mathcal{G}_m$.
$\mathcal{N}_m^L$	The number of labeled nodes for the m-th client.
$\mathcal{N}_m^U$	The number of unlabeled nodes for the m-th client.
f_{θ_m}	The classifier of the m-th client.
$\hat{\mathcal{Y}}_m$	The prediction matrix of $\mathcal{G}_m$.
$\hat{\mathcal{Y}}_m^N$	The noisy label matrix of $\mathcal{G}_m$.
$\mathcal{L}$	The loss function.
ϵ	The fixed probability that the true label is flipped to a specific paired class.
z_{m_i}	The hyperspherical embedding representation of node i in $\mathcal{G}_m$.
$\mathbf{p}_m^{c,k}$	The k-th prototype of the c-th class of the m-th client.
$\omega_m^{c,k}$	The k-th prototype weight of the c-th class of the m-th client.
κ	The concentration parameter.
$\mathbf{S}_m$	The similarity metrix between prototypes in the m-th client.
$\mathbf{S}_m^{pos}$	The similarity matrix between intra-class prototypes in the m-th client
$\mathbf{S}_m^{neg}$	The similarity matrix between inter-class prototypes in the m-th client
Λ_c	The covariance matrix of all prototypes in class c.
η	The pruning ratio of detected nodes.
ϕ^{pos}	The clean clients set.
ϕ^{neg}	The malicious clients set.
α	The struct loss weight.
β	The semantic loss weight.
λ	The hyperparameter controlling the entropy regularization term.

Table 4: Notation and Definitions.

B Related work.

Federated Graph Learning(FGL). FGL enhances Federated Learning (FL) by extending it to graph-structured data, facilitating decentralized training while safeguarding raw graph data, thereby bolstering privacy protection[14, 19, 30, 4, 53, 52]. Current research primarily focuses on addressing the non-IID data problem in FGL. For instance, FedGCN [61] employs an attention mechanism to dynamically reweight local model parameters, mitigating the impact of data distribution heterogeneity. FGSSL [16] further decomposes the Non-IID issue into node-level semantic divergence and graph-level structural discrepancy, calibrating them separately. However, these approaches overlook a critical practical challenge: label noise caused by annotator negligence or bias may extensively exist in local data, significantly degrading the global model's generalization performance. To address this limitation, we propose a robust FGL framework grounded in hypersphere representation learning, which enhances the model's capacity to capture subtle structural differences in graph data, thereby maintaining stable classification performance under label noise.

Robust Federated Learning and Graph Learning. The existing solutions to the noisy label problem in FL can be broadly classified into two categories: label correction and self-supervised learning. The first category involves label correction mechanisms, which reassign noisy labels based on representations extracted from the training data. These include methods like nearest neighbors in the embedding space [48] and predictions from the global model [60]. The second category leverages self-supervised learning to obtain more robust representations, as seen in methods such as RoFL [10] and FedNed [33]. For instance, FedNed [33] reduces the risk of propagating incorrect information by using noisy client negative distillation, while FedDPCont [8] promotes robust learning by randomly selecting contrastive labels and sharing them with the server. Analogous challenges are also prevalent in the domain of graphs, where noisy labels and structural complexities similarly hinder model performance. In recent years, a growing body of research has focused on developing GNN methods tailored for robust graph learning under label noise. Some methods have achieved significant success by incorporating techniques such as loss modulation [39, 66, 29, 9], robust training strategies [56], graph structure augmentation [6, 40, 67], and contrastive learning [64, 27]. However, all these methods are not conducive to a more fine-grained structural learning, leading to the mixing of valid signals and noise components in the feature space, and our method is the first attempt at leveraging hypersphere learning for robust federated graph learning.

C Experimental Details.

C.1 Dataset Details

To assess the effectiveness of `HYPERION`, we conduct experiments on eight real-world graph datasets: Cora, CiteSeer, PubMed, Physics, and Amazon-ratings. 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. 5. 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-ratings.** This dataset is derived from the Amazon product co-purchasing network metadata in the SNAP repository. Nodes represent products (books, music CDs, DVDs, VHS tapes), and edges connect products that are frequently purchased together. The task is to predict the average rating given to a product by reviewers. The authors categorize the possible rating values into five classes. For node features, they use the average of fastText embeddings of the words in the product descriptions. To reduce the size of the graph, only the largest connected component of the 5-core subgraph is considered.
- **Coauthor-Physics.** Coauthor-Physics is an academic network containing co-authorship relationships based on the Microsoft Academic Graph. Nodes in the graph represent authors and edges represent co-authorship relationships. In the dataset, authors are categorized into five classes based on their research areas, and the nodes are characterized as bag-of-words representations of keywords of papers.

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
Amazon-ratings	24,492	97,050	5	300
Coauthor-Physics	34,493	530,417	5	8,415

Table 5: **Statistics** of datasets used in experiments.

C.2 Counterpart Details

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

- **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 [26, 38].
- **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.
- **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]. FedTAD addresses subgraph heterogeneity in FL by decomposing local graph variations into label and structural differences, preventing inconsistent model aggregation. It enhances knowledge transfer via topology-aware distillation, boosting FL reliability and efficiency.
- **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 [69]. By leveraging graph structures in aggregation, it enhances scalability and performance in federated graph learning.
- **MOON** [CVPR21]. MOON adopts a model-contrastive approach to address data heterogeneity in federated learning. The framework utilizes similarities between model representations to correct local training through model-level contrastive learning, providing an effective solution for collaborative training with deep learning models on image datasets while preserving data privacy.
- **FedNoRo** [IJCAI23]. FedNoRo adopts a two-stage framework to address class-imbalanced global data with heterogeneous label noise in federated learning. The method first identifies noisy clients through per-class loss indicators and Gaussian Mixture Modeling, then performs noise-robust federated updates via joint knowledge distillation and distance-aware aggregation, specifically designed for realistic medical scenarios with data imbalance and complex noise patterns.
- **FedNed** [AAAI24]. FedNed adopts a negative distillation framework to effectively leverage extremely noisy clients in federated learning. The method first identifies noisy clients, then innovatively utilizes them as 'bad teachers' through a dual-training approach: one model trained on original noisy labels for reverse knowledge distillation, and another on global model-generated pseudo-labels for conditional participation in aggregation. This approach transforms noisy clients from detrimental elements into valuable contributors while progressively enhancing their trustworthiness through pseudo-label refinement
- **FedCorr** [CVPR22]. FedCorr adopts a multi-stage framework to address heterogeneous label noise in federated learning while preserving data privacy. The method first dynamically identifies noisy clients through model prediction subspace analysis and per-sample loss evaluation, then employs an adaptive local proximal regularization to handle data heterogeneity. After fine-tuning on clean clients and correcting labels for noisy ones, FedCorr performs final training across all clients to fully utilize available data, effectively handling varying noise levels without requiring prior assumptions about client noise models.
- **CRGNN** [NN24]. CRGNN addresses label noise in GNNs by combining neighborhood-based label correction and contrastive learning. It utilizes message passing neural networks to update

Category	Methods	20% Label Noise						70% Label Noise					
		Cora		CiteSeer		PubMed		Cora		CiteSeer		PubMed	
		Uniform	Pair	Uniform	Pair	Uniform	Pair	Uniform	Pair	Uniform	Pair	Uniform	Pair
FL	FedAvg [ASTAT17]	48.00$_{\uparrow 00.00}$	48.41$_{\uparrow 00.00}$	48.57$_{\uparrow 00.00}$	45.00$_{\uparrow 00.00}$	80.18$_{\uparrow 00.00}$	77.08$_{\uparrow 00.00}$	18.34$_{\uparrow 00.00}$	21.31$_{\uparrow 00.00}$	23.24$_{\uparrow 00.00}$	23.53$_{\uparrow 00.00}$	24.56$_{\uparrow 00.00}$	19.29$_{\uparrow 00.00}$
	FedNova [NeurIPS20]	53.50$_{\uparrow 05.50}$	53.47$_{\uparrow 05.06}$	44.15$_{\downarrow 04.42}$	49.05$_{\uparrow 04.05}$	81.71$_{\uparrow 01.53}$	79.86$_{\uparrow 02.78}$	21.89$_{\uparrow 03.55}$	18.12$_{\downarrow 03.19}$	23.45$_{\uparrow 00.21}$	22.03$_{\downarrow 01.50}$	24.24$_{\downarrow 00.32}$	17.43$_{\downarrow 01.86}$
	FedProx [MLSys20]	52.25$_{\uparrow 04.25}$	53.76$_{\uparrow 05.35}$	39.64$_{\downarrow 08.93}$	46.20$_{\uparrow 01.20}$	79.61$_{\downarrow 00.57}$	78.00$_{\uparrow 00.92}$	21.05$_{\uparrow 02.71}$	**23.26**$_{\uparrow 01.95}$	24.04$_{\uparrow 00.80}$	**25.28**$_{\uparrow 01.75}$	24.11$_{\downarrow 00.45}$	18.81$_{\downarrow 00.48}$
	MOON [CVPR21]	48.51$_{\uparrow 00.51}$	49.29$_{\uparrow 00.88}$	38.75$_{\downarrow 09.82}$	45.12$_{\uparrow 00.12}$	79.33$_{\downarrow 00.85}$	78.34$_{\uparrow 01.26}$	18.28$_{\downarrow 00.06}$	20.62$_{\downarrow 00.69}$	22.43$_{\downarrow 00.81}$	21.81$_{\downarrow 01.72}$	21.64$_{\downarrow 02.92}$	19.05$_{\downarrow 00.24}$
FGL	FedGTA [VLDB24]	49.00$_{\uparrow 01.00}$	48.60$_{\downarrow 00.19}$	37.32$_{\downarrow 11.25}$	45.16$_{\uparrow 00.16}$	79.69$_{\downarrow 00.49}$	76.59$_{\downarrow 00.49}$	18.93$_{\uparrow 00.59}$	21.75$_{\uparrow 00.44}$	22.18$_{\downarrow 01.06}$	23.20$_{\downarrow 00.33}$	$\underline{26.43}$$_{\uparrow 01.87}$	19.46$_{\uparrow 00.17}$
	FedTAD [IJCAI24]	47.72$_{\downarrow 00.28}$	47.61$_{\downarrow 00.80}$	38.02$_{\downarrow 10.55}$	44.36$_{\downarrow 00.64}$	79.97$_{\downarrow 00.21}$	78.04$_{\uparrow 00.96}$	20.29$_{\uparrow 01.95}$	21.14$_{\downarrow 00.17}$	21.10$_{\downarrow 02.14}$	22.30$_{\downarrow 01.23}$	19.07$_{\downarrow 05.49}$	19.40$_{\uparrow 00.11}$
	FGSSL [IJCAI23]	54.70$_{\uparrow 06.70}$	53.33$_{\uparrow 04.92}$	45.71$_{\downarrow 2.86}$	48.54$_{\uparrow 03.54}$	84.81$_{\uparrow 04.63}$	$\underline{83.35}$$_{\uparrow 06.27}$	21.75$_{\uparrow 03.41}$	20.04$_{\downarrow 01.27}$	22.25$_{\downarrow 00.99}$	20.42$_{\downarrow 03.11}$	11.51$_{\downarrow 13.05}$	13.60$_{\downarrow 05.69}$
Robust FL	FedNoRo [IJCAI23]	47.86$_{\downarrow 00.14}$	48.37$_{\downarrow 00.04}$	38.79$_{\downarrow 09.78}$	44.88$_{\uparrow 00.12}$	80.12$_{\downarrow 00.06}$	77.44$_{\uparrow 00.36}$	18.34$_{\uparrow 00.00}$	21.30$_{\downarrow 00.01}$	22.77$_{\downarrow 00.47}$	23.35$_{\downarrow 00.18}$	25.01$_{\uparrow 00.45}$	18.87$_{\downarrow 00.42}$
	FedNed [AAAI24]	51.63$_{\uparrow 03.63}$	48.30$_{\downarrow 00.11}$	43.19$_{\downarrow 05.38}$	48.99$_{\uparrow 03.99}$	78.86$_{\downarrow 01.32}$	79.60$_{\uparrow 02.52}$	19.05$_{\uparrow 00.71}$	21.26$_{\downarrow 00.05}$	24.48$_{\uparrow 01.24}$	23.71$_{\downarrow 00.18}$	25.06$_{\uparrow 00.50}$	18.44$_{\downarrow 00.85}$
	FedCorr [CVPR22]	35.94$_{\downarrow 12.06}$	38.78$_{\downarrow 09.63}$	47.55$_{\downarrow 01.02}$	37.32$_{\downarrow 07.68}$	70.45$_{\downarrow 09.73}$	71.80$_{\downarrow 05.28}$	19.70$_{\uparrow 01.36}$	18.70$_{\downarrow 02.61}$	20.21$_{\downarrow 03.03}$	24.51$_{\uparrow 00.98}$	11.27$_{\downarrow 13.29}$	21.59$_{\uparrow 02.30}$
Robust GL	CRGNN [NN24]	$\underline{58.15}$$_{\uparrow 10.15}$	$\underline{61.98}$$_{\uparrow 13.57}$	48.30$_{\downarrow 00.27}$	53.42$_{\uparrow 08.42}$	$\underline{84.41}$$_{\uparrow 04.23}$	82.81$_{\uparrow 05.73}$	$\underline{24.81}$$_{\uparrow 06.47}$	22.99$_{\uparrow 01.68}$	$\underline{26.11}$$_{\uparrow 02.87}$	22.95$_{\downarrow 00.58}$	11.27$_{\downarrow 13.29}$	12.07$_{\downarrow 07.22}$
	RTGNN [WWW23]	50.14$_{\uparrow 02.14}$	43.06$_{\downarrow 05.35}$	**53.48**$_{\uparrow 04.91}$	$\underline{53.53}$$_{\uparrow 08.53}$	83.35$_{\uparrow 03.17}$	82.59$_{\uparrow 05.51}$	18.99$_{\uparrow 00.65}$	14.68$_{\downarrow 06.63}$	19.30$_{\downarrow 03.94}$	14.50$_{\downarrow 09.03}$	11.82$_{\downarrow 12.74}$	$\underline{26.15}$$_{\uparrow 06.86}$
	CLNode [WSDM23]	49.62$_{\uparrow 01.62}$	49.95$_{\uparrow 01.54}$	43.64$_{\downarrow 04.93}$	48.51$_{\uparrow 03.51}$	78.33$_{\downarrow 01.85}$	78.58$_{\uparrow 01.50}$	20.47$_{\uparrow 02.13}$	19.78$_{\downarrow 01.53}$	22.22$_{\downarrow 01.02}$	22.91$_{\downarrow 00.62}$	25.83$_{\uparrow 01.27}$	17.85$_{\downarrow 01.44}$
Robust FGL	HYPERION	**62.18**$_{\uparrow 14.18}$	**64.67**$_{\uparrow 16.26}$	$\underline{53.10}$$_{\uparrow 04.53}$	**54.47**$_{\uparrow 09.47}$	**85.35**$_{\uparrow 05.17}$	**85.47**$_{\uparrow 08.39}$	**29.73**$_{\uparrow 11.39}$	$\underline{23.05}$$_{\uparrow 01.74}$	**27.19**$_{\uparrow 03.95}$	$\underline{24.59}$$_{\uparrow 01.06}$	**30.67**$_{\uparrow 06.11}$	**31.12**$_{\uparrow 11.83}$

Table 6: **Comparison with the state-of-the-art methods on three selected real-world datasets.** The noise is set to **20%** and **70%**, and the number of clients M is set to 10 throughout all experiments. The best and second-best results are highlighted with **bold** and $\underline{\text{underline}}$, respectively.

node representations, integrating graph contrastive learning for consistent representations across augmented graph views. Finally, CGNN employs an MLP for prediction distributions and iteratively corrects noisy labels by comparing them with their neighbors and choosing the most labels.

- **RTGNN** [WWW23]. RTGNN proposes a noise governance framework that combines self-reinforcement supervision for noisy label correction and consistency regularization to prevent overfitting. The method categorizes labels into clean and noisy types, then applies adaptive supervision by rectifying inaccurate labels and generating pseudo-labels for unlabeled nodes, enabling effective learning from clean labels while mitigating noise impact.
- **CLNode** [WSDM24]. CLNode adopt a curriculum learning strategy to mitigate the impact of label noise. To be specific, it first utilize a multi-perspective difficulty measurer to accurately measure the quality of training nodes. Then employ a training scheduler that selects appropriate training nodes to train GNN in each epoch based on the measured qualities. The authors demonstrated this method enhances the robustness of backbone GNN to label noise.

C.3 Implementation Details.

The experiments are conducted using NVIDIA GeForce RTX 4090 GPUs as the hardware platform, coupled with Intel(R) Xeon(R) Platinum 8336C CPU @ 2.30GHz. The deep learning framework employed was Pytorch, version 2.5.1, alongside CUDA version 12.2. Our network features a four-layer GCN backbone with uniform 384-dimensional hidden layers throughout the first three layers, each employing symmetric normalization (normalize=True) and ReLU activation, followed by 0.2 dropout for regularization. The final GCN layer produces compact 32-dimensional graph embeddings without activation. These embeddings are processed through a two-layer MLP head with ReLU activation in the hidden layer. The architecture optionally incorporates prototype learning with configurable parameters: each class maintains multiple 32-dimensional prototype vectors, and the prototype contrastive loss operates with a temperature parameter $\tau = 0.07$ to control the similarity distribution sharpness. All GCN layers implement symmetric normalization (normalize=True), and consistent dropout ($p = 0.2$) is applied after each intermediate layer to prevent overfitting. TP-HSL parameter α is set in the range $\{0.40, 0.50, 0.60\}$, β Is set in the range $\{0.65, 0.70, 0.75\}$.As for GA-SHP parameter λ and η, we set λ in the range $\{0.03, 0.04\}$, η in the range $\{0.92, 0.94, 0.96\}$. The number of communication rounds is 100 for all methods. The number of clients M is set to 10 throughout all experiments, except for Figure 3 (Third).

D Additional Experimental Results.

We place additional F1-macro score results under 0.2 and 0.7 noisy label ratios in Tab. 6.

E Broader Impact.

Our work is an important step in overcoming the widespread and imperceptible labeling noise in FGL. This approach can effectively enhance the topological attention of the model to discriminate the noise. This could lead to more robust and trustworthy graph learning systems in real-world federated environments, where data quality and consistency are often difficult to guarantee.

F Discussion on Limitations.

Although `HYPERION` has demonstrated significant success in efficiently capturing subtle topological differences between nodes of the same class and mitigating malicious noise through a hyperspherical representation, it still faces some limitations. Specifically, our current formulation primarily addresses class label noise, while other noise types (e.g., feature noise or adversarial edge perturbations) may require additional mechanisms beyond the proposed purification framework.

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]

   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, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Details are included in Appendix C.3.

   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: We have provided the societal impacts of the work (See Appendix E).

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.

References

[1] A. Abbas, E. Rusak, K. Tirumala, W. Brendel, K. Chaudhuri, and A. S. Morcos. Effective pruning of web-scale datasets based on complexity of concept clusters, 2024.

[2] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs, 2014.

[3] K. Canese and S. Weis. Pubmed: the bibliographic database. *The NCBI handbook*, 2(1), 2013.

[4] J. Chen, B. Li, Q. He, and K. He. Pamt: A novel propagation-based approach via adaptive similarity mask for node classification. *IEEE Transactions on Computational Social Systems*, 11(5):5973–5983, 2024.

[5] X. Chen and S. Li. A good view for graph contrastive learning. *Entropy*, 26, 2024.

[6] E. Dai, C. Aggarwal, and S. Wang. Nrgnn: Learning a label noise-resistant graph neural network on sparsely and noisily labeled graphs, 2021.

[7] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song. Adversarial attack on graph structured data. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[8] Z. Di, Z. Zhu, X. Li, and Y. Liu. Federated learning with local openset noisy labels. In A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, editors, *Computer Vision – ECCV 2024*. Springer Nature Switzerland, 2025.

[9] X. Du, T. Bian, Y. Rong, B. Han, T. Liu, T. Xu, W. Huang, Y. Li, and J. Huang. Noise-robust graph learning by estimating and leveraging pairwise interactions, 2023.

[10] X. Fang and M. Ye. Robust federated learning with noisy and heterogeneous clients. In *CVPR*, 2022.

[11] 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.

[12] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017.

[13] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[14] C. He, K. Balasubramanian, E. Ceyani, C. Yang, H. Xie, L. Sun, L. He, L. Yang, P. S. Yu, Y. Rong, P. Zhao, J. Huang, M. Annavaram, and S. Avestimehr. Fedgraphnn: A federated learning system and benchmark for graph neural networks, 2021.

[15] S. Huang, Y. Xu, H. Zhang, and X. Li. Learn beneficial noise as graph augmentation, 2025.

[16] W. Huang, G. Wan, M. Ye, and B. Du. Federated graph semantic and structural learning. 2023.

[17] W. Huang, M. Ye, B. Du, and X. Gao. Few-shot model agnostic federated learning. In *ACM MM*, pages 7309–7316, 2022.

[18] W. Huang, M. Ye, Z. Shi, H. Li, and B. Du. Rethinking federated learning with domain shift: A prototype view. In *CVPR*, 2023.

[19] 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.

[20] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[21] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, 2020.

[22] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[23] A. Konovalov, B. Strauss, A. Ritter, and B. O'Connor. Learning to extract events from knowledge base revisions. In *WWW*, 2017.

[24] Q. Li, B. He, and D. Song. Model-contrastive federated learning. In *CVPR*, pages 10713–10722, 2021.

[25] 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.

[26] 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.

[27] X. Li, Q. Li, D. Li, H. Qian, and J. Wang. Contrastive learning of graphs under label noise. *Neural Networks*, 172:106113, 2024.

[28] 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.

[29] Y. Li, J. yin, and L. Chen. Unified robust training for graph neuralnetworks against label noise, 2021.

[30] R. Liu and H. Yu. Federated graph neural networks: Overview, techniques and challenges. *arXiv preprint arXiv:2202.07256*, 2022.

[31] Z. Liu, G. Wan, B. A. Prakash, M. S. Y. Lau, and W. Jin. A review of graph neural networks in epidemic modeling, 2024.

[32] H. Lu, D. Gong, S. Wang, J. Xue, L. Yao, and K. Moore. Learning with mixture of prototypes for out-of-distribution detection, 2024.

[33] Y. Lu, L. Chen, Y. Zhang, Y. Zhang, B. Han, Y. ming Cheung, and H. Wang. Federated learning with extremely noisy clients via negative distillation, 2024.

[34] P. M. Mammen. Federated learning: Opportunities and challenges, 2021.

[35] K. V. Mardia, P. E. Jupp, and K. Mardia. *Directional statistics*, volume 2. Wiley Online Library, 2000.

[36] 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.

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

[38] 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.

[39] H. NT, C. J. Jin, and T. Murata. Learning graph neural networks with noisy labels, 2019.

[40] S. Qian, H. Ying, R. Hu, J. Zhou, J. Chen, D. Z. Chen, and J. Wu. Robust training of graph neural networks via noise governance. In *WSDM*, 2023.

[41] X. Ren, Z. Wu, W. He, M. Qu, C. R. Voss, H. Ji, T. F. Abdelzaher, and J. Han. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17. International World Wide Web Conferences Steering Committee, 2017.

[42] V. Sehwag, M. Chiang, and P. Mittal. Ssd: A unified framework for self-supervised outlier detection, 2021.

[43] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[44] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8135–8153, 2023.

[45] Y. Tan, Y. Liu, G. Long, J. Jiang, Q. Lu, and C. Zhang. Federated learning on non-iid graphs via structural knowledge sharing. In *AAAI*, 2023.

[46] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *AAAI*, 2022.

[47] V. Tsouvalas, A. Saeed, T. Ozcelebi, and N. Meratnia. Federated learning with noisy labels. *ArXiv*, 2022.

[48] V. Tsouvalas, A. Saeed, T. Ozcelebi, and N. Meratnia. Labeling chaos to learning harmony: Federated learning with noisy labels. *ACM Trans. Intell. Syst. Technol.*, 2024.

[49] G. Wan, W. Huang, and M. Ye. Federated graph learning under domain shift with generalizable prototypes. *AAAI*, 38, 2024.

[50] G. Wan, Z. Huang, W. Zhao, X. Luo, Y. Sun, and W. Wang. Rethink graphode generalization within coupled dynamical system. In *Forty-second International Conference on Machine Learning*, 2025.

[51] G. Wan, Z. Liu, X. Shan, M. S. Lau, B. A. Prakash, and W. Jin. Epidemiology-aware neural ode with continuous disease transmission graph. In *Forty-second International Conference on Machine Learning*, 2025.

[52] G. Wan, Z. Shi, W. Huang, G. Zhang, D. Tao, and M. Ye. Energy-based backdoor defense against federated graph learning. In *International Conference on Learning Representations*, 2025.

[53] G. Wan, Y. Tian, W. Huang, N. V. Chawla, and M. Ye. S3gcl: Spectral, swift, spatial graph contrastive learning. In *Forty-first International Conference on Machine Learning*, 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] T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[56] X. Wei, X. Gong, Y. Zhan, B. Du, Y. Luo, and W. Hu. Clnode: Curriculum learning for node classification. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, WSDM '23, page 670–678, New York, NY, USA, 2023. Association for Computing Machinery.

[57] H. Wu, C. Wang, Y. Tyshetskiy, A. Docherty, K. Lu, and L. Zhu. Adversarial examples for graph data: Deep insights into attack and defense. In *IJCAI19*, 2019.

[58] N. Wu, L. Yu, X. Jiang, K.-T. Cheng, and Z. Yan. Fednoro: Towards noise-robust federated learning by addressing class imbalance and label noise heterogeneity, 2023.

[59] H. Xie, J. Ma, L. Xiong, and C. Yang. Federated graph classification over non-iid graphs. *NeurIPS*, 34:18839–18852, 2021.

[60] J. Xu, Z. Chen, T. Q. Quek, and K. F. E. Chong. Fedcorr: Multi-stage federated learning for label noise correction. In *CVPR*, pages 10184–10193, 2022.

[61] Y. Yao, W. Jin, S. Ravi, and C. Joe-Wong. Fedgcn: Convergence and communication tradeoffs in federated training of graph convolutional networks. *arXiv preprint arXiv:2201.12433*, 2022.

[62] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen. Graph contrastive learning with augmentations, 2021.

[63] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama. How does disagreement help generalization against label corruption? In *Proceedings of the 36th International Conference on Machine Learning*, pages 7164–7173. PMLR, 2019.

[64] J. Yuan, X. Luo, Y. Qin, Y. Zhao, W. Ju, and M. Zhang. Learning on graphs under label noise, 2023.

[65] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao. A survey on federated learning. *Knowledge-Based Systems*, 2021.

[66] M. Zhang, L. Hu, C. Shi, and X. Wang. Adversarial label-flipping attack and defense for graph neural networks. In *2020 IEEE International Conference on Data Mining (ICDM)*, 2020.

[67] Y. Zhu, L. Feng, Z. Deng, Y. Chen, R. Amor, and M. Witbrock. Robust node classification on graph data with graph and label noise. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):17220–17227, Mar. 2024.

[68] Y. Zhu, Y.-T. K. Lai, K. Zhao, X. Luo, M. Yuan, J. Ren, and K. Zhou. Binarizedattack: Structural poisoning attacks to graph-based anomaly detection. *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 2021.

[69] Y. Zhu, X. Li, M. Hu, and D. Wu. Federated continual graph learning. *arXiv preprint arXiv:2411.18919*, 2024.

[70] 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.

[71] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang. Deep graph contrastive representation learning, 2020.

[72] D. Zügner, A. Akbarnejad, and S. Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, 2018.