HGOOD-D: Hyperbolic Hierarchical Exploration for Graph Out-of-Distribution Detection

Yuntai Ding, Tao Ren, Yiwei Fu, Yifan Wang, Haodong Zhang, Chong Chen, Wei Ju, Xiao Luo, and Xian-Sheng Hua, *Fellow, IEEE*

***Abstract*—Out-of-distribution (OOD) detection has garnered increasing concern for identifying test samples that exhibit a distributional shift from the training dataset in practical deep learning applications. With the significant advancements in graph deep learning for graph representation, graph OOD detection has emerged as a research problem. Graph contrastive learning (GCL) is applied to graph OOD detection due to its capacity for learning discriminative representations in a self-supervised manner, thereby eliminating the need for time-consuming and labor-intensive label information. However, existing methods often neglect the explicit consideration of underlying semantics behind graph data distribution for OOD detection. We observe that naive data augmentations in GCL may inadvertently compromise the intrinsic graph structure while retaining redundant structural information, which hinders semantic discrimination between graphs. Additionally, euclidean space embedding struggles to maintain hierarchical structural consistency, making it challenging to meaningfully capture the hierarchical semantic distribution of graph data. In response to these issues, we propose a novel framework termed HGOOD-D, which aims to explore latent semantic hierarchies in hyperbolic space for graph OOD detection. Specifically, we design a bottleneck graph extractor grounded in the information bottleneck (IB) principle, which captures the minimal sufficient information to distinguish graph patterns. Based on this, we introduce hierarchical contrastive learning to capture the hierarchical semantics within graph data distribution. These methods are based on hyperbolic space embedding that can preserve complex inter-relationships in graph hierarchies, thereby mitigating data distortion. Comprehensive evaluations on ten widely used benchmark datasets show that HGOOD-D consistently surpasses current state-of-the-art approaches in graph OOD detection.**

***Index Terms*—Graph out-of-distribution detection, graph contrastive learning, hyperbolic embedding.**

I. Introduction

GRAPHS serve as a fundamental structure for modeling complex relational data and are widely applied across diverse fields, such as molecular structures [1], biological networks [2] and recommender systems [3]. Especially graph neural networks (GNNs), which leverage neighborhood aggregation and hierarchical pooling strategies to iteratively update node representations and generate a final graph-level representation, have attracted considerable interest over the past decade [4], [5], [6]. Their powerful representational capabilities have enabled successful applications in tasks such as node/graph classification [7], [8], link prediction [9], and graph generation [10].

Despite their great success, similar to other modern machine-learning methods, existing GNN models are generally constructed with the assumption that the testing data is sampled from the same in-distribution (ID) as the training set. This assumption often fails in practical applications, especially when processing complex graphs with insufficient labeling. As a result, GNN models always struggle with out-of-distribution (OOD) test data that were not encountered during the training phase. Ideally, a trustworthy model should not only work effectively with ID graphs but also be capable of identifying OOD input samples, allowing it to take appropriate precautions. This underscores the essential role of graph OOD detection in real-world settings.

As a fundamental problem in machine learning, OOD detection has been investigated across various domains, particularly in safety-critical applications such as medical diagnosis [11] and autonomous driving [12]. However, most existing works focus on image/text data [13], [14], [15], and the graph-structured data has been less explored. Generally, identifying OOD graph samples is inherently challenging, as non-euclidean graph data involves both diverse node features and intricate structural information. Recently, several pioneering studies have started to study graph OOD detection. For example, GraphDE [16] takes a graph-generative approach to train a supervised graph OOD detection model. Since the scarcity of OOD samples, GOOD-D [17], AAGOD [18] and GOODAT [19] further focus on unsupervised graph OOD detection models trained from scratch, post-hoc and at test-time respectively.

Received 25 April 2025; revised 20 March 2026; accepted 27 April 2026. Date of publication 5 May 2026; date of current version 2 June 2026. The work of Tao Ren was supported in part by the National Natural Science Foundation of China under Grant 62276058 and Grant 41774063, and in part by the Fundamental Research Funds for the Central Universities under Grant N25GFZ011. The work of Yifan Wang was supported in part by the Fundamental Research Funds for the Central Universities in UIBE under Grant 23QN02, and in part by the Humanities and Social Sciences Research Fund of the Ministry of Education of China under Grant 25YJCZH275. Recommended for acceptance by A. Zufle.*(Corresponding author: Yifan Wang.)*

Yuntai Ding, Tao Ren, and Haodong Zhang are with Software College, Northeastern University, Shenyang 110169, China (e-mail: chinaytding@163.com; chinarentao@163.com; 2110496@stu.neu.edu.cn).

Yiwei Fu and Wei Ju are with Peking University, Beijing 100871, China (e-mail: fuyw@stu.pku.edu.cn; juwei@pku.edu.cn).

Yifan Wang is with the School of Artificial Intelligence and Data Science, University of International Business and Economics, Beijing 100029, China (e-mail: yifanwang@uibe.edu.cn).

Chong Chen is with Hesicare Technology Co. Ltd, Shanghai 201210, China (e-mail: chenchong@hesicare.com).

Xiao Luo is with the Department of Statistics, University of Wisconsin–Madison, Madison, WI 53706 USA (e-mail: xiao.luo@wisc.edu).

Xian-Sheng Hua is with the Institute of AI for Engineering, Tongji University, Shanghai 200092, China (e-mail: huaxiansheng@gmail.com).

Digital Object Identifier 10.1109/TKDE.2026.3690579

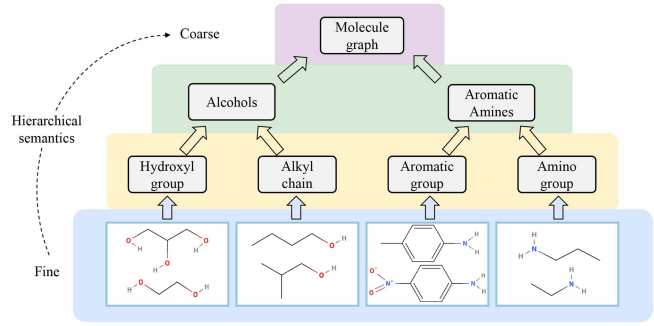


Fig. 1. An example of hierarchical semantic relations in molecular data. The semantic hierarchy exists in the inherent properties of the molecular structure, e.g., "molecular instance → hydroxyl group → alcohols → molecule graph" ranging from fine-level to coarse-level semantics.

However, these existing graph OOD detection approaches continue to be inadequate for the following reasons: **(i) Essential substructures for graph OOD detection are poorly identified.** OOD graphs differ from ID graphs in several ways, including feature-level variations (i.e., descriptive features of nodes) and structural deviations (i.e., graph topologies). These distinctions can be effectively identified through the examination of the graph's substructure, which plays a crucial role in characterizing graph properties and capturing the underlying patterns of the data. For example, the OH group at the end of the compound is crucial in distinguishing between Alcohol and Alkane [20]. **(ii) Fail to retain potential dependencies and inter-relationships in the hierarchical semantic relationships among graphs.** In the real world, graph datasets inherently exhibit hierarchical semantic characteristics, which are essential for capturing the underlying data distribution. As shown in Fig. 1, starting from the hydroxyl group (OH-group) at the bottom, progressing through alcohols, and finally reaching the molecular graph at the top, we can extract tree-structured hierarchies with progressively coarser semantic levels. These structures, which allow the learned representation to preserve data relationships, offer valuable prior information for graph OOD detection.

Recent advancements in hyperbolic geometry facilitate the embedding of hierarchical trees into hyperbolic spaces [21], [22]. Moreover, due to the conformal property of the Poincaré model and its local approximation to euclidean geometry [21], the mapping from euclidean to hyperbolic space can largely preserve the semantic relationships captured in euclidean representations. Unlike the polynomial growth in euclidean spaces, hyperbolic space follows a conformal scaling factor that tends to infinity adjacent to the boundary of the ball, leading to exponential volume growth with respect to the radius. This characteristic facilitates the learning of lower-dimensional embeddings for graph representation, yielding more compact yet powerful embedding spaces. Meanwhile, hyperbolic spaces naturally represent data with tree-like structures [23], [24], which could effectively address the high distortion issue encountered in euclidean embeddings when modeling hierarchical relational data.

Based on these insights, in this paper, we propose a **H**yperbolic **G**raph-**O**ut-**O**f-**D**istribution **D**etection method termed HGOOD-D, which identifies critical graph substructures and organizes hierarchical relationships within hyperbolic geometry (e.g., the Poincaré model) to capture meaningful semantic relationships among ID graphs. Specifically, given the input graph, we design a bottleneck graph masker to compress the graph based on the information bottleneck principle [25], [26], [27], thereby extracting informative key subgraphs that capture its ID-specific structural patterns. Then, for both the original graph and the extracted graph rationale, we holistically encode the graph features in euclidean space and subsequently map them into hyperbolic space to construct a hierarchical multi-view. For each view, hierarchical semantic structures among graphs are constructed via the proposed k-means algorithm in hyperbolic space, which allows for an effective representation of the hierarchical dependencies and inter-relationships across the graph dataset. Building on this, we design a hierarchical contrastive learning framework, which aligns graph instances with their corresponding prototypes across different semantic granularities, effectively mining accurate cross-granularity affiliations within graph hierarchies. Thus, given each input graph sample, the disagreement between the two views and their semantic disconfirmation across varying granularities in hyperbolic space can be utilized as an indicative scoring function for OOD detection.

To summarize, the key contributions of our proposed HGOOD-D are outlined as follows:

- *Conceptual:* We propose to learn the hierarchical semantic structure to capture the underlying ID graph distribution. From what we have surveyed, this represents the first endeavor to explicitly consider the underlying semantics for graph OOD detection.
- *Methodological:* We introduce a bottleneck graph masker to extract subgraphs that capture ID patterns, and explore latent semantic hierarchies in hyperbolic space, enabling hierarchical contrastive learning to effectively measure OOD scores.
- *Experimental:* To assess the performance of our proposed HGOOD-D, we perform comprehensive evaluations across multiple benchmark datasets. Experiments show the effectiveness and interpretability of our framework for graph OOD detection.

II. Related Work

A. Out-of-Distribution Detection

OOD detection has drawn significant attention in machine learning, aiming to identify test examples that diverge from the distribution of training datasets in real-world applications. Existing OOD detection methods efforts are mostly in the fields of computer vision [13], [14], [28] and natural language processing [29]. For example, ODIN [13] distinguishes OOD image data by leveraging the softmax scores obtained from a pre-trained model categorized by classes. SSD [14] employs contrastive learning to obtain image feature representations, followed by a clustering-based detection method that relies on the Mahalanobis distance metric. NGC [28] selects clean ID data using confidence-based and geometry-based sample

selection approaches for contrastive learning to obtain image feature representations. Contra-OOD [29] proposes to fine-tune the Transformers with contrastive learning, which improves the compactness of representations, thereby benefiting OOD detection on NLP datasets.

Recently, there has been a growing focus on the application of the OOD detection task to graph-structured data. In addition, graph anomaly detection aligns with the objectives of OOD detection and can be viewed as one of its subdomains [30]. Due to the significant cost involved in manual label annotation (e.g., drug-target interactions), we adopt a self-supervised framework to learn graph feature representations. GOOD-D [17] designs a multi-level hierarchical graph contrastive learning approach to capture semantically rich graph feature representations, enabling the distinction between ID and OOD data, while also establishing a benchmark for graph-level OOD detection. AAGOD [18] designs a learnable adaptive amplifier generator as a key pattern to enhance graph OOD detection. GOODAT [19] implements a test-time graph OOD detection method by extracting informative subgraphs related to the predicted pseudo-labels on the test set. HGOE [31] enhances OOD detection performance through hybrid graph outlier exposure. Although these methods adopt different strategies for graph OOD detection, they generally ignore the extraction of essential substructures and the exploration of latent semantic relationships across graphs. Consequently, they fail to sufficiently capture the underlying ID graph distribution, leading to suboptimal performance. In contrast, we introduce a bottleneck subgraph extraction module to extract an informative subgraph that captures ID patterns and construct a hierarchical multi-view to enforce the consistency of hierarchical relationships within ID graphs, thereby distinguishing them from OOD samples.

B. Hyperbolic Embedding

Hyperbolic embedding techniques have been successfully applied to computer vision [24], [32], natural language processing [33] and recommender systems [34] due to their suitability for representing data that inherently exhibits complex hierarchical structures. The characteristic that the distance in hyperbolic ball space increases exponentially with the increase in radius allows hyperbolic embeddings in lower dimensions to represent more samples compared to euclidean embeddings. HHCH [24] introduces hyperbolic space into hierarchical hash learning to minimize information distortion and accurately capture the hierarchical relationships in visual data. HPDR [32] incorporates prototype learning to uncover the inherent hierarchical semantics in hyperbolic space and refine domain alignment in face anti-spoofing data. HIT [33] maps the output of language models (LMs) to hyperbolic space and retrains transformer encoder-based LMs for the explicit encoding and interpretability of hierarchies. HNCR [34] leverages hyperbolic geometry, which naturally aligns with the power-law distribution of real-world user-item interactions, to minimize data distortion and enhance recommendation performance. Motivated by the superior ability of hyperbolic geometry to model hierarchical structures, our work departs from existing graph OOD detection methods based on euclidean space by leveraging the radial structure of hyperbolic space to encode fine-to-coarse semantic hierarchies—an aspect that has been underexplored in previous research.

III. Problem Definition and Preliminaries

A. Problem Definition

We consider an ID dataset during training phase $\mathcal{D}_{train} = \{G_1, \ldots, G_N\}$, in which each graph is sampled from a specific distribution $\mathbb{P}^{in}$. Here, each graph $G_i = \{\mathcal{V}_i, \mathcal{E}_i, \boldsymbol{X}_i\}$ denotes the set of nodes $\mathcal{V}_i$, the set of edges $\mathcal{E}_i$ and the node features $\boldsymbol{X}_i \in \mathbb{R}^{|\mathcal{V}_i| \times d}$ with a dimensionality of d. We utilize adjacency matrix $\boldsymbol{A}_i \in \mathbb{R}^{|\mathcal{V}_i| \times |\mathcal{V}_i|}$ to describe the structure information of graph G_i, where each entry $\boldsymbol{A}_i(u, v) = 1$ if edge $(u, v) \in \mathcal{E}_i$, otherwise $\boldsymbol{A}_i(u, v) = 0$. Then, we can define the testing dataset as $\mathcal{D}_{test} = \mathcal{D}^{in}_{test} \cup \mathcal{D}^{out}_{test}$, where $\mathcal{D}^{in}_{test}$ and $\mathcal{D}^{out}_{test}$ contain graphs sampled from distribution $\mathbb{P}^{in}$ and OOD distribution $\mathbb{P}^{out}$, respectively. Given a test sample $G'_i \in \mathcal{D}_{test}$, the goal of graph OOD detection is to generate a score s to identify the source distribution (i.e., $\mathbb{P}^{in}$ or $\mathbb{P}^{out}$) of G'_i, defined as:

$$\text{Detection label} = \begin{cases} 1 \text{ (OOD)}, & \text{if } s_{G'_i} \geqslant \eta \\ 0 \text{ (ID)}, & \text{if } s_{G'_i} < \eta \end{cases}, \tag{1}$$

where η denotes the threshold.

B. Graph Neural Networks

Graph Neural Networks (GNNs) have manifested as a powerful model for extracting the representation of data with a graph topology. In a typical GNN, each graph updates its node representation by aggregating neighboring node information. This process can be represented as:

$$\boldsymbol{h}_v^{(l)} = \mathcal{C}^{(l)}\left(\boldsymbol{h}_v^{(l-1)}, \mathcal{A}^{(l)}\left(\left\{\boldsymbol{h}_u^{(l-1)}\right\}_{u \in \mathcal{N}(v)}\right)\right), \tag{2}$$

where $\boldsymbol{h}_v^{(l)}$ denotes the updated feature representation of node v after l iterations, with information aggregated from the neighboring set $\mathcal{N}(v)$, and $\boldsymbol{h}_v^0$ is initialized as the input x_v. $\mathcal{A}^{(l)}(\cdot)$ and $\mathcal{C}^{(l)}(\cdot)$ serve as two basic operations responsible for aggregating and combining neighbor information at layer $l-1$. In this manner, the GNN updating framework can be constructed as $\boldsymbol{H}^{(L)} = \text{GNN}(\boldsymbol{X}, \boldsymbol{A})$ after stacking L such layers. The graph-level representation is derived by combining all updated node features from each graph with a readout function, defined as:

$$\boldsymbol{z} = \text{READOUT}(\{\boldsymbol{H}^{(L)}\}), \tag{3}$$

where $\boldsymbol{z}$ denotes the entire graph's feature representation, READOUT$(\cdot)$ could involve mean readout or other graph-level pooling strategies, determined by the specific model architecture and task requirements [35], [36], [37].

C. Hyperbolic Geometry

Compared to euclidean geometry, hyperbolic geometry allows multiple lines through a single point that do not intersect a given line. Among the various hyperbolic geometry models, the most basic one is the Poincaré ball model $(\mathbb{B}^d_c, g^c)$, where d and $-c$ $(c > 0)$ are the dimension and constant negative curvature of the model, $\mathbb{B}^d_c = \{\boldsymbol{x} \in \mathbb{R}^d \mid c\|\boldsymbol{x}\|^2 < 1\}$ is an open ball

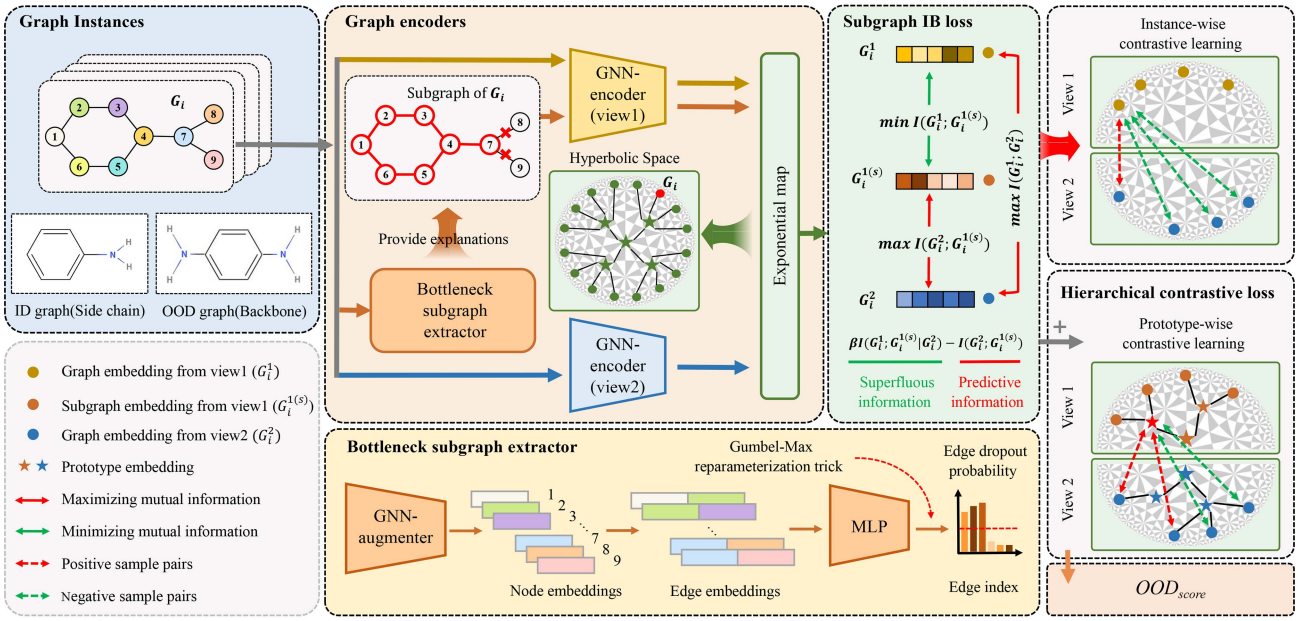


Fig. 2. The general framework of HGOOD-D. First, we construct a bottleneck graph extractor in parallel with the entire graph to generate two views for GCL. Then, different graph views are encoded in hyperbolic space via the exponential map. Finally, we introduce the subgraph IB loss and the hierarchical contrastive loss for graph-level contrast.

with radius $c^{-\frac{1}{2}}$. The corresponding Riemannian metric can be defined as:

$$g_{\boldsymbol{x}}^c = (\lambda_{\boldsymbol{x}}^c) I_d,\ \lambda_{\boldsymbol{x}}^c = \frac{2}{1 - c\|\boldsymbol{x}\|^2}, \tag{4}$$

where identify matrix I_d denotes the euclidean metric tensor and $\lambda_{\boldsymbol{x}}^c$ is a conformal factor defining how vectors are interpreted in the tangent space at a given point $\boldsymbol{x}$, which increases exponentially and tends to an infinite value when $\boldsymbol{x}$ approaching the surface of the ball.

IV. The Proposed Model

A. Overview

The core concept of our framework is to identify the critical substructures as rationale within the ID graphs and explore the corresponding semantics in a self-supervised manner for graph OOD detection. Since hyperbolic space effectively captures hierarchical relationships, it allows for a more natural modeling of complex graph pattern semantics. In this way, the absence of these patterns in OOD graph samples places them far from ID patterns in the feature space, making them easy to detect.

As shown in Fig. 2, our framework comprises four components. Given the input graph, we first extract a subgraph as the second view based on the multi-view information bottleneck (IB) principle, which captures the minimal sufficient information to explicitly distinguish the ID patterns of training data. Then, both views are projected into hyperbolic space (i.e., the Poincaré ball), and we construct hierarchical semantic structures through hyperbolic k-means clustering. Following this, we perform hierarchical contrastive learning to maximize semantic agreement across different granularities between the two views. For the test sample, we can finally estimate the OOD detection score by assessing the disagreement between two views and their semantic disconfirmation in hyperbolic space.

B. Bottleneck Subgraph Extraction

For a graph instance G_i, we aim to eliminate redundant information, focusing on a substructure to facilitate explicit OOD detection. However, general strategies often employ data augmentation methods via stochastic graph perturbations (e.g., node/edge dropping and feature modification), which may disrupt structural and semantic patterns and introduce undesired OOD samples. Drawing inspiration from the concept of IB, we introduce a bottleneck graph extractor to derive the subgraph from the original input $G_i = \{\mathcal{V}_i, \mathcal{E}_i, \boldsymbol{X}_i\}$. Specifically, we assign each edge $e \in \mathcal{E}_i$ with a random variable $p_e \sim \text{Bernoulli}(\omega_e)$, such that the edge e is retained in the subgraph when $p_e = 1$ and dropped otherwise, resulting in an extracted subgraph $G_i^{(s)}$. And ω_e denotes the Bernoulli weight, which can be defined as:

$$\omega_e = \text{MLP}([\boldsymbol{h}_u^{(L)}; \boldsymbol{h}_v^{(L)}]),\ \text{with}\ e = (u, v), \tag{5}$$

where $\{\boldsymbol{h}_v^{(L)}\}_{v \in \mathcal{V}_i} = \boldsymbol{H}_i^{(L)} = \text{GNN}_\phi(\boldsymbol{X}_i, \boldsymbol{A}_i)$ denotes the node embeddings refined by a GNN-based augmenter. To train the subgraph extractor in a continuously differentiable process, we compute the continuous variable p_e constrained within [0,1] and leverage the Gumbel-Max reparameterization trick to approximate the extraction process. Formally,

$$p_e = \text{Sigmoid}\left(\frac{\log \sigma - \log(1 - \sigma) + \omega_{uv}}{\tau}\right), \tag{6}$$

where $\sigma \sim \text{Uniform}(0,1)$ and τ is the temperature parameter. As τ approaches 0, p_{uv} converges to a binary value. In addition, we compute the average probability of edge removal as a regularization term, assigning corresponding weights λ_{reg} to control the extent of edge perturbations. We seek to extract the bottleneck subgraph, which serves as the graph rationale. Given two distinct and distinguishable views of the graph, G_i^1 and G_i^2, the mutual information (MI) between a graph and its subgraph can be decomposed as $I(G_i^1; G_i^{1(s)}) = I(G_i^1; G_i^{1(s)}|G_i^2) + I(G_i^2; G_i^{1(s)})$. Here, the first and second terms represent the superfluous and predictive information, which are expected to be minimized and maximized, respectively. The objective can be:

$$\max_{G_i^{1(s)}} I(G_i^2; G_i^{1(s)}) - \beta I(G_i^1; G_i^{1(s)}|G_i^2), \tag{7}$$

where $I(x;y)$ denotes the mutual information of two graph representations x and y, and β is the Lagrange multiplier. Since directly handling the second conditional MI term is challenging, we apply a transformation using the chain rule of the conditional MI as follows:

$$\begin{aligned} I(G_i^1; G_i^{1(s)}|G_i^2) &= I(G_i^1; G_i^{1(s)}, G_i^2) - I(G_i^1; G_i^2) \\ &= I(G_i^1; G_i^{1(s)}) + I(G_i^2; G_i^1|G_i^{1(s)}) - I(G_i^1; G_i^2) \\ &= I(G_i^1; G_i^{1(s)}) - I(G_i^1; G_i^2), \end{aligned} \tag{8}$$

where we suppose that when given $G_i^{1(s)}$, the amount of information G_i^2 can capture from G_i^1 is zero, i.e., $I(G_i^2; G_i^1|G_i^{1(s)}) = 0$, indicating that $G_i^{1(s)}$ can capture the important information of G_i^1. So in this way, we can rewrite the subgraph IB loss as:

$$\mathcal{L}_{IB} = \sum_{i=1}^{N} \left\{ -I(G_i^2; G_i^{1(s)}) + \beta \left[I(G_i^1; G_i^{1(s)}) - I(G_i^1; G_i^2) \right] \right\}. \tag{9}$$

We adapt the NCE-based MI lower bound [38] for approximately estimating the MI, defined as:

$$I(G_i, G_j) := \sum_{i=1}^{N} \log \frac{\exp(-D_c(G_i, G_j)/\tau)}{\sum_{j'=1}^{N} \exp(-D_c(G_i, G_{j'})/\tau)}, \tag{10}$$

where τ is the temperature parameter, one positive and $N-1$ negative samples are selected as denominators. In our paper, we define the distance $D_c(\cdot,\cdot)$ in hyperbolic space.

C. Hyperbolic Space Learning

To exploit the characteristics of hyperbolic geometry in capturing the latent hierarchical semantics within graph-structured data, we define a reversible transformation between euclidean space and the Poincaré ball model of hyperbolic space (*exponential*), and vice versa (*logarithmic*). Specifically, given two views $G_i^{1(s)}$ and G_i^2 of the graph, we leverage another GNN to extract the corresponding feature vector $\boldsymbol{z}_i^1$ and $\boldsymbol{z}_i^2$ in euclidean space, take $\boldsymbol{z}_i^1$ as an example:

$$\boldsymbol{z}_i^1 = \text{READOUT}(\text{GNN}_\theta(\boldsymbol{X}_i, \boldsymbol{A}_i^{1(s)})), \tag{11}$$

where $\boldsymbol{A}_i^{1(s)}$ denotes the adjacency matrix of the extracted subgraph. Then, for the feature $\boldsymbol{z}_i \in T_{\boldsymbol{x}}\mathbb{B}_c^d \setminus \{0\}$, where $\boldsymbol{x}$ is the base point in the hyperbolic manifold $\mathbb{B}_c^d$ and $T_{\boldsymbol{x}}\mathbb{B}_c^d$ denotes its tangent space, we project it onto the Poincaré disk via *exponential* mapping for $\boldsymbol{x} \neq 0$, which is defined as:

$$\exp_{\boldsymbol{x}}^c(\boldsymbol{z}_i) = \boldsymbol{x} \oplus_c \left(\tanh(\sqrt{c}\frac{\lambda_{\boldsymbol{x}}^c \|\boldsymbol{z}_i\|}{2}) \frac{\boldsymbol{z}_i}{\sqrt{c}\|\boldsymbol{z}_i\|} \right), \tag{12}$$

where $\|\cdot\|$ denote the euclidean norm and $\oplus_c$ is the Möbius addition for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{B}_c^d$ as:

$$\boldsymbol{x} \oplus_c \boldsymbol{y} := \frac{(1 + 2c\langle \boldsymbol{x}, \boldsymbol{y} \rangle + c\|\boldsymbol{y}\|^2)\boldsymbol{x} + (1 - c\|\boldsymbol{x}\|^2)\boldsymbol{y}}{1 + 2c\langle \boldsymbol{x}, \boldsymbol{y} \rangle + c^2\|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2}, \tag{13}$$

where $\langle \cdot \rangle$ denotes the inner product. In practice, $\boldsymbol{x}$ is set to 0, and the *exponential* map can be:

$$\boldsymbol{z}_i^c = \exp_0^c(\boldsymbol{z}_i) = \tanh(\sqrt{c}\|\boldsymbol{z}_i\|) \frac{\boldsymbol{z}_i}{\sqrt{c}\|\boldsymbol{z}_i\|}. \tag{14}$$

Nonetheless, we note that hyperbolic embeddings may approach the boundary of the Poincaré ball during training, resulting in gradient vanishing due to the degeneration of the Riemannian metric. To ensure stable optimization, we adopt a feature clipping strategy [39] by constraining the embedding norm with a clip_radius r before mapping into hyperbolic space, defined as:

$$\text{CLIP}(\boldsymbol{z}_i; r) = \min \left\{ 1, \frac{r}{\|\boldsymbol{z}_i\|} \right\} \cdot \boldsymbol{z}_i. \tag{15}$$

We then define the hyperbolic distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ as:

$$D_c(\boldsymbol{x}, \boldsymbol{y}) = \frac{2}{\sqrt{c}} \text{arctanh}(\sqrt{c}\| -\boldsymbol{x} \oplus_c \boldsymbol{y}\|). \tag{16}$$

In the limit $c \to 0$, $D_c(\boldsymbol{x}, \boldsymbol{y}) \to 2\|\boldsymbol{x} - \boldsymbol{y}\|$, recovering the euclidean distance up to a constant scaling factor. While euclidean space expands polynomially as radius increases, we can observe an exponential expansion in hyperbolic space during radius increase [21]. This intrinsic property of space expansion naturally aligns with tree-structured hierarchical semantic structures, since the number of nodes in a tree with branching factor K grows as $O(K^D)$ at depth D, which is analogous to the discrete form of hyperbolic radius growth and enables hyperbolic space to effectively embed hierarchical relationships. Remarkably, (16) shows that the distance of the point $\boldsymbol{x}$ to the origin is determined by its norm $\|\boldsymbol{x}\|$, which in hyperbolic space corresponds to its hierarchical level, while distances between embeddings capture similarity.

D. Hierarchical Contrastive Learning

We aim to construct a hierarchical semantic structure to capture the underlying data distribution of the ID graph. Toward this end, we propose a hierarchical clustering algorithm in the Poincaré ball model, which constructs the semantic structures in a bottom-up manner. Specifically, for hyperbolic embeddings $\boldsymbol{Z}^c$ obtained in the previous epoch, we perform hyperbolic k-means iteratively from the 1-st to the M-th layer, resulting in a prototype set $\mathcal{P} = \{\{\boldsymbol{p}_k^m\}_{k=1}^{K_m}\}_{m=1}^{M}$ with K_m denotes the prototype count at m-th layer. Since existing k-means variants rely on euclidean averaging to define the prototype of each cluster,

which is not compatible with hyperbolic space, we address this by maintaining the alternating optimization of prototypes and cluster assignments, while introducing two key modifications: 1) a distance metric specifically designed for hyperbolic space, and 2) a method for computing prototypes that is consistent with hyperbolic geometry. The first modification is addressed in (16). For the second, motivated by the previous work [24], [40], [41], we calculate the Einstein midpoint [42] as the prototype via the Klein model of hyperbolic space, which is defined as:

$$Klein\_Proto(\boldsymbol{z}_1^k,\ldots,\boldsymbol{z}_N^k)=\sum_{i=1}^{N}\gamma_i\boldsymbol{z}_i^k\Big/\sum_{i=1}^{N}\gamma_i, \tag{17}$$

where $\boldsymbol{z}_i^k$ is the point in Klein coordinates, $\gamma_i=\frac{1}{\sqrt{1-c\|\boldsymbol{z}_i^k\|^2}}$ denotes the Lorentz factor. Note that the Poincaré and Klein models are isomorphic [40]. Therefore, the transition formula holds:

$$\begin{aligned}\boldsymbol{z}_i^c&=\frac{\boldsymbol{z}_i^k}{1+\sqrt{1-c\|\boldsymbol{z}_i^k\|^2}},\\ \boldsymbol{z}_i^k&=\frac{2\boldsymbol{z}_i^c}{1+c\|\boldsymbol{z}_i^c\|^2}.\end{aligned} \tag{18}$$

Thus, by mapping to the Klein model, we compute the averaged prototypes of the given graph points and transform them back to the Poincaré ball model. In this way, the prototypes computed by hyperbolic k-means lie closer to the origin of the hyperbolic space, corresponding to coarser semantic levels. Through multi-layer iterations, the model progressively learns hierarchical semantics from fine to coarse. Then, for the two views $G_i^{1(s)}$ and G_i^2, we introduce a hierarchical contrastive learning framework to further distinguish the semantic patterns at multiple levels. Formally, for each graph point $\boldsymbol{z}_i^{1c}$ and $\boldsymbol{z}_i^{2c}$ from two views, we define them and its prototype $\mathcal{P}_m(\boldsymbol{z}_i^c)$ in m-th layer as positive pair and the remaining prototypes $\mathcal{N}_m(\boldsymbol{z}_i^c)$ as negative pairs. Take contrastive loss for $G_i^{1(s)}$ as an example:

$$\mathcal{L}_i^{1m}=-\log\frac{\exp(-D_c(\boldsymbol{z}_i^{1c},\mathcal{P}_m(\boldsymbol{z}_i^{2c}))/\tau)}{\sum_{\boldsymbol{p}_m\in\mathcal{P}_m(\boldsymbol{z}_i^{2c})+\mathcal{N}_m(\boldsymbol{z}_i^{2c})}\exp(-D_c(\boldsymbol{z}_i^{1c},\boldsymbol{p}_m)/\tau)}. \tag{19}$$

Note that we use the negative of the hyperbolic distance defined in (16) and apply the exponential function as a measure of similarity. In this way, graph embeddings with similar semantics—i.e., belonging to the same cluster prototype—are pulled closer together in hyperbolic space, whereas embeddings with different semantics are pushed apart. This allows the model to learn the underlying graph distributions across different hierarchical semantic. By considering the two views, we can define the prototype-wise hierarchical contrastive loss at different layers as:

$$\mathcal{L}_{PCL}=\frac{1}{M}\sum_{i=1}^{N}\sum_{m=1}^{M}\frac{1}{m}(\mathcal{L}_i^{1m}+\mathcal{L}_i^{2m}). \tag{20}$$

E. Adaptive Training and OOD Scoring

By combining the subgraph IB and hierarchical contrastive losses, the overall training objective can be defined as:

$$\begin{aligned}\mathcal{L}&=\mathcal{L}_{IB}+\mathcal{L}_{PCL}\\&=\mathcal{L}_{ICL}+\mathcal{L}_{PCL}+\sum_{i=1}^{N}\beta(I(G_i^1;G_i^{1(s)})-I(G_i^1;G_i^2)),\end{aligned} \tag{21}$$

where the first term $\mathcal{L}_{ICL}=\sum_{i=1}^{N}-I(G_i^2;G_i^{1(s)})$ can be seen as an instance-wise contrastive loss. However, manually adjusting the trade-off weights between these two contrastive losses is a challenging task, and treating them equally would ignore the varying sensitivities of the score, potentially leading to suboptimal performance. Referring to the previous work [17], we introduce adaptive training loss with the standard deviations of predicted errors as:

$$\begin{aligned}\mathcal{L}=&(\sigma_{ICL})^{\alpha}\mathcal{L}_{ICL}+(\sigma_{PCL})^{\alpha}\mathcal{L}_{PCL}\\&+\sum_{i=1}^{N}\beta(I(G_i^1;G_i^{1(s)})-I(G_i^1;G_i^2)),\end{aligned} \tag{22}$$

where σ_{ICL} and σ_{PCL} represent the standard deviations of the term for the training data, $\alpha>0$ is a hyper-parameter that governs the degree of automatic adaptation. In this way, the model penalizes loss terms with larger deviations, allowing it to focus more effectively on capturing the shared patterns of ID graph data.

During the test phase, we directly employ the trained subgraph extractor and GNN-based encoder to obtain the graph representations of different views. We do not perform hierarchical clustering on the test data and instead directly assign each test graph to the nearest prototype in the set $\mathcal{P}$ obtained during the training phase. The instance-wise and hierarchical prototype-wise contrastive loss of test graph data G_i' can be further used as the OOD score of test data, namely, $s_{G_i'}^{ICL}$ and $s_{G_i'}^{PCL}$. In this manner, the ID graph typically yields a lower consistency loss by aligning with the learned ID patterns, whereas the OOD graph can be effectively identified through the disagreement between its two views and their semantic inconsistency. To ensure the balance of scores, we further apply z-score normalization based on the mean and standard deviation of the term calculated from the training samples. The OOD score is then defined as follows:

$$s_{G_i'}=\frac{s_{G_i'}^{ICL}-\mu_{ICL}}{\sigma_{ICL}}+\frac{s_{G_i'}^{PCL}-\mu_{PCL}}{\sigma_{PCL}}, \tag{23}$$

where μ_{ICL} and μ_{PCL} are the mean values of the corresponding term for training data.

F. Complexity Analysis

To ensure scalability to large-scale graph-level datasets, we update the gradients using a mini-batch of size B. The computational consumption of our HGOOD-D mainly comprises the three components: (i) graph encoder module; (ii) prototype updating module; (iii) contrastive learning module. We consider

a graph dataset with N graphs, where each graph contains an average of $|\mathcal{V}|$ nodes and $|\mathcal{E}|$ edges. The node representation dimension is d and the number of GNN layers is L. Note that, we take the clustering layer count as M and the prototype count at the m-th layer as K_m. For (i), the time complexity of GNN-based encoder is $O(NL(|\mathcal{E}|d+|\mathcal{V}|d^2))$, with an additional cost of $O(N|\mathcal{E}|d^2)$ incurred by subgraph extraction, followed by a hyperbolic exponential mapping with time complexity $O(Nd)$. For (ii), the computational complexity of hierarchical hyperbolic clustering is $O(NK_1dt+K_1K_2dt+\cdots+K_{M-1}K_Mdt)$, where t denotes the maximum number of iterations in hyperbolic k-means. Since the number of prototypes satisfies $K_m \ll N$, the computational complexity of (ii) can be approximated as $O(NK_1dt)$. For (iii), we perform subgraph IB and hierarchical consistency regularization within mini-batch, which takes $O(B^2d)$ and $O(BK_1d+\cdots+BK_Md)\approx O(BK_1d)$. To sum up, the overall complexity of our HGOOD-D per training epoch is $O((|\mathcal{E}|(L+d)+|\mathcal{V}|Ld+B+K_1t+K_1+1)Nd)$, which scales linearly with the number of graphs and aligns with the latest self-supervised graph OOD detection methods (GOOD-D [17] and HGOE [31]).

Moreover, HGOOD-D requires subgraph extractor, prototypes and GNN-based encoders in contrastive learning training phase, the space complexity is approximately $O((|\mathcal{E}|+|\mathcal{V}|+d)Ld+|\mathcal{E}|+Nd+NMd+B^2+BK_1)$. Thus, our method is primarily influenced by the mini-batch size, enabling it to scale to graph-level datasets of varying sizes.

G. Theoretical Analysis

Theorem 4.1 (Hyperbolic Bound for Hierarchical Clustering [43], [44]): Let $\mathbb{B}_c^d$ be the Poincaré ball model with dimension d and curvature $-c$ ($c>0$), where the hyperbolic distance D_c is defined by (16). Denote $\mathcal{Z}^c=\{\boldsymbol{z}_i^c\}_{i=1}^N$ be the set of hyperbolic embeddings, and assume that an M-layer hierarchical clustering is performed on $\mathcal{Z}^c$. The prototype set is represented by $\mathcal{P}=\{\{\boldsymbol{p}_k^m\}_{k=1}^{K_m}\}_{m=1}^M$, where $\mathcal{C}^m=\{C_k^m\}_{k=1}^{K_m}$ is the clustering at layer m, and the number of clusters K_m decreases as m increases. For any pair of points $\boldsymbol{z}_i^c,\boldsymbol{z}_j^c$, we define their hierarchical distance as:

$$d_H(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c):=\begin{cases}\delta(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c), & \text{if such } m \text{ exists},\\ M+1, & \text{otherwise.}\end{cases} \tag{24}$$

Here, $\delta(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c):=\min\{\,m\in\{1,\ldots,M\}\mid \exists\, k \text{ s.t. } \boldsymbol{z}_i^c,\boldsymbol{z}_j^c\in C_k^m\,\}$. Then there exist positive constants $\alpha,\beta>0$ such that for any pair of points $\boldsymbol{z}_i^c,\boldsymbol{z}_j^c\in\mathbb{B}_c^d$, the hyperbolic distance D_c can be bounded by the hierarchical distance, that is,

$$\alpha\cdot d_H(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c)\le D_c(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c)\le\beta\cdot d_H(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c). \tag{25}$$

Proof: We begin by defining the cluster radius, which is estimated by the maximum distance from any point $\boldsymbol{z}\in C_k^m$ to its corresponding prototype:

$$R_m=\max_{k=1,\ldots,K_m}\max_{\boldsymbol{z}\in C_k^m} D_c(\boldsymbol{z},\boldsymbol{p}_k^m). \tag{26}$$

Similarly, the minimum and maximum distances between prototypes at the m-th layer are defined as follows:

$$\Delta_m^{\min}=\min_{k\neq l} D_c(\boldsymbol{p}_k^m,\boldsymbol{p}_\ell^m), \tag{27}$$

$$\Delta_m^{\max}=\max_{k\neq l} D_c(\boldsymbol{p}_k^m,\boldsymbol{p}_\ell^m). \tag{28}$$

Naturally, we assume the clustering is well-separated, i.e., $\Delta_m^{\min}>2R_m$ for all m, ensuring nontrivial lower bounds. Subsequently, we construct two positive constants $\alpha,\beta>0$, which are expressed as:

$$\alpha=\min_m\frac{\Delta_m^{\min}-2R_m}{m+1}, \tag{29}$$

$$\beta=\max_m\frac{2R_m+\Delta_m^{\max}}{m}, \tag{30}$$

where $\alpha>0$ holds by the assumption $\Delta_m^{\min}>2R_m$. Subsequently, we consider two cases: (i) $1\le d_H(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c)\le M$, and (ii) $1\le d_H(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c)=M+1$.

Case 1. Considering $d_H(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c)=m_0\in\{1,\ldots,M\}$: By definition of d_H, $\boldsymbol{z}_i^c$ and $\boldsymbol{z}_j^c$ belong to the same cluster $C_k^{m_0}$ for some k, but they are assigned to different clusters at layer m_0-1 (if $m_0>1$).

Upper bound: Since $\boldsymbol{z}_i^c,\boldsymbol{z}_j^c\in C_k^{m_0}$, we have:

$$\begin{aligned}D_c(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c)&\le D_c(\boldsymbol{z}_i^c,\boldsymbol{p}_k^{m_0})+D_c(\boldsymbol{p}_k^{m_0},\boldsymbol{z}_j^c)\\&\le 2R_{m_0}\le\beta\cdot m_0=\beta\cdot d_H,\end{aligned} \tag{31}$$

where the first equality holds by norm of Möbius addition and the last inequality follows from the definition of β in (30).

Lower bound: Let $\boldsymbol{p}_a^{m_0-1}$ and $\boldsymbol{p}_b^{m_0-1}$ be the prototypes of the distinct clusters containing $\boldsymbol{z}_i^c$ and $\boldsymbol{z}_j^c$ at layer m_0-1 (for $m_0=1$, interpret this as the trivial partition where each point forms its own cluster; then $\Delta_0^{\min}=\min_{i\neq j}D_c(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c)$ and $R_0=0$, so the bound still holds). By the triangle inequality for hyperbolic distance,

$$\begin{aligned}D_c(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c)&\ge D_c(\boldsymbol{p}_a^{m_0-1},\boldsymbol{p}_b^{m_0-1})-D_c(\boldsymbol{z}_i^c,\boldsymbol{p}_a^{m_0-1})\\&\quad-D_c(\boldsymbol{z}_j^c,\boldsymbol{p}_b^{m_0-1})\\&\ge\Delta_{m_0-1}^{\min}-2R_{m_0-1}\ge\alpha\cdot m_0=\alpha\cdot d_H,\end{aligned} \tag{32}$$

where the final inequality by the definition of α by (29).

Case 2. Considering $d_H(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c)=M+1$: The two points will never belong to the same cluster at any layer. Let $\boldsymbol{p}_a^M,\boldsymbol{p}_b^M$ be the prototypes of their corresponding clusters at the M-th layer. Similarly, by the triangle inequality, we could derive the inequality:

$$\begin{aligned}D_c(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c)&\le D_c(\boldsymbol{z}_i^c,\boldsymbol{p}_a^M)+D_c(\boldsymbol{p}_a^M,\boldsymbol{p}_b^M)+D_c(\boldsymbol{p}_b^M,\boldsymbol{z}_j^c)\\&\le 2R_M+\Delta_M^{\max}\le\beta\cdot(M+1)=\beta\cdot d_H,\end{aligned} \tag{33}$$

$$\begin{aligned}D_c(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c)&\ge D_c(\boldsymbol{p}_a^M,\boldsymbol{p}_b^M)-D_c(\boldsymbol{z}_i^c,\boldsymbol{p}_a^M)-D_c(\boldsymbol{z}_j^c,\boldsymbol{p}_b^M)\\&\ge\Delta_M^{\min}-2R_M\ge\alpha\cdot(M+1)=\alpha\cdot d_H.\end{aligned} \tag{34}$$

In summary, we could conclude that

$$\alpha\cdot d_H(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c)\le D_c(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c)\le\beta\cdot d_H(\boldsymbol{z}_i^c,\boldsymbol{z}_j^c). \tag{35}$$

□

V. Experiment

This section provides a detailed evaluation of HGOOD-D, focusing on the following research objectives:

- *RQ1:* How effectively does our proposed HGOOD-D fare against other baselines on graph OOD detection and anomaly detection tasks?
- *RQ2:* How do the individual key components of HGOOD-D impact the model's effectiveness?
- *RQ3:* How is the model's capability sensitive to various hyperparameters in HGOOD-D?
- *RQ4:* Are there illustrations of the impact of HGOOD-D on critical graph identification and hierarchical semantics?

A. Experimental Settings

1) Datasets: Building upon previous research [17], we utilize ten pairs of publicly available graph OOD detection datasets from the OGB [45] and TU [46] benchmarks to evaluate the effectiveness of our method. Each pair of datasets includes ID and OOD data originating from the same domain but exhibiting a certain degree of domain shift. Additionally, we validate our method on fifteen datasets selected from the TU benchmark [46] for graph anomaly detection task. In accordance with the settings in [17], [47], samples from the true anomalous or minority class are treated as abnormalities, while the remaining samples are classified as normal.

2) Baselines: We evaluate our model in comparison to various approaches on graph OOD detection and anomaly detection tasks. Concretely, the baselines could be grouped into three categories:

Two-Step Methods Based on Graph Kernel: We adopt a two-step framework to integrate diverse graph encoder and detector methods. In the first step, the encoder utilizes the propagation kernel (PK) [48] and the Weisfeiler-Lehman kernel (WL) [49] as graph kernels [50] to extract relevant graph features. In the second step, the detector employs isolation forest (iF) [51], one-class SVM (OCSVM) [52] and local outlier factor (LOF) [53] to identify OOD/anomalous graph samples based on the extracted graph representations.

Two-Step Methods Based on GCL: This type of method follows the same structure as described above, with the difference being that we employ general self-supervised GCL methods (i.e., InfoGraph [54] and GraphCL [55]) to learn graph features. Additionally, the detector methods are substituted by Mahalanobis distance-based (MD) approaches [14] and isolation forest (iF) [51].

End-to-end Methods: We also evaluate our approach against four popular end-to-end deep learning methods. Specifically, OCGIN [56] designs a GIN-based graph-level outlier detector by optimizing a one-class deep SVDD objective. GLocalKD [47] employs knowledge distillation to capture rich global and local normal pattern information. GOOD-D [17] utilizes multi-level graph contrastive learning to capture multi-scale consistency for OOD scoring. HGOE [31] adopts an external and internal hybrid outlier training strategy to enrich the training set.

3) Evaluation and Implementation: Following the previous work [17], we adopt Area under the ROC Curve (AUC) as evaluation metric on graph OOD detection and anomaly detection tasks. For each dataset pair, we calculate the average AUC along with its standard deviation across five trials with different random seeds. A higher AUC value reflects improved detection performance. For the graph OOD detection, we construct the training set by randomly selecting 90% of the ID graph dataset, and build the test set by including the remaining 10% of the ID graph dataset along with the same quantity of OOD graph dataset. For the graph anomaly detection, we use five-fold cross-validation to partition the training and test sets.

For all datasets, we use 5-layer GINs with 16 hidden dimensions as the bottleneck graph extractor. We then adopt non-shared 5-layer GINs with 16 hidden dimensions to extract feature information separately from both the original graph and subgraphs for hierarchical contrastive learning. During this process, we set the curvature and clip radius parameters of the Poincaré ball model to $c = 0.01$ and $r = 2.3$, respectively, and the dimensionality of hyperbolic embeddings to $d = 128$, thus converting graph features from euclidean space to hyperbolic space. For baseline methods, we adopt their source with original settings and reproduce the results. The source code of our HGOOD-D is available at https://github.com/BBDing-DYT/HGOOD-D.

B. Performance Comparison (RQ1)

We perform extensive experiments for graph OOD detection and anomaly detection tasks. The comparison results for these tasks are shown in Tables I and II. The best-performing method is indicated in **bold** while the runner-up is underlined. The findings indicate the following insights:

- In contrast to two-step methods, end-to-end methods demonstrate generally better performance on graph OOD detection task. This indicates that consistent learning objectives, which can globally account for training loss throughout the process and reduce the loss of graph feature information, play a critical role on OOD detection task. Additionally, among two-step methods, the self-supervised GCL-based feature extraction methods significantly outperform the methods based on graph kernel encoder, especially the GraphCL-MD method. This demonstrates that self-supervised GCL methods can extract more comprehensive and distinguishable graph feature representations without relying on data labels.
- Our HGOOD-D achieves the best performance compared to other baselines across all datasets on graph OOD detection task. Specifically, our HGOOD-D achieves 15.87% and 14.08% increment over the runner-up baseline HGOE on PTC-MR/MUTAG and ClinTox/LIPO. The significant improvement can be attributed to the nature of drug molecular graphs, where distinct molecular structures naturally exhibit discriminative properties. Moreover, the original datasets are formulated as relatively simple binary classification tasks, indicating that the semantic patterns of ID graphs are easier to learn. In contrast, the performance gain on Esol/MUV is relatively smaller, possibly because the original dataset is designed for a regression task, resulting in more complex semantic relationships within the data. The overall performance improvement demonstrates that subgraph extraction and the construction of

TABLE I
PERFORMANCE (%) OF GRAPH OOD DETECTION IN TERMS OF AUC (IN PERCENT, MEAN ± STD)

ID dataset OOD dataset	BZR COX2	PTC-MR MUTAG	AIDS DHFR	ENZYMES PROTEIN	IMDB-M IMDB-B	Tox21 SIDER	FreeSolv ToxCast	BBBP BACE	ClinTox LIPO	Esol MUV	Avg. Rank
PK-LOF	42.22±8.39	51.04±6.04	50.15±3.29	50.47±2.87	48.03±2.53	51.33±1.81	49.16±3.70	53.10±2.07	50.00±2.17	50.82±1.48	12.9
PK-OCSVM	42.55±8.26	49.71±6.58	50.17±3.30	50.46±2.78	48.07±2.41	51.33±1.81	48.82±3.29	53.05±2.10	50.06±2.19	51.00±1.33	12.8
PK-iF	51.46±1.62	54.29±4.33	51.10±1.43	51.67±2.69	50.67±2.47	49.87±0.82	52.28±1.87	51.47±1.33	50.81±1.10	50.85±3.51	11.1
WL-LOF	48.99±6.20	53.31±8.98	50.77±2.87	52.66±2.47	52.28±4.50	51.92±1.58	51.47±4.23	52.80±1.91	51.29±3.40	51.26±1.31	10.4
WL-OCSVM	49.16±4.51	53.31±7.57	50.98±2.71	51.77±2.21	51.38±2.39	51.08±1.46	50.38±3.81	52.85±2.00	50.77±3.69	50.97±1.65	11.0
WL-iF	50.24±2.49	51.43±2.02	50.10±0.44	51.17±2.01	51.07±2.25	50.25±0.96	52.60±2.38	50.78±0.75	50.41±2.17	50.61±1.96	12.3
InfoGraph-iF	63.17±9.74	51.43±5.19	93.10±1.35	60.00±1.83	58.73±1.96	56.28±0.81	56.92±1.69	53.68±2.90	48.51±1.87	54.16±5.14	8.5
InfoGraph-MD	86.14±6.77	50.79±8.49	69.02±11.67	55.25±3.51	81.38±1.14	59.97±2.06	58.05±5.46	70.49±4.63	48.12±5.72	77.57±1.69	7.6
GraphCL-iF	60.00±3.81	50.86±4.30	92.90±1.21	61.33±2.27	59.67±1.65	56.81±0.97	55.55±2.71	59.41±3.58	47.84±0.92	62.12±4.01	8.7
GraphCL-MD	83.64±6.00	73.03±2.38	93.75±2.13	52.87±6.11	79.09±2.73	58.30±1.52	60.31±5.24	75.72±1.54	51.58±3.64	78.73±1.40	5.3
OCGIN	76.66±4.17	80.38±6.84	86.01±6.59	57.65±2.96	67.93±3.86	46.09±1.66	59.60±4.78	61.21±8.12	49.13±4.13	54.04±5.50	8.0
GLocalKD	75.75±5.99	70.63±3.54	93.67±1.24	57.18±2.03	78.25±4.35	66.28±0.98	64.82±3.31	73.15±1.26	55.71±3.81	86.83±2.35	5.2
GOOD-D	94.99±2.25	81.21±2.65	99.07±0.40	61.84±1.94	79.94±1.09	66.50±1.35	80.13±3.43	82.91±2.58	69.18±3.61	91.52±0.70	3.1
HGOE	95.00±2.70	82.06±1.63	99.28±0.34	64.44±2.19	81.74±2.25	68.24±0.60	82.89±2.33	83.46±1.79	70.09±1.52	92.64±2.44	2.0
HGOOD-D	**98.73**±0.44	**96.82**±0.13	**99.96**±0.04	**66.25**±0.34	**83.82**±1.55	**69.29**±0.65	**84.28**±0.51	**85.14**±0.62	**79.96**±0.59	**93.28**±0.17	**1.0**

TABLE II
PERFORMANCE (%) OF GRAPH ANOMALY DETECTION IN TERMS OF AUC (IN PERCENT, MEAN ± STD)

Method	PK-OCSVM	PK-iF	WL-OCSVM	WL-iF	InfoGraph-iF	GraphCL-iF	OCGIN	GLocalKD	GOOD-D	HGOE	HGOOD-D
PROTEINS-full	50.49±4.92	60.70±2.55	51.35±4.35	61.36±2.54	57.47±3.03	60.18±2.53	70.89±2.44	77.30±5.15	71.97±3.86	73.13±0.46	**77.59**±1.28
ENZYMES	53.67±2.66	51.30±2.01	55.24±2.66	51.60±3.81	53.80±4.50	53.60±4.88	58.75±5.98	61.39±8.81	63.90±3.69	67.28±0.99	**71.63**±1.77
AIDS	50.79±4.30	51.84±2.87	50.12±3.43	61.13±0.71	70.19±5.03	79.72±3.98	78.16±3.05	93.27±4.19	97.28±0.69	**97.84**±0.55	90.17±0.34
DHFR	47.91±3.76	52.11±3.96	50.24±3.13	50.29±2.77	52.68±3.21	51.10±2.35	49.23±3.05	56.71±3.57	62.67±3.11	64.39±0.68	**70.37**±1.13
BZR	46.85±5.31	55.32±6.18	50.56±5.87	52.46±3.30	63.31±8.52	60.24±5.37	65.91±1.47	69.42±7.78	75.16±5.15	80.54±1.35	**86.15**±1.28
COX2	50.27±7.91	50.05±2.06	49.86±7.43	50.27±0.34	53.36±8.86	52.01±3.17	53.58±5.05	59.37±12.67	62.65±8.14	69.52±2.68	**72.99**±2.50
DD	48.30±3.98	71.32±2.41	47.99±4.09	70.31±1.09	55.80±1.77	59.32±3.92	72.27±1.83	**80.12**±5.24	73.25±3.19	76.95±2.24	73.38±1.72
NCI1	49.90±1.18	50.58±1.38	50.63±1.22	50.74±1.70	50.10±0.87	49.88±0.53	71.98±1.21	68.48±2.39	61.12±2.21	65.82±1.43	**72.86**±1.95
IMDB-B	50.75±3.10	50.80±3.17	54.08±5.19	50.20±0.40	56.50±3.58	56.50±4.90	60.19±8.90	52.09±3.41	65.88±0.75	69.82±1.37	**78.16**±0.86
REDDIT-B	45.68±2.24	46.72±3.42	49.31±2.33	48.26±0.32	68.50±5.56	71.80±4.38	75.93±8.65	77.85±2.62	88.67±1.24	89.41±1.21	**89.47**±0.40
COLLAB	49.59±2.24	50.49±1.72	52.60±2.56	50.69±0.32	46.27±0.73	47.61±1.29	60.70±2.97	52.94±0.85	72.08±0.90	74.24±0.73	**76.50**±0.43
HSE	57.02±8.42	56.87±10.51	62.72±10.13	53.02±5.12	53.56±3.98	51.18±2.71	64.84±4.70	59.48±1.44	69.65±2.14	74.50±3.73	**75.24**±0.96
MMP	46.65±6.31	50.06±3.73	55.24±3.26	52.68±3.34	54.59±2.01	54.54±1.86	71.23±0.16	67.84±0.59	70.51±1.56	71.94±0.54	**72.42**±2.03
p53	46.74±4.88	50.69±2.02	54.59±4.46	50.85±2.16	52.66±1.95	53.29±2.32	58.50±0.37	64.20±0.81	62.99±1.55	**64.70**±1.16	64.36±0.82
PPAR-gamma	53.94±6.94	45.51±2.58	57.91±6.13	49.60±0.22	51.40±2.53	50.30±1.56	71.19±4.28	64.59±0.67	67.34±1.71	71.92±4.17	**72.00**±1.47
Avg. Rank	9.7	8.6	7.8	8.4	7.4	7.8	4.8	4.0	3.4	2.0	**1.4**

hierarchical semantic relationships increase the distributional gap between ID and OOD data, thereby enhancing the model's OOD detection capability.

- Moreover, our HGOOD-D also outperforms all baselines on 12 datasets and performs well on 3 remaining datasets on graph anomaly detection task. Specifically, HGOOD-D obtains 11.95% and 9.29% improvement against the runner-up baseline on IMDB-B and DHFR. Overall, HGOOD-D achieves the top average rank among all baselines. The main reason is that our HGOOD-D graph extractor effectively captures the critical subgraph structures of normal graphs, enhancing the local structural features of the graphs, thereby enabling the distinction of a small number of anomalous samples within a large dataset.

C. Ablation Study (RQ2)

We attribute the improvements in our HGOOD-D to four key components: 1) **subgraph**: bottleneck subgraph extraction; 2) **prototype**: prototype-wise hierarchical contrastive learning; 3) **adaptive**: adaptive training and OOD scoring; 4) **embedding**: graph feature embedding space (i.e., euclidean space, hyper-sphere space and hyperbolic space). To examine the impact of each component, we conduct ablation studies by individually removing each designed module, as well as by employing different feature encoding spaces independently. We use OOD detection task as the basis for our ablation study, with detailed experimental results presented in Table III, leading to the following conclusions:

- Among the designed modules, each component contributes significantly to the detection results. The performance notably drops when the clustering of the prototype set is removed. This indicates that the construction of prototypes facilitates an underlying data distribution with hierarchical semantics through prototype-wise contrastive learning, significantly improving the model's capabilities for graph representations.
- Among the three embedding spaces, our model in the hyperbolic space outperforms all others by employing the corresponding distance measures to assess similarity. The average performance of HGOOD-D shows improvements of 2.61% and 1.42% when compared to euclidean and hyper-sphere space. This demonstrates that hyperbolic space provides a more expressive feature representation within the same dimensionality, thereby aiding the construction of hierarchical semantics.
- Our HGOOD-D, which incorporates all components, exhibits the optimal outcome over all datasets. This

TABLE III
ABLATION STUDY RESULTS (%) OF HGOOD-D AND ITS VARIANTS IN TERMS OF AUC (IN PERCENT, MEAN ± STD)

Setting	BZR COX2	PTC-MR MUTAG	AIDS DHFR	ENZYMES PROTEIN	IMDB-M IMDB-B	Tox21 SIDER	FreeSolv ToxCast	BBBP BACE	ClinTox LIPO	Esol MUV
HGOOD-D *w/o subgraph*	98.38±1.05	95.93±1.50	99.94±0.01	66.17±1.64	83.07±0.57	67.25±0.71	82.33±0.76	83.85±0.48	77.90±0.38	92.91±1.00
HGOOD-D *w/o prototype*	98.02±0.54	95.36±2.67	99.95±0.02	64.20±1.10	82.12±0.34	68.74±0.35	81.57±1.04	84.76±1.10	79.90±1.17	91.90±0.80
HGOOD-D *w/o adaptive*	98.22±0.32	96.12±1.30	99.96±0.02	66.10±0.86	83.28±1.89	66.09±0.68	80.49±2.05	83.45±0.72	79.68±2.02	93.13±0.43
Euclidean	97.94±0.27	96.01±1.04	99.77±0.12	65.47±1.11	75.98±0.74	68.91±0.36	80.04±0.08	82.15±0.88	78.83±1.63	91.62±1.46
Hyper-sphere	98.61±0.56	96.69±3.61	99.96±0.01	63.95±0.40	81.95±0.51	68.33±0.10	82.20±2.08	83.73±1.78	79.64±1.71	91.49±1.25
HGOOD-D	**98.73**±0.44	**96.82**±0.13	**99.96**±0.04	**66.25**±0.34	**83.82**±1.55	**69.29**±0.65	**84.28**±0.51	**85.14**±0.62	**79.96**±0.59	**93.28**±0.17

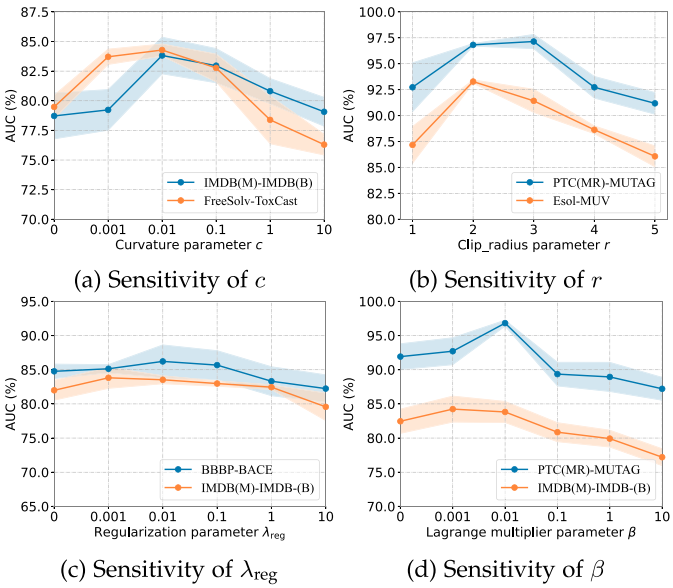


Fig. 3. Parameter sensitivity of c, r, λ_{reg} and β.

observation proves the efficacy of jointly extracting critical subgraph features and mapping them into hyperbolic space to construct hierarchical semantic relationships, while employing an adaptive strategy during training to optimize multiple objectives comprehensively.

D. Parameter Sensitivity (RQ3)

We also explore the dependency of the proposed HGOOD-D on various hyperparameters. Specifically, we explored the performance of our proposed HGOOD-D with the curvature parameter c and the clip _ radius parameter r of the hyperbolic space, edge dropout probability regularization weight parameter λ_{reg} and Lagrange multiplier β of bottleneck subgraph extraction during training phase, as well as K_m and M of the hierarchical hyperbolic k-means.

1) Effect of the Parameter of Hyperbolic Space: By leveraging hyperbolic space to construct the hierarchical structure, we additionally explore the effect of varying parameter values of curvature and clip _ radius. We select one dataset pair each from TU and OGB datasets to analyze the effectiveness of HGOOD-D for OOD detection task.

- Fig. 3(a) illustrates the performance of HGOOD-D with different settings of curvature c within a scope of $\{0, 0.001, 0.01, 0.1, 1, 10\}$ on IMDB(M)-IMDB(B) and FreeSolv-ToxCast dataset pair. From the result, we observe that when $c = 0$, the graph features effectively regress to the euclidean space, which has weaker expressive power, leading to a suboptimal performance, as discussed in Section IV-C. With increasing c, the features of the graph begin to be fully expanded. When $c = 0.01$, the model generally achieves optimal performance, illustrating that graph feature representation under hyperbolic space contributes to uncovering hierarchical semantic structures on the OOD detection task. However, too large curvature (i.e., $c > 0.01$) may weaken the model's performance. This is likely due to the larger curvature causing the distances between graph feature vectors near the boundary to be excessively compressed, thereby impairing the model's ability to effectively distinguish between graph features.
- Fig. 3(b) shows the performance results with varying values of clip _ radius r in a range of $\{1, 2, 3, 4, 5\}$ on PTC(MR)-MUTAG and Esol-MUV dataset pair. From the result, we discover that when $r = 1$, the poor performance suggests that the graph feature vectors are compressed into a smaller-radius spherical space, causing the model to focus on more localized structures, which may result in information loss. Generally, when $r = 2$ or 3, the model exhibits optimal efficacy. However, as r increases without appropriate constraints, accuracy suffers a significant drop. Such a trend may be caused by the relaxation of the distance constraint of graph feature vectors to the origin, leading to a more dispersed distribution of feature vectors. This results in instability in the distance calculations between feature vectors, leading to difficulties in model convergence.

2) Effect of the Weight of Optimization Objective: To analyze whether HGOOD-D can benefit from bottleneck subgraph extraction, we investigate the impact of different weights of edge dropout probability regularization λ_{reg} and Lagrange multiplier β in subgraph IB loss. We select one social dataset pair IMDB(M)-IMDB(B) and one other molecular and protein dataset pair (i.e., BBBP-BACE or PTC(MR)-MUTAG) to evaluate HGOOD-D for OOD detection task.

- We evaluate model performance during edge dropout probability regularization across different λ_{reg} ranges of $\{0, 0.001, 0.01, 0.1, 1, 10\}$. As shown in Fig. 3(c), when parameter $\lambda_{\text{reg}} = 0$, the model demonstrates poor performance, suggesting that the absence of constraints on edge dropout probability during subgraph extraction leads to unstable subgraph structures, resulting in suboptimal and inconsistent outcomes. As λ_{reg} increases, the model shows continuous performance enhancement, reaching its optimal level at $\lambda_{\text{reg}} = 0.01$. This indicates that the model effectively learns to extract key subgraph structures, which enhances its ability to distinguish semantic relationships

TABLE IV
PARAMETER ANALYSIS (%) ON CLUSTERING LAYER COUNT M AND PROTOTYPE COUNT AT DIFFERENT LAYERS K_m. THE SETTING OF K_m AND M IS PRESENTED AS $K_1 \rightarrow K_2 \rightarrow \cdots \rightarrow K_M$.

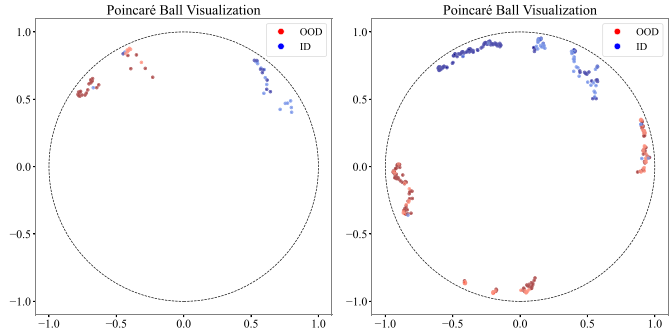
Dataset	Configuration of K_m and M	AUC
ENZYMES-PROTEIN	5	63.63±3.83
	$10 \rightarrow 2$	64.69±0.95
	$10 \rightarrow 5$	**66.25±0.34**
	$20 \rightarrow 5$	61.06±0.27
	$10 \rightarrow 5 \rightarrow 2$	65.90±0.15
	$20 \rightarrow 10 \rightarrow 5 \rightarrow 2$	62.42±2.28
Tox21-SIDER	5	67.27±2.01
	$10 \rightarrow 2$	68.29±0.14
	$10 \rightarrow 5$	68.72±0.54
	$20 \rightarrow 5$	66.14±0.34
	$10 \rightarrow 5 \rightarrow 2$	**69.29±0.65**
	$20 \rightarrow 10 \rightarrow 5 \rightarrow 2$	68.54±1.50

between different graphs, thereby improving OOD detection performance. However, excessively large weight λ_{reg} leads the model to extract structures almost identical to the original graph, thereby weakening its ability to capture the underlying semantic relationships of the graph.

- We investigate the impact of different Lagrange multipliers within the range of $\{0, 0.001, 0.01, 0.1, 1, 10\}$ on the model's ability to extract subgraph structures. As shown in Fig. 3(d), when Lagrange multiplier $\beta = 0$, the model neglects the superfluous information between the original graph and the subgraph, which prevents it from capturing the key structural information essential for distinguishing graphs with different semantics, thereby resulting in suboptimal performance. As the weight increases, the model's performance reaches a peak and then declines. This is likely due to the larger Lagrange multiplier causing the model to overly focus on the discrepancy loss between the extracted subgraph and the original graph, thereby leading to excessive loss of original information in the subgraph features and impairing the model's stable convergence.

3) Effect of the Clustering Hierarchy of Hyperbolic Space: By performing hierarchical clustering on graph embeddings in hyperbolic space to construct semantic structures, we further investigate the model's performance by varying clustering layer count M and prototype count at each layer K_m. We also select a pair of datasets from different databases (i.e., ENZYMES-PROTEIN and Tox21-SIDER) as test cases for the OOD detection task. From the results in Table IV, we can draw the subsequent observations:

- Relative to clustering with just one layer, HGOOD-D leads to improvement of 4.12% and 3.01% on ENZYMES-PROTEIN and Tox21-SIDER dataset pairs with optimal settings (i.e., $10 \rightarrow 5$ and $10 \rightarrow 5 \rightarrow 2$), respectively. However, when clustering with four layers, the model does not perform consistently as expected. This observation suggests that, on one hand, different datasets inherently contain hierarchical semantic information of varying complexity; on the other hand, as the number of layers increases, the high-level semantic features tend to introduce excessive noise due to the accumulated clustering errors from the lower-level features, thereby impairing the model's ability to capture meaningful high-level semantics.



(a) PTC(MR)-MUTAG (b) AIDS-DHFR

Fig. 4. The visualization of test samples in 128-d embeddings from PTC(MR)-MUTAG and AIDS-DHFR dataset pairs within the Poincaré ball.

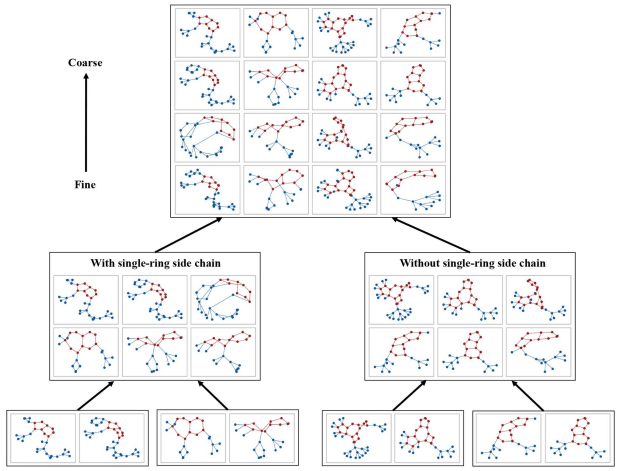


Fig. 5. Visualization of the hierarchical semantic structure. We select a subset of samples from the BZR dataset for visualization, where the direction of the arrow indicates the aggregation from fine-grained clustering at the lower layers to coarse-grained clustering at the higher layers.

- Compared with clustering with two layers, the model achieves the best performance in the settings ($10 \rightarrow 5$). This finding demonstrates that performance does not correlate linearly with prototype count. This may be due to the excessive number of clusters, which hinders the model's ability to effectively capture the distinction of lower-level semantics of the graph, leading to the misclassification of OOD data within the ID range. Meanwhile, an excessive number of prototypes may lead to a more complex objective optimization during hierarchical contrastive learning, which can hinder the model's convergence.

E. Visualization (RQ4)

1) Visualization of Hyperbolic Embeddings: To analyze how our HGOOD-D affects graph representations, we visualize the distribution of the learned features on PTC(MR)-MUTAG and AIDS-DHFR in hyperbolic space. Using the UMAP method with the distance metric set to "hyperboloid", we project the graph's hyperbolic features into 2D and map them onto the Poincaré disk for visualization. As illustrated in Fig. 4, the samples sharing the same distribution labels are grouped together, with each cluster being driven towards the circle's boundary.

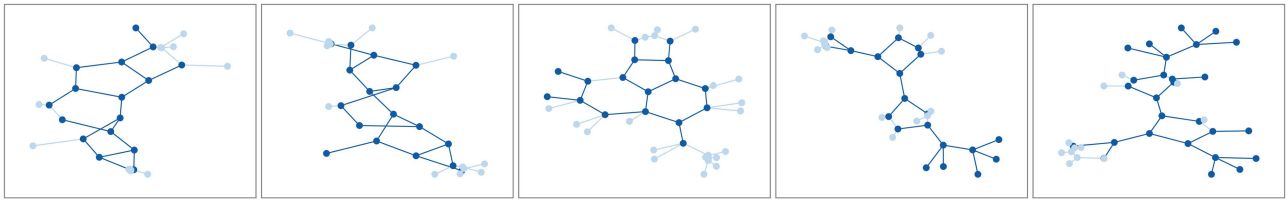


Fig. 6. Visualization of different subgraph extraction on BZR-COX2 dataset pair. The extracted ID subgraphs tend to focus on the backbone instead of side chains, where adjacent rings share common edges. In contrast, adjacent rings in OOD subgraphs are connected by additional edges.

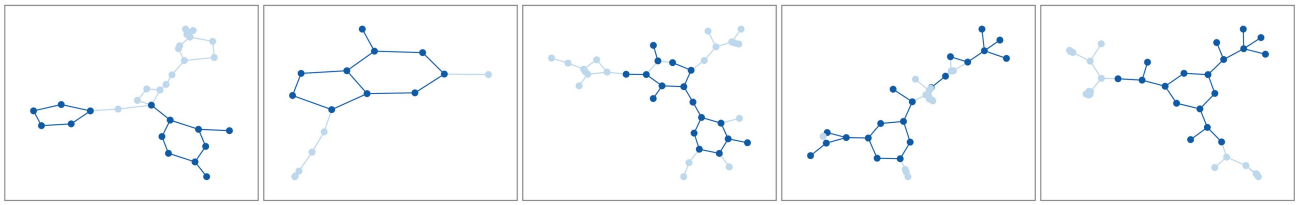


Fig. 7. Visualization of different subgraph extraction on BBBP-BACE dataset pair. The extracted ID subgraphs tend to focus on rings rather than chains, while highlighting the functional group with a hexagonal structure at the end of the backbone. In contrast, the hexagonal structure is embedded within the backbone in OOD subgraphs.

This observation indicates that our method effectively learns sufficiently separable feature representations. Meanwhile, samples within ID/OOD categories also exhibit clustering according to their original class labels, indicating that our method is able to learn a semantic relationship distribution that aligns with the inherent graph properties.

2) Visualization of Hierarchical Semantics: To further demonstrate the effectiveness of our HGOOD-D on the hierarchical semantics of graph representations, we present a visualization of the partitioned results of the hierarchical hyperbolic k-means on the BZR dataset. As shown in Fig. 5, molecules with different numbers of adjacent rings sharing common edges (marked in red) are clustered into a single group at the bottom layer. Meanwhile, molecules with two adjacent rings are divided into two clusters: one consisting of pentagon-hexagon combinations and the other of heptagon-hexagon combinations. Furthermore, the lower-layer clusters are grouped into two categories: molecules with single-ring side chains and those without. In the uppermost layer, these clusters are merged into a molecular graph class characterized by significant diversity. It is evident that molecules exhibit a progression from fine-grained to coarse-grained hierarchical semantics as they are iteratively clustered from lower to higher layers, e.g., molecules at the lowest level of the hierarchy demonstrate greater structural similarity, while those at the top reveal a broader diversity in structural composition. This indicates that our proposed HGOOD-D could effectively capture the hierarchical structure and semantic segmentation that aligns with the inherent nature of the data.

3) Visualization of Subgraph Extraction: Moreover, to validate the effectiveness of our HGOOD-D in extracting key subgraph structures, we selected the edges with weights greater than a set threshold in each graph and deepened the corresponding edges and nodes for visualization on BZR-COX2 and BBBP-BACE dataset pairs. As shown in Fig. 6, we observe that the extracted ID subgraphs focus more on the backbone structure, while highlighting adjacent rings that share common edges as key structures to distinguish them from OOD graphs. As shown in Fig. 7, the extracted ID subgraphs place more emphasis on the edge side chains, particularly those containing a functional group with a hexagonal structure, distinguishing them from OOD graphs that feature a hexagonal structure within the backbone. This indicates that our proposed HGOOD-D could learn more discriminative ID patterns of training data, which is beneficial for distinguishing between ID and OOD graphs for OOD detection task.

VI. Conclusion

In this paper, we propose a novel framework termed HGOOD-D for graph OOD detection, which aims at explicitly learning latent semantic hierarchies to capture the underlying ID graph distribution and distinguish OOD graphs. Specifically, we propose a subgraph extraction module to obtain critical subgraph structures while preserving minimal sufficient information. Based on the hyperbolic space, we introduce hierarchical contrastive learning to encode graph representations that preserve latent tree-like hierarchical semantic relationships, providing useful prior knowledge for graph OOD detection. We evaluate our model in comparison to various approaches on popular graph OOD detection datasets to demonstrate the efficacy of our HGOOD-D.

We hope our hyperbolic hierarchical exploration can offer new insights into GCL. Moving forward, we intend to generalize this architecture to broader domains, including drug discovery, knowledge graphs and recommendation systems, to further verify its real-world applicability.

REFERENCES

[1] Z. Hao et al., "ASGN: An active semi-supervised graph neural network for molecular property prediction," in *Proc. Int. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2020, pp. 731–752.

[2] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, pp. i47–i56, 2005.

[3] Y. Wang, S. Tang, Y. Lei, W. Song, S. Wang, and M. Zhang, "DisenHAN: Disentangled heterogeneous graph attention network for recommendation," in *Proc. Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1605–1614.

[4] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[5] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.

[6] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," in *Proc. Int. Conf. Learn. Representations*, 2019.

[7] X. Luo, Y. Zhao, Y. Qin, W. Ju, and M. Zhang, "Towards semi-supervised universal graph classification," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 1, pp. 416–428, Jan. 2024.

[8] Y. Wang, X. Luo, C. Chen, X.-S. Hua, M. Zhang, and W. Ju, "DisenSemi: Semi-supervised graph classification via disentangled representation learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 5, pp. 8192–8204, May 2025.

[9] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 5171–5181.

[10] C. Shi, M. Xu, Z. Zhu, W. Zhang, M. Zhang, and J. Tang, "GraphAF: A flow-based autoregressive model for molecular graph generation," in *Proc. Int. Conf. Learn. Representations*, 2020.

[11] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for HealthCare: Predicting pneumonia risk and hospital 30-day readmission," in *Proc. Int. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2015, pp. 1721–1730.

[12] K. Eykholt et al., "Robust physical-world attacks on deep learning visual classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1625–1634.

[13] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *Proc. Int. Conf. Learn. Representations*, 2018.

[14] V. Sehwag, M. Chiang, and P. Mittal, "SSD: A unified framework for self-supervised outlier detection," in *Proc. Int. Conf. Learn. Representations*, 2021.

[15] W. Zhou, F. Liu, and M. Chen, "Contrastive out-of-distribution detection for pretrained transformers," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 1100–1111.

[16] Z. Li, Q. Wu, F. Nie, and J. Yan, "GraphDE: A generative framework for debiased learning and out-of-distribution detection on graphs," in *Proc. Conf. Neural Inf. Process. Syst.*, 2022, pp. 30277–30290.

[17] Y. Liu, K. Ding, H. Liu, and S. Pan, "GOOD-D: On unsupervised graph out-of-distribution detection," in *Proc. Int. ACM Conf. Web Search Data Mining*, 2023, pp. 339–347.

[18] Y. Guo, C. Yang, Y. Chen, J. Liu, C. Shi, and J. Du, "A data-centric framework to endow graph neural networks with out-of-distribution detection ability," in *Proc. Int. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2023, pp. 638–648.

[19] L. Wang et al., "GOODAT: Towards test-time graph out-of-distribution detection," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 15537–15545.

[20] R.-C. Tzeng and S.-H. Wu, "Distributed, egocentric representations of graphs for detecting critical structures," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6354–6362.

[21] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6341–6350.

[22] H. Li, Z. Chen, Y. Xu, and J. Hu, "Hyperbolic anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 17511–17520.

[23] T. Long and N. van Noord, "Cross-modal scalable hyperbolic hierarchical clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 16655–16664.

[24] R. Wei, Y. Liu, J. Song, Y. Xie, and K. Zhou, "Exploring hierarchical information in hyperbolic space for self-supervised image hashing," *IEEE Trans. Image Process.*, vol. 33, pp. 1768–1781, 2024.

[25] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. IEEE Inf. Theory Workshop*, 2015, pp. 1–5.

[26] S. Suresh, P. Li, C. Hao, and J. Neville, "Adversarial graph augmentation to improve graph contrastive learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 15920–15933.

[27] S. Wang, C. Li, Y. Li, Y. Yuan, and G. Wang, "Self-supervised information bottleneck for deep multi-view subspace clustering," *IEEE Trans. Image Process.*, vol. 32, pp. 1555–1567, 2023.

[28] Z.-F. Wu, T. Wei, J. Jiang, C. Mao, M. Tang, and Y.-F. Li, "NGC: A unified framework for learning with open-world noisy data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 62–71.

[29] W. Zhou, F. Liu, and M. Chen, "Contrastive out-of-distribution detection for pretrained transformers," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 1100–1111.

[30] Y. Liu, K. Ding, Q. Lu, F. Li, L. Y. Zhang, and S. Pan, "Towards self-interpretable graph-level anomaly detection," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2023, pp. 8975–8987.

[31] H. Junwei, Q. Xu, Y. Jiang, Z. Wang, Y. Sun, and Q. Huang, "HGOE: Hybrid external and internal graph outlier exposure for graph out-of-distribution detection," in *Proc. ACM Int. Conf. Multimedia*, 2024, pp. 1544–1553.

[32] C. Hu, K.-Y. Zhang, T. Yao, S. Ding, and L. Ma, "Rethinking generalizable face anti-spoofing via hierarchical prototype-guided distribution refinement in hyperbolic space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 1032–1041.

[33] Y. He, M. Yuan, J. Chen, and I. Horrocks, "Language models as hierarchy encoders," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2025, pp. 14690–14711.

[34] A. Li, B. Yang, H. Huo, H. Chen, G. Xu, and Z. Wang, "Hyperbolic neural collaborative recommender," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 9114–9127, Sep. 2023.

[35] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3734–3743.

[36] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 4805–4815.

[37] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4438–4445.

[38] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[39] Y. Guo, X. Wang, Y. Chen, and S. X. Yu, "Clipped hyperbolic classifiers are super-hyperbolic classifiers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11–20.

[40] C. Gulcehre et al., "Hyperbolic attention networks," in *Proc. Int. Conf. Learn. Representations*, 2019.

[41] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky, "Hyperbolic image embeddings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6418–6428.

[42] A. A. Ungar, *Analytic Hyperbolic Geometry: Mathematical Foundations and Applications*. Singapore: World Scientific, 2005.

[43] M. Gromov, "Hyperbolic groups," in *Essays in Group Theory*. Berlin, Germany: Springer, 1987, pp. 75–263.

[44] S. Ge, S. Mishra, S. Kornblith, C.-L. Li, and D. Jacobs, "Hyperbolic contrastive learning for visual representations beyond objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 6840–6849.

[45] W. Hu et al., "Open graph benchmark: Datasets for machine learning on graphs," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 22118–22133.

[46] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "TUDataset: A collection of benchmark datasets for learning with graphs," in *Proc. Int. Conf. Mach. Learn. Workshop*, 2020.

[47] R. Ma, G. Pang, L. Chen, and A. van den Hengel, "Deep graph-level anomaly detection by glocal knowledge distillation," in *Proc. Int. ACM Conf. Web Search Data Mining*, 2022, pp. 704–714.

[48] M. Neumann, R. Garnett, C. Bauckhage, and K. Kersting, "Propagation kernels: Efficient graph Kernels from propagated information," *Mach. Learn.*, vol. 102, no. 2, pp. 209–245, 2016.

[49] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-Lehman graph kernels," *J. Mach. Learn. Res.*, vol. 12, no. 9, pp. 2539–2561, 2011.

[50] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, "Graph kernels," *J. Mach. Learn. Res.*, vol. 11, pp. 1201–1242, 2010.

[51] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. Int. Conf. Data Mining*, 2008, pp. 413–422.

[52] L. M. Manevitz and M. Yousef, "One-class SVMs for document classification," *J. Mach. Learn. Res.*, vol. 2, pp. 139–154, 2001.
[53] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. 2000 ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 93–104.
[54] F.-Y. Sun, J. Hoffman, V. Verma, and J. Tang, "InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *Proc. Int. Conf. Learn. Representations*, 2020.
[55] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 5812–5823.
[56] L. Zhao and L. Akoglu, "On using classification datasets to evaluate graph outlier detection: Peculiar observations and new insights," *Big Data*, vol. 11, pp. 151–180, 2023.

Yuntai Ding is currently working toward the PhD degree in software engineering with the Software College, Northeastern University, Shenyang, China. His research interests include graph representation learning and sequential recommendation.

Tao Ren received the BS, MS, and PhD degrees in engineering from Northeastern University, Shenyang, China, in 2003, 2005, and 2007, respectively. He held a post-doctoral position in computer science, from 2009 to 2013. He is currently a professor with Northeastern University. Recently, he is in charge of 20 projects, such as the National Natural Science Foundation of China. He has authored or coauthored more than 50 high-qualified academic papers in several high-ranking journals or conferences. Furthermore, he has published four books and authorized more than 20 Chinese patents. His main research interests include graph representation learning, machine learning and its applications.

Yiwei Fu received the BS degree in statistics from the School of Mathematical Sciences, Nankai University, in 2021. She is currently working toward the PhD degree in statistics with the School of Mathematical Sciences, Peking University, Beijing, China. Her research interests include graph representation learning, language models, and bioinformatics.

Yifan Wang received the BS and MS degrees in software engineering from Northeastern University, Liaoning, China, in 2014 and 2017, respectively, and the PhD degree in computer science from Peking University, Beijing, China, in 2023. He is currently an assistant professor with the School of Artificial Intelligence and Data Science, University of International Business and Economics. His research interests include graph representation learning, graph neural networks, disentangled representation learning, and corresponding applications such as drug discovery and recommender systems.

Haodong Zhang received the BS and MS degrees in software engineering from Northeastern University, Shenyang, China, in 2019 and 2021, respectively. He is currently working toward the PhD degree in software engineering with the Software College, Northeastern University, Shenyang, China. His research interests include graph representation learning, graph neural networks, and corresponding applications in drug discovery and out-of-distribution detection.

Chong Chen received the BS degree in mathematics from Peking University, in 2013, and the PhD degree in statistics from Peking University, in 2019, under the supervision of Prof. Ruibin Xi. He is currently a research scientist with Hesicare Technology Co., Ltd. His research interests include image understanding, self-supervised learning, and data mining.

Wei Ju received the BS degree in mathematics from Sichuan University, Sichuan, China, in 2017, and the PhD degree from the School of Computer Science, Peking University, Beijing, China, in 2022, where he is a postdoc research fellow. His current research interests lie primarily in the area of machine learning on graphs including graph representation learning and graph neural networks, and interdisciplinary applications such as recommender systems, bioinformatics, drug discovery, and spatio-temporal analysis. He has published more than 60 papers in top-tier venues and has won the best paper finalist in IEEE ICDM 2022.

Xiao Luo is an assistant professor with the Department of Statistics, University of Wisconsin–Madison, Madison, USA. His research interests include machine learning, AI for science, data mining, and bioinformatics. He has published more than 120 papers in refereed journals and conference proceedings such as NeurIPS, ICML, ICLR, CVPR, ICCV, ACL, NAACL, EMNLP, TheWebConf, ICDE, SIGKDD, the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Information Forensics and Security*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Image Processing*, and *Bioinformatics*. He is also an associate editor of the *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Xian-Sheng Hua (Fellow, IEEE) received the BS and PhD degrees in applied mathematics from Peking University, Beijing, in 1996 and 2001, respectively. In 2001, he joined Microsoft Research Asia, as a researcher, and has been a senior researcher with Microsoft Research Redmond since 2013. He became a researcher and the senior director of Alibaba Group, in 2015. He has authored or coauthored more than 250 research articles and has filed more than 90 patents. His research interests include multimedia search, advertising, understanding, and mining, pattern recognition, and machine learning. He was honored as one of the recipients of MIT35. He served as a Program co-chair for the IEEE ICME 2013, the ACM Multimedia 2012, and the IEEE ICME 2012, and on the Technical Directions Board for the IEEE Signal Processing Society. He is an ACM distinguished scientist.