DEEP GRAPH-LEVEL ORTHOGONAL HYPERSPHERE COMPRESSION FOR ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph-level anomaly detection aims to identify abnormal samples of a set of graphs in an unsupervised manner. It is non-trivial to find a reasonable decision boundary between normal data and anomalous data without using any anomalous data in the training stage, especially for data in graphs. This paper first proposes a novel deep graph-level anomaly detection model, which learns the graph representation with maximum mutual information between substructure features and global structure features while exploring a hypersphere anomaly decision boundary. We implement an orthogonal projection layer to keep the training data distribution consistent with the decision hypersphere thus avoiding erroneous evaluations. More importantly, we further propose projecting the normal data into the interval region between two co-centered hyperspheres, which makes the normal data distribution more compact and effectively overcomes the issue of outliers falling close to the center of the hypersphere. The numerical and visualization results on a few graph datasets demonstrate the effectiveness and superiority of our methods in comparison to many baselines and state-of-the-art.

1 INTRODUCTION

Anomaly detection is an essential task with various applications, such as detecting abnormal patterns or actions in credit-card fraud, medical diagnosis, sudden natural disasters (Aggarwal, 2017), etc. Usually, in anomaly detection, the training data only contain normal data and are used to train a model that can distinguish unusual patterns from abnormal ones. Anomaly detection on tabular data and images has been extensively studied recently. In contrast, there is little work on graph data despite the fact that graph data anomaly detection is very useful in various problems, such as identifying abnormal communities in social networks or detecting unusual protein structures in biology experiments. Compared with the other types of data, graph data is inherently complicated and rich in structural and relational information. The complexity of graph structure facilitates us to learn graph-level representations with discriminative patterns in many supervised tasks (e.g., graph classification). As for graph-level anomaly detection, however, the intricate graph structure brings many obstacles to this unsupervised learning problem.

Graph anomaly detection usually composes four families: anomalous edge (Ouyang et al., 2020; Xu et al., 2020), node (Zhu & Zhu, 2020; Bojchevski & Günnemann, 2018), sub-graph (Wang et al., 2018; Zheng et al., 2018), and graph-level detections (Zheng et al., 2019; Chalapathy et al., 2018). Herein, the target of the graph-level algorithms is to explore a regular group pattern and distinguish the abnormal manifestations of the group. Group abnormal behaviors usually foreshadow some unusual events and thus play an important role in practical applications. In the past five years, few approaches have focused on graph-level anomaly detection because of the difficulty of representing graphs into feature vectors without using any label information. Graph kernel can measure the similarity between graphs and regard the result as a representation non-strictly or implicitly. Based on this, graph anomaly detection task usually performs as two-stage. In our experiments (see Section 4), we also find that one-class SVM with graph kernels sometimes yields unsatisfying performances since graph kernels may not be effective enough to quantify the similarity between graphs. So there is a large room for improvement regarding graph anomaly detection to our best knowledge. Concerning end-to-end models, Ma et al. (2022) proposed a global and local knowledge distillation

method for graph-level anomaly detection, which learns rich global and local normal pattern information by random joint distillation of graph and node representations. The method needs to train two GCNs jointly at a high time cost. Zhao & Akoglu (2021) combined the Deep SVDD objective function and graph isomorphism network to learn a hypersphere of normal samples. Qiu et al. (2022) also sought a hypersphere decision boundary and optimized the representations learned by k GNNs close to the reference GNN while maximizing the differences between k GNNs, but did not consider the relationship between the graph-level representation and node features. Collecting all approaches based on the hypersphere assumption in graph anomaly detection, we find that the practical decision region may be an ellipsoid instead of a standard hypersphere, thus causing the error when the standard hypersphere evaluation is employed. Except for that, our experiment also confirms that anomalous data may appear in decision regions that are not filled with normal data, especially near the center of the hypersphere.

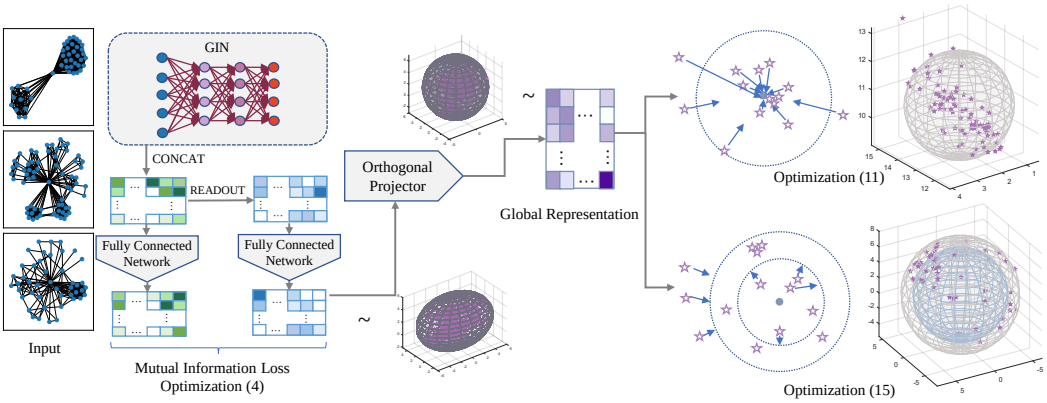


Figure 1: Architecture of the proposed models *(right top: DOHSC; right bottom: DO2HSC)*.

In order to effectively explore a better representation without label information and obtain a more suitable decision boundary with high efficiency, in this paper, we propose a one-class deep graph-level anomaly detection method and its improved version. The first proposed model, Deep Orthogonal Hypersphere Contraction (DOHSC), uses the mutual information of local feature maps and the global representation to learn a high-quality representation and simultaneously optimizes it to distribute in a hypersphere area. An orthogonal projection layer then renders the decision region more hyperspherical and compact to decrease evaluation errors. With regard to phenomenon that anomalous data falling close to the hyperspherical center, an improved graph-level Deep Orthogonal Bi-Hypersphere Compression (DO2HSC) for anomaly detection architecture is proposed. From a cross-sectional point of view, DO2HSC limits the decision area (of normal data) to an interval enclosed by two co-centered hyperspheres and learns the orthogonality-projected representation similarly. The framework of the methods mentioned above is shown in Figure 1 correspondingly. Furthermore, we define a new evaluation way according to DO2HSC, and comprehensive experimental results verify the effectiveness of all proposed methods. In summary, the main contributions of our work are listed as follows.

- First, we present a new graph-level hypersphere contraction algorithm for anomaly detection tasks, which is jointly trained via mutual information loss between local and global representations and hypersphere decision loss.
- Second, we impose an orthogonal projection layer on the proposed model to promote training data distribution close to the standard hypersphere, thus avoiding errors arising from inconsistencies between assessment criteria and actual conditions.
- Finally, we propose an improved graph-level deep orthogonal bi-hypersphere compression model to further explore a decision region enclosed by two co-centered hyperspheres, which can effectively prevent anomalous data falling close to the hyperspherical center and surpass baselines significantly in the experiments.

2 PROPOSED APPROACH

In this section, we first introduce a joint learning architecture in detail, named as Graph-Level Deep Orthogonal Hypersphere Contraction. Then an improved algorithm is proposed to compensate for the underlying assumption's deficiency.

2.1 GRAPH-LEVEL DEEP ORTHOGONAL HYPERSPHERE CONTRACTION

2.1.1 VALLINA MODEL

Given a set of graphs $\mathbb{G} = \{G_1, ..., G_N\}$ with N samples, the proposed model aims to learn a k-dimensional representation and then set a soft-boundary according to it. In this paper, the Graph Isomorphism Network (GIN) (Xu et al., 2019) is employed to obtain the graph representation in three stages: first, input the graph data and integrate neighbors of the current node (AGGREGATE); second, combine neighbor and current node features (CONCAT); finally, integrate all node information (READOUT) into one global representation. Mathematically, the i-th node features of l-th layer and the global features of its affiliated j-th graph would be denoted as

$$\begin{aligned} \mathbf{z}_{\Phi}^{i} &= \text{CONCAT}(\{\mathbf{z}_{i}^{(l)}\}_{l=1}^{L}), \\ \mathbf{Z}_{\Phi}(G_j) &= \text{READOUT}(\{\mathbf{z}_{\Phi}^{i}\}_{i=1}^{|G_j|}), \end{aligned} \tag{1}$$

where $\mathbf{z}_{\Phi}^{i} \in \mathbb{R}^{1\times k}$ and $\mathbf{Z}_{\Phi}(G_j) \in \mathbb{R}^{1\times k}$. To integrate the contained information and enhance the differentiation between node-level and global-level representations, we append additional fully connected layers denoted as the forms $M_{\Upsilon}(\cdot)$ and $T_{\Psi}(\cdot)$, respectively, where Υ and Ψ are the parameters of the added layers. So the integrated node-level and graph-level representations are obtained via

$$\begin{aligned} \mathbf{h}_{\Phi,\Upsilon}^{i} &:= M_{\Upsilon}(\mathbf{z}_{\Phi}^{i}), \\ \mathbf{H}_{\Phi,\Psi}(G_j) &:= T_{\Psi}(\mathbf{Z}_{\Phi}(G_j)), \end{aligned} \tag{2}$$

To better capture the local information, we utilize the batch optimization property of neural networks to maximize the mutual information (MI) between local and global representations in each batch $\mathbf{G} \subseteq \mathbb{G}$, which is defined by Sun et al. (2020) as the following term:

$$\hat{\Phi}, \hat{\Psi}, \hat{\Upsilon} = \underset{\Phi,\Psi,\Upsilon}{\arg\max}\, I_{\Phi,\Psi,\Upsilon}\left(\mathbf{h}_{\Phi,\Upsilon}, \mathbf{H}_{\Phi,\Psi}(\mathbf{G})\right). \tag{3}$$

Specifically, the mutual information estimator $I_{\Phi,\Psi,\Upsilon}$ follows Jensen-Shannon MI estimator (Nowozin et al., 2016) with a positive-negative sampling method as below,

$$\begin{aligned} I_{\Phi,\Psi,\Upsilon}\left(\mathbf{h}_{\Phi,\Upsilon}, \mathbf{H}_{\Phi,\Psi}(\mathbf{G})\right) :&= \sum_{G_j\in\mathbf{G}} \frac{1}{|G_j|} \sum_{u\in G_j} I_{\Phi,\Psi,\Upsilon}\left(\mathbf{h}_{\Phi,\Upsilon}^{u}(G_j), \mathbf{H}_{\Phi,\Psi}(\mathbf{G})\right) \\ &= \sum_{G_j\in\mathbf{G}} \frac{1}{|G_j|} \sum_{u\in G_j} \Big[\mathbb{E}\big(-\sigma\left(-\mathbf{h}_{\Phi,\Upsilon}^{u}(\mathbf{x}^{+}) \times \mathbf{H}_{\Phi,\Psi}(\mathbf{x})\right)\big) \\ &\qquad - \mathbb{E}\big(\sigma\left(\mathbf{h}_{\Phi,\Upsilon}^{u}(\mathbf{x}^{-}) \times \mathbf{H}_{\Phi,\Psi}(\mathbf{x})\right)\big)\Big], \end{aligned} \tag{4}$$

where a softplus function $\sigma(z) = \log(1 + e^{z})$ is activated after vector multiplication between node and graph representations. For $\mathbf{x}$ as an input sample graph, we calculate the expected mutual information with its positive samples $\mathbf{x}^{+}$ and negative samples $\mathbf{x}^{-}$, which are generated from distribution across all graphs in a subset. Given each $G = (\mathcal{V}_G, \mathcal{E}_G)$ and node set $\mathcal{V}_G = \{v_i\}_{i=1}^{|G|}$, the positive and negative samples are divided in this way:

$$\mathbf{x}^{+} = \begin{cases} \mathbf{x}_{ij}, & \text{if } v_i \in G_j, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

And $\mathbf{x}^{-}$ produces the opposite result in each of the conditions above. In the next step, a data-enclosing decision boundary is required for our anomaly detection task. According to the assumption that most normal data can locate in a hypersphere, the center of this decision boundary should

be initialized through

$$\mathbf{c} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{H}_{\Phi,\Psi}(G_i). \tag{6}$$

With this center, we expect to optimize the learned representation of normal data to be distributed as close to it as possible, so that the unexpected anomalous data falling out of this hypersphere would be detected. Besides, the regularization term is adopted to avoid over-fitting problems. Collectively denote the weight parameters of Φ, Ψ and Υ as $\mathcal{Q} := \Phi \cup \Psi \cup \Upsilon$, we formulate the training loss with two joint objectives – Hypersphere Contraction and MI:

$$\min_{\Phi,\Psi,\Upsilon} \frac{1}{|\mathbf{G}|}\sum_{i=1}^{|\mathbf{G}|}\|\mathbf{H}_{\Phi,\Psi}(G_i)-\mathbf{c}\|^2 + \lambda I_{\Phi,\Psi,\Upsilon}\left(\mathbf{h}_{\Phi,\Upsilon}, \mathbf{H}_{\Phi,\Psi}(\mathbf{G})\right) + \frac{\mu}{2}\sum_{\mathbf{Q}\in\mathcal{Q}}\|\mathbf{Q}\|_F^2, \tag{7}$$

where $|\mathbf{G}|$ denotes the number of graphs in batch $\mathbf{G}$ and λ is a trade-off factor, the third term is a network weight decay regularizer with the hyperparameter μ.

2.1.2 Orthogonal Projection Layer

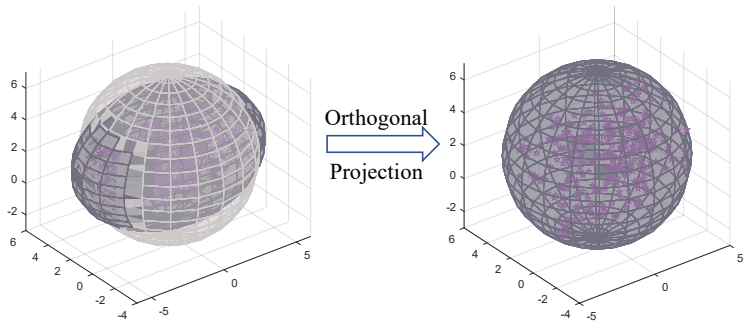


Figure 2: Variations in decision boundary with and without the orthogonal projection layer. The ellipsoid in dark grey denotes the real decision region that training data distribution and the hypersphere in light grey is the decision boundary set by Optimization 7. After orthogonal projection, the ellipsoid is expected to be transformed into a standard hypersphere, which avoids the evaluation error when practical data do not fit the decision boundary.

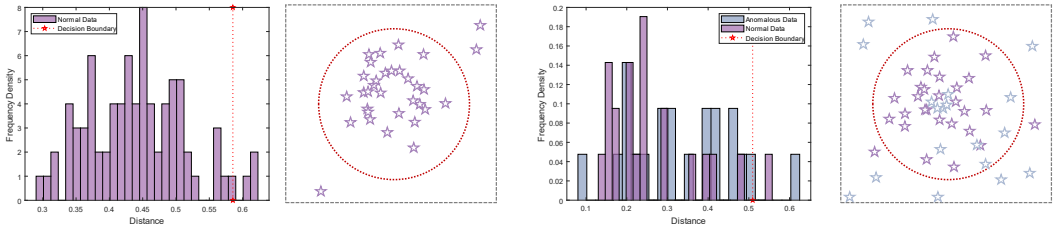
However, an empirical study shows that a hyperellipsoid is commonly observed during deep representation learning. This phenomenon would lead to inaccuracies in the final test because the evaluation results are based on the hypersphere decision region. Problem (7) obviously cannot guarantee the soft-boundary of learned representation to be a standard hypersphere like Figure 2. (Note that this figure is not drawing based on the real data.) Accordingly, we append an orthogonal projection layer after obtaining the global representation. Our experiments also justify the necessity of this projection step, which will be discussed further in Section 4.4. Specifically, the projection layer can be formulated as

$$\tilde{\mathbf{H}}_{\Phi,\Psi,\Theta}(G) = \text{Proj}_{\Theta}(\mathbf{H}_{\Phi,\Psi}(G)), \;\; \text{subject to} \;\; \tilde{\mathbf{H}}_{\Phi,\Psi,\Theta}^{\top}\tilde{\mathbf{H}}_{\Phi,\Psi,\Theta} = \mathbf{I}_{k'} \tag{8}$$

where $\Theta := \{\mathbf{W} \in \mathbb{R}^{k\times k'}\}$ are the projection parameters, $\mathbf{I}_{k'}$ denotes an identity matrix, and k' is the projected dimension.

Note that to achieve (8), one may consider adding a regularization term $\frac{\alpha}{2}\|\tilde{\mathbf{H}}_{\Phi,\Psi,\Theta}^{\top}\tilde{\mathbf{H}}_{\Phi,\Psi,\Theta} - \mathbf{I}_{k'}\|_F^2$ with large enough α to the objective, which is not very effective and will lead to one more tuning hyperparameter. Instead, we propose to achieve (8) via singular value decomposition, i.e.,

$$\begin{aligned} \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\top} &= \mathbf{H}_{\Phi,\Psi}, \\ \mathbf{W} &:= \mathbf{V}_{k'}\mathbf{\Lambda}_{k'}^{-1}, \end{aligned} \tag{9}$$



(a) Training Data (b) Testing Data

Figure 3: Illustration of inevitable flaws in DOHSC on both the training and testing data of COX2. Left: the ℓ_2-norm distribution of 4-dimensional distances learned from the real dataset; Right: the pseudo-layout in two-dimensional space sketched by reference to the empirical distribution.

where $\mathbf{\Lambda} = \text{diag}(\rho_1, \rho_2, ..., \rho_{|\mathbf{G}|})$ and $\mathbf{V}$ are the diagonal matrix with sigular values and right-singular matrix of $\mathbf{H}_{\Phi,\Psi}$, respectively. It needs to be emphasized that $\mathbf{V}_{k'} := [\mathbf{v}_1, ..., \mathbf{v}_{k'}]$ denotes the first k' right-singular vectors and $\mathbf{\Lambda}_{k'} := \text{diag}(\rho_1, ..., \rho_{k'})$. In each forward propagation epoch, the original weight parameter is substituted to a new matrix $\mathbf{W}$ in subsequent loss computations.

2.1.3 Anomaly Detection

Attaching with an orthogonal projection layer, the improved initialization of the center is rewritten in the following form

$$\tilde{\mathbf{c}} = \frac{1}{N} \sum_{i=1}^{N} \tilde{\mathbf{H}}_{\Phi,\Psi,\Theta}(G_i) \tag{10}$$

and the final objective function for anomaly dectection tasks in a mini-batch would become

$$\min_{\Theta,\Phi,\Psi,\Upsilon} \frac{1}{|\mathbf{G}|} \sum_{i=1}^{|\mathbf{G}|} \|\tilde{\mathbf{H}}_{\Phi,\Psi,\Theta}(G_i) - \tilde{\mathbf{c}}\|^2 + \lambda \sum_{\mathbf{G}\in\mathbb{G}} I_{\Phi,\Psi,\Upsilon}\left(\mathbf{h}_{\Phi,\Upsilon}, \tilde{\mathbf{H}}_{\Phi,\Psi,\Theta}(\mathbf{G})\right) + \frac{\mu}{2} \sum_{\mathbf{Q}\in\mathcal{Q}} \|\mathbf{Q}\|_F^2. \tag{11}$$

After the training stage, a decision boundary $\hat{r}$ will be fixed, which is calculated based on the $1-\nu$ percentile of the training data distance distribution:

$$\hat{r} = \arg\min_{r} \mathcal{P}(\mathbf{D} \leq r) \geq \nu \tag{12}$$

where $\mathbf{D} := \{d_i\}_{i=1}^{N}$ follows a sampled distribution $\mathcal{P}$, and $d_i = \|\tilde{\mathbf{H}}_{\Phi,\Phi,\Theta}(G_i) - \tilde{\mathbf{c}}\|$. Accordingly, the anomalous score of i-th instance is defined as follows:

$$s_i = d_i^2 - \hat{r}^2 \tag{13}$$

where $\mathbf{s} = (s_1, s_2, \ldots, s_N)$. It is evident that when the score is positive, the instance is identified as abnormal, and the opposite is considered normal.

The detailed procedures of algorithm are summarized into Algorithm 1 (see Appendix B). It starts with graph representation learning and promotes the training data to approximate the center of a hypersphere while adding an orthogonal projection layer. Unfortunately, it can be observed from Figure 3 that the anomalous data would appear in partial regions of the learned decision hypersphere, which are not filled by the training data, especially the region close to the center. To handle this particular situation, an improved graph-level anomaly detection approach, which terms as Graph-Level Deep Orthogonal Bi-Hypersphere Compression, is designed in the next section.

2.2 Graph-Level Deep Orthogonal Bi-Hypersphere Compression

According to our empirical study in Figure 3, we found that when obtained training data distributions cannot perfectly satisfy the assumed hypersphere, anomalous data may appear in the empty inner decision region and lead to the wrong detection results, especially in regions near the center. Hence, we propose an improved approach, which only sets the decision boundary as the interval region

between two co-centered hyperspheres. This refinement can narrow the decision area's scope so as to induce normal data to fill the entire interval area as much as possible.

After the same graph representation learning stage, we firstly utilize the DOHSC model for a few epochs and initialize the large radius r_{max} and the small radius r_{min} of the interval area according to the $1-\nu$ percentile and ν of the sample distances distribution, respectively. The aforementioned descriptions can be denoted mathematically as below.

$$\begin{aligned} r_{\text{max}} &= \arg\min_{r} \mathcal{P}(\mathbf{D} \leq r) \geq \nu, \\ r_{\text{min}} &= \arg\min_{r} \mathcal{P}(\mathbf{D} \leq r) \geq 1-\nu. \end{aligned} \tag{14}$$

After fixing the decision boundaries r_{max} and r_{min}, the improved training loss is also set with a trade-off factor λ, which implicitly emphasizes the importance of the max-min term:

$$\begin{aligned} \min_{\Theta,\Phi,\Psi,\Upsilon} \frac{1}{|\mathbf{G}|} \sum_{i=1}^{|\mathbf{G}|} (\max\{d_i, r_{max}\} - \min\{d_i, r_{min}\}) + \lambda \sum_{\mathbf{G}\in\mathbb{G}} I_{\Phi,\Psi,\Upsilon}\left(\mathbf{h}_{\Phi,\Upsilon}, \tilde{\mathbf{H}}_{\Phi,\Psi,\Theta}(\mathbf{G})\right) \\ + \frac{\mu}{2} \sum_{\mathbf{Q}\in\mathcal{Q}} \|\mathbf{Q}\|_F^2. \end{aligned} \tag{15}$$

This decision loss has the lowest bound $r_{max} - r_{min}$ and can be jointly minimized with mutual information effectively. Besides, the evaluation standard of test data is also needed to change based on this interval structure. More specifically, all instances located in the inner hypersphere and out of the outer hypersphere should be identified as anomalous graphs; only those located in the interval area should be regarded as normal data. Compared with equation 13, we reset a new score function to award the positive samples beyond $[r_{min}, r_{max}]$ and meanwhile punishing the negative samples within the range. Accordingly, the distinctive scores are calculated by

$$s_i = (d_i - r_{max}) \cdot (d_i - r_{min}), \tag{16}$$

where $i \in \{1, ..., N\}$. This way, we can also effectively identify a sample's abnormality by its score. In general, the improved deep graph-level anomaly detection algorithm changes the decision boundary and effectively makes the normal area more compact. Correspondingly, the new practical evaluation is raised to adapt to the improved detection way. Eventually, we summarize the detailed procedures of the optimization into Algorithm 2 (see Appendix B).

3 Connection with previous work

Actually, few studies have been undertaken in graph-level anomaly detection (GAD). Existing solutions to GAD tasks can be categorized into two types: a two-stage approach and an end-to-end one. Two-stage GAD methods first transform graphs into graph embeddings by graph neural networks or into similarities between graphs by graph kernels, and then apply off-the-shelf anomaly detectors such as local outlier factor (LOF) (Breunig et al., 2000), one-class support vector machine (OCSVM) (Schölkopf et al., 1999), etc. The drawback of the two-stage method is that the graph feature extractor and outlier detector are independent, and some graph kernels produce "hand-crafted" features that are deterministic without much space to improve. End-to-end approaches overcome this problem by utilizing deep graph learning techniques, such as graph convolutional network (GCN) (Welling & Kipf, 2016) and graph isomorphism network (GIN) (Xu et al., 2019). With an anomaly measure as the objective, end-to-end approaches jointly learn an effective graph representation for GAD task (Zhao & Akoglu, 2021; Qiu et al., 2022; Ma et al., 2022). Zhao & Akoglu (2021) optimized one-class model based on deep support vector data description (Deep SVDD) as the anomaly measure. Here we clarify the differences between (Zhao & Akoglu, 2021) and ours. First, the proposed model employed mutual information loss in the graph learning stage to obtain the graph representation incorporating local and global information. On the contrary, Zhao & Akoglu (2021) directly utilized the readout result of GIN. We also impose an orthogonal projection on the learned representation to maintain consistency between the learned decision boundary and normal data distribution. More importantly, we present a new approach to constructing decision boundary. The new approach learns two hyperspheres, between which the region accommodates normal data and hence yields more space for abnormal data. As for graph kernel, we summarized the previous work in Appendix A.

4 EXPERIMENTS

4.1 DATASET

In this work, we test our method on six real-world graph datasets[1], which contain three social networks datasets (COLLAB, COX2, and IMDB-Binary) and three bioinformatics datasets (DD, ER_MD, and MUTAG). The details of the datasets are shown in Table 1.

Table 1: Description for six datasets.

Datasets	# Graphs	Avg. # Nodes	Avg. # Edges	# Classes	# Graph Labels
COLLAB	5000	74.49	2457.78	3	2600 / 775 / 1625
COX2	467	42.43	44.54	2	365 / 102
ER_MD	446	21.33	234.85	2	265 / 181
MUTAG	188	17.93	19.79	2	63 / 125
DD	1178	284.32	715.66	2	691 / 487
IMDB-Binary	1000	19.77	96.53	2	500 / 500

4.2 BASELINES

We compare our method with the following unsupervised graph-level anomaly detection methods: Random Walk (RW) (Gärtner et al., 2003; Kashima et al., 2003), Shortest Path Kernel (SP) (Borgwardt & Kriegel, 2005), Weisfeiler-Lehman Sub-tree Kernel (WL) (Shervashidze et al., 2011), Neighborhood Hash Kernel (NH) (Hido & Kashima, 2009). Besides graph kernels, we also compare three graph-level representation learning methods: Deep One Class Model with GIN network (OCGIN) (Zhao & Akoglu, 2021), Graph-level embedding Learning via Mutual Information Maximization+Deep SVDD (infoGraph+Deep SVDD) (Sun et al., 2020; Ruff et al., 2018), Global and Local Knowledge Distillation for Graph-level Anomaly Detection (GLocalKD) (Ma et al., 2022) and One Class Graph Transformation Learning (OCGTL) (Qiu et al., 2022).

4.3 RESULTS

In this part, extensive experimental results are displayed to validate the effectiveness of the proposed models. Here, the averages and standard deviations of the Area Under Operating Characteristic Curve (AUC) are used to support the comparable experiments by repeating each algorithm ten times. The higher value of the AUC metric represents better performance. Tables 2–4 report the AUC metric and its standard deviations. It can be seen that the proposed methods basically achieve the best AUC values compared to other algorithms on all datasets. Both approaches outperform other state-of-the-art baselines, and DO2HSC obtains superior performance and get 5% higher performance than other algorithms on many datasets, such as MUTAG, COLLAB Class 1, and ER_MD Class 0, IMDB-Binary Class 1, etc. It is woth mentioning that we defeat infoGraph+Deep SVDD with a large improvement, which is a degraded version of the proposed models, thus showing that promoting representation learning towards the anomaly detection goal is very meaningful and well targeted.

The anomaly detection visualization results of DO2HSC are displayed in Figure 4 and those of DOHSC are also shown in Appendix D. We draw them by setting the projection dimension to three directly. Results of different perspectives are given to avoid blind spots in the field of vision, demonstrating excellent performance. Hence, it can be concluded that the effect of the improved model is in line with our motivation and shows much potential.

4.4 ABLATION STUDY

In this section, we display the ablation study of the orthogonal projection layer on three datasets. From quantitive comparisons, we concluded that orthogonality positively influences all performance to some extent. It also well supports our assumption discussed in Section 2.1.2.

[1]https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets

Table 2: Average AUCs with standard deviation (10 trials) of different graph-level anomaly detection algorithms. We assess models by regarding every data class as normal data, respectively. The best results in are marked in **bold** and '–' means out of memory.

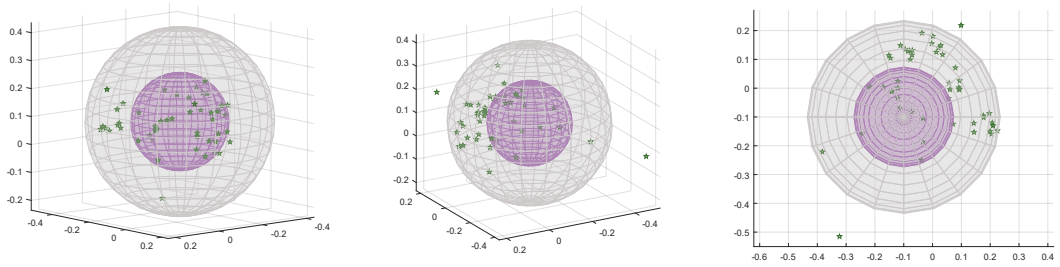
	COLLAB			COX2	
	0	1	2	0	1
SP+OCSVM	0.5910 ± 0.0000	0.8397 ± 0.0000	0.7902 ± 0.0000	0.5408 ± 0.0000	0.5760 ± 0.0000
WL_2+OCSVM	0.5051 ± 0.0000	0.7989 ± 0.0000	0.6977 ± 0.0000	0.5736 ± 0.0000	0.4286 ± 0.0000
WL_5+OCSVM	0.5079 ± 0.0000	0.8021 ± 0.0000	0.7884 ± 0.0000	0.5990 ± 0.0000	0.4376 ± 0.0000
WL_8+OCSVM	0.5106 ± 0.0000	0.8035 ± 0.0000	0.7953 ± 0.0000	0.5979 ± 0.0000	0.5057 ± 0.0000
WL_{10}+OCSVM	0.5122 ± 0.0000	0.8031 ± 0.0000	**0.7996 ± 0.0000**	0.5937 ± 0.0000	0.5034 ± 0.0000
NH+OCSVM	0.5976 ± 0.0000	0.8054 ± 0.0000	0.6414 ± 0.0000	0.4841 ± 0.0000	0.4717 ± 0.0000
RW+OCSVM	–	–	–	0.5243 ± 0.0000	0.6553 ± 0.0000
OCGIN	0.4217 ± 0.0606	0.7565 ± 0.2035	0.1906 ± 0.0857	0.5964 ± 0.0578	0.5683 ± 0.0768
infoGraph+Deep SVDD	0.5662 ± 0.0597	0.7926 ± 0.0986	0.4062 ± 0.0978	0.4825 ± 0.0624	0.5029 ± 0.0700
GLocalKD	0.4638 ± 0.0003	0.4330 ± 0.0016	0.4792 ± 0.0004	0.3861 ± 0.0131	0.3143 ± 0.0383
OCGTL	0.6504 ± 0.0433	0.8908 ± 0.0239	0.4029 ± 0.0541	0.5541 ± 0.032	0.4862 ± 0.0224
DOHSC (Ours)	0.5338 ± 0.0337	0.8694 ± 0.0285	0.5450 ± 0.0469	0.6263 ± 0.0333	**0.6805 ± 0.0168**
DO2HSC (Ours)	**0.6718 ± 0.0353**	**0.9153 ± 0.0070**	0.7188 ± 0.0260	**0.6329 ± 0.0292**	0.6518 ± 0.0481

Table 3: Average AUCs with standard deviation (10 trials) of different graph-level anomaly detection algorithms. We assess models by regarding every data class as normal data, respectively. The best results are marked in **bold**.

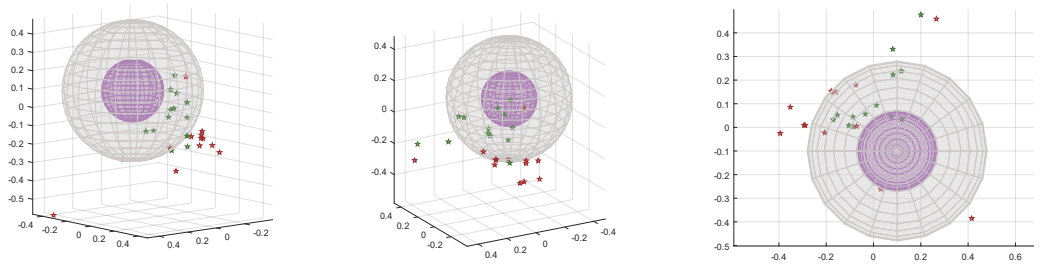
	ER_MD		MUTAG	
	0	1	0	1
SP+OCSVM	0.4092 ± 0.0000	0.3824 ± 0.0000	0.5917 ± 0.0000	0.2608 ± 0.0000
WL_2+OCSVM	0.3702 ± 0.0000	0.3262 ± 0.0000	0.5976 ± 0.0000	0.2960 ± 0.0000
WL_5+OCSVM	0.4446 ± 0.0000	0.2933 ± 0.0000	0.6509 ± 0.0000	0.2352 ± 0.0000
WL_8+OCSVM	0.4571 ± 0.0000	0.3199 ± 0.0000	0.4556 ± 0.0000	0.2176 ± 0.0000
WL_{10}+OCSVM	0.4297 ± 0.0000	0.2761 ± 0.0000	0.5089 ± 0.0000	0.2048 ± 0.0000
NH+OCSVM	0.5155 ± 0.0200	0.3648 ± 0.0000	0.7959 ± 0.0274	0.1679 ± 0.0062
RW+OCSVM	0.4820 ± 0.0000	0.3484 ± 0.0000	0.8698 ± 0.0000	0.1504 ± 0.0000
OCGIN	0.5645 ± 0.0323	0.4358 ± 0.0538	0.8491 ± 0.0424	0.4933 ± 0.1525
infoGraph+Deep SVDD	0.5312 ± 0.1545	0.5682 ± 0.0704	0.8805 ± 0.0448	0.6166 ± 0.2052
GLocalKD	0.5781 ± 0.1790	0.7154 ± 0.0000	0.3952 ± 0.2258	0.2965 ± 0.2641
OCGTL	0.2755 ± 0.0317	0.6915 ± 0.0207	0.6570 ± 0.0210	0.7579 ± 0.2212
DOHSC (Ours)	0.6620 ± 0.0308	0.5184 ± 0.0793	0.8822 ± 0.0432	0.8115 ± 0.0279
DO2HSC (Ours)	**0.6867 ± 0.0226**	**0.7351 ± 0.0159**	**0.9089 ± 0.0609**	**0.8250 ± 0.0790**

Table 4: Average AUCs with standard deviation (10 trials) of different graph-level anomaly detection algorithms. We assess models by regarding every data class as normal data, respectively. The best results are marked in **bold** and '–' means out of memory.

	DD		IMDB-Binary	
	0	1	0	1
SP+OCSVM	0.6856 ± 0.0000	0.4474 ± 0.0000	0.4592± 0.0000	0.4716 ± 0.0000
WL_2+OCSVM	0.6225 ± 0.0000	0.4946 ± 0.0000	0.4294 ± 0.0000	0.4607 ± 0.0000
WL_5+OCSVM	0.7127 ± 0.0000	0.4666 ± 0.0000	0.4940 ± 0.0000	0.4243 ± 0.0000
WL_8+OCSVM	0.7343 ± 0.0000	0.4455 ± 0.0000	0.5066 ± 0.0000	0.4143 ± 0.0000
WL_{10}+OCSVM	0.7397 ± 0.0000	0.4302 ± 0.0000	0.5157 ± 0.0000	0.4115 ± 0.0000
NH+OCSVM	**0.7424 ± 0.0000**	0.3684 ± 0.0000	0.5321 ± 0.0000	0.4652 ± 0.0000
RW+OCSVM	–	–	0.4951 ± 0.0000	0.5311 ± 0.0000
OCGIN	0.6659 ± 0.0444	0.6003 ± 0.0534	0.4047 ± 0.1083	0.3332 ± 0.0649
infoGraph+Deep SVDD	0.3942 ± 0.0436	0.6484 ± 0.0236	0.6353 ± 0.0277	0.5836 ± 0.0995
GLocalKD	0.1952 ± 0.0000	0.2203 ± 0.0001	0.5383 ± 0.0124	0.4812 ± 0.0101
OCGTL	0.6990 ± 0.0260	0.6767 ± 0.0280	0.6510 ± 0.0180	0.6412 ± 0.0127
DOHSC (Ours)	0.7083 ± 0.0188	0.7579 ± 0.0154	**0.6609 ± 0.0033**	**0.7705 ± 0.0045**
DO2HSC (Ours)	0.7320 ± 0.0194	**0.7651 ± 0.0317**	0.6406 ± 0.0642	0.7101 ± 0.0429



(a) Training Result



(b) Testing Result

Figure 4: Visualization results of the DO2HSC on MUTAG Class 0 in different perspectives.

Table 5: Ablation study of the orthogonal projection layer. We test models by regarding every data class as normal data, respectively. The best performance is highlighted in **bold**.

	Orthogonal Projection Layer	Class	MUTAG	COX2	IMDB-Binary
DOHSC	×	0	0.8521	0.5817	0.5880
		1	0.8912	0.5737	0.6082
	✓	0	**0.8639**	**0.6433**	**0.6192**
		1	**0.9088**	**0.6077**	**0.6830**
DO2HSC	×	0	0.8934	0.6281	0.6252
		1	0.7504	0.6178	0.6541
	✓	0	**0.9467**	**0.7158**	**0.6933**
		1	**0.9184**	**0.7074**	**0.6700**

Except for the aforementioned contents, please see the detailed experiment configurations, parameter sensitivity and robustness of the proposed models, supplemented visualizations of distance distributions for anomaly detection and visualization comparison between proposed models with orthogonal projection layer and without orthogonal projection layer in Appendix C, which can also strongly support our theory and validate the effectiveness.

5 Conclusion

This paper has proposed two novel end-to-end graph-level AD methods, DOHSC and DO2HSC, which combined the effectiveness of mutual information between node-level and global features to learn graph representation and the power of hypersphere compression. DOHSC and DO2HSC mitigate the possible shortcomings of hypersphere boundary learning by applying an orthogonal projection for global representation. Furthermore, DO2HSC projects normal data between the interval area of two co-centered hyperspheres to significantly alleviate the "performance flip" issue. Our experiments' comprehensive results strongly demonstrate the superiority of DOHSC and DO2HSC on multifarious datasets. In future work, we will explore more efficient and excellent graph representations and refine our learning process for decision regions.

REFERENCES

Charu C Aggarwal. *An introduction to outlier analysis*. 2017.

Aleksandar Bojchevski and Stephan Günnemann. Bayesian robust attributed graph clustering: Joint learning of partial anomalies and group structure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Karsten M Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pp. 8–pp, 2005.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: Identifying density-based local outliers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 93–104, 2000.

Raghavendra Chalapathy, Edward Toth, and Sanjay Chawla. Group anomaly detection using deep generative models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 173–189, 2018.

Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*, pp. 129–143. 2003.

Shohei Hido and Hisashi Kashima. A linear-time graph kernel. In *Ninth IEEE International Conference on Data Mining*, pp. 179–188, 2009.

Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the 20th International Conference on Machine Learning*, pp. 321–328, 2003.

Nils M Kriege, Fredrik D Johansson, and Christopher Morris. A survey on graph kernels. *Applied Network Science*, 5(1):1–42, 2020.

Rongrong Ma, Guansong Pang, Ling Chen, and Anton van den Hengel. Deep graph-level anomaly detection by glocal knowledge distillation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 704–714, 2022.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Linshu Ouyang, Yongzheng Zhang, and Yipeng Wang. Unified graph embedding-based anomalous edge detection. In *International Joint Conference on Neural Networks*, pp. 1–8, 2020.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Chen Qiu, Marius Kloft, Stephan Mandt, and Maja Rudolph. Raising the bar in graph-level anomaly detection. In Luc De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 2196–2203, 2022.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International Conference on Machine Learning*, pp. 4393–4402, 2018.

Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. *Advances in Neural Information Processing Systems*, 12, 1999.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

Giannis Siglidis, Giannis Nikolentzos, Stratis Limnios, Christos Giatsidis, Konstantinos Skianis, and Michalis Vazirgiannis. Grakel: A graph kernel library in Python. *Journal of Machine Learning Research*, 21(54):1–5, 2020.

Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

Haibo Wang, Chuan Zhou, Jia Wu, Weizhen Dang, Xingquan Zhu, and Jilong Wang. Deep structure learning for fraud detection. In *IEEE International Conference on Data Mining*, pp. 567–576, 2018.

Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2016.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S Yu. Icane: Interaction content-aware network embedding via co-embedding of nodes and edges. *International Journal of Data Science and Analytics*, 9(4):401–414, 2020.

Lingxiao Zhao and Leman Akoglu. On using classification datasets to evaluate graph outlier detection: Peculiar observations and new insights. *Big Data*, 2021.

Mengyu Zheng, Chuan Zhou, Jia Wu, Shirui Pan, Jinqiao Shi, and Li Guo. Fraudne: A joint embedding approach for fraud detection. In *International Joint Conference on Neural Networks*, pp. 1–8, 2018.

Panpan Zheng, Shuhan Yuan, Xintao Wu, Jun Li, and Aidong Lu. One-class adversarial nets for fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1286–1293, 2019.

Mengxiao Zhu and Haogang Zhu. Mixedad: A scalable algorithm for detecting mixed anomalies in attributed graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 1274–1281, 2020.

A Related Work on Graph Kernel

To learn with graph-structured data, graph kernels that measure the similarity between graphs become an established and widely-used approach (Kriege et al., 2020). A large body of work has emerged in the past years, including kernels based on neighborhood aggregation techniques and walks and paths. (Shervashidze et al., 2011) introduced Weisfeiler-Lehman (WL) algorithm, a well-known heuristic for graph isomorphism. In (Hido & Kashima, 2009), Neighborhood Hash kernel was introduced, where the neighborhood aggregation function is binary arithmetic. The most influential graph kernel for paths-based kernels is the shortest-path (SP) kernel by (Borgwardt & Kriegel, 2005). For walks-based kernels, Gärtner et al. (2003) and Kashima et al. (2003) simultaneously proposed graph kernels based on random walks, which count the number of label sequences along walks that two graphs have in common. These graph kernel methods have the desirable property that they do not rely on the vector representation of data explicitly but access data only via the Gram matrix.

B Supplemented Algorithm

Algorithm 1 summarizes the procedure of DOHSC in detail. It starts with graph representation learning and promotes the training data to approximate the center of a hypersphere while adding an orthogonal projection layer. Also, DO2HSC is recapped in Algorithm 2 and also starts with same graph representation learning. Differently, DOHSC is utilized of few epochs to initial the decision boundaries and after that, improved optimization is applied.

Algorithm 1 Graph-Level Deep Orthogonal Hypersphere Contraction (DOHSC)

Input: The input graph set $\mathbb{G}$, dimensions of GIN hidden layers k and orthogonal projection layer k', a trade-off parameter λ and the coefficient of regularization term μ, pretraining epoch $\mathcal{T}$, learning rate η.

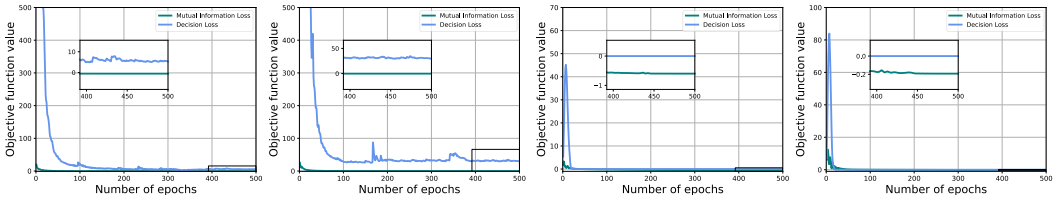
Output: The anomaly detection scores $\mathbf{s}$.

Initialize the network parameters Φ, Ψ, Υ and the orthogonal projection layer parameter Θ;
for $t \rightarrow \mathcal{T}$ **do**
  for each batch $\mathbf{G}$ **do**
    Calculate $I_{\Phi,\Psi,\Upsilon}\left(\mathbf{h}_{\Phi,\Upsilon}, \mathbf{H}_{\Phi,\Psi}(\mathbf{G})\right)$ via equation 4;
    Back-propagation GIN and update Φ, Ψ and Υ, respectively;
  end for
end for
Update the orthogonal parameter Θ of orthogonal projection layer by equation 9;
Obtain the global orthogonal latent representation by equation 8;
Initialize the center of hypersphere by equation 10;
repeat
  for each batch $\mathbf{G}$ **do**
    Calculate total loss via 11;
    Back-propagation and update Φ, Ψ, Υ and Θ, respectively;
    Further update the orthogonal parameter Θ of orthogonal projection layer by equation 9;
  end for
until convergence
Compute decision boundary r by equation 12;
Calculate the anomaly detection scores $\mathbf{s}$ through equation 13;
return The anomaly detection scores $\mathbf{s}$.

C Experiment Configuration

In this part, the experiment settings are listed for reproducing. First, each dataset is divided into two parts: training and testing sets. We randomly sample eighty percent of the normal data as the training set, and the remaining normal data together with the randomly sampled abnormal data in a one-to-one ratio to form the testing set.

With regard to the experiment settings of compared graph-kernel baselines, we adopted the classical AD method, One-Class SVM (OC-SVM) (Schölkopf et al., 2001) and used 10-fold cross-validation to make a fair comparison. All graph kernels via GraKel (Siglidis et al., 2020) to extract a Kernel matrix and apply OC-SVM in scikit-learn (Pedregosa et al., 2011) are implemented. Specifically, we selected Floyd Warshall as the SP kernel's algorithm and set lambda as 0.01 for the RW kernel. WL kernel algorithm is sensitive to the number of iterations, so we test four types with the iteration of $\{2, 5, 8, 10\}$ and denote them as WL_2, WL_5, WL_8, WL_{10}. For all graph kernels, the outputs are normalized. About infoGraph+Deep SVDD, the first stage runs in 20 epochs, and the second stage pretrains 50 epochs and trains 100 epochs. In OCGIN, GLocalKD, and OCGTL, their default or reported parameter settings are adopted to ensure experimental fairness.



(a) DOHSC (b) DO2HSC

Figure 5: Convergence curves of the proposed models on the MUTAG dataset.

Regarding our model DOHSC, we firstly set 1 epoch in the pretraining stage to initialize the center of the decision boundary and then train the model in 500 epochs. The convergence curves are given in Figure 5 to indicate that the final optimized results are adopted. The improved method

Algorithm 2 Graph-Level Deep Orthogonal Bi-Hypersphere Compression (DO2HSC)

Input: The input graph set $\mathbb{G}$, dimensions of GIN hidden layers k and orthogonal projection layer k', a trade-off parameter λ and the coefficient of regularization term μ, pretraining epoch $\mathcal{T}_1$, iterations of initializing decision boundaries $\mathcal{T}_2$, learning rate η.

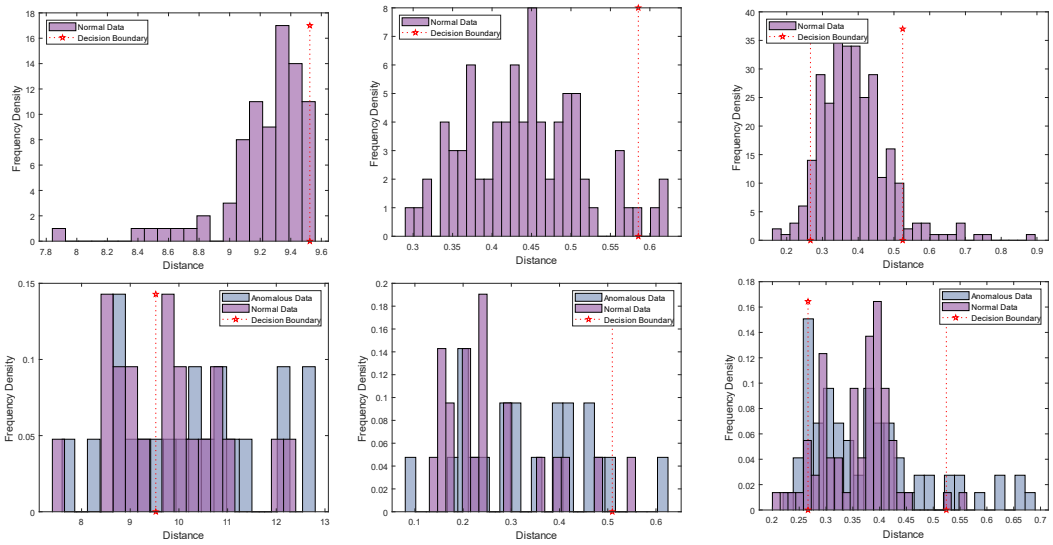
Output: The anomaly detection scores $\mathbf{s}$.

```
Initialize the network parameters Φ, Ψ, ϒ and the orthogonal projection layer parameter Θ;
for t → T_1 do
    for each batch G do
        Calculate I_{Φ,Ψ,ϒ}(h_{Φ,ϒ}, H_{Φ,Ψ}(G)) via equation 4;
        Back-propagation GIN and update Φ, Ψ and ϒ, respectively;
    end for
end for
Update the orthogonal parameter Θ of orthogonal projection layer by equation 9;
Obtain the global orthogonal latent representation by equation 8;
Initialize the center of hypersphere by equation 10;
for t → T_2 do
    for each batch G do
        Calculate total loss via 11;
        Back-propagation and update Φ, Ψ, ϒ and Θ, respectively;
        Further update the orthogonal parameter Θ of orthogonal projection layer by equation 9;
    end for
end for
Initialize decision boundaries r_max and r_min via equation 14;
repeat
    for each batch G do
        Calculate the improved total loss via 15;
        Back-propagation and update Φ, Ψ, ϒ and Θ, respectively;
        Further update the orthogonal parameter Θ of orthogonal projection layer by equation 9;
    end for
until convergence
Calculate the anomaly detection scores s through equation 16;
return The anomaly detection scores s.
```

DO2HSC is also set 1-epoch pretraining stage and trains DOHSC 5 epochs to initialize a suitable center and decision boundaries r_{max} and r_{min}, where the percentile ν of r_{max} is fixed to 0.05. After initializing, the model is trained in 500 epochs. For both proposed approaches, the trade-off factor λ is set to 10 to ensure decision loss as the main optimization objective. Dimensions of the GIN hidden layer and orthogonal projection layer are fixed as 16 and 8, respectively. About the backbone network, a 4-layer GIN and a 3-layer fully connected neural network are adopted. Besides, the averages and standard deviations of the Area Under Operating Characteristic Curve (AUC) are used to support the comparable experiments by repeating each algorithm ten times. The higher value of the AUC metric represents better performance. When calculating the AUC of graph-kernel baselines, we estimated the radius of the hypersphere as 99 percentile of all squared distances to the separating hyperplane and then determined the score as the difference between squared distances and its square radius. Regarding our model DOHSC, we firstly set 1 epoch in the pretraining stage to initialize the center of the decision boundary and then train the model in 500 epochs. The convergence curves are given in Figure 5. The improved method DO2HSC is also set 1-epoch pretraining stage and trains DOHSC 5 epochs to initialize a suitable center and decision boundaries r_{max} and r_{min}, where the percentile ν of r_{max} is fixed to 0.1. After initializing, the model is also trained in 500 epochs. For both proposed approaches, the trade-off factor λ is set to 10. Dimensions of the GIN hidden layer and orthogonal projection layer are fixed as 16 and 8, respectively. About the backbone network, a 4-layer GIN and a 3-layer fully connected neural network are adopted. Besides, the averages and standard deviations of the Area Under Operating Characteristic Curve (AUC) are used to support the comparable experiments by repeating each algorithm ten times. The higher value of the AUC metric represents better performance. When calculating the AUC of graph-kernel baselines, we estimated the radius of the hypersphere as 99 percentile of all squared distances to the separating hyperplane and then determined the score as the difference between squared distances and its square radius.

D SUPPLEMENTED VISUALIZATION

This part shows the related supplemented visualization results of the anomaly detection task.



(a) infoGraph+Deep SVDD (b) DOHSC (c) DO2HSC

Figure 6: Distance distributions were obtained by infoGraph+Deep SVDD, the proposed model, and the improved proposed model on COX2. The first row represents the distance distribution of the training samples in relation to the decision boundary. The last row indicates the distance distribution of the test data with respect to the decision boundary.

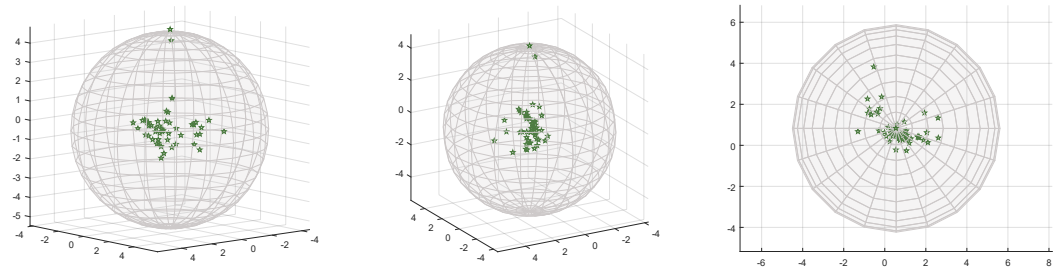
Figure 6 shows the distance distributions of the two-stage method, the proposed model DOHSC, and the improved DO2HSC. Here, the "distance" is defined as the distance between the sample and the center of the decision hypersphere. The distance distribution denotes the sample proportion in this distance interval to the corresponding total samples. It can be intuitively observed that most distances of instances are close to the decision boundary because of the fixed learned representation. As mentioned earlier, the jointly-trained algorithm has mitigated the situation, and the obtained representation makes many instances have smaller distances from the center of the sphere.

However, like we wrote in Section 2, anomalous data may occur in regions with less training data, especially the region close to the center, which is also confirmed by (a) and (b) of Figure 6. Differently, DO2HSC effectively shrinks the decision area, and we find that the number of outliers is obviously reduced due to a more compact distribution of training data.

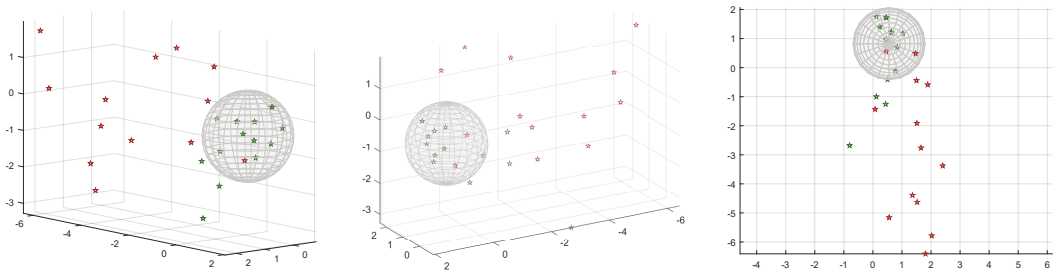
D.1 PARAMETER SENSITIVITY AND ROBUSTNESS

To claim the stability of our models, we analyze the parameter sensitivity and robustness of DOHSC and DO2HSC, respectively. Consider that the projection dimension varies in $\{4, 8, 16, 32, 64, 128\}$ while the hidden layer dimension of the GIN module ranges from 4 to 128. In Figure 8, the DO2HSC model has less volatile performances than DOHSC, especially when the training dataset is sampled from COX2 class 0, as Subfigure (d) shows. Noticeably, a higher dimension of the GIN hidden layer usually displays a better AUC result since the quality of learned graph representations improves when the embedding space is large enough.

In addition, we assess different aspects of model robustness. More specifically, the AUC results about two "ratios" are displayed: 1) Different sampling ratios for the training set; 2) Different ratios of noise disturbance for the learned representation. In Subfigures (c) (f), the purple bars regard normal data as class 0, while green bars treat normal data as class 1. Notice that most AUC results are elevating along with a higher ratio of authentic data in the training stage, demonstrating our models' potential in the unsupervised setting. On the other hand, when more noise is blended into



(a) Training Result



(b) Testing Result

Figure 7: Visualization results of the DOHSC with MUTAG in different perspectives.

the training dataset, the AUC performances of yellow line and blue line always stay stable at a high level. This outcome verifies our models' robustness in response to the alien data.

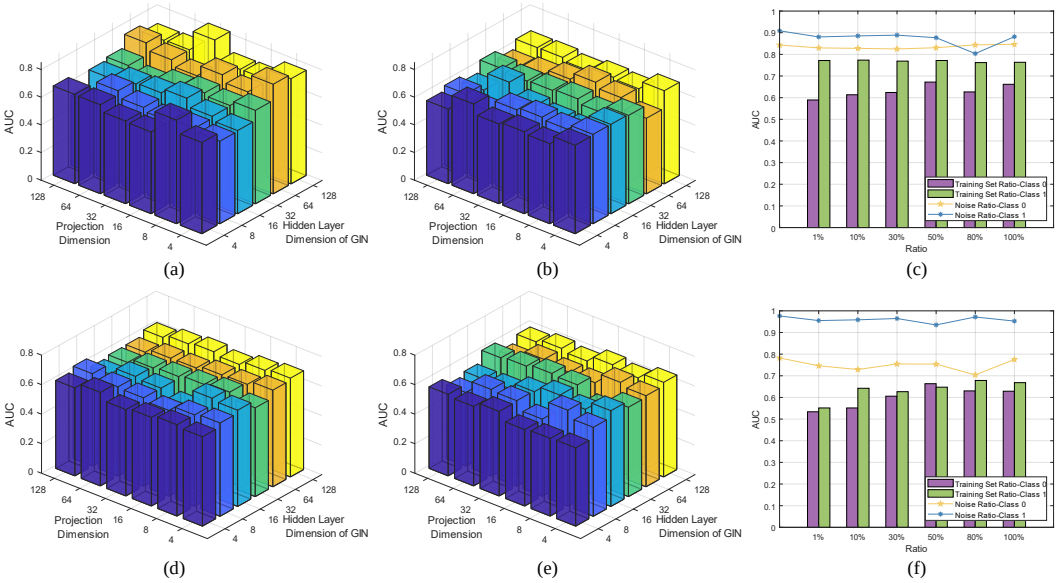
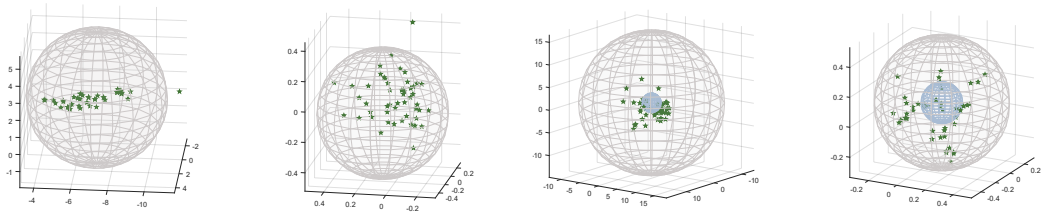


Figure 8: Parameter sensitivity and robustness of the proposed models. (a)-(b) Parameter sensitivity of DOHSC with different hidden layer dimensions of GIN and projection dimensions on COX2 with Class 0 and Class 1, respectively. (d)-(e) Parameter sensitivity of DO2HSC with the same settings. (c) and (f) shows the performance impacts with different ratios of the training set on the IMDB-Binary dataset and added noise disturbances on the MUTAG dataset while training DOHSC and DO2HSC, respectively.

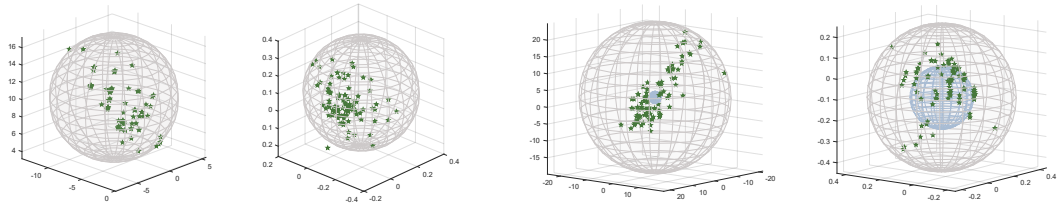
E SUPPLEMENTED VISUALIZATION OF ABLATION STUDY

To demonstrate the effectiveness of the orthogonal projection layer (OPL), we conduct ablation studies and visualize the comparison of 3-dimensional results produced with OPL and without OPL, respectively. For each model trained on a particular dataset class, we show the result without OPL on the left side, while the result with OPL is displayed on the right. As Figure 9 illustrates, OPL drastically improves the distribution of the embeddings to be more spherical rather than elliptical. Similarly, with the help of OPL, other embeddings show a more compact and rounded layout.



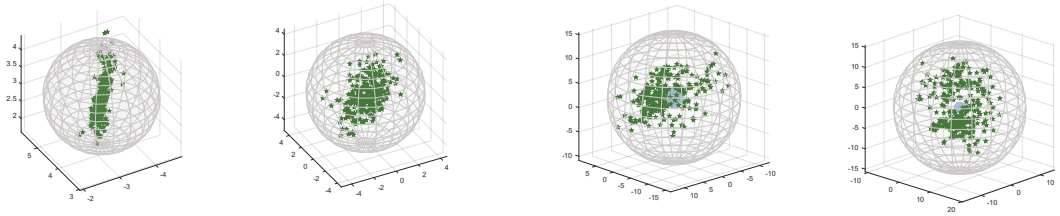
(a) DOHSC (b) DO2HSC

Figure 9: Visualizations on the MUTAG dataset Class 0 (left: with OPL; right: without OPL).



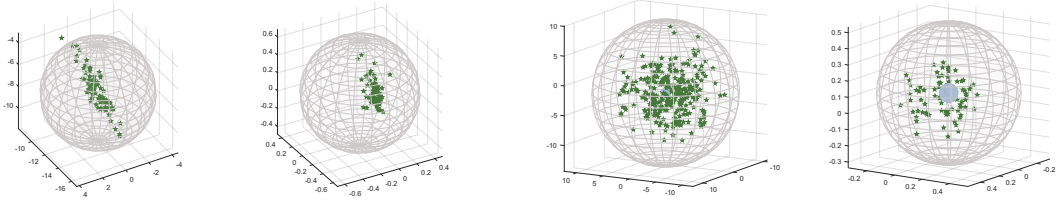
(a) DOHSC (b) DO2HSC

Figure 10: Visualizations on the MUTAG dataset Class 1 (left: with OPL; right: without OPL).



(a) DOHSC (b) DO2HSC

Figure 11: Visualizations on the COX2 dataset Class 0 (left: with OPL; right: without OPL).



(a) DOHSC (b) DO2HSC

Figure 12: Visualizations on the COX2 dataset Class 1 (left: with OPL; right: without OPL).