

Conformal Graph-level Out-of-distribution Detection with Adaptive Data Augmentation

Anonymous Author(s)*

ABSTRACT

Graph-level out-of-distribution (OOD) detection, which attempts to identify OOD graphs originated from an unknown distribution, is a vital building block for safety-critical applications in Web and society. Current approaches concentrate on how to learn better graph representations, but fail to provide any statistically guarantee on detection results, therefore impeding their deployments in the scenario where detection errors would result in serious consequences. To overcome this critical issue, we propose the Conformal Graph-level Out-of-distribution Detection (CGOD), extending the theory of conformal prediction to graph-level OOD detection with a rigorous control over the false positive rate. In CGOD, we develop a new aggregated non-conformity score function based on the proposed adaptive data augmentation. Through the guidance from two designed metrics, i.e., score consistency and representation diversity, our augmentation strategy can generate multiple non-conformity scores, and aggregating these generated non-conformity scores together is robust to the misleading information. Meanwhile, our score function can perceive the subsequent process of conformal inference, enabling the aggregated non-conformity score to be adaptive to different input graphs and deriving a more accurate score estimation. We conduct experiments on multiple real-world datasets with different empirical settings. Extensive results and model analyses demonstrate the superior performance of our approach over several competitive baselines.

CCS CONCEPTS

- Computing methodologies → Neural networks; • Information systems → Data mining.

KEYWORDS

Graph-level out-of-distribution detection, conformal prediction, graph neural networks

ACM Reference Format:

Anonymous Author(s). 2018. Conformal Graph-level Out-of-distribution Detection with Adaptive Data Augmentation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX')*. ACM, New York, NY, USA, 11 pages. <https://doi.org/XXXXXX.XXXXXXX>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/XXXXXX.XXXXXXX>

1 INTRODUCTION

Graph-level applications are ubiquitous in diverse set of research fields, such as social network analysis [22, 27, 41], molecular property prediction [6, 33, 40] and intelligent traffic forecasting [9, 13, 39]. Recently, a significant amount of approaches for these graph-level applications have been proposed [4, 44, 50], among which graph neural networks (GNNs) have received a great deal of attention [11, 16, 35]. Typically, GNNs are trained with a closed-world assumption that training graphs and test graphs follow the same data distribution [5, 20, 42]. Nevertheless, many real-world situations violate this closed-world assumption and, instead, involve out-of-distribution (OOD) graphs which have not been seen during the training process [8, 21, 37]. In fact, for an ideal graph machine learning model, it should not only make predictions on in-distribution (ID) graphs correctly, but also be able to detect such OOD samples during the inference phase to mitigate some unexpected risks, when being deployed on practical scenarios [14, 48].

On the basis of this high security demand, **graph-level OOD detection**, which aims to determine whether an input graph is ID or OOD, has become an important research direction. Previous works adopt different learning paradigms, e.g., contrastive learning [28] and prompt learning [23] to learn effective ID features, with the goal of enlarging the score gap between ID and OOD graphs. For example, GOOD-D [24] designs a method of hierarchical contrastive learning to capture the common patterns of ID graphs in different granularities (node-, graph- and group-levels) so that OOD graphs which violate these patterns can be easily exposed. AAGOD [10] proposes a post-hoc method, enabling a pre-trained GNN to detect OOD graphs without modifying model parameters. It leverages a graph prompt on the adjacency matrix to amplify the structural difference between ID and OOD graphs.

Despite their promising results, existing works of graph-level OOD detection remain at designing more advanced models, but fail to provide any statistically guarantee on detection results. Such a lack of rigor impedes their applications in the scenario where detection errors would result in serious consequences. In this paper, we attempt to establish the connection between graph-level OOD detection and conformal prediction (CP) [36] for filling this important gap. CP is a useful tool of generating prediction sets with a coverage guarantee that such sets cover the true label with a user-specified threshold. To be specific, we propose the **Conformal Graph-level Out-of-distribution Detection** named **CGOD**, a novel framework that extends CP to graph-level OOD detection with a rigorous control over the false positive rate (FPR). Additionally, CGOD is a flexible and light-weighted architecture which can be paired with different detection models to identify OOD graphs in a post-processing manner.

The non-conformity score function, which quantifies how different the input graph is from the training distribution, is the primary

factor for CP [1]. However, when handling graph-level OOD detection, it suffers from two critical limitations: (1) It follows a single point estimation of the non-conformity score, which is easily affected by the misleading information including the high estimation variance and the noise feature of recognizing OOD characteristics; (2) It is a pre-determined function, which cannot make adaptive adjustments to perceive the subsequent estimation process, leading to an inaccurate calculation. In this paper, we develop a new **aggregated non-conformity score function** to overcome above limitations, based on the proposed adaptive data augmentation.

Specifically, our method *differentiates* the process of conformal inference. Through optimizing two designed metrics, i.e., score consistency and representation diversity, CGOD can utilize our augmentation strategy to generate multiple non-conformity scores. By aggregating these scores together, CGOD derives an aggregated non-conformity score to enhance the robustness of the misleading information. Furthermore, due to the fact that our score function is trainable, it can be aware of the calculation of the non-conformity score, enabling the aggregated non-conformity score to be adaptive to different input graphs and deriving a more accurate score estimation. In general, the main contributions of this paper are summarized here:

- To the best of our knowledge, we are the first work that attempts to formulate the task of graph-level OOD detection from the perspective of CP, facilitating a rigorous control over the FPR while ensuring detection performance.
- We design two novel metrics to guide the proposed adaptive data augmentation to generate multiple non-conformity scores leading to an aggregated non-conformity score function for boosting model performance effectively.
- Experiments are conducted on multiple real-world datasets in different empirical settings. Extensive results and detailed analyses validate the superiority of our method over multiple strong counterparts¹.

2 RELATED WORK

2.1 Conformal Prediction on Graphs

CP [1, 36] is an uncertainty quantification framework which can produce statistically valid prediction sets (or intervals) for any pre-trained machine learning models, only assuming exchangeability of the data. The basic idea of CP is to estimate the p -value for each possible label of a new sample and exclude from the prediction set those labels having a p -value less than a user-specified threshold ϵ . Similar to statistical hypothesis testing [15], CP aims to reject the most unlikely labels at significance level ϵ . For estimating these p -values, CP leverages the *non-conformity score function* to measure how different a given sample is relative to a set of training samples. Thus, the non-conformity score function plays a crucial role of determining the usefulness of CP.

CP has proven effective in numerous domains. For example, ICAD [18, 19] exploits CP to detect anomalous trajectories. Recent works investigate how to use CP to solve problems on graphs. DAPS [47] and CF-GNN [12] study the exchangeability of GNNs under the semi-supervised node classification, and propose different

¹The source code is available at <https://anonymous.4open.science/r/CGOD/>

Table 1: Used notations.

Notation	Description
$\mathbb{P}^{in}, \mathbb{P}^{out}$	ID and OOD distributions of graphs
$\mathcal{D}^{in}, \mathcal{D}^{out}$	ID and OOD datasets
$\mathcal{D}_{tr}, \mathcal{D}_{te}$	training and test datasets
$\mathcal{D}_{ptr}, \mathcal{D}_{cal}$	proper training and calibration dataset
$\mathcal{D}_{cal-tr}, \mathcal{D}'_{cal}$	trainable and remaining calibration datasets
G^{in} and G_{te}	a ID graph from \mathcal{D}^{in} and a test graph from \mathcal{D}_{te}
$\text{detect}(\cdot)$	detection function
$F(\cdot, \cdot)$	transformation function
$\Phi(\cdot)$	a GNN encoder
$V(\cdot, \cdot)$ and s	non-conformity score function with its score
$\psi^{j,1:k}$	k data augmentations of the j -th graph
$\widehat{V}(\cdot, \cdot; \cdot)$	aggregated non-conformity score function
\hat{s}	aggregated non-conformity score
$\mathbf{g}_j, \hat{\mathbf{g}}_j^i$	graph representation with its i -th augmentation
p_{te}, \hat{p}_{te}	p -value and aggregated p -value

strategies to improve efficiency. CoDrug [17] introduces conformal molecular graph prediction, which adopts kernel density estimation to handle the problem of covariate shift. Different from above approaches, our method extends CP to graph-level OOD detection and develops a new aggregated non-conformity score function to improve model effectiveness.

2.2 OOD Detection on Graphs

OOD detection [29] attempts to identify OOD samples from ID data, which is an essential problem for deploying machine learning models on safety-critical applications in social networks. Many methods have studied the problem of OOD detection on graphs [43]. Among them, GOOD-D [24] is the first work focusing on graph-level OOD detection. As described in Introduction, GOOD-D designs a self-supervised method that contrasts different granularities to capture the ID patterns from both feature and structure views, so as to detect OOD graphs based on the discrepancy in these granularities. GOODAT [38] is one of follow-up works, which uses the information bottleneck to capture informative sub-graphs for achieving OOD detection in test time.

Graph-level anomaly detection [26] is a sub-area of graph-level OOD detection, since anomalous or malicious graphs can be regarded as a certain type of OOD data. There are many promising works on graph-level anomaly detection. For instance, OCGIN [49] is an end-to-end model which adopts GNNs to learn graph representations simultaneously optimizes an anomaly detection objective, e.g., one-class classification or reconstruction loss function. GlobalKD [25] learns global- and local-sensitive graph normality to detect anomalous graphs by the joint random distillation of graph and node representations. However, all above works fall short in providing any statistically guarantee on detection results.

3 PRELIMINARY

In this section, we first provide the preliminary in our paper and the used notations are summarized in Table 1. Let $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ denote an undirected graph, and \mathcal{V} and \mathcal{E} represent the sets of

nodes and edges. $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ represents the feature matrix, where d is the feature dimension. $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ is the adjacency matrix and each element $A_{i,j} \in \{0, 1\}$ denotes the connectivity between nodes i and j . Following the previous work [24], the definition of graph-level OOD detection is given as follows,

DEFINITION 1 (GRAPH-LEVEL OOD DETECTION). Assuming that we have an ID dataset \mathcal{D}^{in} where each input graph is originated from the distribution \mathbb{P}^{in} and an OOD dataset \mathcal{D}^{out} where each input graph is originated from the distribution \mathbb{P}^{out} . The training dataset $\mathcal{D}_{tr} = \{G_1^{\text{in}}, \dots, G_n^{\text{in}}\}$ is a subset of \mathcal{D}^{in} which only includes ID graphs (n is the size of \mathcal{D}_{tr}), while the test dataset \mathcal{D}_{te} is constituted by two separated datasets $\mathcal{D}_{te}^{\text{in}} \subset \mathcal{D}^{\text{in}}$ and $\mathcal{D}_{te}^{\text{out}} \subset \mathcal{D}^{\text{out}}$, i.e., $\mathcal{D}_{te} = \mathcal{D}_{te}^{\text{in}} \cup \mathcal{D}_{te}^{\text{out}}$ and $\mathcal{D}_{tr} \cap \mathcal{D}_{te}^{\text{in}} = \emptyset$. For an arbitrary test sample $G_{te} \in \mathcal{D}_{te}$, the goal of graph OOD detection is to distinguish which distribution (\mathbb{P}^{in} or \mathbb{P}^{out}) G_{te} belongs to.

Graph-level OOD detection typically follows an *unsupervised* setting in which only unlabeled ID graphs are available for identifying OOD graphs. Most current approaches focus on designing an effective GNN encoder $\Phi(\cdot)$ to learn the graph representation $\mathbf{g} \in \mathbb{R}^c$, i.e., $\mathbf{g} = \Phi(G)$, where c denotes the representation dimension. Afterwards, a transformation function $F(\cdot, \cdot)$ that transforms the graph representation into the OOD detection score can be applied on \mathbf{g} to identify its OOD-ness:

$$\text{detect}(G) = \begin{cases} 1 & F(\Phi(G), \mathcal{D}_{tr}) \geq \gamma \\ 0 & F(\Phi(G), \mathcal{D}_{tr}) < \gamma \end{cases} \quad (1)$$

Here 0 and 1 indicate the ID sample and the OOD sample, respectively. SSD [31] is a well-known transformation function for OOD detection. It uses the k -means algorithm to group graph representations into T clusters and adopts the Mahalanobis distance between \mathbf{g} and the nearest cluster center as the OOD detection score:

$$F(\Phi(G), \mathcal{D}_{tr}) = F(\mathbf{g}, \mathcal{D}_{tr}) = \min_t (\mathbf{g} - \boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_t^{-1} (\mathbf{g} - \boldsymbol{\mu}_t), \quad (2)$$

$\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ are the sample mean and the sample covariance of the cluster t generated from \mathcal{D}_{tr} .

4 METHODOLOGY

In this section, we first delve into how to achieve graph-level OOD detection from the view of CP. This process follows a post-processing manner that can be armed with different pre-trained detection models to achieve prediction. We then introduce the key component of CGOD: the aggregated non-conformity score function based on the proposed adaptive data augmentation. Figure 1 shows a sketch of CGOD. Model analysis including the theoretical detection guarantee and time complexity is given in the end.

4.1 Model Overview

CGOD follows a most widely used case of CP, i.e., split conformal prediction (SCP) [36] for model implementation. Specifically, CGOD includes the following three steps: (1) **Data split**. For improving computational efficiency, CGOD first splits \mathcal{D}_{tr} into a *proper training set* $\mathcal{D}_{ptr} = \{G_j^{\text{in}}\}_{j=1}^m$ and a *calibration set* $\mathcal{D}_{cal} = \{G_j^{\text{in}}\}_{j=m+1}^n$, i.e., $\mathcal{D}_{tr} = \mathcal{D}_{ptr} \cup \mathcal{D}_{cal}$. (2) **Non-conformity score function**. Given an arbitrary graph G , CGOD has to define the non-conformity score function $V(G, \mathcal{D}_{ptr}) = s$ for testing whether G conforms to

\mathcal{D}_{ptr} , where s indicates how different G is relative to the graph samples in \mathcal{D}_{ptr} . A higher s demonstrates that G is more different from \mathcal{D}_{ptr} . To be consistent with Eq.(1-2), we can set

$$V(G, \mathcal{D}_{ptr}) := F(\Phi(G), \mathcal{D}_{ptr}), \quad (3)$$

where $\Phi(\cdot)$ and $F(\cdot, \cdot)$ can be pre-trained models. (3) **P-value estimation**. The non-conformity score function then can be applied to the calibration set \mathcal{D}_{cal} for generating the non-conformity score $s_j = V(G_j^{\text{in}}, \mathcal{D}_{ptr})$ of each graph $G_j^{\text{in}} \in \mathcal{D}_{cal}$. Likewise, the non-conformity score of a test sample G_{te} can be calculated as s_{te} . Building upon these non-conformity scores, the *p*-value of G_{te} is estimated as the ratio of s_{m+1}, \dots, s_n that are at least as large as s_{te} :

$$p_{te} = \frac{|\{j = m+1, \dots, n : s_j \geq s_{te}\}| + 1}{n - m + 1}. \quad (4)$$

When p_{te} is smaller than a given OOD threshold $\epsilon \in (0, 1)$, G_{te} is identified as an OOD sample. As suggested by previous works [2], the above process can be viewed as a *statistical hypothesis testing* where the null hypothesis is that G_{te} is originated from the same distribution as the graph samples in \mathcal{D}_{ptr} , and p_{te} can be interpreted as the probability of rejecting this null hypothesis wrongly.

Furthermore, applying CP to graph-level OOD detection requires a critical assumption that the condition of exchangeability² holds for graph data [1, 12, 47]. We highlight that CGOD can satisfy exchangeability via meeting the following *independence* proposition:

PROPOSITION 1. In the setting of graph-level OOD detection described in Definition 1, the non-conformity score function $V(\cdot, \cdot)$, the calibration set \mathcal{D}_{cal} and the test dataset \mathcal{D}_{te} are independent of each other.

The corresponding proof is evident: the non-conformity score function $V(\cdot, \cdot)$ is built upon the proper training set \mathcal{D}_{ptr} . \mathcal{D}_{ptr} and \mathcal{D}_{cal} are i.i.d. Meanwhile, \mathcal{D}_{te} is another independent fold of the used graph data.

4.2 Aggregated Non-conformity Score Function

According to the above three steps, the non-conformity score function is the primary factor in CP, because it incorporates almost all the information for determining whether the given graphs are OOD samples. However, it still exists two critical limitations as described in Introduction: First, $V(\cdot, \cdot)$ follows a single point estimation so that it is easily affected by the misleading information; Second, $V(\cdot, \cdot)$ is a pre-determine function which cannot perceive the subsequent process of *p*-value estimation.

To overcome these limitations, we differentiate the process of *conformal inference* and introduce a novel aggregated non-conformity score function. In our score function, we propose the adaptive data augmentation where each data augmentation captures different representation diversities and corresponds to a non-conformity score. Aggregating multiple non-conformity scores together can reduce the high variance introduced by the single point estimation and is robust to the noise feature of recognizing OOD characteristics. Meanwhile, the proposed augmentation strategy is trainable, so our score function can be aware of the follow-up *p*-value estimation,

²Exchangeability. This condition requires the distribution to be invariant to permutations of the elements in $\{G_1, \dots, G_n\}$. More precisely, for each finite n , if π is a permutation of $\{1, \dots, n\}$, then: $\mathbb{P}(G_{\pi(1)}, \dots, G_{\pi(n)}) = \mathbb{P}(G_1, \dots, G_n)$.

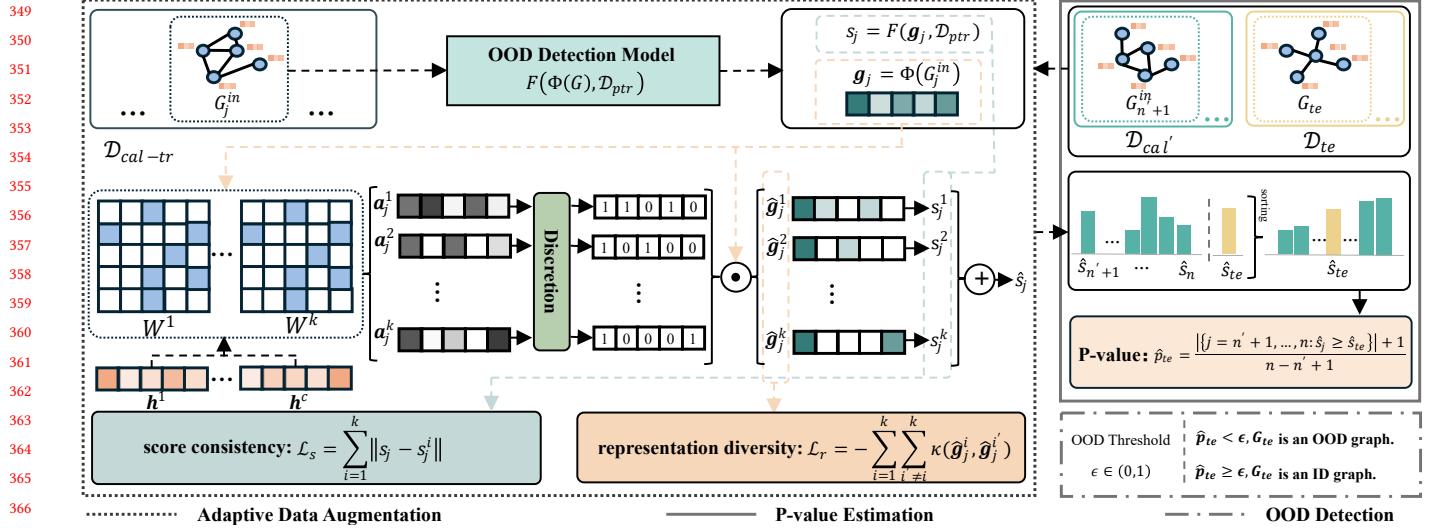


Figure 1: CGOD for graph-level OOD detection. In the training phase, we design the metrics of score consistency and representation diversity to optimize our proposed adaptive data augmentation. In the inference phase, CGOD first calculates these aggregated non-conformity scores of \mathcal{D}'_{cal} and $G_{te} \in \mathcal{D}_{te}$, and then derives the p -value of G_{te} for achieving OOD detection.

enabling the aggregated non-conformity score to be adaptive to different input graphs and thereby deriving a more accurate score estimation.

Specifically, given k data augmentations $\Psi^{1:k} = (\Psi^1, \dots, \Psi^k)$, CGOD first generates k non-conformity scores $s^{1:k} = (s^1, \dots, s^k)$ for G :

$$s^{1:k} = (V(\Psi^1(G), \mathcal{D}_{ptr}), \dots, V(\Psi^k(G), \mathcal{D}_{ptr})). \quad (5)$$

Afterwards, our score function $\widehat{V} = (G, \mathcal{D}_{ptr}; \Psi^{1:k})$ is set as an element-wise strictly increasing function with the following aggregated non-conformity score:

$$\hat{s} = \widehat{V}(G, \mathcal{D}_{ptr}; \Psi^{1:k}) = \sum_{i=1}^k V(\Psi^i(G), \mathcal{D}_{ptr}). \quad (6)$$

We explain why use this function setting in Model Analysis. In the next, we describe how to optimize these data augmentations.

Adaptive Data Augmentation. Instead of using some heuristic augmentations working on graph structures and node features [30], our augmentation strategy follows a trainable fashion which perturbs learned graph representations in the embedding space. This approach has been demonstrated to be a simple yet effective way of introducing noise to graphs [3]. To train these data augmentations, we first split a small number of data from \mathcal{D}_{cal} as the trainable calibration set $\mathcal{D}_{cal-tr} = \{G_j^in\}_{j=m+1}^{n'}$, and the remaining data $\mathcal{D}'_{cal} = \{G_j^in\}_{j=n'+1}^n$ is still used as the calibration set, i.e., $\mathcal{D}_{cal} = \mathcal{D}_{cal-tr} \cup \mathcal{D}'_{cal}$. We then design two metrics, i.e., score consistency and representation diversity as the loss function to train these data augmentations on \mathcal{D}_{cal-tr} .

For an arbitrary graph $G_j^in \in \mathcal{D}_{cal-tr}$, each data augmentation would generate an augmented graph representation $\hat{g}_j^i = \Psi^{j,i}(g_j; \theta^i)$, where g_j denotes the original graph representation of

G_j^in and θ^i denotes the model parameters. We parameterize each data augmentation $\Psi^{j,i}$ as a masking operation that would mask some inessential or redundant dimensions. To effectively determine which dimensions should be masked, we first use the attention mechanism $att(g_j; \theta^i)$ to derive a weighted vector $a_j^i \in \mathbb{R}^c$ where each element $a_j^{i,z}$ represents the importance of z -th dimension:

$$a_j^{i,z} = \frac{\exp(\text{LeakyReLU}(g_j^T W^i h^z))}{\sum_{o=1}^c \exp(\text{LeakyReLU}(g_j^T W^i h^o))}. \quad (7)$$

Here W^i is a parameter matrix in $\Psi^{j,i}$, and $\{h^o\}_{o=1}^c$ represents a set of dimension-wise trainable vectors. We then perform a discretization operation on a_j^i , i.e., $\text{disc}(a_j^i; \xi)$ where the highest values are discretized as ones and the others are discretized as zeros, with a specific ratio ξ for controlling the proportion of zeros. \hat{g}_j^i is finally defined as

$$\hat{g}_j^i = \Psi^{j,i}(g_j; \theta^i) = \text{disc}(\text{att}(g_j; \theta^i); \xi) \odot g_j, \quad (8)$$

where \odot denotes the element-wise multiplication. Once \hat{g}_j^i has been generated, we can further derive its corresponding non-conformity score $s_j^i = F(\hat{g}_j^i, \mathcal{D}_{ptr})$.

Designed Metrics. We design two metrics, i.e., score consistency and representation diversity to train our augmentation strategy, with the goal of improving the robustness of the noise feature of recognizing OOD characteristics. Concretely, given k augmented graph representations $\hat{g}_j^{1:k} = (\hat{g}_j^1, \dots, \hat{g}_j^k)$, the score consistency requires that their corresponding non-conformity scores $s_j^{1:k} = (s_j^1, \dots, s_j^k)$ are consistent with the original non-conformity score s_j of the input graph G_j^in . Meanwhile, representation diversity encourages $\hat{g}_j^{1:k}$ to be dissimilar from each other. Combining these two metrics together, we can define the following loss function:

$$\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_r = \underbrace{\sum_{i=1}^k \|s_j - s_j^i\|}_{\text{score consistency}} - \lambda \underbrace{\sum_{i=1}^k \sum_{i' \neq i}^k \kappa(\hat{g}_j^i, \hat{g}_j^{i'})}_{\text{representation diversity}}, \quad (9)$$

where \mathcal{L}_s and \mathcal{L}_r denote the loss functions of score consistency and representation diversity, λ is a harmonic factor, and κ represents a distance function for measuring the distance between two graph representations. In fact, each non-conformity score corresponds to a p -value:

$$p_j^i = \frac{|\{j' = n' + 1, \dots, n : s_{j'} \geq s_j^i\}| + 1}{n - n' + 1}. \quad (10)$$

But directly optimizing the consistency of the p -values generated by these augmented graph representations is non-differentiable due to Eq.(10). So we minimize the loss function of score consistency to ensure p -value consistency for overcoming the second limitation mentioned in Section 4.2.

Overall, through optimizing these two metrics, CGOD can be aware of the downstream of conformal inference, generating the augmented graph representations that own a high-level consistency of p -value estimation with the original graph representation while incorporating as much representation diversity as possible. Moreover, our proposed adaptive data augmentation is not limited to the graph-level representation. It is a general augmentation strategy, which can be armed with different level representations, e.g., the sub-graph level to calculate the corresponding non-conformity scores. Due to this property, CGOD can be combined with different detection models to recognize OOD graphs.

4.3 Model Analysis

4.3.1 Detection Guarantee. According to the standard principle of CP [7, 36], these generated k non-conformity scores are required to be exchangeable random variables. To approximate this condition, we parameterize a set of graph data augmentations \mathcal{S} via learned graph representations and the attention mechanism as described in Eq.(7). In this way, each data augmentation can be regarded as one data point which is randomly and independently sampled from \mathcal{S} . For avoiding ties, we assume that these aggregated non-conformity scores of an arbitrary test sample and of the samples in \mathcal{D}_{cal-tr} are distinct with probability 1. To comply with this assumption, we set the aggregated non-conformity score function as an element-wise strictly increasing function, and suppose that the data distribution is absolutely continuous w.r.t Lebesgue measure.

THEOREM 4.1. *Given that \mathcal{S} represents a set of graph data augmentations, the aggregated non-conformity score of an arbitrary graph $G_j^{in} \in \mathcal{D}'_{cal}$ is denoted as*

$$\hat{s}_j = \widehat{V}(G_j^{in}, \mathcal{D}_{ptr}; \Psi^{j,1:k}) = \sum_{i=1}^k V(\Psi^{j,i}(G_j^{in}), \mathcal{D}_{ptr}), \quad (11)$$

where $\Psi^{j,i}$ is sampled from \mathcal{S} independently. The aggregated non-conformity score \hat{s}_{te} of the test graph G_{te} follows the same computational procedure with \hat{s}_j . If G_{te} belongs to the distribution \mathbb{P}^{in} , then

its p -value is defined as

$$\hat{p}_{te} = \frac{|\{j = n' + 1, \dots, n : \hat{s}_j \geq \hat{s}_{te}\}| + 1}{n - n' + 1}, \quad (12)$$

which follows a discrete uniform distribution:

$$\hat{p}_{te} \sim \text{Uniform}\left(\left\{\frac{1}{n - n' + 1}, \frac{2}{n - n' + 1}, \dots, 1\right\}\right). \quad (13)$$

Here we denote the new p -value of G_{te} as the aggregated p -value, i.e., \hat{p}_{te} to distinguish from the original p -value p_{te} in Eq.(4). The corresponding proof has been provided in Appendix A.1. If \hat{p}_{te} is smaller than a given OOD detection threshold ϵ , then G_{te} is recognized as an OOD sample. Based on Theorem 4.1, we have the following detection guarantee:

THEOREM 4.2. *Suppose the calibration set \mathcal{D}'_{cal} sampled from the distribution \mathbb{P}^{in} , the test dataset \mathcal{D}_{te} and the aggregated non-conformity score function $\widehat{V}(\cdot, \cdot; \cdot)$ are independent of each other. Given a test graph $G_{te} \in \mathcal{D}_{te}$ with its aggregated p -value \hat{p}_{te} calculated by Eq.(12) and a specified OOD threshold ϵ , then we have the following upper bound:*

$$\Pr(\hat{p}_{te} < \epsilon | G_{te} \sim \mathbb{P}^{in}) \leq \epsilon. \quad (14)$$

Theorem 4.2 demonstrates that our proposed aggregated non-conformity score function can control the type I error rate guaranteeing a bounded FPR for graph-level OOD detection. The corresponding proof is provided in Appendix A.2.

4.3.2 Time Complexity. Appendix A.3 shows the training and inference pseudo-codes of CGOD. CGOD is a light-weighted architecture, which adopts the batch manner for model training and inference. The time complexity of model training is $O(B(|\mathcal{D}'_{cal}|k(c^2 + c) + k^2 + c))$, where B denotes the batchsize. In the inference phase, we can calculate these aggregated non-conformity scores of \mathcal{D}'_{cal} in advance, which largely reduces the time complexity. Based on this operation, the time complexity of model inference is $O(B(k(c^2 + c) + |\mathcal{D}'_{cal}|))$. Since the above parameters typically take small values, so CGOD keeps good model efficiency. In Appendix A.8, we provide the concrete runtime comparison.

5 EXPERIMENTS

In this section, we conduct extensive experiments to answer the following research questions (RQs):

- **RQ1:** Can our method be combined with different detection models to improve empirical performance?
- **RQ2:** Can our method provide a rigorous control over the false positive rate?
- **RQ3:** Does our method achieve the supreme performance in comparison with multiple strong baselines?
- **RQ4:** What are the contributions of the proposed different components in our method?
- **RQ5:** How sensitive is our method with respect to different hyper-parameters?

To answer these above questions, we conduct a detailed comparative analysis.

581 **Table 2: OOD performance comparison (%) of adapting different detection methods (GIN_D, IG_D and GOOD-D) to our framework**
 582 **on six ID & OOD datasets.**

ID	OOD	Metric	GIN _D	GIN _D +CP	CGOD _S	IG _D	IG _D +CP	CGOD _U	GOOD-D	GOOD-D+CP	CGOD _E
IMDB-M	IMDB-B	AUC	74.43	74.08	77.07	74.39	74.27	75.81	77.40	77.06	79.02
		AUPR	65.05	64.32	69.08	68.08	68.12	66.78	67.84	70.29	71.95
		FPR95	69.20	65.16	45.12	54.44	54.00	45.00	52.44	62.24	47.71
ClinTox	LIPO	AUC	52.40	52.19	58.60	48.63	46.04	53.09	62.17	61.99	65.13
		AUPR	45.15	48.78	52.97	42.73	43.97	50.41	57.49	55.69	62.41
		FPR95	92.09	93.07	82.88	90.46	93.92	88.42	79.28	75.70	70.72
BBBP	BACE	AUC	76.88	75.35	78.61	78.18	78.16	78.51	79.72	79.35	81.97
		AUPR	68.33	67.83	72.00	69.44	69.37	69.34	72.18	71.59	79.65
		FPR95	68.44	68.06	65.98	65.20	63.13	62.85	59.80	54.31	53.61
Esol	MUV	AUC	84.43	83.98	85.59	80.61	79.67	82.41	87.43	86.86	89.68
		AUPR	78.80	78.62	81.68	72.53	72.36	75.19	86.81	86.70	90.74
		FPR95	46.90	50.65	44.63	51.43	55.03	47.72	65.19	58.99	48.09
AIDS	DHFR	AUC	95.41	95.04	96.04	93.51	93.42	96.28	96.06	96.02	98.17
		AUPR	91.32	93.10	95.20	90.24	91.81	94.78	89.96	95.82	98.07
		FPR95	12.53	13.73	15.67	22.85	23.39	13.83	6.83	12.11	7.08
ENZYMES	PROTEIN	AUC	54.69	54.55	57.81	57.60	57.75	60.42	58.73	56.78	61.53
		AUPR	60.34	63.44	67.93	63.40	63.00	66.57	61.89	64.15	68.98
		FPR95	97.06	97.22	93.53	95.00	94.31	93.02	92.78	93.82	92.50

604 5.1 Experimental Setting

605 **5.1.1 Datasets.** According to previous works [24], we use six pairs
 606 of graph datasets as ID and OOD data respectively, i.e., **IMDB-M & IMDB-B**, **ClinTox & LIPO**, **BBBP & BACE**, **Esol & MUV**,
 607 **AIDS & DHFR**, and **ENZYMES & PROTEIN**. The descriptions
 608 of these dataset pairs are provided in Appendix A.4. Each dataset
 609 pair comes from the same domain and has a moderate domain shift
 610 between them. For each dataset pair, we treat the first dataset as
 611 the ID dataset and the second dataset as the OOD dataset. 60%
 612 of and 10% of graphs from the ID dataset are used for training
 613 and calibration respectively, i.e., \mathcal{D}_{ptr} and \mathcal{D}_{cal} . The remaining
 614 30% of ID graphs and the same number of OOD graphs from the
 615 second dataset are used for constructing \mathcal{D}_{te} . \mathcal{D}_{cal} is further evenly
 616 divided into \mathcal{D}_{cal-tr} and \mathcal{D}'_{cal} . \mathcal{D}_{ptr} is used to derive a pre-trained
 617 OOD detection model.

618 **5.1.2 Baselines.** To fully evaluate our method, we compare CGOD
 619 with the following three compared categories:

- **Supervised method with detector.** This branch owns two stages. At the first stage, it would train some powerful graph encoders, such as graph kernels and GNNs in a supervised manner. Based on the learned graph representations, these methods adopt an OOD or anomaly detector to identify OOD graphs at the second stage.
- **Unsupervised method with detector.** This branch also owns two stages. The only difference is that this branch adopts the unsupervised or self-supervised manner to train graph encoders. Hence, graph contrastive learning models, such as InfoGraph [34] and GraphCL [46] can be well used at the first stage.
- **End-to-end method.** Current state-of-the-art (SOTA) models belong to this branch, e.g., GOOD-D [24]. In addition,

620 some promising graph anomaly detection methods, such
 621 as GLocalKD [25] are also end-to-end methods.

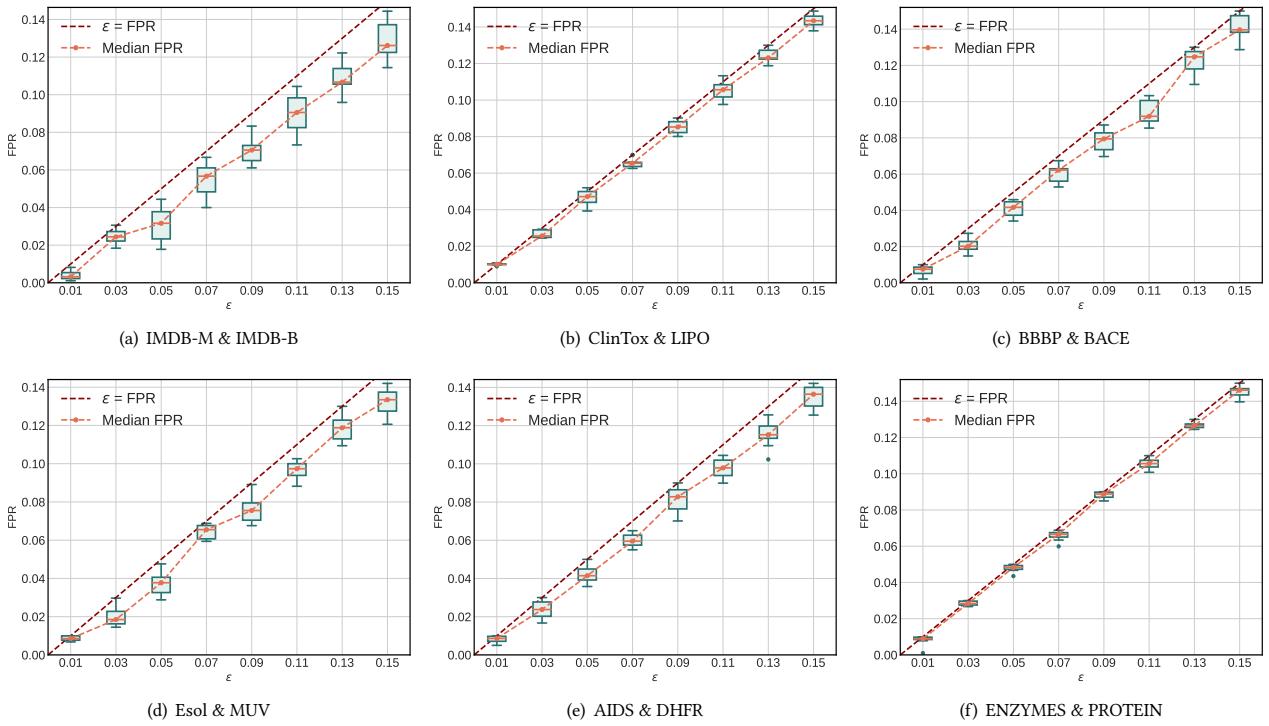
622 The description of these used baselines is provided in Appendix A.5.

623 **5.1.3 Implementation.** There are four main hyper-parameters in
 624 our model: the learning rate for training the proposed adaptive data
 625 augmentation, i.e., lr, the number of data augmentations, i.e., k, the
 626 discretization ratio used in Eq.(8), i.e., ξ , and the harmonic factor
 627 used in Eq.(9) i.e., λ . We adopt the grid search to find their optimal
 628 values. The search intervals of these hyper-parameters are reported
 629 in Appendix A.6. For a fair comparison, all baselines and CGOD
 630 share the same hidden dimensional size. In addition, since \mathcal{D}_{cal} is
 631 not used in all baselines, we integrate \mathcal{D}_{cal} into the training dataset
 632 for their model training.

633 **5.1.4 Evaluation Metrics.** Following previous works [10, 43], we
 634 adopt four OOD detection metrics: **AUC**, **AUPR**, **FPR95** and **FPR**.
 635 The first three metrics are used for evaluating detection results.
 636 A higher AUC, AUPR and a lower FPR95 indicate better model
 637 performance. FPR is used for testing whether our method can effec-
 638 tively control FPR within a user-specified threshold ϵ . The detailed
 639 descriptions of these metrics are provided in Appendix A.7.

639 5.2 Model Adaptation (A1)

640 In this section, we test whether CGOD can be armed with different
 641 detection models to achieve prediction. Three representative meth-
 642 ods from the above categories in Section 5.1.2 are selected: GIN,
 643 InfoGraph and GOOD-D. GIN and InfoGraph are classic GNNs in
 644 supervised and unsupervised settings. We arm them with SSD to
 645 achieve detection which are denoted as GIN_D and IG_D. GOOD-D
 646 is a classic end-to-end detection model. We denote these methods
 647 armed with CGOD as CGOD_S, CGOD_U and CGOD_E, respectively.
 648 We also combine them with the traditional CP framework, and

Figure 2: In our method, the FPR is upper bounded by ϵ on six ID & OOD datasets.

"+CP" is added after the model name. Table 2 shows their empirical results. The best performance is in boldface. From it, we have the following conclusions:

- CGOD can further improve the performance of different detection methods. Compared with three detection methods, the most obvious improvement is that CGOD_U reduces FPR95 by 39.47% on AIDS & DHFR.
- The comparison between CGOD and those armed with the traditional CP framework demonstrates the effectiveness of our proposed aggregated non-conformity score function. The most obvious improvement is that CGOD_U reduces FPR95 by 40.87% on AIDS & DHFR.
- Through optimizing the metrics of score consistency and representation diversity, our proposed adaptive data augmentation can better capture the intrinsic features among ID graphs. Therefore, CGOD can not only control the type I error rate, but also reduce the type II error rate and improve the detection performance.

5.3 FPR Controlling (A2)

In this section, we investigate whether the FPR can be controlled by CGOD. The user-specified threshold ϵ is selected from 0.01 to 0.15 with the step size 0.02. For each ϵ , we perform 10 random splits of \mathcal{D}_{cal-tr} and \mathcal{D}'_{cal} . We select CGOD_E as the test model. The empirical results of all datasets are shown in Figure 2. Besides the theoretical result of Theorem 4.2, Figure 2 shows that the FPR can be empirically upper bounded by ϵ for all datasets in our method.

Table 3: Ablation study on six ID & OOD datasets w.r.t. AUC.

Linear	\mathcal{L}_s	\mathcal{L}_r	IMDB-M IMDB-B	ClinTox LIPO	BBBP BACE	Esol MUV	AIDS DHFR	ENZYME PROTEIN
✓	✓	-	77.39	62.85	80.07	87.49	96.21	58.20
✓	-	✓	76.08	60.62	78.43	85.18	95.51	56.33
✓	✓	✓	78.53	67.08	81.26	88.12	96.44	59.88
-	✓	✓	79.02	65.13	81.97	89.68	98.17	61.53

Moreover, for different datasets, the bounded difficulty is different. The upper bounds in the sub-figures (a), (c), (d) and (e) are more relaxed than those in the sub-figures (b) and (f).

5.4 Performance Comparison (A3)

To further validate the effectiveness of CGOD, we also compare CGOD_E with eight SOTA graph-level OOD detection models which are described in Appendix A.5. We conduct experiments on all datasets and select AUC as the evaluation metric. The empirical results are provided in Figure 3. From it, we can conclude that CGOD_E consistently achieves the best detection performance on all datasets. Particularly, the most significant improvement is 4.76% increase in AUC of CGOD_E compared to the SOTA baseline on ClinTox & LIPO. It is a non-trivial improvement because CGOD_E uses 10% less training data. It also demonstrates the effectiveness of our framework with the proposed aggregated non-conformity score function.

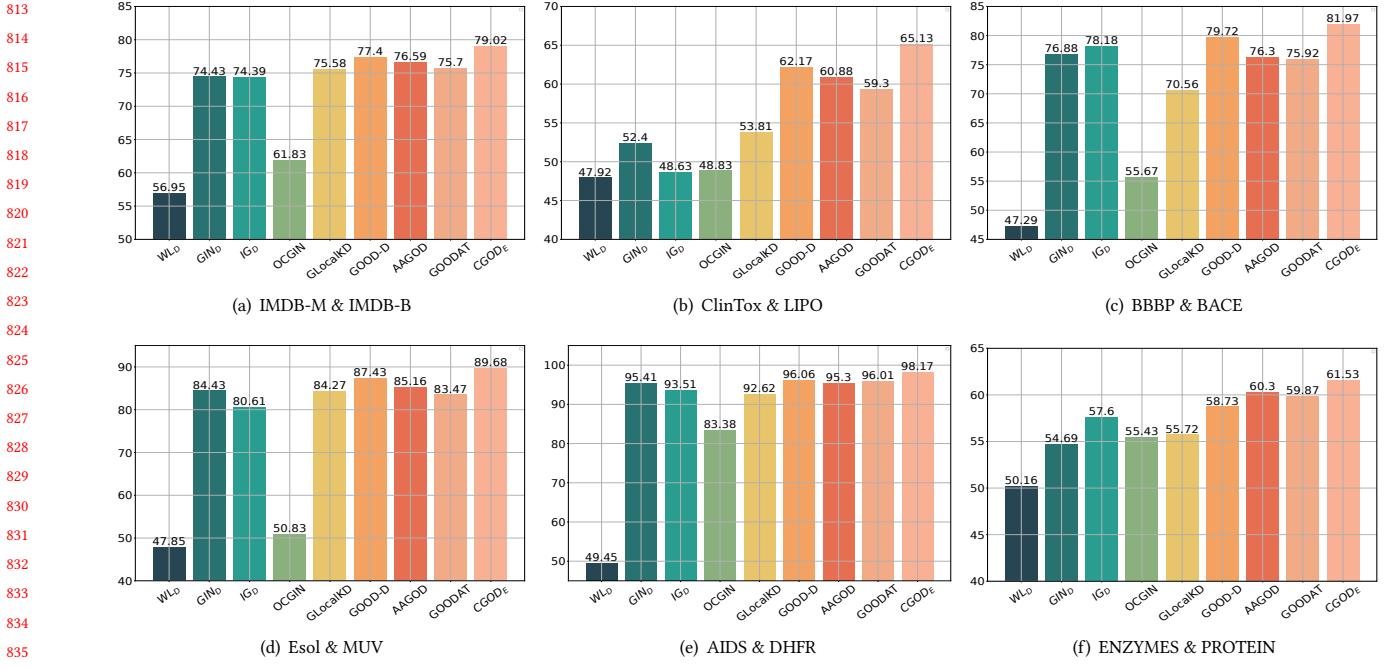


Figure 3: Performance (%) comparison on six ID & OOD datasets w.r.t. AUC.

5.5 Ablation Study (A4)

In this section, we perform the ablation study to validate the effectiveness of different components in CGOD. Specifically, we consider three model variants: 1) the one uses the linear operation to generate the weighted vector a_j^i instead of the attention mechanism in Eq.(7); 2) the one only uses the loss function of score consistency, i.e., \mathcal{L}_s ; 3) the one only uses the loss function of representation diversity, i.e., \mathcal{L}_r . We select CGODE as the backbone, and we denote three model variants as CGODE-ls, CGODE-lr and CGODE-lsr sequentially.

The empirical results are shown in Table 3. From it, we can see that among three model variants CGODE-lsr achieves the best performance. But CGODE-lsr is still inferior to CGODE, which illustrates the effectiveness of the attention mechanism in our data augmentation. In addition, compared with CGODE and CGODE-lsr, the performance of CGODE-ls and CGODE-lr has significantly declined. For CGODE-ls, there may be redundant features among the augmented graph representations without \mathcal{L}_r . For CGODE-lr, the augmented graph representations may deviate too far from the original graph representation, resulting in the introduction of noisy information in model prediction.

5.6 Hyper-parameter Sensibility (A5)

Figure 4 shows the hyper-parameter sensitivity of CGOD in terms of four hyper-parameters as mentioned in Section 5.1.3. We select CGODE as the test case. From it, we have the following observations: A reasonable hyper-parameter setup is to set the learning rate and λ to the smaller values for better designing and optimizing the loss function; k and ξ are suitable to assign the larger values for

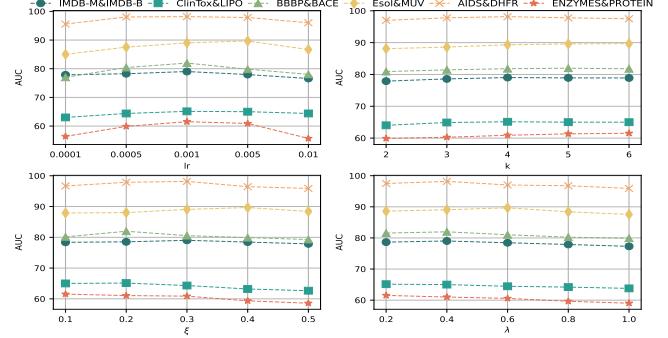


Figure 4: Hyper-parameter sensitivity of CGOD.

improving model expressiveness, but setting them to an excessively large value can also increase the risk of overfitting.

6 CONCLUSION

In this paper, we present the CGOD, a novel framework that extends the theory of CP to graph-level OOD detection for providing a rigorous control over detection results. CGOD differentiates the process of conformal inference and proposes a trainable augmentation strategy, i.e., adaptive data augmentation to generate multiple non-conformity scores. Building on this, CGOD further introduces a new aggregated non-conformity score function that can aggregate these generated scores to improve the robustness and accuracy of the non-conformity score estimation. Extensive experiments show that CGOD outperforms many strong baselines and guarantees a bounded FPR effectively.

REFERENCES

- [1] Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. 2014. *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes.
- [2] Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. 2023. Testing for outliers with conformal p-values. *The Annals of Statistics* 51, 1 (2023), 149–178.
- [3] Nan Chen, Zemin Liu, Bryan Hooi, Bingsheng He, Rizal Fathony, Jun Hu, and Jia Chen. 2024. Consistency Training with Learnable Data Augmentation for Graph Anomaly Detection with Limited Supervision. In *The Twelfth International Conference on Learning Representations*.
- [4] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. 2017. A Survey on Network Embedding. *IEEE Transactions on Knowledge and Data Engineering* 31 (2017), 833–852. <https://api.semanticscholar.org/CorpusID:3951790>
- [5] Shaohua Fan, Xiao Wang, Chuan Shi, Peng Cui, and Bai Wang. 2023. Generalizing graph neural networks on out-of-distribution graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).
- [6] Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. 2022. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence* 4, 2 (2022), 127–134.
- [7] Alexander Gammerman and Vladimir Vovk. 2007. Hedging predictions in machine learning. *Comput. J.* 50, 2 (2007), 151–163.
- [8] Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. 2022. Good: A graph out-of-distribution benchmark. *Advances in Neural Information Processing Systems* 35 (2022), 2059–2073.
- [9] Kan Guo, Yongli Hu, Yanfeng Sun, Sean Qian, Junbin Gao, and Baocai Yin. 2021. Hierarchical graph convolution network for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 151–159.
- [10] Yuxin Guo, Cheng Yang, Yulu Chen, Jixi Liu, Chuan Shi, and Junping Du. 2023. A Data-centric Framework to Endow Graph Neural Networks with Out-Of-Distribution Detection Ability. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 638–648.
- [11] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).
- [12] Kexin Huang, Ying Jin, Emmanuel Candès, and Jure Leskovec. 2023. Uncertainty quantification over graph with conformalized graph neural networks. *Advances in Neural Information Processing Systems* 36 (2023).
- [13] Weiwei Jiang and Jiayun Luo. 2022. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications* 207 (2022), 117921.
- [14] Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Zhengyang Mao, Hourun Li, Yiyang Gu, Yifang Qin, Nan Yin, Senzhang Wang, et al. 2024. A survey of graph neural networks in real world: Imbalance, noise, privacy and ood challenges. *arXiv preprint arXiv:2403.04468* (2024).
- [15] Gopal K Kanji. 2006. 100 statistical tests (2006), 1–256.
- [16] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.
- [17] Siddhartha Laghavarapu, Zhen Lin, and Jimeng Sun. 2024. CoDrug: Conformal Drug Property Prediction with Density Estimation under Covariate Shift. *Advances in Neural Information Processing Systems* 36 (2024).
- [18] Rikard Laxhammar and Göran Falkman. 2011. Sequential conformal anomaly detection in trajectories based on hausdorff distance. In *14th international conference on information fusion*. IEEE, 1–8.
- [19] Rikard Laxhammar and Göran Falkman. 2015. Inductive conformal anomaly detection for sequential detection of anomalous sub-trajectories. *Annals of Mathematics and Artificial Intelligence* 74 (2015), 67–94.
- [20] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2022. Ood-gnn: Out-of-distribution generalized graph neural network. *IEEE Transactions on Knowledge and Data Engineering* (2022).
- [21] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2022. Out-of-distribution generalization on graphs: A survey. *arXiv preprint arXiv:2202.07987* (2022).
- [22] Jia Li, Yu Rong, Hong Cheng, Helen Meng, Wenbing Huang, and Junzhou Huang. 2019. Semi-supervised graph classification: A hierarchical graph perspective. In *The World Wide Web Conference*. 972–982.
- [23] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.
- [24] Yixin Liu, Kaize Ding, Huan Liu, and Shirui Pan. 2023. Good-d: On unsupervised graph out-of-distribution detection. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 339–347.
- [25] Rongrong Ma, Guansong Pang, Ling Chen, and Anton van den Hengel. 2022. Deep graph-level anomaly detection by glocal knowledge distillation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 704–714.
- [26] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. 2021. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering* 35, 12 (2021), 12012–12038.
- [27] Alexandra Marin and Barry Wellman. 2011. Social network analysis: An introduction. *The SAGE handbook of social network analysis* (2011), 11–25.
- [28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).
- [29] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. 2019. Likelihood ratios for out-of-distribution detection. *Advances in neural information processing systems* 32 (2019).
- [30] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2019. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903* (2019).
- [31] Vickash Schwag, Mung Chiang, and Prateek Mittal. 2021. Ssd: A unified framework for self-supervised outlier detection. *arXiv preprint arXiv:2103.12051* (2021).
- [32] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research* 12, 9 (2011).
- [33] Hanne Stärk, Dominique Beaini, Gabriele Corso, Prudencio Tossou, Christian Dallago, Stephan Günnemann, and Pietro Liò. 2022. 3d infomax improves gnns for molecular property prediction. In *International Conference on Machine Learning*. PMLR, 20479–20502.
- [34] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. 2019. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000* (2019).
- [35] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.
- [36] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. 2005. *Algorithmic learning in a random world*. Vol. 29. Springer.
- [37] Fangxin Wang, Yuqing Liu, Kay Liu, Yibo Wang, Sourav Medya, and Philip S Yu. 2024. Uncertainty in Graph Neural Networks: A Survey. *arXiv preprint arXiv:2403.07185* (2024).
- [38] Luzhi Wang, Dongxiao He, He Zhang, Yixin Liu, Wenjie Wang, Shirui Pan, Di Jin, and Tat-Seng Chua. 2024. GOODAT: Towards Test-Time Graph Out-of-Distribution Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 15537–15545.
- [39] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. 2020. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020*. 1082–1092.
- [40] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. 2022. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence* 4, 3 (2022), 279–287.
- [41] Stanley Wasserman and Katherine Faust. 1994. Social network analysis: Methods and applications. (1994).
- [42] Man Wu, Shirui Pan, and Xingguan Zhu. 2020. Openwgl: Open-world graph learning. In *2020 IEEE international conference on data mining (icdm)*. IEEE, 681–690.
- [43] Qitian Wu, Yiting Chen, Chenxiao Yang, and Junchi Yan. 2023. Energy-based out-of-distribution detection for graph neural networks. In *International Conference on Learning Representations*.
- [44] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.
- [45] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).
- [46] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems* 33 (2020), 5812–5823.
- [47] Soroush H Zargarbashi, Simone Antonelli, and Aleksandar Bojchevski. 2023. Conformal prediction sets for graph neural networks. In *International Conference on Machine Learning*. PMLR, 12292–12318.
- [48] He Zhang, Bang Wu, Xingliang Yuan, Shirui Pan, Hanghang Tong, and Jian Pei. 2024. Trustworthy graph neural networks: aspects, methods, and trends. *Proc. IEEE* (2024).
- [49] Lingxiao Zhao and Leman Akoglu. 2021. On using classification datasets to evaluate graph outlier detection: Peculiar observations and new insights. *Big Data* (2021).
- [50] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph Neural Networks: A Review of Methods and Applications. *ArXiv abs/1812.08434* (2018). <https://api.semanticscholar.org/CorpusID:56517517>

929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044

1045 A APPENDIX

1046 A.1 Proof of Theorem 4.1

1048 Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, the set of graph data augmentations \mathcal{S} is a subset of all augmentations on the feature space \mathcal{X} .
 1049 Both \mathcal{S} and \mathcal{X} are equipped with respective σ -algebras $\mathcal{F}_{\mathcal{S}}$ and $\mathcal{F}_{\mathcal{X}}$. Sampling a random data augmentation $\Psi \in \mathcal{S}$ corresponds to a
 1050 measurable function $\tilde{\Psi} : \mathcal{X} \times \Omega \rightarrow \mathcal{S}$, which allocates an element
 1051 $\omega \in \Omega$ to a specific graph data augmentation $\Psi = \tilde{\Psi}(\cdot; \omega)$. The
 1052 distribution of $\tilde{\Psi}(\cdot; \omega)$ denoted as \mathbb{P}^{Ψ} is a probability measure on
 1053 $(\mathcal{S}, \mathcal{F}_{\mathcal{S}})$. Based on this, we can construct the product measure for
 1054 sampling i.i.d data augmentations for graphs.
 1055

1056 First, we have that the proper training set \mathcal{D}_{ptr} and the calibration set \mathcal{D}'_{cal} are i.i.d:
 1057

$$1060 \mathcal{D}_{ptr} \stackrel{\text{i.i.d.}}{\sim} \mathcal{D}'_{cal} \stackrel{\text{i.i.d.}}{\sim} \mathbb{P}^{in}, \quad (15)$$

1062 which follow the same distribution \mathbb{P}^{in} . When the test graph G_{te} is
 1063 from \mathbb{P}^{in} , we then have that
 1064

$$1065 (G_{te}, \Psi^{te,1:k}) \stackrel{\text{i.i.d.}}{\sim} \{(G_j^{in}, \Psi^{j,1:k}) | G_j^{in} \in \mathcal{D}'_{cal}\}_{j=n'+1}^n \stackrel{\text{i.i.d.}}{\sim} \mathbb{P}^{in} \times \mathbb{P}^{\Psi}. \quad (16)$$

1066 Meanwhile, the aggregated non-conformity score function $\widehat{V}(\cdot, \cdot; \cdot)$
 1067 relies on \mathcal{D}_{ptr} , the transformation function $F(\cdot, \cdot)$ in Eq.(2), and the
 1068 sampled data augmentations, i.e., $\Psi^{te,1:k} \cup \{\Psi^{j,1:k}\}_{j=n'+1}^n$. Hence,
 1069 the generated k non-conformity scores $s_{te}^{1:k}$ of G_{te} and those for
 1070 the graphs in \mathcal{D}'_{cal} , i.e., $\{s_j^{1:k}\}_{j=n'+1}^n$ are also i.i.d conditioned on
 1071 $\widehat{V}(\cdot, \cdot; \cdot)$, \mathcal{D}_{ptr} and $\Psi^{te,1:k} \cup \{\Psi^{j,1:k}\}_{j=n'+1}^n$.
 1072

1073 Due to the fact that $\widehat{V}(\cdot, \cdot; \cdot)$ is an aggregation function of the
 1074 generated non-conformity scores for guaranteeing that it is an
 1075 element-wise strictly increasing function, so these $n - n' + 1$ aggregated
 1076 non-conformity scores $(\hat{s}_{te}, \hat{s}_{n'+1}, \dots, \hat{s}_n)$ are also i.i.d:
 1077

$$1078 \hat{s}_{te} \stackrel{\text{i.i.d.}}{\sim} \hat{s}_{n'+1} \stackrel{\text{i.i.d.}}{\sim} \dots \stackrel{\text{i.i.d.}}{\sim} \hat{s}_n. \quad (17)$$

1080 Following the above formulation, we can regard these aggregated
 1081 non-conformity scores $(\hat{s}_{te}, \hat{s}_{n'+1}, \dots, \hat{s}_n)$ as i.i.d $n - n' + 1$ random
 1082 variables with their corresponding continuous densities. Summing
 1083 up the above, we can derive that
 1084

$$1085 |\{\hat{s}_{te} \leq \hat{s}_j | j = n' + 1, \dots, n\}| \sim \text{Uniform}(\{1, \dots, n - n' + 1\}). \quad (18)$$

1086 Therefore, the aggregated p -value in Eq.(12), i.e., \hat{p}_{te} is uniformly
 1087 distributed over $\{\frac{1}{n-n'+1}, \frac{2}{n-n'+1}, \dots, 1\}$, and this proof is com-
 1088 pleted. In addition, besides the strictly increasing constraint, we
 1089 can also add a small amount of random noise to these aggregated
 1090 non-conformity scores.
 1091

1092 A.2 Proof of Theorem 4.2

1093 The independence between \mathcal{D}'_{cal} , \mathcal{D}_{te} and $\widehat{V}(\cdot, \cdot; \cdot)$ is easily to be
 1094 satisfied. If the test graph G_{te} is sampled from \mathbb{P}^{in} , then its corre-
 1095 sponding aggregated p -value \hat{p}_{te} follows a discrete uniform distri-
 1096 bution based on Theorem 4.1, i.e., \hat{p}_{te} is uniformly distributed over
 1097 $\{\frac{1}{n-n'+1}, \frac{2}{n-n'+1}, \dots, 1\}$. The probability of the event that $G_{te} \sim \mathbb{P}^{in}$
 1098

1099 Algorithm 1: Training and inference procedures of CGOD.

```

1100 1: #Training Procedure:
1101 2: Input: training set  $\mathcal{D}_{tr}$ .
1102 3: Output: learned adaptive data augmentations.
1103 4: Split  $\mathcal{D}_{tr}$  into  $\mathcal{D}_{ptr}$  and  $\mathcal{D}_{cal}$ , and further split  $\mathcal{D}_{cal}$  into
1104    $\mathcal{D}_{cal-tr}$  and  $\mathcal{D}'_{cal}$ .
1105 5: Pre-train an OOD detection model on  $\mathcal{D}_{ptr}$ .
1106 6: while not done do
1107   7:   for randomly sampling a graph  $G_j^{in} \in \mathcal{D}_{cal-tr}$  do
1108     8:     for each data augmentation  $\Psi^{j,i} \in \Psi^{j,1:k}$  do
1109       9:         Generate the augmented representation  $\hat{g}_j^i$  by Eq.(7-8).
1110       10:        Calculate its non-conformity score  $s_j^i$  by Eq.(3).
1111     11:   end for
1112   12:   Calculate the loss function  $\mathcal{L}$  in Eq.(9), and update model
1113     parameters by Adam optimizer.
1114 13: end for
1115 14: end while
1116 15: #Inference Procedure:
1117 16: Input: test dataset  $\mathcal{D}_{te}$ ; learned adaptive data augmentations.
1118 17: Output: detection results of  $\mathcal{D}_{te}$ .
1119 18: for each test graph  $G_{te} \in \mathcal{D}_{te}$  do
1120   19:   for each data augmentation  $\Psi^{te,i} \in \Psi^{te,1:k}$  do
1121     20:     Generate the augmented representation  $\hat{g}_{te}^i$  by Eq.(7-8).
1122     21:     Calculate its non-conformity score  $s_{te}^i$  by Eq.(3).
1123   22:   end for
1124   23:   Calculate its aggregated non-conformity score  $\hat{s}_{te}$  by Eq.(6).
1125   24:   Derive its  $p$ -value  $\hat{p}_{te}$  with  $\mathcal{D}'_{cal}$  by Eq.(12).
1126   25:   If  $\mathbb{I}(\hat{p}_{te} < \epsilon)$ :  $G_{te}$  is an OOD graph; otherwise, it is not.
1127 26: end for
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
```

and $\hat{p}_{te} < \epsilon$ is

$$1137 \Pr(\hat{p}_{te} < \epsilon | G_{te} \sim \mathbb{P}^{in}) = \sum_{j=1}^{(n-n'+1)\epsilon} \frac{1}{n - n' + 1} \quad (19) \\ 1138 = \frac{\lfloor (n - n' + 1)\epsilon \rfloor}{n - n' + 1} \quad 1139 \\ 1140 \leq \epsilon. \quad 1141 \\ 1142 \quad 1143$$

If \hat{p}_{te} is smaller than ϵ , then G_{te} is recognized as the OOD graph.
 Hence, if $G_{te} \sim \mathbb{P}^{in}$, the probability of the event that CGOD falsely
 recognizes G_{te} as the OOD graph is upper bounded by ϵ .

A.3 Pseudo-code Description

Algorithm 1 illustrates the training and inference procedures of CGOD. From it, we can see that our model follows a simple computational process.

A.4 Dataset Description

We adopt six pairs of real-world datasets from three domains:

- Social networks datasets: **IMDB-M** & **IMDB-B**. They are movie collaboration datasets, where nodes denote actors and an edge is drawn between two actors if they appear in the same movie.

- Molecule datasets: **ClinTox & LIPO**, **BBBP & BACE**, **Esol & MUV**, and **AIDS & DHFR**. For each dataset, nodes denote atoms and an edge connecting two nodes is a chemical bond of molecule graphs.
- Bioinformatics datasets: **ENZYME & PROTEIN**. They are macromolecule datasets, where nodes denote secondary structure elements and an edge represents that two nodes are neighbors along the amino acid sequence or one of three nearest neighbors in space.

A.5 Baseline Description

We compare our method with the following strong baselines:

- WL** [32] represents the Weisfeiler-lehman graph kernel, which processes a graph by re-labeling each node with a new label compressed from a multiset label consisting of its original label and the sorted labels of its neighbors.
- GIN** [45] is a representative work of supervised GNNs, which designs the new neighbor aggregation and readout functions to improve GNN expressiveness.
- InfoGraph** [34] is a representative work of unsupervised GNNs. It designs a self-supervised objective to maximize the mutual information between the graph-level representation and the sub-structure representations.
- OCGIN** [49] is a one-class anomaly detection model for recognizing abnormal graphs, which adopts GIN and SVDD (support vector data description) as the encoder and the objective at the output layer, respectively.
- GLocalKD** [25] is a popular method of graph-level anomaly detection. It performs the joint random distillation of graph and node representations to capture local- and global-patterns to detect anomalous graphs.
- GOOD-D** [24] is the first work to solve graph-level OOD detection, which proposes a hierarchical graph contrastive learning framework to identify OOD graphs via different granularity scores.
- AAGOD** [10] is a post-hoc method which designs an amplifier as prompts for recognizing the key information from the graph structure, enabling a well-trained GNN to achieve OOD detection.
- GOODAT** [38] considers how to achieve graph-level OOD detection at the test time. It leverages the information bottleneck to learn informative subgraphs for enlarging the gap between ID and OOD graphs.

We use the implementations of GIN and InfoGraph in a well-known GNN library³. For WL⁴, OCGIN⁵, GLocalKD⁶, GOOD-D⁷, AAGOD⁸ and GOODAT⁹, we adopt their public implementations and adapt them into our training and inference pipelines. According to the relevant literature, we respectively arm WL, GIN and InfoGraph with the local outlier factor (LOF), SSD and SSD as the transformation function.

³<https://www.pyg.org/>

⁴<https://ysig.github.io/GraKeL/0.1a8/>

⁵<https://github.com/LingxiaoShawn/GLOD-Issues>

⁶<https://github.com/RongrongMa/GLocalKD>

⁷<https://github.com/yixinliu233/G-OOD-D>

⁸<https://github.com/BUPT-GAMMA/AAGOD>

⁹<https://github.com/Ee1s/GOODAT>

Table 4: Search intervals of hyper-parameters.

Hyper-parameter	Search Interval
learning rate lr	{0.0001, 0.0005, 0.001, 0.005, 0.01}
augmentation number k	{2, 3, 4, 5, 6}
discretization ratio ξ	{0.1, 0.2, 0.3, 0.4, 0.5}
harmonic factor λ	{0.2, 0.4, 0.6, 0.8, 1.0}

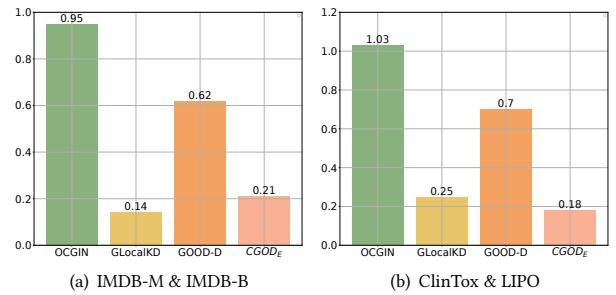


Figure 5: Training time comparison (in seconds).

A.6 Hyper-parameter Setting

As described in Section 5.1.3, there are four hyper-parameters in our method: lr, k , ξ and λ . Their corresponding search intervals are given in Table 4.

A.7 Metric Description

- AUC** refers to area under the receiver operating characteristic curve, which summarizes the ROC curve into a single value to demonstrate the model performance with different thresholds.
- AUPR** refers to area under the precision-recall curve. Compared with AUC, AUPR can adjust different positive and negative class rates to alleviate the problem of imbalanced class.
- FPR95** is the false positive rate at 95% true positive rate. It is a commonly used metric to evaluate the performance of classification models, especially in the tasks of binary classification and anomaly or OOD detection.
- FPR** is a statistical measure that represents the proportion of actual negative samples that are incorrectly classified as positive by a detection model.

A.8 Runtime Comparison

We compare the training runtime between CGOD_E and three end-to-end detection methods including OCGIN, GLocalKD and GOOD-D in an epoch. The experiments are conducted on a Linux server with Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz, 128G RAM and NVIDIA Tesla V100. IMDB-M & IMDB-B and ClinTox & LIPO are selected datasets. Figure 5 shows the concrete results. From it, we conclude that CGOD keeps a good training efficiency.