Graph Out-of-Distribution Detection via Test-Time Calibration with Dual Dynamic Dictionaries

Yue Hou[1,2], **Ruomei Liu**[1], **Yingke Su**[2], **Junran Wu**[1*], **Ke Xu**[1,3]
[1]State Key Laboratory of Complex & Critical Software Environment, Beihang University, Beijing, China
[2]Shen Yuan Honors College, Beihang University, Beijing, China
[3]Key Laboratory of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin, China
{hou_yue, rmliu, suyingke, wu_junran, kexu}@buaa.edu.cn

Abstract

A key challenge in graph out-of-distribution (OOD) detection lies in the absence of ground-truth OOD samples during training. Existing methods are typically optimized to capture features within the in-distribution (ID) data and calculate OOD scores, which often limits pre-trained models from representing distributional boundaries, leading to unreliable OOD detection. Moreover, the latent structure of graph data is often governed by multiple underlying factors, which remains less explored. To address these challenges, we propose a novel test-time graph OOD detection method, termed **BaCa**, that calibrates OOD scores using dual dynamically updated dictionaries without requiring fine-tuning the pre-trained model. Specifically, BaCa estimates graphons and applies a mix-up strategy solely with test samples to generate diverse boundary-aware discriminative topologies, eliminating the need for exposing auxiliary datasets as outliers. We construct dual dynamic dictionaries via priority queues and attention mechanisms to adaptively capture latent ID and OOD representations, which are then utilized for boundary-aware OOD score calibration. To the best of our knowledge, extensive experiments on real-world datasets show that BaCa significantly outperforms existing state-of-the-art methods in OOD detection.

1 Introduction

With remarkable success across various domains, deep learning models are widely known to make overconfident predictions on inputs that differ from the training distribution. This often leads to misclassifying out-of-distribution (OOD) samples as in-distribution (ID) classes. OOD detection (Schreyer et al. 2017; Zhou, Liu, and Chen 2021) aims to identify anomalous inputs and is essential for the safe deployment of models in open-world settings. However, performing OOD detection on graph-structured data is particularly challenging due to the non-Euclidean geometry and complex topology. These challenges highlight the need for robust representation learning methods for graphs (Wu et al. 2022a,b, 2023; Wu, Chen, and Li 2024; Wu, Ooi, and Xu 2025; Hou et al. 2025c,a), especially in the presence of previously unseen samples (Hou et al. 2024).

Recent efforts (Guo et al. 2023; Liu et al. 2023; Hou et al. 2025b) in graph OOD detection fall into two main categories: **(1) End-to-end methods** that optimize an OOD-specific graph neural network (GNN) (Kipf and Welling 2017; Xu et al. 2019) from scratch using only unlabeled ID data, and **(2) Post-hoc approaches** (Guo et al. 2023; Wang et al. 2024) that apply fine-tuned detectors on well-trained GNNs. These methods typically define an OOD score function based on the model's output logits or latent features. A notable extension of end-to-end training includes Outlier Exposure (OE) (Junwei et al. 2024), which leverages auxiliary OOD data during training to encourage the model to output flattened distributions for anomalous inputs. However, OE-based methods assume access to external OOD datasets, which violates the standard assumption of training solely on ID data. Additionally, GOODAT (Wang et al. 2024) introduces a more practical test-time setting by directly modifying test samples without altering the pre-trained model. However, it still requires optimizing a learnable graph masker during inference, which may limit stability in real-time applications.

Despite these advancements, several notorious challenges remain underexplored. Pretrained GNNs, optimized solely on ID data, often struggle to distinguish OOD samples when their representations lie close to the ID manifold, such as when sharing similar topological structures. Moreover, the diversity of latent structural factors makes it difficult for such models to generalize well to unseen data. This limitation manifests in the form of overlapping score distributions between ID and OOD samples (▷ Figure 1(a)), particularly near the decision boundary. We argue that *the key to effective test-time OOD detection lies in modeling the distributional boundary between ID and OOD samples*, especially in identifying those ambiguous cases at the boundary.

Intuitively, if a test sample is more OOD-like than the least OOD sample near the ID boundary, it should be classified as OOD; similarly, if it is more ID-like than the least ID-like OOD sample, it should be treated as ID. Therefore, **a natural solution** is to calibrate OOD scores such that the overlap between ID and OOD samples is reduced (▷ Figure 1(b)), enhancing their separability at the distributional boundary. Thus, this problem is highly challenging in:

- How to model the distributional boundary without relying on training ID or auxiliary OOD data?
- How to enlarge the gap between ID and OOD data distributions through OOD score calibration?

*Corresponding authors

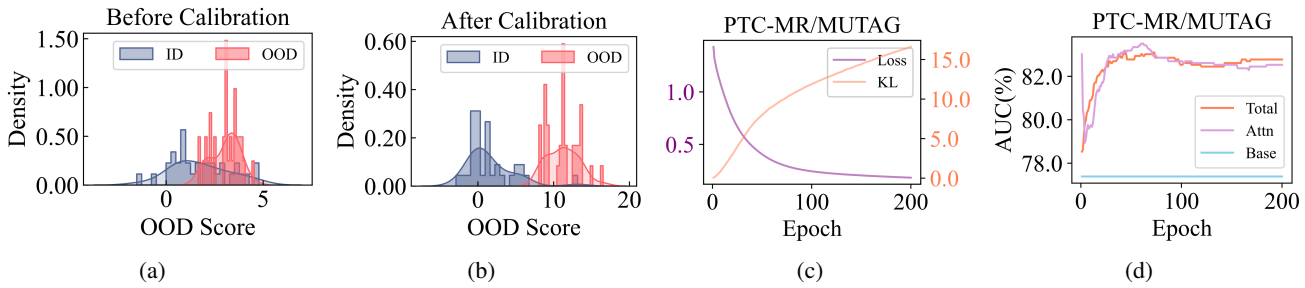


Figure 1: An example of OOD score distribution and detection performance evolution over test-time iterations on the PTC/-MUTAG dataset pair. **(a)** Before calibration, we dynamically feed the lower left tail of OOD score distribution into the OOD dictionary and the higher right tail of ID score distribution into the ID dictionary via two priority queues. **(b)** After calibration, the overlap between the ID and OOD score distributions is significantly reduced. **(c)** KL divergence and the loss of attention-based trainable parameters during the first 200 iterations. **(d)** AUC over the first 200 iterations, where *Total*, *Attn*, and *Base* denote our full method with S_{BaCa}, attention-based calibration with S_{Attn}, and the pre-trained baseline with S_{Pre}, respectively.

To address these challenges, we propose a novel framework, **B**oundary-**a**ware **Ca**libration for test-time graph OOD detection, termed **BaCa**. Our BaCa solves the aforementioned challenges and achieves adaptive OOD score calibration target through the following design. **Firstly**, to model ID and OOD distributional boundaries, we perform partitioning based on initial judgment from the pre-trained model, and estimate graphons separately for ID and OOD subgroups. To capture diverse latent topological factors, we apply a graphon mixup strategy to generate synthetic samples that enhance the expressiveness of discriminative typologies and improve robustness, particularly in the early stages of detection. **Then**, we propose the adaptive score calibration for the separation between ID and OOD distributions. Specifically, BaCa continuously collects synthetic latent representations during test time, especially those near the decision boundary, such as ID samples with OOD-like characteristics and vice versa, and dynamically inserts them into ID and OOD dictionaries maintained as priority queues. By incorporating a learnable attention mechanism, we adaptively calibrate OOD scores in a boundary-aware manner, reducing distributional overlap and ambiguity. We utilize KL divergence to measure the distributional difference of OOD scores between ID and OOD samples. As iteration progresses (shown in Figure 1(c)), the KL divergence gradually increases, and the calibrated AUC consistently improves over the pre-trained encoder (see Figure 1(d)). Extensive experiments on real-world graph datasets demonstrate the superiority of BaCa over state-of-the-art (SOTA) baselines. Notably, under the same test-time setting, BaCa outperforms GOODAT (Wang et al. 2024) on all 10 datasets, with an average AUC improvement of 8.37%, especially on ClinTox/LIPO with gains up to 20.11%. Contributions of this paper are as follows:

- We propose BaCa, a novel boundary-aware OOD score calibration framework for test-time graph OOD detection. Unlike previous approaches, it does not require prior outlier samples from auxiliary data or pre-trained model fine-tuning.
- We generate diverse samples with discriminative typology and develop dual dynamic dictionaries maintained as priority queues, enabling adaptive OOD score calibration.
- Extensive experiments validate the effectiveness of BaCa, demonstrating the superior performance over SOTA baselines in unsupervised OOD detection.

2 Notations and Preliminaries

Before formulating the research problem, we first provide some necessary notations. Let $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ represent a graph, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. The node features are represented by the feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, where $n = |\mathcal{V}|$ is the number of nodes and d is the feature dimension. The structure information can also be described by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, so a graph can be alternatively represented by $G = (\mathbf{A}, \mathbf{X})$.

Test-time Graph-level OOD Detection. For graph-level OOD detection at test-time, following GOODAT (Wang et al. 2024), we consider an unlabeled ID dataset $\mathcal{D}^{id} = \{G_1^{id}, \cdots, G_{N_1}^{id}\}$ where graphs are sampled from distribution $\mathbb{P}^{id}$ and an OOD dataset $\mathcal{D}^{ood} = \{G_1^{ood}, \cdots, G_{N_2}^{ood}\}$ sampled from a different distribution $\mathbb{P}^{ood}$. Given a test sample G from $\mathcal{D}_{test}^{id} \cup \mathcal{D}_{test}^{ood}$, test-time graph OOD detection aims to detect whether G originates from $\mathbb{P}^{id}$ or $\mathbb{P}^{ood}$ utilizing a GNN encoder f pre-trained on ID graphs $\mathcal{D}_{train}^{id} \subset \mathcal{D}^{id}$. Specifically, the objective is to learn an OOD detector $D(\cdot, \cdot)$ that assigns an OOD detection score $S = D(f, G)$, with a higher S indicating a greater probability that G is from $\mathbb{P}^{ood}$ (note that $\mathcal{D}_{test}^{id} \cap \mathcal{D}_{train}^{id} = \emptyset$, $\mathcal{D}_{test}^{id} \subset \mathcal{D}^{id}$, and $\mathcal{D}_{test}^{ood} \subset \mathcal{D}^{ood}$). It should be emphasized that graph data sourced from $\mathbb{P}^{in}$ and $\mathbb{P}^{out}$ might fall into multiple categories. However, in the unsupervised graph-level OOD task, the model is not provided with any category-specific labels.

Graphon. A graphon is a symmetric, bounded, and measurable function widely used to model the generative process of graphs (Airoldi, Costa, and Chan 2013; Lovász 2012). It serves as a limit object for sequences of dense graphs and captures the probability of edge existence between latent node representations in a continuous domain. Formally, a graphon is defined as a two-dimensional symmetric Lebesgue measurable function $W : \Omega^2 \to [0, 1]$, where Ω is a probability space, typically taken as the unit interval $[0, 1]$. The value

$W(x, y)$ indicates the probability of an edge between two nodes associated with latent positions x and y in Ω. Graphons provide a principled framework for capturing the structural characteristics of graphs beyond discrete representations. By sampling latent variables from Ω and forming edges according to $W(x, y)$, one can generate synthetic graphs that share topological properties with the original graph distribution.

In real-world applications, the closed-form expression of the underlying graphon is generally unavailable and must be approximated from observed graphs. A common estimation approach is to approximate the graphon using a step function, which can be represented as a matrix $W \in [0, 1]^{N\times N}$, where N corresponds to the number of aligned latent positions or nodes. This matrix-form approximation enables efficient sampling of synthetic graphs and supports downstream tasks such as generation, augmentation, and structure comparison. In this work, we adopt the USVT estimator (Chatterjee 2015) due to its theoretical guarantees and empirical effectiveness.

3 Methodology

In this section, we elaborate on the proposed adaptive redundancy-aware OOD score calibration for test-time graph OOD detection, termed **BaCa**.

3.1 Overall Framework

In general, the basic objective in OOD detection for obtaining a GNN encoder f is defined as:

$$\min_{f} \mathbb{E}_{G\sim\mathcal{D}^{\text{in}}_{\text{train}}} \mathcal{L}_{\text{Pre}}(f; G), \tag{1}$$

where $\mathcal{L}_{\text{Pre}}$ denotes the pretraining loss function. For end-to-end OOD detection methods (Liu et al. 2023), the OOD score of a test sample is typically derived directly from the output of this pre-trained model. However, the initial judgment made by the pre-trained model regarding a sample's distribution may be unreliable, due to its lack of exposure to true OOD samples. This can lead to inaccurate OOD scores, especially near the boundary between ID and OOD distributions.

To enable test-time OOD score calibration without updating the pre-trained model, we identify two key challenges: **(C1)** how to effectively model the boundary between ID and OOD samples, and **(C2)** how to design a robust score calibration mechanism. To address **(C1)**, we first partition test samples into two groups based on the initial score estimation, and then estimate graphons separately for each group. A graphon mixup strategy is applied within each group to generate diverse discriminative typologies that enhance the representation of boundary distributions. To address **(C2)**, we maintain dual dynamic dictionaries using priority queues and perform adaptive score calibration via attention mechanisms. The overall pipeline of BaCa is illustrated in Figure 2.

3.2 Boundary-Aware Latent Pattern Modeling

Subgroup Partitioning Based on Initial Judgment. We utilize the pre-trained model f to extract the representation of each test sample $G \in \mathcal{D}_{\text{test}}$ and compute its initial OOD score $S_{\text{Pre}} = \mathcal{L}_{\text{Pre}}(f; G)$ using Eq. (1). This score serves as an initial judgment of the sample's distributional status.

Graphon Estimation for Latent Factor Construction. To capture the structural differences among test samples and model their distributional variation, we employ graphons to estimate the characteristic topologies in different subsets of graphs. A graphon $W : \Omega^2 \to [0, 1]$ defines the probability of edge existence between any two latent positions sampled from a base space Ω. Given a graphon, a random graph can be generated as follows:

$$\begin{aligned} v_n &\sim \text{Uniform}(\Omega), \quad \text{for } n = 1, \ldots, N, \\ a_{nn'} &\sim \text{Bernoulli}(W(v_n, v_{n'})), \quad \text{for } n, n' = 1, \ldots, N, \end{aligned} \tag{2}$$

where v_n denotes the latent position of node n, and $a_{nn'}$ indicates whether an edge exists between nodes n and n'. This process results in an adjacency matrix $\mathbf{A} \in \{0, 1\}^{N\times N}$, which defines the structure of a sampled graph $\tilde{G}(\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$ with $\tilde{\mathcal{V}} = \{1, \ldots, N\}$ and $\tilde{\mathcal{E}} = \{(n, n') \mid a_{nn'} = 1\}$.

Since the true graphon is an unknown function and cannot be recovered in closed form, we adopt the step-function approximation commonly used in prior work (Chatterjee 2015; Han et al. 2022; Xu et al. 2021). A step-function graphon $W^P : [0, 1]^2 \to [0, 1]$ is expressed as: $W^P(x, y) = \sum_{n,n'=1}^{N} w_{nn'} \mathbb{1}_{\mathcal{P}_n\times\mathcal{P}_{n'}}(x, y)$, where $\mathcal{P} = (\mathcal{P}_1, \ldots, \mathcal{P}_N)$ is a uniform partition of $[0, 1]$ into N intervals, and $w_{nn'} \in [0, 1]$ represents the estimated connection probability between intervals $\mathcal{P}_n$ and $\mathcal{P}_{n'}$. The indicator function $\mathbb{1}_{\mathcal{P}_n\times\mathcal{P}_{n'}}(x, y)$ equals 1 if $(x, y) \in \mathcal{P}_n \times \mathcal{P}_{n'}$ and 0 otherwise.

Based on S_{Pre}, we partition all samples in the current test-time batch into two mutually exclusive subsets:$\mathcal{D}^{\text{batch}}_{\text{test}} = \mathcal{C}^{\text{id}} \cup \mathcal{C}^{\text{ood}}$, where $\mathcal{C}^{\text{ood}} = \{W_{i,m}\}_{\tilde{y}=1}^{M}$ and $\mathcal{C}^{\text{id}} = \{W_{i,m'}\}_{\tilde{y}=0}^{M'}$, with M and M' denoting the number of samples initially predicted as OOD and ID, respectively. This partitioning allows ID and OOD candidate samples to be processed separately during downstream graphon mixup and dictionary construction, relying only on the pre-trained model and soft predictions, without requiring ground-truth supervision.

Graphon Mixup for Discriminative Typology Expansion. After the partitioning step, test-time samples are divided into two disjoint subgroups and separate sets of graphons are estimated to model the structural patterns within each group. However, the discriminative topological factors responsible for distributional differences are often multifaceted rather than governed by a single mode. Moreover, the estimated graphons may not sufficiently capture structures near the boundary regions, leading to unstable detection and poor generalization, especially in early-stage inference.

To alleviate this issue, we propose a graphon-level mixup strategy performed within each subgroup (i.e., among ID graphons and among OOD graphons separately). This approach interpolates between graphons derived from structurally distinct samples within the same class, thereby enhancing internal structural diversity and enriching the boundary space. Formally, let W_i and W_j be two graphons estimated from the same group (e.g., $\mathcal{C}^{\text{ood}}$). We define their mixed graphon as:

$$W_s = \lambda W_i + (1 - \lambda)W_j, \quad \lambda \in [0, 1], \tag{3}$$

where λ is a balancing hyperparameter. The resulting W_s lies in the convex hull of W_i and W_j and can be interpreted as a

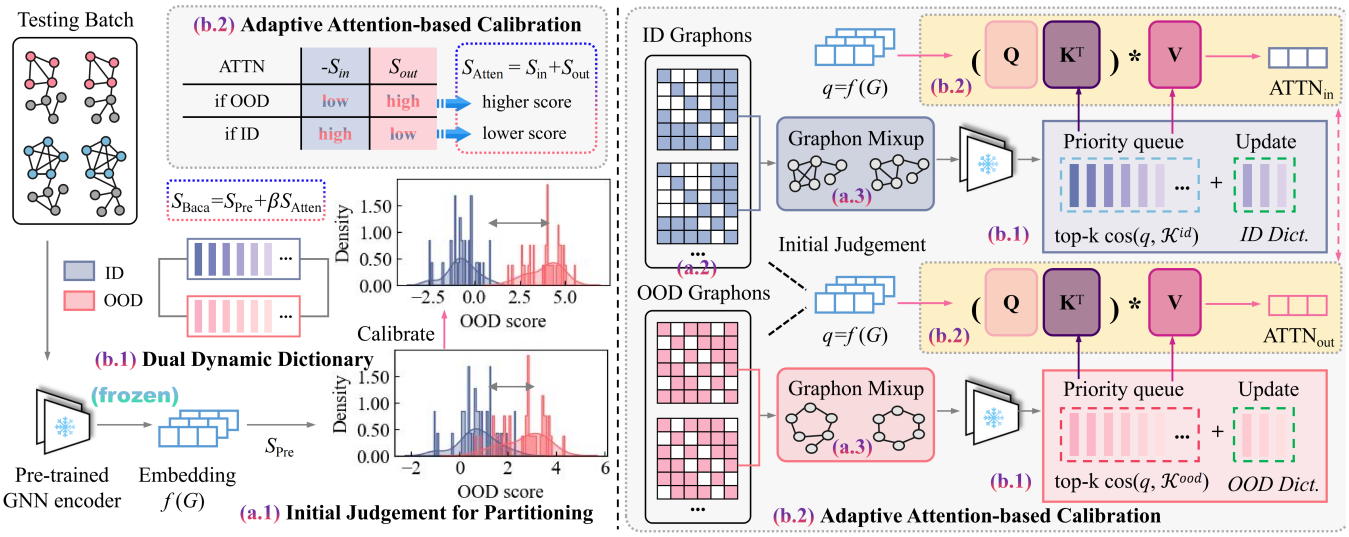


Figure 2: Overview of our proposed BaCa framework. **(a.1)** Given a pre-trained GNN encoder and test samples, we first compute the initial OOD scores and partition the samples into two preliminary subgroups. **(a.2–a.3)** Within each subgroup, diverse discriminative typologies are generated via graphon mixup and stored in dual dynamic dictionaries maintained as priority queues. **(b.1–b.2)** The priority queue–based dictionaries are used to support adaptive, attention-based score calibration. **(b.3)** The adaptive attention module is optimized during inference to compute the final calibrated OOD score.

new generative process that inherits structural traits from both sources. Sampling from W_s generates graphs located in the interpolated region between the two subgroups, which helps bridge discontinuities in the estimated structure space and populate low-density zones near the ID/OOD boundary. To formalize this notion, we introduce the concept of a *discriminative typology*, which characterizes the essential structural properties that determine a graph's subgroup membership.

Definition 1 (**Discriminative Typology**). *Given a graph G, a discriminative typology T_G is a structural pattern that reflects the most representative and characteristic features of G with respect to its latent distribution, i.e., ID or OOD.*

Intuitively, typologies summarize structural traits that differentiate subgroups within a distribution and intuitively capture the generative semantics of graph samples. Our hypothesis is that graphons estimated from a group of graphs encode their typological characteristics, and linear combinations of such graphons preserve essential features from the source groups.

Theorem 1. *Let W_G and W_H be graphons estimated from two subgroups G and H of the same distribution type (i.e., both ID or both OOD). Let the interpolated graphon be defined as $W_s = \lambda W_G + (1-\lambda) W_H$, where $\lambda \in [0,1]$. Then, for any discriminative typology T_G and T_H:*

$$\begin{aligned} |t(T_G, W_s) - t(T_G, W_G)| &\le (1-\lambda)\cdot \delta_{GH}, \\ |t(T_H, W_s) - t(T_H, W_H)| &\le \lambda \cdot \delta_{GH}, \end{aligned} \tag{4}$$

where $t(F,W) = \int_{[0,1]^{|\tilde{\mathcal{V}}|}} \prod_{(i,j)\in|\tilde{\mathcal{E}}|} W(x_i,x_j) \prod_{i\in|\tilde{\mathcal{V}}|} dx_i$ denotes the homomorphism density of structure F in graphon W, and $\delta_{GH} = \|W_G - W_H\|_\square$ is the cut norm distance between W_G and W_H.

Remark: The theorem indicates that the mixed graphon W_s retains the key structural characteristics from both W_G and W_H, with bounded deviations depending on the mixing ratio λ and the structural dissimilarity between the original subgroups. Since W_G and W_H originate from the same distribution (either ID or OOD), the synthetic graphs sampled from W_s remain typologically consistent with their source distribution, enabling meaningful boundary exploration without compromising distributional integrity. Through this graphon-level mixup procedure, we can generate an arbitrary number of graphs at test-time, filling in the low-density regions between known ID and OOD modes and improving the robustness of boundary estimation.

Random Size Sampling for Boundary Diversity. To enhance structural diversity and better approximate the true distributional variability among graphs, we introduce a random size-based sampling strategy. Although an interpolated graphon $W_s \in [0,1]^{N\times N}$ allows infinite graph generation, naive sampling typically results in graphs of size close to N, limiting diversity. To mitigate this, we randomly select a target size $r \in [2, N]$ and generate the graph from sampled graphon $W_s' \in [0,1]^{r\times r}$. The existence of an edge between nodes i and j is determined by sampling from a Bernoulli distribution in Eq. (2) with the parameter $W_s'(i,j)$.

3.3 Adaptive Calibration via Dual Dynamic Dictionary

Dual Priority Queues for Dynamic Dictionary Maintenance. As test-time samples arrive in successive batches, the boundary between ID and OOD samples evolves dynamically. To adaptively track this boundary, we maintain two separate dynamic dictionaries for ID and OOD samples, each implemented as a priority queue. These dictionaries are updated online according to the sample's relative position to the ID/OOD boundary, as estimated from the OOD score.

Intuitively, if a sample is more OOD-like than the least OOD sample (*i.e.*, on the boundary side of the OOD distribution), it is added to the OOD dictionary; similarly, if a sample is more ID-like than the least ID-like sample, it is added to the ID dictionary. In practice, this means that the right tail of the ID score distribution, where ID samples are most similar to OOD, is inserted into the ID dictionary, while the left tail of the OOD score distribution, where OOD samples are most similar to ID, is inserted into the OOD dictionary. We refer to these as *latent ID features* and *latent OOD features*, respectively, as they represent boundary-side discriminative typologies. The initial dictionaries are constructed based on the pre-trained model's score, and as test-time progresses, these dictionaries are continuously enriched by newly generated synthetic samples from graphon mixup, which increases the diversity of latent patterns near the boundary.

During inference, the ID and OOD dictionaries are maintained as fixed-length priority queues. This design allows encoded features from previous batches to be reused, decoupling the dictionary size from the mini-batch size. The queue size l is a tunable hyperparameter and enables storage of more diverse and representative structures. Taking the OOD dictionary as an example, we denote it as $\mathcal{K}_l^{\text{ood}} = \{k_1^{\text{ood}}, k_2^{\text{ood}}, \ldots, k_{l'}^{\text{ood}}\}$ with $l \geq l'$. New candidates are added to the queue only if their OOD score exceeds that of the front element. In this setup, the front of the OOD queue always corresponds to the sample closest to the ID/OOD boundary. Similarly, we maintain the ID dictionary $\mathcal{K}_l^{\text{id}}$ using the same mechanism, where the front represents the least ID-like inlier.

In summary, we dynamically feed the lower left tail of the OOD score distribution into the OOD dictionary, and the higher right tail of the ID score distribution into the ID dictionary. This dual-priority-queue mechanism ensures that both dictionaries retain the most representative and boundary-sensitive graphon-derived features, allowing for adaptive and efficient modeling of the evolving ID/OOD structure during test time.

Adaptive Attention-based Score Calibration. To enhance calibration adaptively to capture boundary-aware representations, we introduce an attention mechanism over the ID and OOD dictionaries. Since attention scores are often concentrated on a small subset of keys, we compute attention over only the top-$\mathbb{K}$ most relevant entries, improving efficiency and reducing noise from irrelevant matches. Taking OOD dictionary as an example, we first derive the query $q = f(G)$ for a test sample $G \in \mathcal{D}_{test}$ and compute the cosine similarity $\cos(k_i^{ood}, q)$ with each key k_i^{ood} in OOD dictionary $\mathcal{K}_{n'}^{ood}$. Then, we denote the sorted list of these similarities in ascending order as $\cos(k_{(1)}^{ood}, q) \leq \cos(k_{(2)}^{ood}, q) \leq \cdots \leq \cos(k_{(n')}^{ood}, q)$. The top $\mathbb{K}$ entries are selected to form the candidate set $\hat{\mathcal{K}}_{(:\mathbb{K})}^{\text{ood}}$. We construct the attention components as:

$$\begin{aligned} \mathbf{Q} = q\mathbf{W}_Q, \mathbf{K} = \hat{\mathcal{K}}_{(:\mathbb{K})}^{\text{ood}}\mathbf{W}_K, \mathbf{V} = \hat{\mathcal{K}}_{(:\mathbb{K})}^{\text{ood}}\mathbf{W}_V, \\ \text{ATTN}_{\text{out}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})\mathbf{V}, \end{aligned} \tag{5}$$

where $\mathbf{W}_Q \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{\mathbb{K}\times d}$ are learnable matrices. The calibrated OOD score based on OOD dictionary is then defined as:

$$S_{\text{out}}(G) = \text{ATTN}_{\text{out}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}). \tag{6}$$

The complete OOD dictionary includes both the priority queue and memory bank: $\mathcal{K}_{\text{total}}^{\text{ood}} = \mathcal{K}_l^{\text{ood}} \cup \mathcal{K}_{\text{mb}}^{\text{ood}}$, where $\mathcal{K}_{\text{mb}}^{\text{ood}}$ denotes a fixed-size memory buffer. Similarly, we calculate the negative cosine similarity between the query and each key in the ID dictionary:

$$S_{\text{in}}(G) = -\text{ATTN}_{\text{in}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}), \tag{7}$$

where $\mathbb{K}$-th largest cosine similarity is selected, and the ID dictionary is composed as $\mathcal{K}_{total}^{id} = \mathcal{K}_l^{id} \cup \mathcal{K}_{mb}^{id}$. The final boundary-aware calibrated score is then given by:

$$S_{\text{Atten}}(G) = S_{\text{in}}(G) + S_{\text{out}}(G), \tag{8}$$

If G is an ID sample, it will typically have high similarity with the ID dictionary and low similarity with the OOD dictionary, resulting in a low $S_{\text{Attn}}(G)$. Conversely, OOD samples yield higher values. This calibration mechanism encourages a clearer separation of score distributions between ID and OOD samples by modeling diverse features and structural boundaries. We integrate $S_{Attn}(G)$ into the overall objective:

$$S_{\text{BaCa}} = S_{\text{Pre}} + \beta \cdot S_{\text{Attn}}(G), \tag{9}$$

where β is a trade-off hyperparameter controlling the influence of test-time similarity calibration.

Training Objective. To optimize the learnable parameters $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$, we employ a dual binary cross-entropy loss that supervises the attention-based similarity scores. Formally, the training objective is defined as:

$$\begin{aligned} \mathcal{L} = &- \mathbb{E}_{\mathcal{K}^{id}} \left[\log(\text{ATTN}_{\text{in}}) + \log(1 - \text{ATTN}_{\text{out}})\right] \\ &- \mathbb{E}_{\mathcal{K}^{ood}} \left[\log(1 - \text{ATTN}_{\text{in}}) + \log(\text{ATTN}_{\text{out}})\right]. \end{aligned} \tag{10}$$

This loss encourages ID samples to yield high attention scores with ID dictionary and low scores with OOD dictionary, while OOD samples are trained to exhibit the opposite pattern.

Computational Complexity Analysis. For graph generation, suppose we aim to generate l graphs with N nodes. The complexity is $\mathcal{O}(lN)$ for node sampling and $\mathcal{O}(lN^2)$ for edge construction, resulting in a total complexity of $\mathcal{O}(lN^2)$. For dynamic dictionary construction, BaCa relies solely on dot-product operations between test-time samples and stored entries. This is equivalent to adding a linear transformation layer, introducing a per-sample complexity of $\mathcal{O}(dl)$, where d is the feature dimension and l denotes the priority queue size. Updating the priority queue has a complexity of $\mathcal{O}(\log l)$ per insertion. For the attention-based score calibration, given query $\mathbf{Q} \in \mathbb{R}^{1\times d}$ and key-value matrices $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{\mathbb{K}\times d}$ from the top-$\mathbb{K}$ dictionary entries, the main computation involves $\mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{1\times\mathbb{K}}$ and its softmax weighting over $\mathbf{V}$, resulting in $\mathcal{O}(2\mathbb{K}d)$ complexity per test sample. Since $\mathbb{K}$ is typically small, this module introduces negligible overhead and scales well during inference.

4 Experiment

In this section, we empirically evaluate the effectiveness of the proposed BaCa.[1]

[1]The code of BaCa is available at https://github.com/name-is-what/BaCa.

ID dataset OOD dataset	BZR COX2	PTC-MR MUTAG	AIDS DHFR	ENZYMES PROTEIN	IMDB-M IMDB-B	Tox21 SIDER	FreeSolv ToxCast	BBBP BACE	ClinTox LIPO	Esol MUV	A.A.
Graph Kernel Based Methods											
PK-LOF	42.22±8.39	51.04±6.04	50.15±3.29	50.47±2.87	48.03±2.53	51.33±1.81	49.16±3.70	53.10±2.07	50.00±2.17	50.82±1.48	49.63
PK-OCSVM	42.55±8.26	49.71±6.58	50.17±3.30	50.46±2.78	48.07±2.41	51.33±1.81	48.82±3.29	53.05±2.10	50.06±2.19	51.00±1.33	49.52
PK-iF	51.46±1.62	54.29±4.33	51.10±1.43	51.67±2.69	50.67±2.47	49.87±0.82	52.28±1.87	51.47±1.33	50.81±1.10	50.85±3.51	51.45
WL-LOF	48.99±6.20	53.31±8.98	50.77±2.87	52.66±2.47	52.28±4.50	51.92±1.58	51.47±4.23	52.80±1.91	51.29±3.40	51.26±1.31	51.68
WL-OCSVM	49.16±4.51	53.31±7.57	50.98±2.71	51.77±2.21	51.38±2.39	51.08±1.46	50.38±3.81	52.85±2.00	50.77±3.69	50.97±1.65	51.27
WL-iF	50.24±2.49	51.43±2.02	50.10±0.44	51.17±2.01	51.07±2.25	50.25±0.96	52.60±2.38	50.78±0.75	50.41±2.17	50.61±1.96	50.87
Anomaly Detection Methods											
OCGIN	76.66±4.17	80.38±6.84	86.01±6.59	57.65±2.96	67.93±3.86	46.09±1.66	59.60±4.78	61.21±8.12	49.13±4.13	54.04±5.50	63.87
GLocalKD	75.75±5.99	70.63±3.54	93.67±1.24	57.18±2.03	78.25±4.35	66.28±0.98	64.82±3.31	73.15±1.26	55.71±3.81	86.83±2.35	72.23
Self-supervised Training Methods											
InfoGraph-iF	63.17±9.74	51.43±5.19	93.10±1.35	60.00±1.83	58.73±1.96	56.28±0.81	56.92±1.69	53.68±2.90	48.51±1.87	54.16±5.14	59.60
InfoGraph-MD	86.14±6.77	50.79±8.49	69.02±11.67	55.25±3.51	81.38±1.14	59.97±2.06	58.05±5.46	70.49±4.63	48.12±5.72	77.57±1.69	65.68
GraphCL-iF	60.00±3.81	50.86±4.30	92.90±1.21	61.33±2.27	59.67±1.65	56.81±0.97	55.55±2.71	59.41±3.58	47.84±0.92	62.12±4.01	60.65
GraphCL-MD	83.64±6.00	73.03±2.38	93.75±2.13	52.87±6.11	79.09±2.73	58.30±1.52	60.31±5.24	75.72±1.54	51.58±3.64	78.73±1.40	70.70
GOOD-D	93.00±3.20	78.43±2.67	98.91±0.41	61.89±2.51	79.71±1.19	65.30±1.27	70.48±2.75	81.56±1.97	66.13±2.98	91.39±0.46	78.68
HGOE	–	–	99.28±0.34	64.44±2.19	**81.74±2.25**	68.24±0.60	82.89±2.33	83.46±1.79	70.09±1.52	92.64±2.44	–
Test-time and Data-centric Methods											
AAGOD-GIN$_S$+	76.75	–	–	66.22	59.00	64.26	–	67.80	–	–	–
AAGOD-GIN$_L$+	76.00	–	–	65.89	62.70	57.59	–	57.13	–	–	–
GOODAT	82.16±0.15	81.84±0.57	96.43±0.25	66.29±1.54	79.03±0.03	68.92±0.01	68.83±0.02	77.07±0.03	62.46±0.54	85.91±0.27	76.89
BaCa	**94.23±0.42**	**86.53±1.39**	**99.86±0.03**	**67.10±1.43**	80.93±0.69	**69.82±0.59**	**83.12±0.42**	**93.11±0.29**	**82.57±0.23**	**95.31±0.14**	**85.26**
Improve	△+1.23	△+4.69	△+0.58	△+0.81	▽-0.81	△+0.90	△+0.23	△+9.65	△+12.48	△+2.67	△+6.58

Table 1: OOD detection results in terms of AUC (%, mean ± std). The best and runner-up results are highlighted with **bold** and underline, respectively. The results of baselines are derived from the published works, with unreported results denoted by '–'.

Datasets. For OOD detection, we employ 10 pairs of datasets from two mainstream graph data benchmarks (i.e., TUDataset (Morris et al. 2020) and OGB (Hu et al. 2020)) following GOOD-D (Liu et al. 2023). Each pair of datasets belongs to the same field and shares similar features, but exhibits distribution shifts between two datasets in the pair.

Baselines. We compare BaCa with a wide range of graph OOD detection baselines, grouped into the following categories: (1) **graph kernel based methods** (Neumann et al. 2016; Shervashidze et al. 2011), (2) **anomaly detection methods** (Ma et al. 2022; Zhao and Akoglu 2021), (3) **self-supervised methods** (Sun et al. 2020; You et al. 2020), and (4) **test-time and data-centric methods** (Guo et al. 2023; Wang et al. 2024; Junwei et al. 2024).

Evaluation and Implementation. We evaluate BaCa with a popular OOD detection metric, i.e., area under receiver operating characteristic Curve (AUC). Higher AUC values indicate better performance. The reported results are the mean performance with standard deviation after 5 runs.

Performance on OOD Detection. We compare BaCa with representative baselines on graph OOD detection tasks in Table 1. BaCa achieves the best performance on 7 out of 10 dataset pairs, and runner-up performance on two others. Compared with end-to-end baselines such as GOOD-D (Liu et al. 2023) and HGOE (Junwei et al. 2024), our method consistently yields higher detection accuracy. Notably, under the same test-time setting, BaCa outperforms GOODAT (Wang et al. 2024) on all 10 datasets, with an average AUC improvement of 8.37%. We also observe that both GOODAT and BaCa perform relatively poorly on the IMDB-M/IMDB-B pair. This is likely due to their structural similarity, as both originate from the same dataset source.

Ablation Study. We perform ablation studies by selectively removing the ID dictionary and OOD dictionary (denoted as *ID Dict.* and *OOD Dict.*, respectively). The results are summarized in Table 2. We first observe that BaCa with both dictionaries (last row) consistently achieves the best performance across all dataset pairs, highlighting the effectiveness of our dual-dictionary design. The first row corresponds to removing both dictionaries, which reduces the model to the pretrained baseline without score calibration. Notably, using only one of the two dictionaries (either ID or OOD) leads to a clear drop in performance, indicating that both are necessary to enable accurate boundary-aware score calibration.

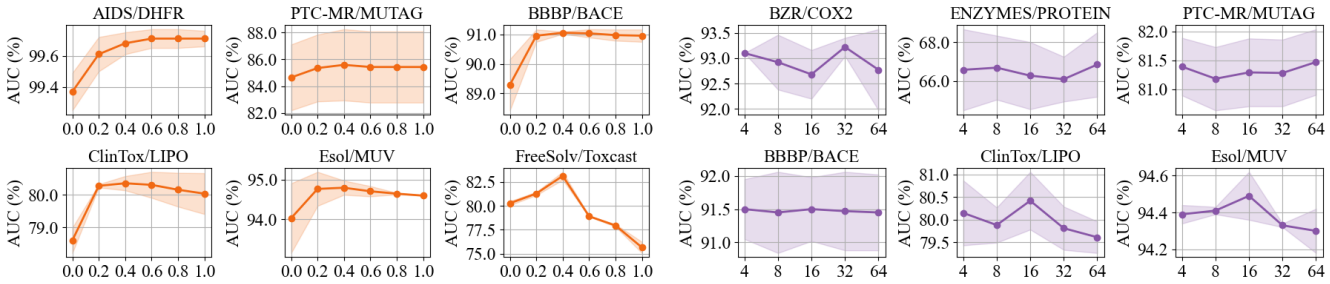
ID Dict.	*OOD Dict.*	BZR COX2	PTC-MR MUTAG	AIDS DHFR	ENZYMES PROTEIN
✗	✗	92.95±0.15	77.59±4.37	99.24±0.06	63.14±0.00
✗	✓	93.22±0.12	85.71±1.88	99.80±0.04	65.51±2.39
✓	✗	93.66±0.03	84.65±2.45	99.50±0.01	66.30±2.43
✓	✓	**94.23±0.42**	**86.53±1.39**	**99.86±0.03**	**67.10±1.43**

Table 2: Ablation study results of BaCa and its variants.

λ	AIDS DHFR	BZR COX2	PTC-MR MUTAG	Esol MUV	ClinTox LIPO
$[0.01, 0.2]$	99.83±0.04	92.89±0.33	85.63±1.47	94.55±0.16	79.62±0.05
$[0.2, 0.4]$	99.81±0.06	92.95±0.62	86.00±1.43	94.46±0.02	79.99±0.79
$[0.4, 0.6]$	99.80±0.07	92.71±0.21	85.92±1.84	94.38±0.11	80.09±0.66
$[0.6, 0.8]$	99.80±0.06	**92.95±0.03**	**86.04±1.55**	94.49±0.27	**80.13±0.84**
$[0.8, 1.0]$	**99.83±0.05**	92.89±0.62	85.92±1.35	**94.53±0.23**	79.96±0.61

Table 3: Performance of BaCa with different λ ranges.

Sensitivity Analysis of λ. We explore the sensitivity of BaCa with respect to the interpolation coefficient λ in Eq. (3). In the main results, λ was randomly sampled from the interval $[0.01, 1]$ for generating mixed graphons between ID and OOD subgroups. Here, we conduct a finer-grained analysis by fixing λ to specific values within this range and examining its impact on detection performance. As shown in Table 3, the



(a) The sensitivity of β on calibration.

(b) The sensitivity of $\mathbb{K}$ on calibration.

Figure 3: Sensitivity analysis on calibration.

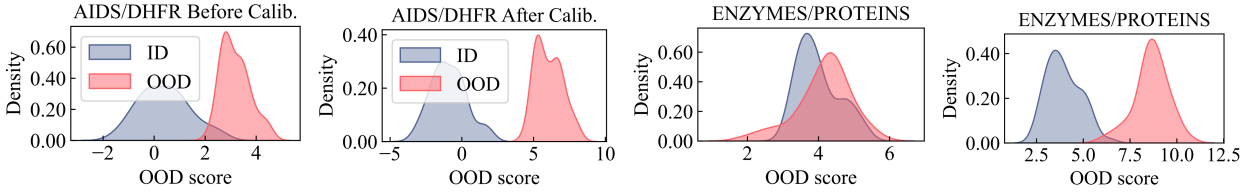


Figure 4: Score distributions on several dataset pairs. The first and third columns show the score distribution **before** calibration (abbreviated as Calib.), while the second and fourth columns present the score distribution **after** applying our calibration on the corresponding dataset. The overlap area between ID and OOD samples is significantly reduced after calibration using BaCa.

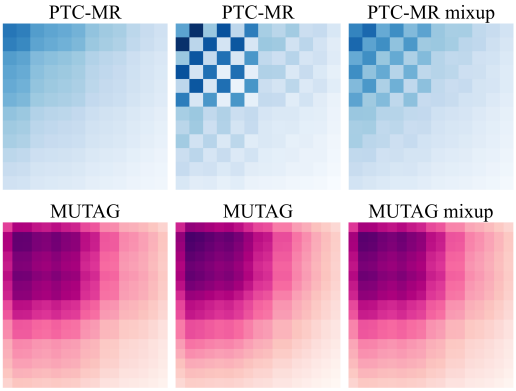


Figure 5: Estimated graphons and their mixup results.

performance sensitivity to λ varies across different dataset pairs. However, we observe that moderate values of λ generally lead to stronger results on most benchmarks. This suggests that a balanced interpolation effectively preserves discriminative topological patterns from both source graphons and enhances the diversity of boundary-aware samples.

Sensitivity Analysis of β. We further study the effect of β, the weight assigned to the calibration term in the final score. As shown in Figure 3(a), we vary β from 0.1 to 1.0. While performance is relatively stable in a mid-range band, too small or too large values of β may suppress or over-amplify the influence of similarity-based score correction.

Sensitivity Analysis of $\mathbb{K}$. To analyze the sensitivity of $\mathbb{K}$ for BaCa, we alter the value from 4 to 64. The AUC w.r.t different selections of $\mathbb{K}$ is plotted in Figure 3(b). Results demonstrate the performance is sensitive to changes in $\mathbb{K}$ and contains a reasonable range across different datasets.

Graphon Mixup Visualization. We estimate graphons of ID and OOD samples and perform graphon mixup visualized as heatmaps on the PTC/MUTAG (PTC as ID, MUTAG as OOD) in Figure 5. We can observe clear structural differences between graphons from different distributions. In contrast, mixup within the same distribution preserves key structural properties while generating new graphons, effectively enhancing the diversity of discriminative typologies.

Score Distribution Visualization. We visualize the OOD score distributions before and after applying our calibration in Figure 4. Compared to the uncalibrated setting, the overlap between ID and OOD score distributions is significantly reduced. This demonstrates that our structure-aware calibration effectively amplifies the distributional differences between ID and OOD samples, leading to more reliable detection.

5 Conclusion

In this paper, we propose BaCa, a boundary-aware OOD score calibration framework for test-time graph OOD detection that calibrates scores without modifying pre-trained GNNs or using auxiliary outlier data. We first partition test samples into subgroups based on pre-trained scores and estimate graphons separately for ID and OOD groups. To handle structural diversity and improve representations near the distribution boundary, we introduce a graphon mixup strategy that generates discriminative topologies stored in dual dynamic priority-queue dictionaries. A learnable attention mechanism is then used for boundary-aware score calibration, which reduces the overlap between ID and OOD score distributions, particularly for boundary samples. Extensive experiments across multiple benchmarks demonstrate the superiority of BaCa over state-of-the-art baselines.

Acknowledgments

This work has been supported by CCSE project (CCSE-2024ZX-09).

References

Airoldi, E. M.; Costa, T. B.; and Chan, S. H. 2013. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. *Advances in Neural Information Processing Systems*, 26.

Chatterjee, S. 2015. Matrix estimation by universal singular value thresholding.

Guo, Y.; Yang, C.; Chen, Y.; Liu, J.; Shi, C.; and Du, J. 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.

Han, X.; Jiang, Z.; Liu, N.; and Hu, X. 2022. G-Mixup: Graph Data Augmentation for Graph Classification. In *ICML*, 8230–8248.

Hou, Y.; Chen, X.; Zhu, H.; Liu, R.; Shi, B.; Liu, J.; Wu, J.; and Xu, K. 2024. NC2D: Novel Class Discovery for Node Classification. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 849–859.

Hou, Y.; Su, Y.; Wu, J.; and Xu, K. 2025a. Test-time Graph OOD Detection via Dynamic Dictionary Expansion and OOD Score Calibration. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 8845–8853.

Hou, Y.; Zhu, H.; Liu, R.; Su, Y.; Xia, J.; Wu, J.; and Xu, K. 2025b. Redundancy-Aware Test-Time Graph Out-of-Distribution Detection. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Hou, Y.; Zhu, H.; Liu, R.; Su, Y.; Xia, J.; Wu, J.; and Xu, K. 2025c. Structural Entropy Guided Unsupervised Graph Out-Of-Distribution Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 17258–17266.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, volume 33, 22118–22133.

Junwei, H.; Xu, Q.; Jiang, Y.; Wang, Z.; Sun, Y.; and Huang, Q. 2024. HGOE: Hybrid External and Internal Graph Outlier Exposure for Graph Out-of-Distribution Detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 1544–1553.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Liu, Y.; Ding, K.; Liu, H.; and Pan, S. 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.

Lovász, L. 2012. *Large networks and graph limits*, volume 60. American Mathematical Soc.

Ma, R.; Pang, G.; Chen, L.; and van den Hengel, A. 2022. Deep Graph-level Anomaly Detection by Glocal Knowledge Distillation. In *WSDM*.

Morris, C.; Kriege, N. M.; Bause, F.; Kersting, K.; Mutzel, P.; and Neumann, M. 2020. TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML Workshop*.

Neumann, M.; Garnett, R.; Bauckhage, C.; and Kersting, K. 2016. Propagation kernels: efficient graph kernels from propagated information. *Machine Learning*, 102(2): 209–245.

Schreyer, M.; Sattarov, T.; Borth, D.; Dengel, A.; and Reimer, B. 2017. Detection of anomalies in large scale accounting data using deep autoencoder networks. *arXiv preprint arXiv:1709.05254*.

Shervashidze, N.; Schweitzer, P.; Van Leeuwen, E. J.; Mehlhorn, K.; and Borgwardt, K. M. 2011. Weisfeiler-lehman graph kernels. *JMLR*, 12(9).

Sun, F.-Y.; Hoffman, J.; Verma, V.; and Tang, J. 2020. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *ICLR*.

Wang, L.; He, D.; Zhang, H.; Liu, Y.; Wang, W.; Pan, S.; Jin, D.; and Chua, T.-S. 2024. GOODAT: Towards Test-Time Graph Out-of-Distribution Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 15537–15545.

Wu, J.; Chen, X.; and Li, S. 2024. Uncovering capabilities of model pruning in graph contrastive learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 6510–6519.

Wu, J.; Chen, X.; Shi, B.; Li, S.; and Xu, K. 2023. SEGA: Structural entropy guided anchor view for graph contrastive learning. In *International Conference on Machine Learning*. PMLR.

Wu, J.; Chen, X.; Xu, K.; and Li, S. 2022a. Structural entropy guided graph hierarchical pooling. In *International Conference on Machine Learning*, 24017–24030. PMLR.

Wu, J.; Li, S.; Li, J.; Pan, Y.; and Xu, K. 2022b. A simple yet effective method for graph classification. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, July 23-29, 2022*. ijcai.org.

Wu, J.; Ooi, B. C.; and Xu, K. 2025. Toward Robust Signed Graph Learning through Joint Input-Target Denoising. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 8721–8729.

Xu, H.; Luo, D.; Carin, L.; and Zha, H. 2021. Learning graphons via structured gromov-wasserstein barycenters. In *AAAI*, volume 35, 10505–10513.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *ICLR*.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. In *NeurIPS*, volume 33, 5812–5823.

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

Zhou, W.; Liu, F.; and Chen, M. 2021. Contrastive Out-of-Distribution Detection for Pretrained Transformers. In *EMNLP*, 1100–1111.