LLM-Enhanced Energy Contrastive Learning for Out-of-Distribution Detection in Text-Attributed Graphs

Xiaoxu Ma[1], Dong Li[2], Minglai Shao[1,3*], Xintao Wu[4], Chen Zhao[2]
[1]School of New Media and Communication, Tianjin University, China
[2]Department of Computer Science, Baylor University, USA
[3]Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, China
[4]Electrical Engineering and Computer Science Department, University of Arkansas, USA
{maxiaoxu, shaoml}@tju.edu.cn, {dong_li1, chen_zhao}@baylor.edu, xintaowu@uark.edu

Abstract

Text-attributed graphs, where nodes are enriched with textual attributes, have become a powerful tool for modeling real-world networks such as citation, social, and transaction networks. However, existing methods for learning from these graphs often assume that the distributions of training and testing data are consistent. This assumption leads to significant performance degradation when faced with out-of-distribution (OOD) data. In this paper, we address the challenge of node-level OOD detection in text-attributed graphs, with the goal of maintaining accurate node classification while simultaneously identifying OOD nodes. We propose a novel approach, **L**LM-Enhanced **E**nergy **C**ontrastive Learning for Out-of-Distribution Detection in **T**ext-Attributed Graphs (**LECT**), which integrates large language models (LLMs) and energy-based contrastive learning. The proposed method involves generating high-quality OOD samples by leveraging the semantic understanding and contextual knowledge of LLMs to create dependency-aware pseudo-OOD nodes, and applying contrastive learning based on energy functions to distinguish between in-distribution (IND) and OOD nodes. The effectiveness of our method is demonstrated through extensive experiments on six benchmark datasets, where our method consistently outperforms state-of-the-art baselines, achieving both high classification accuracy and robust OOD detection capabilities.

Introduction

Text-attributed graphs, which are a type of graph structure where nodes are associated with textual attributes, have gained significant attention due to their wide applicability in real-world scenarios, such as citation networks, social networks, e-commerce transaction networks, and hyperlink networks (Yan et al. 2023). Current approaches to learning from text-attributed graphs (Yang et al. 2021; Chen et al. 2021; He et al. 2023a; Zhao, Chen, and Thuraisingham 2021; Wu et al. 2025; Lin et al. 2025) typically follow a two-step process: first, extracting textual embeddings (Devlin 2018; Liu 2019; Wang et al. 2020), and then applying Graph Neural Networks (GNNs) (Kipf and Welling 2016; Veličković et al. 2017; Hamilton, Ying, and Leskovec 2017) for message passing. While these methods have achieved notable success, they rely on the assumption that the distribution of training and test data is consistent. In the presence of out-of-distribution (OOD) data, these methods often misclassify OOD nodes as part of existing categories, undermining the model's robustness and reliability. Thus, detecting OOD anomalies while ensuring effective graph learning has become a critical challenge.

We address the problem of node-level OOD detection in text-attributed graphs, with a focus on achieving accurate node classification while effectively identifying OOD nodes. Unlike OOD detection in vision (Yang et al. 2024b; Shao et al. 2024) and language (Lang et al. 2023), where OOD samples are assumed to be independent and identically distributed (i.i.d.), OOD detection in graph data is significantly more complex due to the non-Euclidean nature of graphs and the interdependence of their data. Existing methods for graph OOD detection (Ju et al. 2024) can be categorized into propagation-based and classification-based approaches. Propagation-based methods (Stadler et al. 2021; Song and Wang 2022; Huang, Wang, and Fang 2022) leverage the message-passing paradigm of GNNs to propagate uncertainty estimates, while classification-based methods (Hendrycks and Gimpel 2016; Liang, Li, and Srikant 2017; Wu et al. 2023) design OOD score metrics for detection. However, most current approaches overlook textual attributes and rely on shallow models, such as bag-of-words, which may fail to capture the rich semantic information embedded in the text. Furthermore, these models often focus on complex propagation mechanisms and uncertainty quantification, while neglecting the critical interplay between textual features and node connectivity. With the rapid advancement of large language models (LLMs) (Li et al. 2025b,a; Zhao et al. 2025), which provide extensive contextual knowledge and semantic understanding, textual attribute information can now be more effectively captured and leveraged for OOD detection (Cao et al. 2024; Li et al. 2024a; Xu and Ding 2024).

To address OOD detection in text-attributed graphs, we propose a novel LLM-enhanced contrastive learning method, named **LECT**. Our method consists of two key stages: (1) LLM-based sample generation and (2) energy-based contrastive learning. In the first stage, we generate pseudo-OOD nodes by randomly initializing edges. Using the textual understanding and generation capabilities of

*Corresponding author.

LLMs, we create OOD samples related to connected IND nodes through chain-of-thought (COT) prompts, producing high-quality pseudo-data with meaningful dependency relationships. In the second stage, we train the model using Linked IND-OOD Pairs and Triplet Contrastive Pairs, applying an energy-based objective function to effectively distinguish between IND and OOD samples. Our contributions are as follows:

- We propose a novel method for generating high-quality OOD node textual attributes using LLMs, ensuring dependency relationships with connected IND nodes.
- We design an energy-based contrastive learning algorithm for OOD detection, which enables multi-dimensional sample pairs to effectively capture distinctions between IND and OOD samples.
- We demonstrate that our model outperforms state-of-the-art baselines on six benchmark datasets while maintaining strong node classification performance.

Related Work

LLM-based Graph Mining. Large language models (LLMs) have shown strong capabilities in text representation and generation, leading to their broad adoption in graph learning tasks (Ren et al. 2024; Chen et al. 2024b). Their roles can be categorized as enhancers (He et al. 2023b; Chien et al. 2022; Liu et al. 2023), predictors (Tang et al. 2024; Chen et al. 2024a; Zhao et al. 2023), and aligners (Zhao et al. 2023, 2022). As enhancers, LLMs enrich node features (e.g., TAPE (He et al. 2023b)); as predictors, they perform reasoning-based tasks (e.g., GraphGPT (Tang et al. 2024)); and as aligners, they ensure consistency between language and graph modalities (e.g., GLEM (Zhao et al. 2022)). Among these, using LLMs to enhance node attributes is notably efficient and stable (Li et al. 2024b).

While prior work has focused on node classification and link prediction, little attention has been given to OOD detection in text-attributed graphs. Motivated by the enhancer role, we propose leveraging LLMs to improve node-level OOD detection by generating diverse textual attributes and extracting high-quality representations from pretrained models.

OOD Detection on Graphs. Despite the strong representational capabilities of GNN, their performance often degrades or becomes overconfident when test samples differ significantly from the training distribution, leading to misclassification. Consequently, OOD detection on graphs has attracted considerable attention and can be categorized into graph-level and node-level tasks (Ju et al. 2024; Li et al. 2023). Node-level OOD detection focuses on OOD nodes within a single graph. Due to the interdependence among nodes, it presents greater challenges compared to graph-level detection (Bazhenov et al. 2022; Li et al. 2022), which involves separate graphs.

Currently, node-level OOD detection methods can be broadly divided into classification-based and propagation-based approaches. Classification-based methods, such as MSP (Hendrycks and Gimpel 2016) and ODIN (Liang, Li, and Srikant 2017), utilize maximum softmax probability or temperature scaling to estimate OOD probabilities. However, these methods struggle to handle complex distributions and are prone to overconfidence issues. Propagation-based methods, such as GPN (Stadler et al. 2021) and OODGAT (Song and Wang 2022), enhance model reliability by propagating uncertainty within the graph. GraphSAFE (Wu et al. 2023) achieves promising results using energy-based techniques. However, it overlooks node textual attributes, and the scarcity of OOD samples during training, along with the complexity of setting upper and lower thresholds in real-world scenarios, limits its ability to effectively address complex node dependencies and distribution shifts.

To tackle these challenges, we propose a novel approach that leverages the generative and reasoning capabilities of LLMs to capture textual attribute dependencies among nodes, generating pseudo-OOD samples. Subsequently, energy-based contrastive learning is employed to capture the relationship between IND and OOD samples.

Preliminaries

The node-level OOD detection task aims to determine whether each node in a graph belongs to the in-distribution or out-of-distribution. Specifically, given a graph with textual features, the goal is to classify each node based on its textual features and the graph's structural information, thereby identifying whether the node belongs to the training dataset's distribution or to an OOD distribution.

Predictive Tasks on Text-Attributed Graphs. Consider a text-attributed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{Q})$, where $\mathbf{V}$ is the set of nodes, $\mathbf{E} = \{e_{ij}\}$ represents the set of edges, and the adjacency matrix $\mathbf{A} \in \mathbb{R}^{|\mathbf{V}|\times|\mathbf{V}|}$ can be derived from the edge set. The adjacency matrix $a_{ij} = 1$ if there is an edge connecting nodes i and j, and $a_{ij} = 0$ otherwise. Each node $n \in \mathbf{V}$ is associated with a textual feature q_n, typically a sentence or description. The set $\mathbf{Q} = \{q_n\}_{n\in\mathbf{V}}$ denotes the collection of these textual features for all nodes. The node classification problem is formulated as follows: given labeled nodes $\mathbf{Y}_{train}$ (where $\mathbf{V}_{train} \subset \mathbf{V}$), the goal is to predict the labels of the unlabeled nodes $\mathbf{Y}_{test}$ (where $\mathbf{V}_{test} = \mathbf{V} \setminus \mathbf{V}_{train}$).

OOD Detection Task on Text-Attributed Graphs. Besides achieving strong predictive performance on IND nodes, the learned classifier is also expected to detect OOD instances that originate from a distinct data-generating distribution. For the node-level OOD detection task, we assume a training dataset $\mathbf{V}_{train} = \{V^1_{train}, V^2_{train}, \ldots, V^{N_1}_{train}\}$ consisting solely of IND nodes, and a testing dataset $\mathbf{V}_{test} = \{V^1_{test_{in}}, V^2_{test_{in}}, V^1_{test_{out}}, \ldots, V^{N_2}_{test_{out}}\}$, where $V_{test_{in}}$ and $V_{test_{out}}$ are drawn from distinct distributions P_{in} and P_{out}, respectively. For each node V^n_{test}, the goal of OOD detection is to determine, based on its textual feature Q_n and the graph's structural information $\mathbf{A}$, whether the node originates from the training data distribution P_{in} or from an OOD distribution P_{out}, while ensuring accurate node classification.

We define a detector d and a scoring function $s(V^n_{test}, f_\theta)$, where f_θ represents the model trained on $\mathbf{V}_{train}$ with pa-

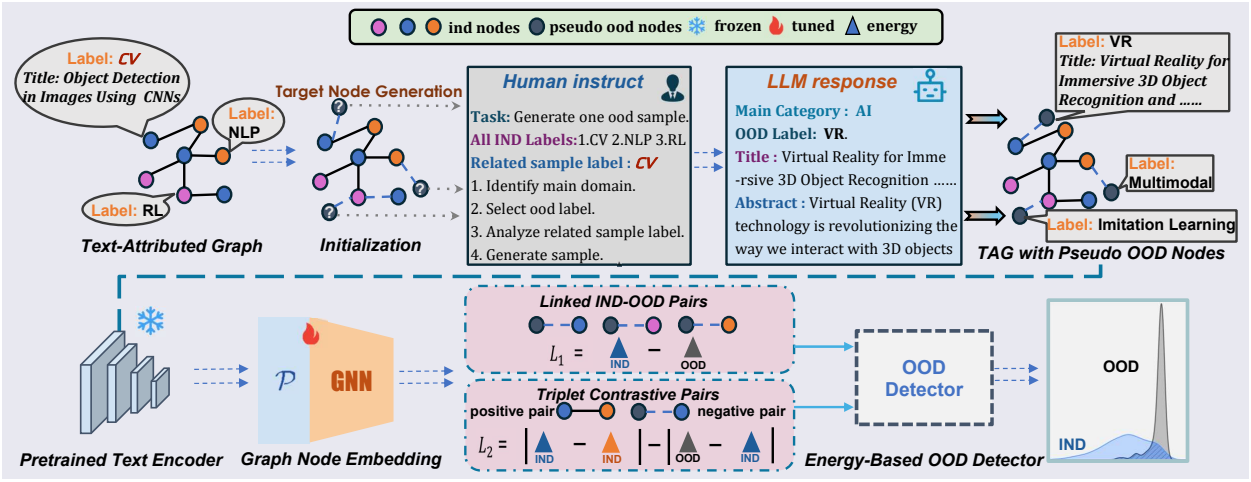


Figure 1: The overall pipeline of LECT. Given a text-attributed graph, we first construct pseudo-OOD nodes by injecting random edges and generating their textual content with an LLM. We then derive node representations using a text encoder and a GNN with a projector. Energy scores are subsequently computed to form Linked IND–OOD Pairs and Triplet Contrastive Pairs for training. Finally, the model identifies OOD samples based on the resulting energy levels.

rameters θ. The OOD detection task can be formalized as:

$$d(V_{test}^n; \tau, s, f_\theta) = \begin{cases} V_{test}^n \in P_{in}, & \text{if } s(V_{test}^n, f_\theta) \leq \tau, \\ V_{test}^n \in P_{out}, & \text{if } s(V_{test}^n, f_\theta) > \tau, \end{cases} \quad (1)$$

where $s(V_{test}^n, f_\theta)$ is the scoring function that computes the score of node V_{test}^n based on its textual features and graph structure. The threshold τ determines whether the node belongs to the in-distribution or out-of-distribution. P_{in} denotes the distribution of the training data, while P_{out} represents the OOD data distribution. The optimization goal of OOD detection is to learn an appropriate scoring function $s(V_{test}^n, f_\theta)$ and well-trained model parameters f_θ.

Methodology

In this section, we introduce the proposed LECT algorithm for text-attributed graph OOD detection. The architecture of our proposed model is shown in Figure 1.

Pseudo-OOD Sample Generation

Initialization of Pseudo-OOD Nodes. Assume we have the original training graph $\mathbf{G}_s = (\mathbf{V}_s, \mathbf{E}_s, \mathbf{Q_s})$. First, we initialize the pseudo-OOD node set, resulting in $\mathbf{V}_o = \{v_o^1, v_o^2, \ldots, v_o^{N_o}\}$, where N_o is the number of generated pseudo-OOD nodes.

To simulate the generation of OOD samples while modeling the dependencies between IND and OOD samples, we initialize the structural relationships between nodes using a random edge connection method. Specifically, each pseudo-OOD node v_o is randomly connected to one or more IND nodes in the graph. The number of IND nodes connected to each OOD node is controlled by a predefined upper limit C_{max}, which means that each pseudo-OOD node can connect to a maximum of C_{max} IND nodes. This step aims to control the complexity of the graph, preventing excessive connections from introducing noise and increasing the inference difficulty for the subsequent large model. To control complexity, we initialize C_{max} as the number of classes in the training samples, that is, $|Y_{\text{train}}|$.

Thus, we define the edge set $\mathbf{E}_o$ as:

$$\mathbf{E}_o = \{e_{ij} = (v_i, v_j) \mid v_i \in \mathbf{V}_s, v_j \in \mathbf{V}_o\}. \quad (2)$$

Chain-of-Thought (COT) Construction. Given the initialized pseudo-node structure $\mathbf{V}_o$ and edge set $\mathbf{E}_o$, we employ LLM to generate OOD text attributes $\mathbf{Q}_o$ based on the dependency relationships of the nodes. To address potential challenges in generating accurate and coherent responses for unfamiliar patterns, we incorporate the COT technique to guide step-by-step reasoning. This enables the model to produce high-quality OOD samples through logical and structured reasoning. The COT-guided generation process is as follows:

1. ***Main Domain Classification:*** Using the labels of all IND samples, the LLM classifies the samples into known domains, aiding in categorizing OOD samples based on existing IND labels.
2. ***Selection of OOD Categories:*** The LLM selects an OOD category significantly different from the IND distribution to ensure the generated samples effectively represent out-of-distribution data.
3. ***Analysis of Connected Sample Labels:*** To simplify reasoning, only the labels of IND samples connected to the OOD nodes are provided. This allows the LLM to select OOD labels correlated with connected IND labels, simulating real-world graph dependencies.
4. ***Sample Generation:*** Based on the selected OOD label, the LLM generates textual features for the pseudo-OOD nodes.

To simulate diverse OOD scenarios, the COT approach generates both near-OOD and far-OOD samples:

- *Near-OOD samples*: OOD categories are selected close to the main IND domain to capture subtle distributional shifts.
- *Far-OOD samples*: Categories are chosen farther from the IND domain to introduce more diverse and loosely related samples.

For each pseudo-OOD node, we repeat the above LLM generation step, resulting in the pseudo-OOD node text features $\mathbf{Q}_{\text{o}}$. Thus, the LLM-enhanced graph can be represented as:

$$\mathbf{G}_{\text{en}} = (\mathbf{V}_s \cup \mathbf{V}_o, \mathbf{E}_s \cup \mathbf{E}_o, \mathbf{Q}_s \cup \mathbf{Q}_o). \tag{3}$$

Energy-based Contrastive Learning

Text-Attributed Graphs Feature Extraction. After generating the pseudo-OOD samples and obtaining the enhanced graph $\mathbf{G}_{\text{en}}$, we use a pre-trained language model (PLM) to extract the textual features for each node. Specifically, for each node $v_i \in \mathbf{G}_{\text{en}}$, the textual feature q^i is mapped to a textual embedding h^i via the PLM:

$$h^i = \text{PLM}(q^i), \quad \forall i \in V_{\text{en}}. \tag{4}$$

To align the textual features with the graph structure and capture richer representations, we input all node embeddings $\mathbf{H} = [h^1, h^2, \ldots, h^{|V_{\text{en}}|}]$ into a projector $\mathcal{P}$, obtaining the projected features $\mathbf{H}'$:

$$\mathbf{H}' = \mathcal{P}(\mathbf{H}). \tag{5}$$

We then pass the projected textual features $\mathbf{H}'$ along with the graph structure information $\mathbf{A}$ into a GNN for message passing and node updating, resulting in the final node representations z^i:

$$z^i = \text{GNN}(\mathbf{H}', \mathbf{A}), \quad \forall i \in \mathbf{V}_{\text{en}}. \tag{6}$$

Thus, the node representations $\mathbf{Z} = [z^1, z^2, \ldots, z^{|\mathbf{V}_{\text{en}}|}]$ are obtained for each node in the graph $\mathbf{G}_{\text{en}}$.

Energy-Based OOD Detection. We use the energy function as the OOD detection score, which is grounded in the principles of Energy-Based Models (EBM) (Liu et al. 2020). Specifically, the energy function quantifies the degree of alignment between a sample and the training data. The relationship between the energy function $\mathcal{E}(x, y)$ and probability is described by the Boltzmann distribution, which can be written as:

$$p(y|x) = \frac{e^{-\mathcal{E}(x,y)}}{\mathcal{Z}(x)} = \frac{e^{-\mathcal{E}(x,y)}}{\sum_{y'} e^{-\mathcal{E}(x,y')}}, \tag{7}$$

where $\mathcal{E}(x, y)$ represents the energy for input x and class label y, and $\mathcal{Z}(x)$ is the partition function, which sums over all possible class labels. A lower energy $\mathcal{E}(x, y)$ indicates a higher degree of alignment between sample x and class label y, suggesting that the sample is more likely to belong to the training data distribution (i.e., IND samples). Hence, IND samples have lower energy values, while OOD samples exhibit higher energy values.

For the embedding representations z_i obtained from the PLM and graph neural network, we calculate the energy for each node using the following formula:

$$\mathcal{E}_i = -\log\left(\sum_{c=1}^{C} \exp(z_c^i)\right), \tag{8}$$

where C is the number of classes, and z_c^i denotes the score for node v_i belonging to class c.

Linked IND-OOD Pairs. We first construct pairs of linked IND and OOD samples. Using the energy values $\mathbf{G}_{\text{en}}$ from Equation 3, we randomly select a pair e_{ij} from the set E_o, where v_i is an IND sample and v_j is an OOD sample, forming a linked IND-OOD sample pair $\mathbb{P}_{\text{ind-ood}}$. Then, applying the vector z^i from Equation 6 and using Equation 8, we compute the corresponding energy values $\mathcal{E}_i$ and $\mathcal{E}_j$ for the IND and OOD samples, respectively.

For all IND-OOD pairs $\mathbb{P}_{\text{ind-ood}}$, the loss is computed as:

$$\mathcal{L}_{\text{ind-ood}} = \mathbb{E}_{(v_i,v_j)\in\mathbb{P}_{\text{ind-ood}}}\left[\max\left(0, \gamma - (\mathcal{E}_j - \mathcal{E}_i)\right)\right], \tag{9}$$

where $\mathcal{E}_i$ and $\mathcal{E}_j$ are the energy values for the IND and OOD samples, respectively. $\mathbb{P}_{\text{ind-ood}}$ denotes the set of all IND and OOD sample pairs, and (v_i, v_j) represents a specific pair. The energy difference $\mathcal{E}_j - \mathcal{E}_i$ captures the dependency between the IND and OOD samples. The margin γ is a hyperparameter that enforces a lower bound on the energy difference, preventing misclassification of challenging pairs as OOD. According to Equation 7, larger energy values indicate lower confidence, increasing the likelihood that a sample belongs to the OOD class. This loss function aims to minimize the energy difference to correctly distinguish between IND and OOD samples. Additionally, a constraint term is introduced to regulate the energy difference between the mean energies of all IND and OOD samples.

Triplet Contrastive Pairs. In this step, we construct triplet contrastive node pairs. Specifically, for a given IND node v_{c} in $\mathbf{G}_{\text{en}}$, we first identify the edge $e_{ci} = (v_c, v_i)$ in $\mathbf{E}_s$, where v_i is an IND node, and the edge $e_{cj} = (v_c, v_j)$ in $\mathbf{E}_o$, where v_j is an OOD node. The pair e_{ci} is treated as a positive sample, and e_{cj} as a negative sample, thus forming the triplet contrastive pair $\mathcal{T} = \{v_i, v_c, v_j\}$, where v_i and v_c are IND nodes, and v_j is an OOD node. The energy values for each node are then computed using Equation 6 and Equation 8.

The contrastive learning loss is defined as follows:

$$\mathcal{L}_{\text{triplet}} = \mathbb{E}_{v_i,v_j,v_c\in\mathcal{T}}\left[\max\left(0, |\mathcal{E}_i - \mathcal{E}_{\text{c}}| - (\mathcal{E}_j - \mathcal{E}_{\text{c}})\right)\right], \tag{10}$$

where $|\mathcal{E}_i - \mathcal{E}_{\text{c}}|$ represents the absolute energy difference between the positive sample nodes, and $\mathcal{E}_j - \mathcal{E}_{\text{c}}$ represents the energy difference between the negative sample nodes.

The design of this loss function aims to capture the relationships and dependencies between IND and OOD nodes in the graph effectively. For the positive sample pair (the relationship between the IND node and the center node), we expect the energy difference to be as small as possible, since connected IND nodes should exhibit high similarity, implying a small energy difference. For the negative sample pair (the relationship between the OOD node and the center node), we expect a larger energy difference. OOD nodes typically exhibit lower confidence than IND nodes, so their energy values should be significantly higher, leading to a more pronounced energy difference.

Model Learning. The main task of the model is node classification, optimized using a supervised loss function $\mathcal{L}_{\text{sup}}$. This is formulated as a cross-entropy loss over the labeled nodes $\mathbf{V}_s$ in the training set. Let $\hat{\mathbf{y}}_v = (\hat{y}_{v,1}, \hat{y}_{v,2}, \ldots, \hat{y}_{v,C})$ denote the predicted class probabilities for node v, and $y_v \in \{1, \ldots, C\}$ be the true class label. The cross-entropy loss is defined as:

$$\mathcal{L}_{\text{sup}} = -\sum_{v \in \mathbf{V}_s} \log \hat{y}_{v,y_v}, \tag{11}$$

where $\hat{y}_{v,y_v}$ is the predicted probability of the true class y_v for node v.

Combining the supervised loss, IND-OOD loss, and triplet contrastive loss, the overall loss function $\mathcal{L}_{\text{all}}$ is formulated as:

$$\mathcal{L}_{\text{all}} = \mathcal{L}_{\text{sup}} + \lambda_1 \mathcal{L}_{\text{ind-ood}} + \lambda_2 \mathcal{L}_{\text{triplet}}, \tag{12}$$

where $\mathcal{L}_{\text{ind-ood}}$ accounts for the pairwise loss between IND and OOD samples, and $\mathcal{L}_{\text{triplet}}$ is the triplet contrastive loss. The hyperparameters λ_1 and λ_2 control the contributions of these auxiliary losses.

This combined loss enables the model to optimize for both node classification and OOD detection. During testing, node embeddings are extracted and their energy values are computed to distinguish IND and OOD samples via a threshold. To avoid overfitting to pseudo-OOD samples, we freeze the pretrained language model (PLM) and update only the projection and GNN parameters. This reduces training overhead while preserving the PLM's generalization ability.

Experiments

Experimental Settings

Dataset. We selected six datasets for the experiments: the citation datasets `Cora`, `Citeseer`, `Pubmed` (Li et al. 2024b), and `Arxiv-2023` (Zhu et al. 2024), the web link dataset `Wikics` (Li et al. 2024b), and the transaction network dataset `Photo` (Yan et al. 2023). Since there were no existing benchmarks for text graph OOD detection, in order to evaluate the effectiveness of semantic OOD detection at the node level in text graphs, we primarily construct OOD samples through label shifting. Specifically, we treat one or more classes of samples from the graph as OOD samples, masking them in the training set and treating them as OOD samples in the test set.

Evaluation Metrics. We use AUROC, AUPR, and FPR95 as evaluation metrics for OOD detection, along with accuracy for assessing the classification performance of IND nodes. AUROC evaluates the model's ability to distinguish IND from OOD samples. AUPR emphasizes the balance between precision and recall, addressing class imbalance. FPR95 measures the misclassification rate of OOD samples as IND at a 95% IND true positive rate.

Baselines. We compare two types of OOD detection models. The first type consists of widely used OOD detection models in the visual domain, such as **MSP** (Hendrycks and Gimpel 2016) and **ODIN** (Liang, Li, and Srikant 2017). In deployment, we replace the convolutional neural networks with GCNs that extract graph representations. The second type includes advanced baseline models specifically designed for node-level OOD detection in graph data, including **GPN** (Stadler et al. 2021), **OOD-GAT** (Song and Wang 2022), **OSSNC** (Huang, Wang, and Fang 2022) , **EMP** (Yang, Lu, and Gan 2023) , **GNNSAFE** (Wu et al. 2023), **GRASP**(Gong and Sun 2024) and **NODESAFE**(Yang et al. 2025).

Experimental Setup. We implement both our model and all baseline models using PyTorch (Paszke et al. 2019). Each method utilizes MiniLM (Wang et al. 2020) for text feature extraction and a two-layer GCN to extract node representations, with a hidden layer dimension of 64. A batch normalization layer is applied between the two layers, and the dropout rate is set to 0.5. The Adam optimizer (Kingma 2014) is used with a learning rate of 0.001 and weight decay regularization set to 0.0005. All models are trained for 300 epochs. For our model, we employ LLAMA3 (8b) (Dubey et al. 2024), Qwen2.5 (7b) (Yang et al. 2024a), Gemma2 (9b) (Team et al. 2024), and Chatgpt-4omini (Achiam et al. 2023) to generate near-OOD and far-OOD data during the large model generation phase. In the contrastive learning phase, the projection layer dimension is set to 128. All experiments are conducted on an Nvidia 3090 GPU.

Overall Performance

We conduct experiments on six real-world graph datasets. Due to space constraints, we show three representative datasets (`Cora`, `Citeseer`, and `Pubmed`) in Table 1. For each dataset, we perform five repeated experiments and report the average results along with the standard deviations for ind-accuracy, AUROC, AUPR, and FPR95. For our method, we select four representative large models to generate pseudo-OOD data.

As shown in Table 1, our proposed method outperforms all baseline models on all datasets in OOD detection. Specifically, on the `Cora`, the AUROC is improved by 1.3% compared to the best baseline model, AUPR is improved by 7.9%, and FPR95 is reduced by 2.1%. On the `Citeseer`, AUROC is improved by 3.6%, AUPR is improved by 11.2%, and FPR95 is reduced by 7.2%. On the `Pubmed`, all metrics show a performance improvement ranging from 4.7% to 11%. These results demonstrate the effectiveness of our proposed method, which combines LLM-generated pseudo-samples with energy-based contrastive learning for OOD detection. Additionally, we observe that, compared to other methods, the accuracy of our IND samples is even improved, with only a slight performance loss on the `Pubmed`. This indicates that our method can accurately detect OOD samples while maintaining strong node classification performance. As shown in Figure 4, LECT outperforms baseline models by maintaining high accuracy in classifying IND nodes and significantly improving the distinction of OOD samples.

Our method achieves notable performance improvements by integrating high-quality pseudo-OOD sample generation using LLMs with an energy-based contrastive learning framework. This joint design facilitates more effective modeling of the relationship between IND and OOD samples, improving detection accuracy without the need for manually tuned thresholds. Unlike prior energy-based approaches that

	Cora				Citeseer				Pubmed			
	IND-Acc↑	AUROC↑	AUPR↑	FPR95↓	IND-Acc↑	AUROC↑	AUPR↑	FPR95↓	IND-Acc↑	AUROC↑	AUPR↑	FPR95↓
MSP	86.72±0.47	81.96±1.38	43.37±1.23	99.21±0.43	75.36±0.76	71.47±1.29	30.81±0.57	97.64±0.89	**92.68±0.18**	60.53±3.38	48.67±3.71	96.65±0.94
ODIN	86.78±0.21	80.76±1.96	43.92±4.48	62.74±11.61	76.40±0.50	70.60±2.27	32.97±3.78	79.77±3.09	92.26±0.29	59.86±3.00	49.09±4.07	92.39±1.32
OSSNC	86.22±0.41	82.32±1.76	47.01±4.34	97.51±0.63	75.89±0.40	72.14±1.57	33.27±1.67	96.23±2.44	92.50±0.28	61.14±2.32	48.84±2.38	97.22±0.36
EMP	86.14±0.58	82.12±2.49	47.00±5.30	60.91±4.58	75.89±0.51	70.16±2.06	30.87±2.27	79.26±2.46	92.59±0.31	61.44±4.26	51.24±4.63	92.90±1.86
OODGAT	88.08±0.65	80.31±1.47	37.51±4.25	49.09±5.82	76.90±0.65	68.43±1.26	27.33±2.49	97.72±2.04	91.52±0.26	70.08±1.39	32.67±3.99	96.55±1.78
GPN	86.84±1.16	79.71±4.50	45.33±5.00	76.66±4.46	76.76±0.38	55.35±1.05	46.38±0.61	92.51±1.17	92.03±0.23	60.72±1.94	49.04±2.79	95.80±1.97
NODESAFE	88.05±0.41	92.54±0.77	71.74±2.37	37.59±3.21	75.89±0.53	84.29±0.93	48.41±0.73	52.92±2.37	91.42±0.33	83.60±1.38	79.34±3.01	63.92±4.77
GRASP	87.81±0.43	92.71±0.82	72.04±3.18	31.40±3.31	76.52±0.39	82.01±0.87	54.79±0.66	71.45±5.37	90.39±0.77	83.79±0.93	79.77±1.88	72.17±5.21
GNNSAFE	88.13±0.37	92.23±0.90	70.09±3.50	34.49±4.35	76.70±0.62	81.20±1.31	51.17±3.65	69.17±2.47	90.72±0.46	82.55±1.22	75.67±2.80	71.26±6.03
GNNSAFE++	87.26±0.63	93.03±0.50	71.37±2.64	30.77±5.52	73.76±2.30	84.51±1.98	49.23±10.93	54.77±6.29	81.89±2.59	84.62±2.25	70.76±5.77	58.98±2.33
LECT(llama)	88.14±0.24	**94.30±0.78**	**79.22±3.24**	28.61±3.14	76.65±0.58	**88.09±0.70**	62.35±5.16	**47.51±3.11**	92.19±0.07	**89.36±1.07**	**86.70±1.89**	**51.72±3.17**
LECT(qwen)	**88.55±0.23**	93.20±1.12	73.18±7.03	**27.16±3.24**	**77.54**±0.58	87.65±1.51	**62.46±3.35**	58.58±5.39	91.81±0.20	87.41±0.48	83.28±1.32	58.44±0.93
LECT(gemma)	88.49±0.42	93.43±0.53	75.06±3.29	29.68±4.72	76.27±0.37	87.86±0.84	60.01±4.41	55.25±2.96	91.83±0.24	87.20±0.58	82.96±0.72	57.80±1.35
LECT(chatgpt)	87.48±0.70	93.61±0.50	73.30±3.16	30.42±3.68	76.52±**1.02**	87.17±1.39	61.09±5.46	55.84±3.00	92.01±0.17	87.90±0.94	84.56±1.03	57.01±3.86

Table 1: OOD detection performance in terms of Ind-Acc, AUROC, AUPR, and FPR95 (mean ± std). Best results are bolded, second best are marked with a dash, and ↑ (↓) indicates that larger (smaller) values are better.

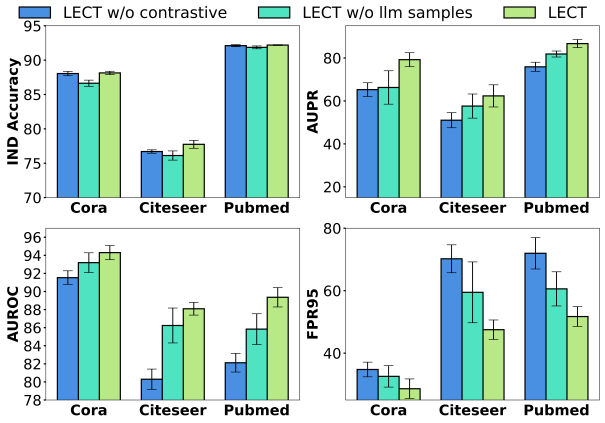


Figure 2: Ablation study results of LECT on the Cora, Citeseer, and Pubmed datasets, showing the performance without contrastive learning and without LLM-generated samples, respectively.

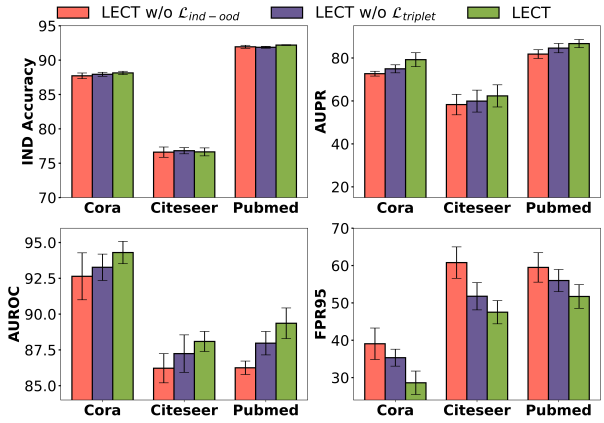


Figure 3: Ablation study results of LECT on the Cora, Citeseer, and Pubmed datasets, showing the performance without $\mathcal{L}_{\text{ind-ood}}$ and without $\mathcal{L}_{\text{triplet}}$, respectively.

rely on static bounds and often compromise IND classification performance, our method maintains a better balance between IND classification and OOD detection. Specifically, we leverage four LLMs to generate pseudo-OOD node attributes that exhibit both distributional and semantic shifts from the IND data, as illustrated in Figure 5.

Ablation Study

Ablation Study On the Role of LLM-Generated Pseudo-OOD Samples and Contrastive Learning. To investigate the contributions of LLM-generated pseudo-OOD samples and contrastive learning in OOD detection, we perform ablation experiments by removing these components individually: (i) LECT *w/o contrastive learning* excludes the proposed contrastive learning method, relying solely on energy scores for OOD detection without using OOD samples and contrastive learning; (ii) LECT *w/o LLM-generated samples* replaces the LLM-generated pseudo-OOD samples with randomly generated text attributes from other datasets. The experimental results are presented in Figure 2. **(i) Removal of Contrastive Learning:** When only the energy score is used for OOD detection, the performance is significantly worse compared to the case where pseudo-OOD samples generated from random text are combined with contrastive learning. This demonstrates that incorporating pseudo-OOD samples and contrastive learning significantly enhances OOD detection performance. **(ii) Removal of LLM-generated Pseudo-OOD Samples:** We observe that using randomly generated text attributes results in poorer performance compared to using LLM-generated attributes. This highlights the effectiveness of our COT template in guiding large models to generate high-quality pseudo-OOD data with meaningful dependencies.

Ablation Study on the Role of $\mathcal{L}_{\text{ind-ood}}$ and $\mathcal{L}_{\text{triplet}}$. To better explore the roles of $\mathcal{L}_{\text{ind-ood}}$ and $\mathcal{L}_{\text{triplet}}$, we conducted ablation experiments, with the results shown in Figure 3. **(i) Removal of $\mathcal{L}_{\text{ind-ood}}$:** When training with only $\mathcal{L}_{\text{triplet}}$ and

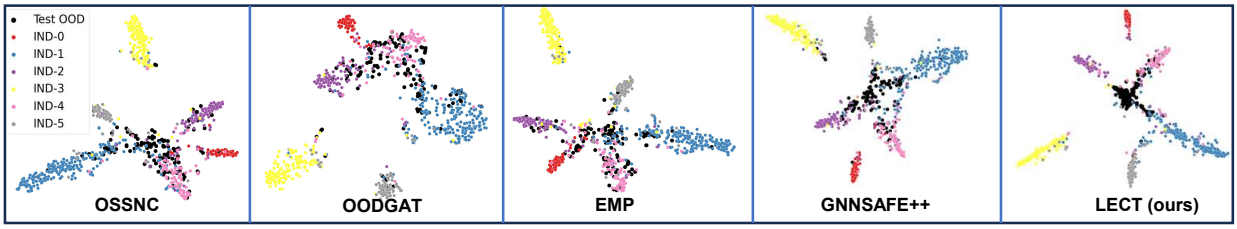


Figure 4: t-SNE visualization of node embeddings on the Cora for different baseline models and LECT.

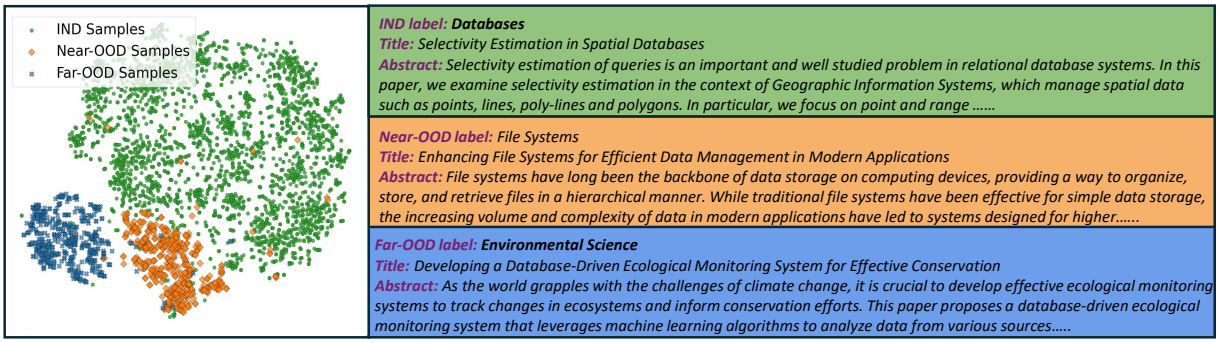
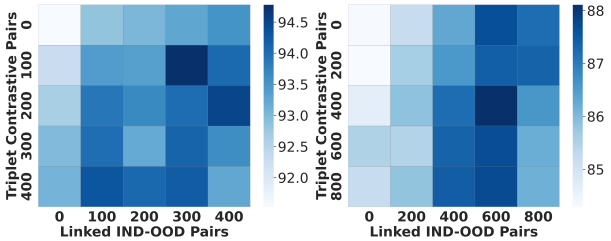


Figure 5: t-SNE visualization and textual representation of the generated near-OOD and far-OOD samples on the Citeseer.

$\mathcal{L}_{\text{sup}}$, the overall performance is inferior to that achieved with $\mathcal{L}_{\text{ind-ood}}$ and $\mathcal{L}_{\text{sup}}$. This indicates that the Linked IND-OOD Pairs play a critical role in the overall model performance. This is mainly because the LLM-generated samples are inherently dependent on the linked IND samples, and this loss directly captures such dependencies.**(ii) Removal of $\mathcal{L}_{\text{triplet}}$:** When training with only $\mathcal{L}_{\text{ind-ood}}$ and $\mathcal{L}_{\text{sup}}$, the performance is inferior to the joint training with $\mathcal{L}_{\text{ind-ood}}$, $\mathcal{L}_{\text{triplet}}$, and $\mathcal{L}_{\text{sup}}$. This highlights the necessity of the designed Triplet Contrastive Pairs, as this loss captures the differences between normal connections (IND-OOD) and OOD connections (IND-OOD) in the graph. Through contrastive learning, it enhances the energy gap between OOD samples and ind samples.

Sensitivity Analysis

To evaluate the impact of Linked IND-OOD pairs and Triplet Contrastive Pairs in contrastive learning, we sampled different quantities of sample pairs, and the results are presented in Figure 6. It is observed that the model performs worst when the sample sizes for both types of pairs are zero, indicating that both sampling strategies contribute positively to the model's ability to learn OOD patterns. For the Cora dataset, the optimal performance is achieved with 300 and 100 sample pairs, while for the Citeseer dataset, the best performance occurs with 600 and 400 sample pairs, respectively. This suggests that the number of sampled pairs influences the results across different datasets. In general, the number of sampled pairs is positively correlated with the size of the dataset. However, we also observe that an excessive number of samples does not always lead to improved performance, as it may result in overfitting, ultimately reducing the model's effectiveness.



(a) Cora (b) Citeseer

Figure 6: Heatmap of AUROC values for different sampling sizes of Linked IND-OOD pairs and Triplet Contrastive Pairs.

Conclusion

In this paper, we address the critical challenge of OOD detection in text-attributed graphs, proposing a novel method that integrates LLM with contrastive learning. Our approach improves OOD detection by generating high-quality pseudo-OOD samples that preserve dependency relationships between nodes. We introduce an energy-based contrastive learning framework that effectively distinguishes between IND and OOD nodes while maintaining robust node classification performance. Experimental results demonstrate that our method outperforms existing state-of-the-art approaches across six benchmark datasets, highlighting its effectiveness in both OOD detection and node classification tasks.

Acknowledgments

This work done by Xiaoxu Ma and Minglai Shao is supported by the National Natural Science Foundation of China (No. 62272338) and the Research Fund of the Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education (EBME25-F-06). Dong Li, Xintao Wu, and Chen Zhao did not receive any financial support for this work and contributed only by developing the research ideas, participating in discussions, and providing feedback on the manuscript.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bazhenov, G.; Ivanov, S.; Panov, M.; Zaytsev, A.; and Burnaev, E. 2022. Towards ood detection in graph classification from uncertainty estimation perspective. *arXiv preprint arXiv:2206.10691*.

Cao, C.; Zhong, Z.; Zhou, Z.; Liu, Y.; Liu, T.; and Han, B. 2024. Envisioning Outlier Exposure by Large Language Models for Out-of-Distribution Detection. *arXiv preprint arXiv:2406.00806*.

Chen, R.; Zhao, T.; Jaiswal, A.; Shah, N.; and Wang, Z. 2024a. Llaga: Large language and graph assistant. *arXiv preprint arXiv:2402.08170*.

Chen, X.; Wang, C.; Li, D.; and Sun, X. 2021. A new early rumor detection model based on bigru neural network. *Discrete Dynamics in Nature and Society*, 2021(1): 2296605.

Chen, Z.; Mao, H.; Li, H.; Jin, W.; Wen, H.; Wei, X.; Wang, S.; Yin, D.; Fan, W.; Liu, H.; et al. 2024b. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2): 42–61.

Chien, E.; Chang, W.-C.; Hsieh, C.-J.; Yu, H.-F.; Zhang, J.; Milenkovic, O.; and Dhillon, I. S. 2022. Node Feature Extraction by Self-Supervised Multi-scale Neighborhood Prediction. In *International Conference on Learning Representations (ICLR)*.

Devlin, J. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Gong, Z.; and Sun, Y. 2024. An energy-centric framework for category-free out-of-distribution node detection in graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 908–919.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

He, X.; Bresson, X.; Laurent, T.; Hooi, B.; et al. 2023a. Explanations as features: Llm-based features for text-attributed graphs. *arXiv preprint arXiv:2305.19523*, 2(4): 8.

He, X.; Bresson, X.; Laurent, T.; Perold, A.; LeCun, Y.; and Hooi, B. 2023b. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. *arXiv preprint arXiv:2305.19523*.

Hendrycks, D.; and Gimpel, K. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Huang, T.; Wang, D.; and Fang, Y. 2022. End-to-end open-set semi-supervised node classification with out-of-distribution detection. IJCAI.

Ju, W.; Yi, S.; Wang, Y.; Xiao, Z.; Mao, Z.; Li, H.; Gu, Y.; Qin, Y.; Yin, N.; Wang, S.; et al. 2024. A survey of graph neural networks in real world: Imbalance, noise, privacy and ood challenges. *arXiv preprint arXiv:2403.04468*.

Kingma, D. P. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Lang, H.; Zheng, Y.; Li, Y.; Sun, J.; Huang, F.; and Li, Y. 2023. A survey on out-of-distribution detection in nlp. *arXiv preprint arXiv:2305.03236*.

Li, D.; Wan, G.; Wu, X.; Wu, X.; Chen, X.; He, Y.; Lian, C. G.; Sorger, P. K.; Semenov, Y. R.; and Zhao, C. 2025a. Multi-Modal Foundation Models for Computational Pathology: A Survey. *arXiv preprint arXiv:2503.09091*.

Li, D.; Wang, W.; Shao, M.; and Zhao, C. 2023. Contrastive representation Learning based on multiple Node-centered Subgraphs. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 1338–1347.

Li, D.; Zhao, C.; Shao, M.; and Wang, W. 2024a. Learning fair invariant representations under covariate and correlation shifts simultaneously. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 1174–1183.

Li, D.; Zhao, X.; Yu, L.; Liu, Y.; Cheng, W.; Chen, Z.; Chen, Z.; Chen, F.; Zhao, C.; and Chen, H. 2025b. SolverLLM: Leveraging Test-Time Scaling for Optimization Problem via LLM-Guided Search. *arXiv preprint arXiv:2510.16916*.

Li, Y.; Wang, P.; Zhu, X.; Chen, A.; Jiang, H.; Cai, D.; Chan, V. W. K.; and Li, J. 2024b. Glbench: A comprehensive benchmark for graph with large language models. *arXiv preprint arXiv:2407.07457*.

Li, Z.; Wu, Q.; Nie, F.; and Yan, J. 2022. Graphde: A generative framework for debiased learning and out-of-distribution detection on graphs. *Advances in Neural Information Processing Systems*, 35: 30277–30290.

Liang, S.; Li, Y.; and Srikant, R. 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

Lin, Y.; Li, D.; Wu, X.; Shao, M.; Zhao, X.; Chen, Z.; and Zhao, C. 2025. Face4FairShifts: A Large Image Benchmark for Fairness and Robust Learning across Visual Domains. *arXiv preprint arXiv:2509.00658*.

Liu, H.; Feng, J.; Kong, L.; Liang, N.; Tao, D.; Chen, Y.; and Zhang, M. 2023. One for all: Towards training one graph model for all classification tasks. *arXiv preprint arXiv:2310.00149*.

Liu, W.; Wang, X.; Owens, J.; and Li, Y. 2020. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33: 21464–21475.

Liu, Y. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Ren, X.; Tang, J.; Yin, D.; Chawla, N.; and Huang, C. 2024. A survey of large language models for graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 6616–6626.

Shao, M.; Li, D.; Zhao, C.; Wu, X.; Lin, Y.; and Tian, Q. 2024. Supervised algorithmic fairness in distribution shifts: A survey. *In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 8225–8233.

Song, Y.; and Wang, D. 2022. Learning on graphs with out-of-distribution nodes. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1635–1645.

Stadler, M.; Charpentier, B.; Geisler, S.; Zügner, D.; and Günnemann, S. 2021. Graph posterior network: Bayesian predictive uncertainty for node classification. *Advances in Neural Information Processing Systems*, 34: 18033–18048.

Tang, J.; Yang, Y.; Wei, W.; Shi, L.; Su, L.; Cheng, S.; Yin, D.; and Huang, C. 2024. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 491–500.

Team, G.; Riviere, M.; Pathak, S.; Sessa, P. G.; Hardin, C.; Bhupatiraju, S.; Hussenot, L.; Mesnard, T.; Shahriari, B.; Ramé, A.; et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; and Zhou, M. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33: 5776–5788.

Wu, Q.; Chen, Y.; Yang, C.; and Yan, J. 2023. Energy-based out-of-distribution detection for graph neural networks. *arXiv preprint arXiv:2302.02914*.

Wu, X.; Chen, X.; Wu, X.; Li, D.; Chen, Z.; He, Y.; and Zhao, C. 2025. Explainable Image-Centric Forgery Detection: A Survey. *Authorea Preprints*.

Xu, R.; and Ding, K. 2024. Large language models for anomaly and out-of-distribution detection: A survey. *arXiv preprint arXiv:2409.01980*.

Yan, H.; Li, C.; Long, R.; Yan, C.; Zhao, J.; Zhuang, W.; Yin, J.; Zhang, P.; Han, W.; Sun, H.; et al. 2023. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36: 17238–17264.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024a. Qwen2. 5 Technical Report. *arXiv preprint arXiv:2412.15115*.

Yang, J.; Liu, Z.; Xiao, S.; Li, C.; Lian, D.; Agrawal, S.; Singh, A.; Sun, G.; and Xie, X. 2021. Graphformers: Gnn-nested transformers for representation learning on textual graph. *Advances in Neural Information Processing Systems*, 34: 28798–28810.

Yang, J.; Zhou, K.; Li, Y.; and Liu, Z. 2024b. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12): 5635–5662.

Yang, L.; Lu, B.; and Gan, X. 2023. Graph Open-Set Recognition via Entropy Message Passing. In *2023 IEEE International Conference on Data Mining (ICDM)*, 1469–1474. IEEE.

Yang, S.; Liang, B.; Liu, A.; Gui, L.; Yao, X.; and Zhang, X. 2025. Bounded and uniform energy-based out-of-distribution detection for graphs. *arXiv preprint arXiv:2504.13429*.

Zhao, C.; Chen, F.; and Thuraisingham, B. 2021. Fairness-aware online meta-learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2294–2304.

Zhao, J.; Qu, M.; Li, C.; Yan, H.; Liu, Q.; Li, R.; Xie, X.; and Tang, J. 2022. Learning on large-scale text-attributed graphs via variational inference. *arXiv preprint arXiv:2210.14709*.

Zhao, J.; Zhuo, L.; Shen, Y.; Qu, M.; Liu, K.; Bronstein, M.; Zhu, Z.; and Tang, J. 2023. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*.

Zhao, Q.; Li, D.; Liu, Y.; Cheng, W.; Sun, Y.; Oishi, M.; Osaki, T.; Matsuda, K.; Yao, H.; Zhao, C.; et al. 2025. Uncertainty propagation on llm agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6064–6073.

Zhu, Y.; Wang, Y.; Shi, H.; and Tang, S. 2024. Efficient tuning and inference for large language models on textual graphs. *arXiv preprint arXiv:2401.15569*.