

Adaptive Hierarchical Graph Cut for Multi-granularity Out-of-distribution Detection

Xiang Fang, Arvind Easwaran, Blaise Genest, Ponnuthurai Nagarathnam Suganthan, *Fellow, IEEE*

Abstract—This paper focuses on a significant yet challenging task: out-of-distribution detection (OOD detection), which aims to distinguish and reject test samples with semantic shifts, so as to prevent models trained on in-distribution (ID) data from producing unreliable predictions. Although previous works have had decent success, they are ineffective for real-world challenging applications since these methods regard all unlabeled data as OOD data and ignore the case that different datasets have different label granularity. For example, “cat” on CIFAR-10 and “tabby cat” on Tiny-ImageNet share the same semantics but have different labels due to various label granularity. To this end, in this paper, we propose a novel Adaptive Hierarchical Graph Cut network (AHGC) to deeply explore the semantic relationship between different images. Specifically, we construct a hierarchical KNN graph to evaluate the similarities between different images based on the cosine similarity. Based on the linkage and density information of the graph, we cut the graph into multiple subgraphs to integrate these semantics-similar samples. If the labeled percentage in a subgraph is larger than a threshold, we will assign the label with the highest percentage to unlabeled images. To further improve the model generalization, we augment each image into two augmentation versions, and maximize the similarity between the two versions. Finally, we leverage the similarity score for OOD detection. Extensive experiments on two challenging benchmarks (CIFAR-10 and CIFAR-100) illustrate that in representative cases, AHGC outperforms state-of-the-art OOD detection methods by 81.24% on CIFAR-100 and by 40.47% on CIFAR-10 in terms of “FPR95”.

Impact Statement—As an effective method to detect outliers, out-of-distribution detection (OOD detection) has attracted more and more attention. However, previous works regard all unlabeled data as OOD data and ignore the case that some unlabeled samples might have similar semantics with labeled data, which might render their methods ineffective for real-world challenging benchmarks. We construct a hierarchical KNN graph to evaluate the similarities between different images. Moreover, our method outperforms the state-of-the-art works by about 81.24% in representative cases. With satisfactory performance on multiple datasets, our proposed method has wide potential applications.

Index Terms—Out-of-distribution Detection, Adaptive Hierarchical Graph Cut, Intra-subgraph Label Assignment

This research is part of the programme DesCartes and is supported by the National Research Foundation, Prime Minister’s Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. (Corresponding author: Xiang Fang)

Xiang Fang is with Energy Research Institute @ NTU, Interdisciplinary Graduate Programme, Nanyang Technological University, Singapore, is also with College of Computing and Data Science, Nanyang Technological University, Singapore (e-mail: xiang003@e.ntu.edu.sg).

Arvind Easwaran with College of Computing and Data Science, Nanyang Technological University, Singapore, is also with CNRS@CREATE, Singapore (e-mail: arvinde@ntu.edu.sg).

Blaise Genest is with CNRS and CNRS@CREATE, IPAL IRL 2955, France and Singapore (e-mail: blaise.genest@cnrsatcreate.sg).

Ponnuthurai Nagarathnam Suganthan is with KINDI Computing Research Center, College of Engineering, Qatar University, Doha (e-mail: p.n.suganthan@qu.edu.qa).

I. INTRODUCTION

Deep Neural Networks (DNNs) have achieved great success under a closed-set assumption [1], [2], where all the samples experienced during the test have been seen during training [3], [4]. Traditional DNN methods compulsorily classify every sample as some of the known classes. The wrong classification of outliers will lead to irrecoverable losses in some safety-critical scenarios. To avoid these losses, out-of-distribution (OOD) detection [5] is proposed to accurately detect the outliers from OOD classes and correctly classify the samples from in-distribution (ID) classes during testing.

The main challenge for OOD detection is that no information about OOD classes is available during training, making it difficult to distinguish ID and OOD samples. To address the challenges, many works [7]–[12] calibrate the distribution of the softmax layer for OOD detection. Other methods [13]–[16] target leveraging many OOD samples to learn the discrepancy between ID/OOD samples at training time, then detect the OOD samples during testing. Although most OOD detection methods have achieved remarkable performance, they tend to perform unsatisfactorily in real-world artificial intelligence (AI) applications since they simply treat a labeled dataset as an ID dataset while regarding all other datasets as an OOD dataset. The treatment is inappropriate in many AI applications since an unlabeled dataset usually contains some ID images. As shown in Fig. 1(a) and 1(b), both the Tiny-ImageNet dataset and the CIFAR-10 dataset contain “cat” samples: The Tiny-ImageNet dataset contains three fine-grained classes (“Tabby Cat”, “Egyptian Cat”, and “Persian Cat”), while the CIFAR-10 dataset only contains a coarse-grained class (“Cat”). Although these samples actually share the same semantics, many methods [17]–[21] regard them as different distributions in Fig. 1(c), which will severely limit their real-world performance since they ignore the relationship between labels with different granularity. To overcome the above limitation, [6] re-designs semantically coherent out-of-distribution detection (SC-OOD) benchmarks and proposes the UDG framework based on K-means clustering to dynamically filter ID samples from the unlabeled datasets. Although [6] finds there are some ID samples in unlabeled datasets, the relationship between fine- and coarse-grained classes is still overlooked. Since semantically coherent classes are treated as different classes, the dynamical K-means clustering often assigns wrong labels to unlabeled OOD samples, which results in unsatisfactory performance, as shown in Fig. 1(e). There is still a large room to deal with OOD detection in real-world multi-granularity datasets.

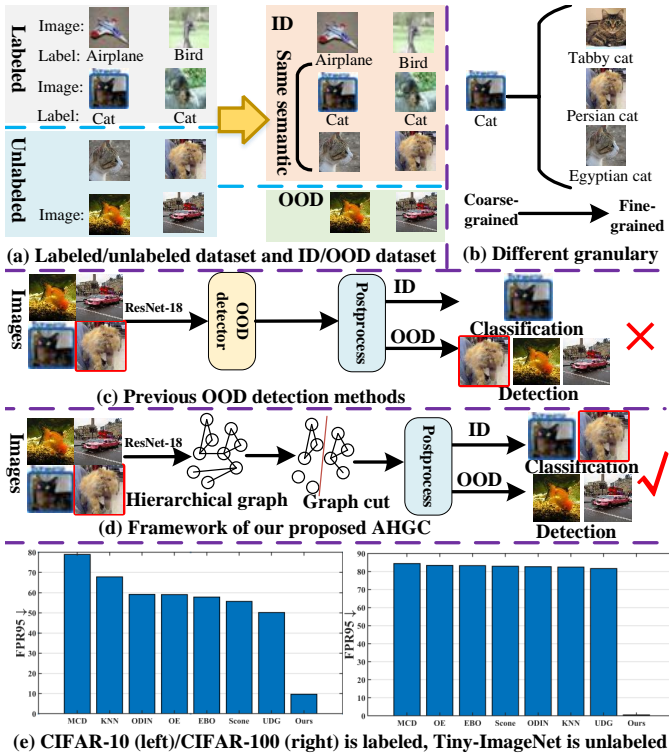


Fig. 1: (a) Relationship between labeled/unlabeled dataset and ID/OOD dataset. (b) Relationship between coarse- and fine-grained labels. In fact, “Tabby cat”, “Persian cat” and “Egyptian cat” belong to the “Cat” class. (c) Previous methods ignore the relationship between coarse- and fine-grained samples, and mistakenly treat unlabeled ID samples (Persian cat) as OOD. (d) Brief framework of our proposed AHGC network, detailed in Fig. 2. (e) Performance comparison (lower FPR95 value means better performance) between state-of-the-art methods and our AHGC on two semantically coherent out-of-distribution detection benchmarks [6]: CIFAR-10 and CIFAR-100. Best viewed in color.

To address the multi-granularity OOD detection task, we design a novel adaptive hierarchical graph cut network (AHGC) to explore the relationship between coarse-grained and fine-grained classes with the same semantics in Fig. 1(d). Specifically, to mine the ID label information, based on the labeled dataset, we first train a feature encoder and a labeled classifier jointly for ID classification. Then, a hierarchical KNN graph is constructed to compute the cosine similarity between different images. Besides, we leverage the ground-truth labels in the labeled dataset as supervision to compute the linkage and density of the graph. Based on the linkage and density information, we cut the graph into multiple subgraphs by removing low-weight edges from the graph. By the above graph cut operation, these high-similarity samples are clustered to the same subgraph, while low-similarity samples are assigned to different subgraphs. If the labeled percentage in a subgraph is larger than a predefined threshold, we will assign the label with the highest percentage to unlabeled images. To enhance the image features, we re-extract the image feature after moving the unlabeled images into the labeled images. To further

improve the model generalization, we first map the enhanced features to a lower-dimensional logit space. Then, we augment each image into two augmentation versions, and maximize the similarity between the two versions. Finally, given any unlabeled sample, we utilize the energy-based approach with temperature-scaled logits for OOD detection.

The main contributions of this paper are as follows: 1) We present a novel adaptive graph cut network, AHGC, which deeply explores multi-granularity semantics between different datasets. To the best of our knowledge, it is the first time that a graph cut network is designed to detect OOD samples and classify ID samples simultaneously. 2) We propose an attention-aware graph cut approach to integrate the images across granularity. Besides, we design an iterative labeling strategy to encourage semantic alignment between different samples in the same subgraph in each epoch. Also, we augment each image into two versions and maximize their similarity for model generalization. 3) We conduct extensive experiments on two challenging benchmarks (CIFAR-10 and CIFAR-100), where our proposed AHGC outperforms state-of-the-art OOD detection methods with clear margins. In representative cases, AHGC outperforms all state-of-the-art OOD detection methods by 81.24% on the CIFAR-100 benchmark and by 40.47% on the CIFAR-10 benchmark in terms of “FPR95”, demonstrating its effectiveness.

II. RELATED WORKS

A. Out-of-distribution Detection

Out-of-distribution detection (OOD detection) targets to detect test samples from distributions that do not overlap with the training distribution. Previous OOD detection methods [7]–[16], [22], [23] can be divided into four types: classification-based methods [5], [7], [19], [24], density-based methods [25], [26], distance-based methods [10], [27] and reconstruction-based methods [28], [29]. 1) Early works mainly refer to a classification framework, which utilizes the maximum softmax probability to determine the ID/OOD samples. For example, by a two-step training process, [11] jointly learns generalizable OOD features and ID features with a minimal twist in a regular multi-class DNN classifier. By using an auxiliary outlier dataset during training, POEM [30] selects informative outliers that are close to the decision boundary between ID and OOD samples. 2) To more explicitly model ID samples, density-based methods leverage the probabilistic models for OOD detection. These methods are based on an operating assumption that OOD samples have low likelihoods whereas ID samples have high likelihoods under the estimated density model. For instance, ConjNorm [31] introduces a novel Bregman divergence-based theoretical framework to design density functions in exponential family distributions. 3) The distance-based methods are under an intuitive idea that OOD samples should be relatively far away from the centroids of ID samples. For example, [10] first utilizes a generative classifier under LDA assumption, and then uses the minimum Mahalanobis distance to all class centroids for OOD detection. 4) The reconstruction-based methods often leverage the encoder-decoder framework, which is trained on only ID samples and

generates different outcomes for OOD detection. For instance, MoodCat [29] is proposed to first mask a random image region and then determine the OOD samples based on classification-based reconstruction results.

The above methods simply define one dataset as ID and all the others as OOD, which neglects the relevant semantic information between different datasets. To overcome the drawback, [6] re-designs these off-the-shelf popular benchmarks into two challenging and proper benchmarks, SC-OOD, and proposes a clustering-based network, UDG, for OOD detection. Unfortunately, although UDG can separate some ID samples from unlabeled datasets, its clustering strategy cannot sufficiently understand the relationship between datasets with different granularity, which causes it to not correctly assign correct labels to unlabeled ID samples. In this paper, we design a novel hierarchical graph cut framework to fully understand the relationship between different samples to widen the gap between ID and OOD samples.

B. Graph Neural Networks

As an important type of DNNs, graph neural networks (GNNs) [32], [33] have been a topic of significant interest in the field of deep learning in recent years. Due to their satisfactory performance in many areas, GNNs have attracted more and more research attention. GNNs target extending neural networks to deal with graph-structured data based on the similarity between different samples. To process these graph-structured data, GNNs often iteratively update each node feature by aggregating feature similarity with its neighbor nodes. Previous GNN-based methods refer to the closed-set assumption, and cannot be used in the challenging OOD detection task. Since GNNs can effectively evaluate the similarity between samples, we design a novel adaptive hierarchical graph cut network to distinguish ID and OOD samples for OOD detection.

III. PROBLEM STATEMENT AND NOTATIONS

The OOD detection task aims to detect outliers from the intended data distribution, to prevent models trained on ID samples from producing arbitrary predictions for OOD samples. For a benchmark with labeled and unlabeled datasets, we denote the labeled dataset as \mathcal{D}^L and the unlabeled dataset as \mathcal{D}^U , respectively¹. Most previous works [7], [8], [34] assume that all the unlabeled samples in \mathcal{D}^U are OOD samples, while all the ID samples are in the labeled dataset. However, real-world unlabeled dataset \mathcal{D}^U is a mixture of ID samples and OOD samples, *i.e.*, some unlabeled samples in \mathcal{D}^U share the same semantics with labeled samples in \mathcal{D}^L , while the others have different semantics. Thus, we aim to mine the deep semantic features of labeled and unlabeled datasets. On the challenging SC-OOD benchmarks, training set \mathcal{D} consists of labeled training set \mathcal{D}^L and unlabeled training set \mathcal{D}^U . For the unlabeled dataset \mathcal{D}^U including ID set \mathcal{I} and OOD set \mathcal{O} , we target to widen their distance for detecting OOD set \mathcal{O} .

All the labeled samples in \mathcal{D}^L belong to \mathcal{I} , while \mathcal{D}^U is a mixture of partial ID set \mathcal{D}_I^U from \mathcal{I} and OOD set \mathcal{D}_O^U from \mathcal{O} . Therefore, the training set is represented as $\mathcal{D} = \mathcal{D}^L \cup \mathcal{D}_I^U \cup \mathcal{D}_O^U$, where $\mathcal{D}^L \subset \mathcal{I}$, $\mathcal{D}_I^U \subset \mathcal{I}$ and $\mathcal{D}_O^U \subset \mathcal{O}$. Similarly, for the test set, we have $\mathcal{T} = \mathcal{T}_I \cup \mathcal{T}_O$, where $\mathcal{T}_I \subset \mathcal{I}$ and $\mathcal{T}_O \subset \mathcal{O}$. For the labeled dataset $\mathcal{D}^L = (X^L, Y^L)$ with N^L samples from R classes, we have $X^L = (x_i^L)_{i=1}^{N^L}$ and labels $Y^L = (y_i^L)_{i=1}^{N^L}$, where x_i denotes the i -th labeled sample and $y_i^L \in \{1, 2, \dots, R\}$ is its corresponding label. Also, we collect an unlabeled dataset $\mathcal{D}^U = (X^U)$ with N^U samples $X^U = (x_j^U)_{j=1}^{N^U}$. A carefully-designed model should not only accurately detect OOD samples from \mathcal{D}^U , but also correctly classify ID samples from \mathcal{D}^L and \mathcal{D}^U into R classes.

IV. ADAPTIVE HIERARCHICAL GRAPH CUT NETWORK

We present our proposed AHGC network in Fig. 2, where we cluster labeled and unlabeled images into multiple sub-graphs and assign proper labels to unlabeled ID images based on density and linkage within each subgraph.

A. Feature Encoder and Classifier on Labeled Datasets

Following [6], we use ResNet-18 [35] to extract image features. For any sample x_i^L (or x_j^U), we denote its feature as $f_i^L \in \mathbb{R}^{d \times 1}$ (or $f_j^U \in \mathbb{R}^{d \times 1}$), where d denotes the feature dimension. Since only the labeled datasets contain the known label information of ID samples, we design a classifier to learn the label information for the ID classification task. We target to train the feature encoder \mathcal{F} and a classification head \mathcal{C}^L simultaneously. Given the image-label pairs (X^L, Y^L) , we train the ID classifier with the feature encoder by the following loss:

$$\mathcal{L}_0^L = \mathcal{L}_{CE}(X^L, Y^L, \theta_F, \theta_C), \quad (1)$$

where θ_F is a learnable parameter in \mathcal{F} , θ_C is a learnable parameter in \mathcal{C}^L , and $\mathcal{L}_{CE}(\cdot)$ denotes the standard cross-entropy loss. Based on \mathcal{L}_0^L , we can jointly train the feature encoder and the classifier for labeled images. After training the classifier, we can understand the relationship between labels and samples for the following process.

B. Single-level Graph Cut Mechanism for Base Model

To measure the similarity between different images, we first construct a graph by combining all images (both labeled samples f_i^L and unlabeled images f_j^U). By jointing labeled image features and unlabeled image features, we can obtain the feature set $F = \{f_i^L\}_{i=1}^{N^L} \cup \{f_j^U\}_{j=1}^{N^U} = \{f_i\}_{i=1}^N$, where $N = N^L + N^U$ denotes the total image number. Then, based on the cosine similarity, we seek K -nearest neighbors of each image to construct an affinity graph $G = \{V, E\}$, where V is the vertex set containing all images (*i.e.*, $|V| = N$), and E is the edge set. In real-world benchmarks (*e.g.*, SC-OOD), each image entails one core object, which is represented as a node in the constructed graph G , with the corresponding image feature as the node feature. Each node is connected to its k neighbors by the edges.

¹In this paper, superscripts L and U denote labeled and unlabeled samples, respectively; superscripts I and O denote ID and OOD samples, respectively.

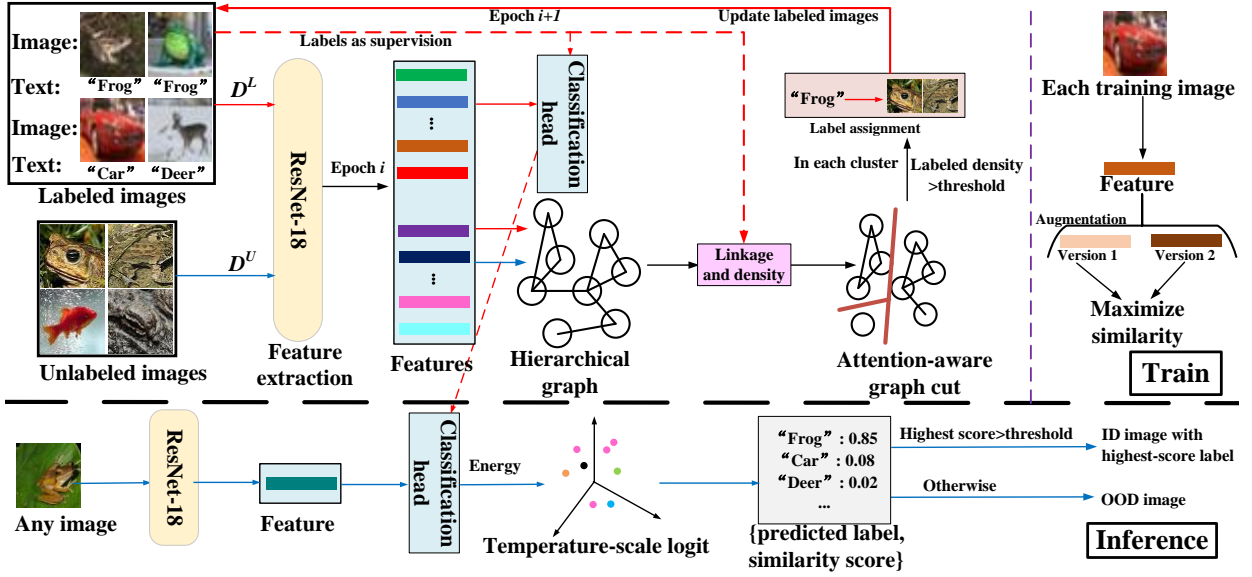


Fig. 2: Overview of our AHGC. We color labeled data flow as red and unlabeled data flow as blue. Best viewed in color.

We introduce a function $\vartheta(\cdot, \cdot)$ to construct the edge subset $E' \subset E$, where $E' = \vartheta(G, F)$. The graph G can be updated as $G' = \{V, E'\}$, which is divided into multiple subgraphs, and each subgraph corresponds to a cluster of nodes. By the above graph cut paradigm, we design the KNN graph cut approach.

C. Hierarchical Framework for Multi-granularity Graph Cut

Real-world AI applications adopt different labeling criteria, and different image datasets often offer various labels from a variety of granularity perspectives. For example, on the CIFAR-10 benchmark in Fig. 1(b), the cat only contains a label (“Cat”), while in the Tiny-ImageNet dataset, the cat is divided into three labels (“Tabby Cat”, “Egyptian Cat”, and “Persian Cat”). Thus, we should deeply understand each image and explore the similarities between images. Unfortunately, previous OOD detection methods [6], [17] ignore this multi-granularity situation, which severely degrades their performance in real-world AI applications. To solve the problem, we propose a hierarchical framework based on the above single-level graph cut paradigm in Section IV-B.

Given initial image features $F = \{f_i\}_{i=1}^N$ and a small k , we design a hierarchical graph framework using a base subgraph function $\vartheta(\cdot, \cdot)$ and an aggregation function $\delta(\cdot, \cdot)$. Especially, we iteratively generate a sequence of graphs $G_l = \{V_l, E_l\}$ and the corresponding node features $F_l = \{f_i\}_{i=1}^{|V_l|}$, where the subscript l denotes the l -th level and we determine it as needed. Firstly, we initialize the graph as G_1 and the corresponding node feature as $F_1 = \{f_i\}_{i=1}^{|V_1|}$. At level l , by introducing the subgraph function $\vartheta(\cdot, \cdot)$, we can obtain the initial edge subset: $E'_l = \vartheta(G_l, F_l)$. By introducing the function $\vartheta(\cdot, \cdot)$, we can obtain the selected edge subsets E'_l based on the node features H_l and k -NN graph G_l . Then, we reserve the selected edge subsets E'_l and remove the rest edges. The graph G_l can be updated as $G'_l = \{V_l, E'_l\}$. Therefore, we can split G'_l into multiple subgraphs $\{c_i^{(l)}\}_{i=1}^{|V_{l+1}|}$, where $c_i^{(l)}$ is the i -th subgraph. To obtain G_{l+1} at the $(l+1)$ -th

level, we denote the i -th node as $v_i^{(l+1)}$, which is an entity representing the subgraph $c_i^{(l)}$. After that, we introduce an aggregation function $\delta(\cdot, \cdot)$ to update the new node feature vectors: $F_{l+1} = \delta(F_l, G'_l)$. Therefore, we can aggregate the node features in each subgraph $c_i^{(l)}$ into a single feature vector. Finally, we can search k -nearest-neighbors on F_{l+1} and connect each node to its k neighbors to obtain the updated E_{l+1} . We repeat the above process until no more new edges are added, i.e., $E'_l = \emptyset$. We denote the length of the converged sequence as Q . Based on G_Q , we assign subgraph index i to the subgraph $c_i^{(Q)}$, which propagates the index i to all its nodes $\{v_j^{(Q)} | v_j^{(Q)} \in c_i^{(Q)}\}$. Then, each node $v_i^{(Q)}$ propagates its label to the subgraph $c_i^{(Q-1)}$ on the previous iteration. By the above process, we can assign a subgraph ID to each node in V_1 , and finish the label assignment on the final level.

In the next sections, we will present the details about the base subgraph function $\vartheta(\cdot, \cdot)$ and the aggregation function $\delta(\cdot, \cdot)$. Besides, we can use our AHGC for an underlying single-level model, akin to a single iteration of AHGC. Also, the inference process will be described.

D. Attention-aware Graph Cut via Linkage and Density

Since the linkage and density information in the GNN model contain a rich relationship between different images, we treat them as supervision for model training. The linkage and density information can be treated as the supervision signals to learn the GNN model [36]–[38]. To understand the semantics of images, we use the graph attention network (GAT) [39] as the backbone to model the similarity between images by the linkage and density information. To extract the complex subgraph structures, we first utilize the graph cut function $\vartheta(\cdot, \cdot)$ for building a learnable graph network, where each node v_i in V comes with a subgraph label y_i . Unlike previous unsupervised graph cut methods [40], [41] that focus on an explicit graph cut, we design the attention-aware graph cut

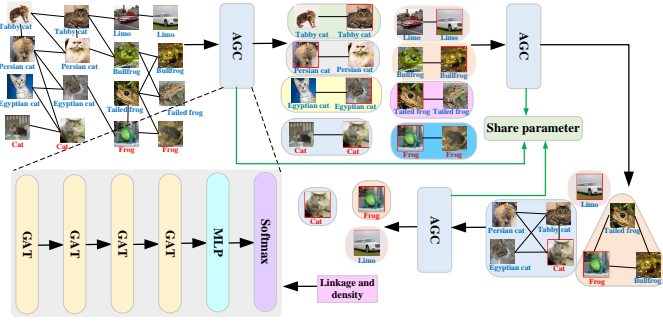


Fig. 3: Illustration of the attention-aware graph cut module, where “AGC” means the “adaptive graph cut” module, the image with a red edge in a subgraph is its peak node, and labels are attached to labeled images. Best viewed in color.

module based on data. To conduct the graph cut efficiently, we leverage a single graph encoder for embeddings to jointly predict these two quantities. Then, we utilize a graph decoding process to estimate the linkage and density to determine edge connectivity and conduct the graph cut. Therefore, we detail the design of our unified joint linkage and density estimation model for graph cut. With the above considerations, we jointly predict the graph edge linkage and node density information based on a single full-graph inference. The subgraph prediction is passed through a graph decoding step.

Linkage and density for graph encoding. To effectively integrate the semantics-similar samples, we explore the relationship between different samples based on linkage and density. For each node feature f_i , we utilize a stack of GAT network layers to encode it as the embedding f'_i . For each edge (v_i, v_j) in the edge set E , we leverage the graph to concatenate the source and destination node features ($[f'_i, f'_j]$), where $i \neq j$ and $[\cdot, \cdot]$ denotes the concatenation operator. Then, we feed the concatenated features into a Multi-Layer Perceptron (MLP) layer [42] followed by a softmax transformation to produce the linkage probabilities: $p_{ij} = P(y_i = y_j)$, where p_{ij} is the estimate of the probability that two nodes linked by the edge share the same label. Besides, we introduce a density proxy d_i to measure the similarity-weighted proportion of same-subgraph nodes in its neighborhood for density estimation.

For any two nodes v_i and v_j , we first compute their similarity s_{ij} as the inner product of their respective node features: $s_{ij} = \langle f_i, f_j \rangle$. Then, we can obtain the corresponding edge coefficients as $e_{ij} = P(y_i = y_j) - P(y_i \neq y_j)$, where j denotes the j -th neighbor of v_i . To approximate the ground-truth density \bar{d}_i , we introduce the estimator $d_i = \frac{1}{k} \sum_{j=1}^k e_{ij} \cdot a_{ij}$. d_i can be obtained by updating e_{ij} with $e_{ij} = \mathbb{1}(y_i = y_j) - \mathbb{1}(y_i \neq y_j)$ using the ground-truth class labels, where $\mathbb{1}$ denotes the indicator function. An ideal density d_i is large if the most similar neighbors share the same labels; otherwise, it is small. By approximating \bar{d}_i based on e_{ij} and p_{ij} , the joint mechanism can reduce parameters for the prediction head during training.

Edge selection for graph decoding. During graph decoding, we convert them into final subgraphs by graph decoding. Especially, we tailor the graph decoding to incorporate our joint density and linkage estimates. Initially, we start with

$E' = \emptyset$. Given e_{ij} , d_i , p_{ij} and an edge connection threshold p_τ , we first define a candidate edge set $\mathcal{Z}(i)$ for node v_i as

$$\mathcal{Z}(i) = \{j | (v_i, v_j) \in E, p_{ij} \geq p_\tau \text{ and } d_i \leq d_j\}. \quad (2)$$

By the condition $d_i \leq d_j$, we introduce an inductive bias to establish connections. For any i , if its candidate set $\mathcal{Z}(i)$ is not empty, we pick $j = \arg\max_{j \in \mathcal{Z}(i)} e_{ij}$, and then add e_{ij} to E' . Based on Eq. (2), each node v_i with a non-empty $\mathcal{Z}(i)$ can add exactly one edge to E' . On the contrary, each node with an empty $\mathcal{Z}(i)$ becomes a peak node with no outgoing edges. The nodes with low density tend to 1) have some neighborhoods that overlap with other classes, 2) be near the boundary among different classes, 3) contain undesirable connections to other nodes. After a full pass over each node, E' forms a set of subgraphs G' , which serve as the designated subgraphs.

E. Intra-subgraph Label Assignment

After the graph cut, the unlabeled ID images and the labeled ID images will be clustered into the same subgraph. Therefore, we aim at assigning labels to unlabeled ID images based on the labeled ID images. After obtaining the subgraph indexes of all the samples $\mathcal{C} = \{c_{ij}\}_{i=1, j=1}^{N, K}$, we assign pseudo labels to unlabeled ID images based on the proportion of each label in a subgraph. At the t -th epoch, we denote the k -th subgraph as $\mathcal{D}_k^{(t)} = \{x_i | c_{ij}^{(t)} = k\}$, where x_i is its i -th sample. The set of ID-class labels is denoted as $\mathcal{Y}^{(t)} = \mathcal{Y}^L \cup \mathcal{Y}_{gc}^{(t)} \in \{0, 1, \dots, R-1\}$, where R denotes the number of ID classes, \mathcal{Y}^L is formed by the ground-truth labels from \mathcal{D}^L , and $\mathcal{Y}_{gc}^{(t)}$ denotes the ID pseudo labels assigned to unlabeled images. When $t = 0$, $\mathcal{Y}_{gc}^{(t)} = \emptyset$, while the ID pseudo labels will be added into $\mathcal{Y}_{gc}^{(t)}$ during training. Then, we define “labeled percentage” as the ratio of images belonging to class $y \in \mathcal{Y}^{(t)}$ in subgraph k : $r_{k,y}^{(t)} = |\mathcal{D}_{k,y}^{(t)}| / |\mathcal{D}_k^{(t)}|$, where $\mathcal{D}_{k,y}^{(t)} = \{x_i | c_{ij}^{(t)} = k, y_i = y\}$.

When most of samples in subgraph k belong to label y , i.e., $r_{k,c}^{(t)}$ is over a threshold ρ , we add all the unlabeled samples in subgraph k into labeled set \mathcal{D}^L to form the updated $\mathcal{D}^{L(t)} = \mathcal{D}^L \cup \mathcal{D}_{new}^{(t)}$, where $\mathcal{D}_{new}^{(t)} = \{x_i | x_i \in \mathcal{D}_k^{(t)}, r_{k,y}^{(t)} > \rho\}$. Therefore, we can utilize $\mathcal{D}^{L(t)}$ to update the label set $\mathcal{Y}^{(t)}$.

On the contrary, we denote the remaining unlabeled samples from \mathcal{D}^U at epoch t as $\mathcal{D}^{U(t)}$. By the reconstructed dataset, we introduce a classification loss \mathcal{L}_0 and an equalization loss \mathcal{L}_1 as follows:

$$\begin{aligned} \mathcal{L}_0 &= - \sum_t \sum_{x_i \in \mathcal{D}^{L(t)}} \sum_{y_i \in \mathcal{Y}^{(t)}} y_i \cdot \log(p(\mathbf{y}|x_i)), \\ \mathcal{L}_1 &= - \sum_t \sum_{x_i \in \mathcal{D}^{U(t)}} v \cdot \log(p(\mathbf{y}|x_i)), \end{aligned} \quad (3)$$

where $p(\mathbf{y}|x_i)$ denotes the prediction probability of x_i over all ID classes, y_i is converted to a one-hot label vector of x_i , $v = \mathbf{1}_R / R$ denotes the uniform posterior distribution over all of R ID classes, where $\mathbf{1}_R$ is an all-ones vector with R -dimension. Thus, $\mathcal{L}_1^{(t)}$ forces samples in $\mathcal{D}^{U(t)}$ to uniformly distribute among R ID classes.

F. Subgraph Aggregation for Cross-level Feature Update

To update the cross-level image features, we design a sub-

graph aggregation strategy. Especially, after label assignment, we can utilize these newly-labeled samples to update the feature encoder for better image features. To obtain $G_{l+1} = \{V_{l+1}, E_{l+1}\}$, we first convert any subgraph $c_i^{(l)}$ in G_l to node $v_i^{(l+1)}$ in V_{l+1} . For $v_i^{(l+1)}$, we introduce its two feature vectors: the identity feature $\tilde{f}_i^{(l+1)}$ and the average feature $\bar{f}_i^{(l+1)}$ defined as follows:

$$\tilde{f}_i^{(l+1)} = \tilde{f}_{m_i}^{(l)} \quad \text{and} \quad \bar{f}_i^{(l+1)} = \frac{1}{|c_i^{(l)}|} \sum_{j \in c_i^{(l)}} \tilde{f}_j^{(l)}, \quad (4)$$

where m_i denotes the peak node index of the subgraph $c_i^{(l)}$, which is defined as: $m_i = \operatorname{argmax}_{j \in c_i^{(l)}} \hat{d}_j^{(l)}$. Especially, for the first level, $\tilde{f}_i^{(0)} = \bar{f}_i^{(0)} = f_i$, where f_i is the visual embedding feature. Thus, we utilize the identity feature $\tilde{f}_i^{(l)}$ to identify similar nodes across hierarchies, and use the average feature $\bar{f}_i^{(l)}$ to extract the global information for all nodes in the subgraph. The next-level input feature of node $v_i^{(l+1)}$ is the concatenation of the peak feature and the average feature, i.e., $f_i^{l+1} = [\tilde{f}_i^{(l+1)}, \bar{f}_i^{(l+1)}]$.

G. Data Augmentation for Model Generalization

In real-world applications, the supervised classification methods only aim to extract semantic representations to meet the minimum necessary, and they often overtrust the given labels. However, in the challenging OOD detection task, we cannot overtrust collected samples including OOD samples. We tend to widen the discrepancy between ID/OOD on similarity metric, which can strengthen the ability to guide ID/OOD samples to different subgraphs. By building an unsupervised training module, we can extract enhanced features.

Therefore, we leverage the data augmentation technology to obtain different augmented images. Then, for each image, we maximize the similarity between its different augmented versions. To obtain the augmented images of all the images, we first integrate N^L labeled samples $\{x_i^L\}_{i=1}^{N^L}$ and N^U unlabeled images $\{x_i^U\}_{i=1}^{N^U}$ into a total set $X = \{x_i\}_{i=1}^{N^L+N^U}$. Then, we augment any x_i into two versions: x_i^0 and x_i^1 , where $x_i^0 = A^0(x_i)$ and $x_i^1 = A^1(x_i)$; A^0 and A^1 denote two augmentations. Finally, we feed x_i^0 and x_i^1 into a shared feature encoder followed by an MLP layer. For convenience, we denote the augmented features as b_i^0 and b_i^1 , respectively.

For the t -th epoch, we maximize the cosine similarity of the same samples in different augmented features ($\{b_i^0\}^{(t)}$ and $\{b_i^1\}^{(t)}$) to enhance the sample feature representation. Thus, we introduce the InfoNCE loss as the objective:

$$\mathcal{L}_2 = -\sum_{i=1}^N \log\left(\frac{\exp(\mu(b_i^0, b_i^1))}{\sum_{j=1}^N \exp(\mu(b_i^0, b_j))} + \frac{\exp(\mu(b_i^1, b_i^0))}{\sum_{g=1}^N \exp(\mu(b_i^1, b_g))}\right), \quad (5)$$

where $b_j \in B^{1,(t)}$, $b_g \in B^{0,(t)}$, and $\mu(\cdot, \cdot)$ denotes the cosine similarity function. Based on \mathcal{L}_2 , we map the enhanced representation into the lower-dimensional logit space, where the class-specific similarity score can illustrate the semantic discrepancy of samples from different classes, and assign a more discriminative subgraph distribution to them.

Therefore, our overall loss is as follows:

$$\mathcal{L} = \mathcal{L}_0 + \beta \mathcal{L}_1 + \alpha \mathcal{L}_2 + \gamma \mathcal{L}_3^L, \quad (6)$$

where α , β and γ are parameters to balance different losses.

H. Inference for Any Image

Given an unlabeled image, a well-designed OOD detection model should correctly generate its prediction. With a classification head h_1 , we leverage a fully-connected layer to map the image features into a logit space. We utilize the energy-based approach with temperature-scaled logits for OOD detection. After training, we can utilize the similarity score from the logit $l(\mathbf{y}|x_i) = h_1(b_i)$ as the OOD score. The similarity score is formulated as: $g(x_i) = T \cdot \log \sum_{c=1}^C \exp(l(y_c|x_i)/T)$, where $l(y_m|x_i)$ is the logit of x_i belonging to the ID class y_m , and ID samples have higher similarity score.

Unlike traditional OOD detection methods [7]–[12] that only train on ID samples, we pay more attention to OOD samples. These OOD samples will be more concentrated at minimum similarity due to the involvement of training OOD data under the SC-OOD setting. Thus, a proper temperature value T can smooth the similarity distribution and widen the discrepancy between ID/OOD on the similarity metric.

Therefore, we compare the similarity score to the predefined threshold δ . Any image x_i is classified as in-distribution if the softmax score is greater than the threshold and vice versa. Mathematically, the out-of-distribution detector can be described as:

$$x_i \text{ is } \begin{cases} \text{OOD sample,} & \text{if } g(x_i) \leq \delta, \\ \text{ID sample,} & \text{if } g(x_i) > \delta, \end{cases} \quad (7)$$

where δ is chosen so that the true positive rate (i.e., the fraction of ID images correctly classified as ID images) is 95%.

V. EXPERIMENTS

A. Experimental Settings

Datasets. Following [6], we use two SC-OOD benchmarks: CIFAR-10 and CIFAR-100. Each benchmark contains seven datasets: CIFAR-10 [43], CIFAR-100 [43], Texture [44], SVHN [45], Tiny-ImageNet (Tiny-I) [46], LSUN [47] and Places365 [48]. Each benchmark treats a dataset (CIFAR-10 or CIFAR-100) as the labeled dataset, and the others as unlabeled datasets. The statistics of these datasets are reported in supplementary materials.

Evaluation metrics. We follow [6] to using the following evaluation metrics: 1) FPR95: It denotes the rate of falsely recognized OOD when 95% ID samples are recalled. 2) AUROC: It calculates the area by plotting the TPR as a function of the FPR. 3) AUPR: It measures the area under the curve by plotting recall as a function of precision. Also, AUPR is a threshold-independent. Based on different selections of positiveness, we consider two evaluation metrics: AUPR(In) (ID samples are positive) and AUPR(Out) (OOD samples are positive). 4) CCR@FPR m : It denotes Correct Classification Rate (CCR) for ID classification when the FPR equals m . Note that the value ranges for all the evaluation metrics are [0%, 100%]. For a remarkable comparison, the performance improvement in our paper denotes the performance difference between two methods in terms of the above evaluation metrics.

TABLE I: Performance comparison on the CIFAR-10 benchmark, where data partitions are identical for all methods. Our AHGC achieves consistently the best performance on all OOD detection metrics. \uparrow/\downarrow means a higher/lower value is better.

Dataset	Method	FPR95 \downarrow	AUROC \uparrow	AUPR(In/Out) \uparrow	CCR@FPR \uparrow			
					10 $^{-4}$	10 $^{-3}$	10 $^{-2}$	10 $^{-1}$
Texture	ODIN	42.52	84.06	86.01 / 80.73	0.02	0.18	3.71	40.14
	ConjNorm	23.78	95.59	97.40 / 95.88	18.42	45.29	70.96	89.73
	UDG	20.43	96.44	98.12 / 92.91	19.90	43.33	69.19	87.71
	Ours	0.89	99.42	99.67 / 99.05	54.03	75.02	87.81	94.62
SVHN	ODIN	52.27	83.26	63.76 / 92.60	1.01	4.00	11.82	44.85
	UDG	13.26	97.49	95.66 / 98.69	36.64	56.81	76.77	89.54
	ConjNorm	9.89	96.32	94.88 / 96.25	33.82	67.94	80.31	91.72
	Ours	0.00	99.99	99.99 / 100.00	91.89	94.97	95.37	95.40
CIFAR-100	ODIN	56.34	78.40	73.21 / 80.99	0.10	0.38	4.43	30.11
	UDG	47.20	90.98	91.74 / 89.36	1.50	10.94	40.34	75.89
	Scone	43.79	91.24	90.68 / 88.76	2.38	8.17	42.96	77.94
	Ours	24.82	92.72	93.50 / 91.69	3.81	21.00	47.87	80.04
Tiny-I	ODIN	59.09	79.69	79.34 / 77.52	0.36	0.63	4.49	34.52
	ConjNorm	52.80	90.74	83.25 / 80.24	9.52	19.37	59.88	79.45
	UDG	50.18	91.91	94.43 / 86.99	0.32	23.15	53.96	78.36
	Ours	9.71	96.45	97.94 / 92.96	72.84	81.85	85.22	86.67
LSUN	ODIN	47.85	84.56	81.56 / 85.58	0.21	0.85	9.92	46.95
	UDG	42.05	93.21	94.53 / 91.03	14.26	37.59	60.62	81.69
	ConjNorm	38.29	93.67	95.82 / 92.89	13.84	41.95	64.31	88.72
	Ours	1.72	98.71	98.84 / 98.63	53.97	69.41	82.55	93.14
Places365	ODIN	53.94	82.01	54.92 / 93.30	0.47	1.68	7.13	39.63
	UDG	44.22	92.64	87.17 / 96.66	10.62	35.05	58.96	79.63
	ConjNorm	36.02	93.58	86.72 / 95.85	12.40	40.10	60.70	82.34
	Ours	10.24	97.24	94.74 / 98.82	37.19	59.12	76.67	89.54
Mean	ODIN	52.00	82.00	73.13 / 85.12	0.36	1.29	6.92	39.37
	UDG	36.22	93.78	93.61 / 92.61	13.87	34.48	59.97	82.14
	ConjNorm	31.95	92.89	91.54 / 91.72	14.84	38.07	62.67	84.58
	Ours	7.90	97.42	97.45 / 96.86	52.29	66.90	79.25	89.90

Implementation details. We utilize the standard ResNet-18 network as the feature encoder. We use the standard SGD optimizer with a weight decay of 0.0005 and a momentum of 0.9 to train it on two SC-OOD benchmarks: the CIFAR-10 benchmark and the CIFAR-100 benchmark. The two benchmarks treat CIFAR-10 and CIFAR-100 as the labeled datasets in their corresponding benchmarks, respectively. Besides, the Tiny-ImageNet training dataset is utilized as the external unlabeled dataset \mathcal{D}^U in both benchmarks during the training process. We utilize a cosine learning rate scheduler with an initial learning rate of 0.1, taking totally 180 epochs. For the labeled dataset \mathcal{D}^L , its batch size is 64, while the batch size for the unlabeled dataset \mathcal{D}^U is 128. The total subgraph number is 1500 and 1800 for CIFAR-10 and CIFAR-100 benchmarks, respectively. For hyper-parameters, we set $\alpha = 0.15, \beta = 0.5, \gamma = 10^{-4}$. All experiments are implemented by PyTorch. Codes are available in [Github](#).

B. Comparison with State-of-the-arts

To evaluate the effectiveness of our proposed AHGC, we follow [6] to compare AHGC with the following open-source state-of-the-art OOD detection methods: ODIN [7], UDG [6], and ConjNorm [31]. ODIN and ConjNorm directly treat one labeled dataset as ID and all unlabeled datasets as OOD. UDG and our AHGC treat some unlabeled samples shared similar

TABLE II: Performance comparison on the CIFAR-100 benchmark, where data partitions are identical for all methods. Our AHGC achieves consistently the best performance on all OOD detection metrics. \uparrow/\downarrow means a higher/lower value is better.

Dataset	Method	FPR95 \downarrow	AUROC \uparrow	AUPR(In/Out) \uparrow	CCR@FPR \uparrow			
					10 $^{-4}$	10 $^{-3}$	10 $^{-2}$	10 $^{-1}$
Texture	ODIN	79.47	77.92	86.69 / 62.97	2.66	4.66	15.09	45.82
	ConjNorm	77.21	76.39	85.72 / 62.83	0.45	3.27	11.80	42.74
	UDG	75.04	79.53	87.63 / 65.49	1.97	4.36	9.49	33.84
	Ours	22.93	84.64	91.58 / 67.42	0.91	9.41	30.64	56.73
SVHN	ODIN	90.33	75.59	65.25 / 84.49	4.98	12.02	23.79	46.61
	UDG	60.00	88.25	81.46 / 93.63	14.90	25.50	38.79	56.46
	ConjNorm	59.32	82.17	79.32 / 90.44	13.05	23.85	30.92	58.40
	Ours	0.46	97.70	97.75 / 97.36	61.34	69.62	72.82	74.36
CIFAR-10	ConjNorm	85.87	72.55	75.50 / 65.38	1.29	4.76	13.85	42.17
	UDG	83.35	76.18	78.92 / 71.15	1.99	5.58	17.27	42.11
	ODIN	81.82	77.90	79.93 / 73.39	0.09	3.69	15.39	47.20
	Ours	43.98	74.47	78.21 / 67.13	2.46	6.88	16.96	44.42
Tiny-I	ODIN	82.74	77.58	86.26 / 61.38	0.20	3.78	15.99	45.56
	ConjNorm	82.30	79.42	87.96 / 60.44	1.73	13.59	20.88	44.10
	UDG	81.73	77.18	86.00 / 61.67	0.67	4.82	17.80	41.72
	Ours	0.49	89.19	95.01 / 73.42	50.02	57.76	59.66	60.79
LSUN	ODIN	80.57	78.22	86.34 / 63.44	1.68	5.59	17.37	45.56
	ConjNorm	79.82	77.29	85.17 / 61.98	2.84	6.08	20.95	44.88
	UDG	78.70	76.79	84.74 / 63.05	1.59	5.34	18.04	44.70
	Ours	25.15	80.42	88.88 / 62.18	7.56	19.15	30.26	49.71
Places365	ODIN	76.42	80.66	66.77 / 89.66	1.45	4.16	18.98	49.60
	UDG	73.86	79.87	65.36 / 89.60	1.96	6.33	22.03	47.97
	ConjNorm	73.34	78.86	67.40 / 88.27	2.87	7.34	21.99	50.43
	Ours	27.10	81.73	71.82 / 89.79	3.93	10.93	27.15	52.44
Mean	ODIN	81.89	77.98	78.54 / 72.56	1.84	5.65	17.77	46.73
	ConjNorm	76.31	77.78	80.18 / 71.56	3.71	9.82	20.07	47.12
	UDG	75.45	79.63	80.69 / 74.10	3.85	8.66	20.57	44.47
	Ours	20.02	84.69	87.21 / 76.22	21.04	28.96	39.58	56.41

semantics with labeled samples as ID.

For convenience, we directly cite some results in [6] from corresponding works, which are trained and tested on the challenging benchmarks. As shown in Table I and Table II, we compare our proposed AHGC with state-of-the-art methods, where AHGC reaches the best results in most cases. Particularly, when we treat the Tiny-ImageNet as the unlabeled dataset, our AHGC achieves the following amazing improvement compared with the second-best method UDG: 1) In the CIFAR-10 benchmark, AHGC improves the performance by 40.47% in terms of “FPR95”. 2) In the CIFAR-100 benchmark, AHGC outperforms compared methods by 81.24% in terms of “FPR95”. The main reason is that our AHGC can well understand the latent semantics between multi-granularity images on different datasets for OOD detection, while UDG only conducts K-means clustering on different images, which fails to understand the latent relationship between labeled and unlabeled samples with different granularity.

Analysis. From the above results, we can observe that: (i) In all the datasets, our proposed AHGC on these two benchmarks (CIFAR-10 and CIFAR-100) outperforms these state-of-the-art methods by a large margin in most cases. There are two main reasons as follows: Firstly, our AHGC is a more effective model, which leverages the graph cut to explore the latent relationship of different samples under the same semantics.

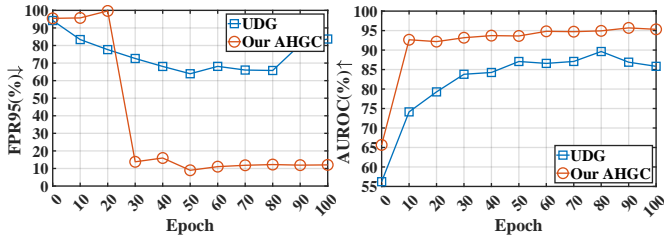


Fig. 4: Performance (%) of our proposed AHGC and UDG [6] on CIFAR-10 benchmark during training process. Since the maximum number of epochs on UDG is 100, we report our performance of the first 100 epochs for fair comparison.

Secondly, we focus on the challenging SC-OOD setting, while many compared methods directly treat one labeled dataset as ID and all unlabeled datasets as OOD. The better performance in the SC-OOD setting shows the effectiveness of our AHGC. (ii) Although UDG is also under the SC-OOD setting, it is defeated by Scone in many cases because UDG cannot mine the latent relationship between multi-granularity images by simple K-means clustering. (iii) In Table II, for the CIFAR-10 dataset, the traditional method (ODIN) performs well in terms of some metrics. The main reason is that all the images in the CIFAR-100 dataset disjoint from the images in the CIFAR-10 dataset, which exactly meets their assumptions that labeled CIFAR-100 images are ID and unlabeled CIFAR-10 images are OOD. However, ODIN achieves unsatisfactory performance in the other unlabeled datasets (Texture, SVHN, Tiny-ImageNet, LSUN, Places365). This illustrates the significance of mining the unlabeled ID samples. (iv) These compared methods generally perform well on SVHN and Texture datasets, while they often suffer a defeat on the Tiny-ImageNet, LSUN and Places365 datasets. It is because the images on SVHN and Texture datasets have relatively flat backgrounds, which are quite different in style from those in CIFAR10/CIFAR-100, and the resulting covariate shifts make it easier for the model to identify them as OOD examples. Our AHGC especially achieves performance improvement on all the datasets in most cases. This indicates that more exploration between multi-granularity images is required to overcome the interference of covariate shifts in the OOD detection tasks. About the model learnability [49], we utilize finite samples to design a hypothesis function with approximately minimal risk. As the sample size increases, we can obtain better approximation based on the proposed AHGC. Therefore, our proposed AHGC satisfies the learnability condition [50], and AHGC can effectively address the multi-granularity OOD detection task.

Performance comparison during training process. To further analyze the performance of our proposed AHGC, we report their performance during the training process. For convenience, we report their performance in every 10 epochs. Fig. 4 shows the results. Obviously, our AHGC outperforms UDG with large margins after 10 epochs for all the cases, which shows the effectiveness of our AHGC. Besides, our AHGC can converge faster than UDG.

TABLE III: Main ablation study on CIFAR-10 benchmark, where we utilize energy-based similarity score with temperature scaling to distinguish ID/OOD samples. We report the mean values of all unlabeled datasets.

Modules	FPR95(%)↓	AUROC(%)↑	AUPR(In/Out)(%)↑
AHGC(a)	41.28	83.80	83.46 / 80.12
AHGC(b)	18.45	87.36	86.47 / 88.72
AHGC(c)	13.78	91.30	92.57 / 92.34
AHGC(full)	7.90	97.42	97.45 / 96.86

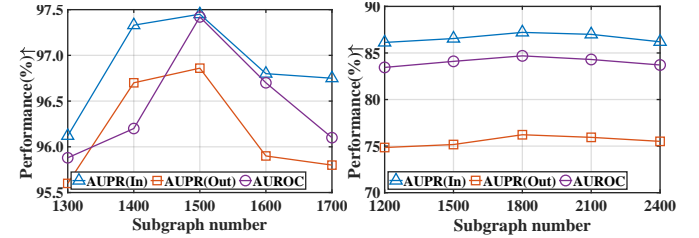


Fig. 5: Effect of subgraph number K on the CIFAR-10 benchmark (left) and for the CIFAR-100 benchmark (right).

TABLE IV: Ablation study on GAT on the CIFAR-10 benchmarks. We report the mean values of all unlabeled datasets.

Modules	FPR95(%)↓	AUROC(%)↑	AUPR(In/Out)(%)↑
GNN	23.85	89.74	90.11 / 88.72
Graphsage	16.08	86.07	87.10 / 86.09
GAT	7.90	97.42	97.45 / 96.86

C. Ablation Study

1) *Main ablation study:* As shown in Table III, we first conduct the main ablation study to examine the effectiveness of all the main modules in our model. Therefore, we design the following ablation models: (i) AHGC(a): we only utilize the classification loss \mathcal{L}_0 for OOD detection. (ii) AHGC(b): we combine \mathcal{L}_0 and \mathcal{L}_1 for OOD detection. (iii) AHGC(c): we joint the classification loss \mathcal{L}_0 and the InfoNCE loss \mathcal{L}_2 in data augmentation for OOD detection. (iv) AHGC(full): we use our full AHGC model. Obviously, our AHGC(full) outperforms the other ablation models on both benchmarks, which shows the effectiveness of each module. It is because our subgraph aggregation module can close the discrepancy between labeled and unlabeled images with the same semantics to transfer the label knowledge; our intra-subgraph label assignment module can reduce the semantic gaps between different datasets in each subgraph semantics; the data augmentation module is able to learn more fine-grained and representative image features for graph cut different datasets.

2) *Effect of subgraph number K :* We further conduct an ablation study to analyze the impact of the subgraph numbers K . As shown in Fig. 5, the optimal subgraph number is $K = 1500$ for the CIFAR-10 benchmark and $K = 1800$ for the CIFAR-100 benchmark.

3) *Impact of graph attention network (GAT):* To explore our utilized graph attention network, we conduct a corresponding ablation study. We compare two popularly-used graph cut net-

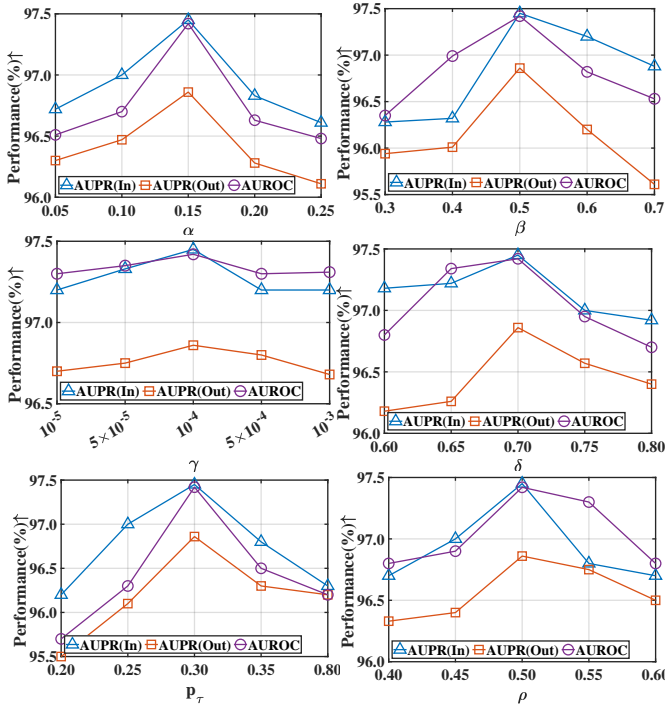


Fig. 6: Effect of parameter on the CIFAR-10 benchmark.

works: GNN [51] and Graphsage [52] in Table IV. Obviously, the GAT module is more effective than GNN and Graphsage on the OOD detection task.

4) *Influence of hyper-parameters* : We conduct an ablation study to analyze hyper-parameters (α , β , γ , δ , p_τ and ρ) in Fig. 6. The optimal values for these parameters are: $\alpha = 0.15$, $\beta = 0.5$, $\gamma = 10^{-4}$, $\delta = 0.7$, $p_\tau = 0.3$ and $\rho = 0.5$.

VI. CONCLUSION

In this paper, we propose a novel Adaptive Hierarchical Graph Cut network (AHGC) to deeply explore the semantics relationship between labeled and unlabeled samples with different granularity. Experimental results show the effectiveness of our AHGC.

REFERENCES

- [1] B. Arslan, S. Memiş, E. B. Sönmez, and O. Z. Batur, “Fine-grained food classification methods on the ucf food-100 database,” *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 2, pp. 238–243, 2021.
- [2] H. Zhang, X. Ning, C. Wang, E. Ning, and L. Li, “Deformation depth decoupling network for point cloud domain adaptation,” *Neural Networks*, vol. 180, p. 106626, 2024.
- [3] L. Jiao, J. Gao, X. Liu, F. Liu, S. Yang, and B. Hou, “Multiscale representation learning for image classification: A survey,” *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 1, pp. 23–43, 2021.
- [4] F. Lu, K. Zhu, W. Zhai, K. Zheng, and Y. Cao, “Uncertainty-aware optimal transport for semantically coherent out-of-distribution detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3282–3291.
- [5] D. Hendrycks and K. Gimpel, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” in *International Conference on Learning Representations*, 2016.
- [6] J. Yang, H. Wang, L. Feng, X. Yan, H. Zheng, W. Zhang, and Z. Liu, “Semantically coherent out-of-distribution detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8301–8309.

- [7] S. Liang, Y. Li, and R. Srikant, “Enhancing the reliability of out-of-distribution image detection in neural networks,” in *International Conference on Learning Representations*, 2018.
- [8] W. Liu, X. Wang, J. Owens, and Y. Li, “Energy-based out-of-distribution detection,” *Advances in neural information processing systems*, vol. 33, pp. 21 464–21 475, 2020.
- [9] Y. Sun, C. Guo, and Y. Li, “React: Out-of-distribution detection with rectified activations,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 144–157, 2021.
- [10] K. Lee, K. Lee, H. Lee, and J. Shin, “A simple unified framework for detecting out-of-distribution samples and adversarial attacks,” *Advances in neural information processing systems*, vol. 31, 2018.
- [11] S. Mohseni, M. Pitale, J. Yadawa, and Z. Wang, “Self-supervised learning for generalizable out-of-distribution detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5216–5223.
- [12] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, and T. L. Willke, “Out-of-distribution detection using an ensemble of self supervised leave-out classifiers,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 550–564.
- [13] Q. Yu and K. Aizawa, “Unsupervised out-of-distribution detection by maximum classifier discrepancy,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9518–9526.
- [14] A. Zaeemzadeh, N. Bisagno, Z. Sambugaro, N. Conci, N. Rahnavard, and M. Shah, “Out-of-distribution detection using union of 1-dimensional subspaces,” in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2021, pp. 9452–9461.
- [15] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira, “Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10951–10960.
- [16] Y. Ming, Y. Sun, O. Dia, and Y. Li, “How to exploit hyperspherical embeddings for out-of-distribution detection?” in *The Eleventh International Conference on Learning Representations*, 2023.
- [17] G. Chen, P. Peng, X. Wang, and Y. Tian, “Adversarial reciprocal points learning for open set recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8065–8081, 2021.
- [18] X. Du, Z. Wang, M. Cai, and Y. Li, “Vos: Learning what you don’t know by virtual outlier synthesis,” in *ICLR*, 2021.
- [19] K. Lee, H. Lee, K. Lee, and J. Shin, “Training confidence-calibrated classifiers for detecting out-of-distribution samples,” in *6th International Conference on Learning Representations, ICLR 2018*, 2018.
- [20] K. Sricharan and A. Srivastava, “Building robust classifiers through generation of confident out of distribution examples,” *arXiv*, 2018.
- [21] S. Vernekar, A. Gaurav, V. Abdelzad, T. Denouden, R. Salay, and K. Czarnecki, “Out-of-distribution detection in classifiers via generation,” *arXiv preprint arXiv:1910.04241*, 2019.
- [22] R. Kaur, S. Jha, A. Roy, S. Park, E. Dobriban, O. Sokolsky, and I. Lee, “idecode: In-distribution equivariance for conformal out-of-distribution detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, 2022, pp. 7104–7114.
- [23] K. Bibas, M. Feder, and T. Hassner, “Single layer predictive normalized maximum likelihood for out-of-distribution detection,” *NeurIPS*, 2021.
- [24] K. Lee, K. Lee, K. Min, Y. Zhang, J. Shin, and H. Lee, “Hierarchical novelty detection for visual object recognition,” in *CVPR*, 2018.
- [25] P. Kirichenko, P. Izmailov, and A. G. Wilson, “Why normalizing flows fail to detect out-of-distribution data,” *NeurIPS*, 2020.
- [26] J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J. F. Núñez, and J. Luque, “Input complexity and out-of-distribution detection with likelihood-based generative models,” in *ICLR*, 2019.
- [27] E. Techapanurak, M. Suganuma, and T. Okatani, “Hyperparameter-free out-of-distribution detection using cosine similarity,” in *ACCV*, 2020.
- [28] Y. Zhou, “Rethinking reconstruction autoencoder-based out-of-distribution detection,” in *CVPR*, 2022, pp. 7379–7387.
- [29] Y. Yang, R. Gao, and Q. Xu, “Out-of-distribution detection with semantic mismatch under masking,” in *European Conference on Computer Vision*. Springer, 2022, pp. 373–390.
- [30] Y. Ming, Y. Fan, and Y. Li, “Poem: Out-of-distribution detection with posterior sampling,” in *International Conference on Machine Learning*, PMLR, 2022, pp. 15 650–15 665.
- [31] B. Peng, Y. Luo, Y. Zhang, Y. Li, and Z. Fang, “Conjnorm: Tractable density estimation for out-of-distribution detection,” in *The Twelfth International Conference on Learning Representations*, 2024.
- [32] C. Zou, A. Han, L. Lin, M. Li, and J. Gao, “A simple yet effective framelet-based graph neural network for directed graphs,” *IEEE Transactions on Artificial Intelligence*, 2023.

- [33] Y. Boykov and O. Veksler, "Graph cuts in vision and graphics: Theories and applications," in *Handbook of mathematical models in computer vision*. Springer, 2006, pp. 79–96.
- [34] H. Zhang, C. Wang, L. Yu, S. Tian, X. Ning, and J. Rodrigues, "Pointgt: A method for point-cloud classification and segmentation based on local geometric transformation," *IEEE Transactions on Multimedia*, 2024.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
- [36] P. Singh, A. Kaul, and S. Ganapathy, "Supervised hierarchical clustering using graph neural networks for speaker diarization," in *ICASSP*, 2023.
- [37] Y. Xing, T. He, T. Xiao, Y. Wang, Y. Xiong, W. Xia, D. Wipf, Z. Zhang, and S. Soatto, "Learning hierarchical graph neural networks for image clustering," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 3467–3477.
- [38] D. Zou, W. Wei, Z. Wang, X.-L. Mao, F. Zhu, R. Fang, and D. Chen, "Improving knowledge-aware recommendation with multi-level interactive contrastive learning," in *CIKM*, 2022.
- [39] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.
- [40] K. Fu, C. Gong, I. Y.-H. Gu, and J. Yang, "Normalized cut-based saliency detection by adaptive multi-level region merging," *IEEE TIP*, vol. 24, no. 12, pp. 5671–5683, 2015.
- [41] N. Khan, U. Chaudhuri, B. Banerjee, and S. Chaudhuri, "Graph convolutional network for multi-label vhr remote sensing scene recognition," *Neurocomputing*, vol. 357, pp. 36–46, 2019.
- [42] F. Rosenblatt *et al.*, *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Spartan books Washington, DC, 1962.
- [43] A. Krizhevsky, "Learning multiple layers of features from tiny images," *CiteSeer*, 2009.
- [44] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *CVPR*, 2014, pp. 3606–3613.
- [45] N. Yuval, "Reading digits in natural images with unsupervised feature learning," in *Proceedings of the NIPS Workshop*, 2011.
- [46] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.
- [47] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.
- [48] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE TPAMI*, vol. 40, no. 6, 2018.
- [49] Z. Fang, Y. Li, J. Lu, J. Dong, B. Han, and F. Liu, "Is out-of-distribution detection learnable?" *Advances in Neural Information Processing Systems*, vol. 35, pp. 37 199–37 213, 2022.
- [50] Z. Fang, Y. Li, F. Liu, B. Han, and J. Lu, "On the learnability of out-of-distribution detection," *Journal of Machine Learning Research*, vol. 25, 2024.
- [51] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.
- [52] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.



Blaise Genest is a former student of ENS Cachan in mathematics and computer science in 1999-2003, and received the PhD degree in computer science from University Paris 7, France in 2004. He is a full time permanent researcher at French CNRS in computer science, and currently in the Joint French-Singapore Laboratory IPAL. His current research interests include verification of distributed systems and computational systems biology.



Ponnuthurai Nagaratnam Suganthan (Fellow, IEEE) received an honorary doctorate (i.e. Doctor Honoris Causa) in 2020 from University of Maribor, Slovenia. Since August 2022, he has been a Research Professor with the KINDI Centre for Computing Research, Qatar University, Doha, Qatar. His research interests include randomization-based learning methods, swarm and evolutionary algorithms, pattern recognition and deep learning.



Xiang Fang received the M.E. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, in 2020. He is currently working toward the Ph.D. degree at the Energy Research Institute @ NTU, Interdisciplinary Graduate Programme, Nanyang Technological University, Singapore. His research interests include out-of-distribution detection, multi-modal learning, etc.



Arvind Easwaran received the PhD degree in computer and information science from the University of Pennsylvania, in 2008. He is currently an associate professor in the College of Computing and Data Science, Nanyang Technological University, Singapore, where he joined in 2013. His research interests include cyber-physical systems, embedded real-time systems, and formal methods.