Topological Structure Learning for Weakly-Supervised Out-of-Distribution Detection

Rundong He∗
School of Software, Shandong University
Jinan, Shandong, China
rundong_he@mail.sdu.edu.cn

Rongxue Li∗
School of Software, Shandong University
Jinan, Shandong, China
rongxue2022@gmail.com

Zhongyi Han†
Machine Learning Department, MBZUAI
Masdar City, Abu Dhabi, United Arab Emirates
hanzhongyicn@gmail.com

Xihong Yang
National University of Defense Technology
Changsha, Hunan, China
yangxihong@nudt.edu.cn

Yilong Yin‡
School of Software, Shandong University
Jinan, Shandong, China
ylyin@sdu.edu.cn

ABSTRACT

Out-of-distribution (OOD) detection is the key to deploying models safely in the open world. For OOD detection, collecting sufficient in-distribution (ID) labeled data is usually more time-consuming and costly than unlabeled data. When ID labeled data is limited, the previous OOD detection methods are no longer superior due to their high dependence on the amount of ID labeled data. Based on limited ID labeled data and sufficient unlabeled data, we define a new setting called Weakly-Supervised Out-of-Distribution Detection (WSOOD). To solve the new problem, we propose an effective method called Topological Structure Learning (TSL). Firstly, TSL uses a contrastive learning method to build the initial topological structure space for ID and OOD data. Secondly, TSL mines effective topological connections in the initial topological space. Finally, based on limited ID labeled data and mined topological connections, TSL reconstructs the topological structure in a new topological space to increase the separability of ID and OOD instances. Extensive studies on several representative datasets show that TSL remarkably outperforms the state-of-the-art, verifying the validity and robustness of our method in the new setting of WSOOD.

KEYWORDS

Out-of-distribution detection, Weakly-supervised learning, Topological structure learning

ACM Reference Format:

Rundong He, Rongxue Li, Zhongyi Han, Xihong Yang, and Yilong Yin. 2023. Topological Structure Learning for Weakly-Supervised Out-of-Distribution Detection. In Proceedings of the 31st ACM International Conference on Multimedia (MM ’23), October 29-November 3, 2023, Ottawa, ON, Canada. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3612195

1 INTRODUCTION

The deep neural networks have achieved considerable success in the scenario where the training and testing data are sampled from an identical distribution [23–25, 30–32]. However, in many real applications, the assumption cannot be satisfied due to the existence of unknowns. A reliable classification model ought to own the ability to say “I do not know” to out-of-distribution (OOD) data that the model has not seen before, which is the key to deploying models safely in the real world [18, 29]. For example, a wildlife monitoring system with the ability to detect OOD data can help doctors discover rare and novel diseases and prevent patients missing the best treatment period. In autonomous driving, OOD detection enables cars to evoke human control of driving in an emergency or unknown scenarios [8, 20], which contributes to safer and more reliable autonomous driving.

OOD detection has received much attention because of its significance, and plenty of methods have emerged. The existing OOD detection methods can be divided into two main categories: classification-based OOD detection methods and density-based OOD detection methods. The classification-based methods contain post-hoc based methods [9, 15] and fine-tuning based methods [10, 16]. Hendrycks and Gimpel [9] detected OOD data with the softmax confidence score. Liang et al. [15] used temperature scaling and input perturbation to amplify the OOD’s separability. Hendrycks et al. [10] and Liu et al. [16] fine-tuned the model by introducing a large-scale
We refer to this problem as Weakly-Supervised Out-of-Distribution (WSOOD) because previous methods assume that ID labeled data is sufficient. The mining process of positive pairs is too crude and it is difficult to judge the connections between positive pairs. Different positive pairs have different levels of closeness. STEP considers all other instances except itself as its negative instances, which is unreasonable and may be in conflict with positive pairs. Moreover, STEP does not fully exploit adequate information in labeled data.

To solve the existing problems and boost WSOOD, we propose Topological Structure Learning (TSL). Firstly, TSL uses SimCLR [2] to obtain reliable feature representations on all labeled and unlabeled data, which help distinguish between ID and OOD and construct points for the topological space. Secondly, to enhance the reliability of mined pairs from all the data for the next topological structure reconstruction, TSL mines close positive pairs and loose positive pairs according to reciprocal and non-reciprocal neighbors. At the same time, TSL mines reliable negative pairs in the complement of positive pairs according to the Mahalanobis [17] distance. Then, TSL reconstructs a new topological structure in a new topological space to increase the separability of ID and OOD. When constructing the new topological structure, TSL performs two steps. The first is the maintenance of the topological skeleton. TSL maintains the topological skeleton by reducing the intra-class Mahalanobis distance of limited ID labeled data. The second is to extend and reconstruct the whole topological structure in the new topological space based on the topological skeleton and topological pairs mined before. TSL adjusts the distance between instances with different credibility depending on the types of topological structure pairs mined before, including close positive pairs, loose positive pairs, and negative pairs.

The following are the main contributions of this work:

- We investigate a new setting called weakly-supervised out-of-distribution detection, analyze why this problem is difficult, and propose a novel method called TSL.
- We mine close positive pairs and loose positive pairs according to reciprocal and non-reciprocal neighbors. And we mine negative pairs in the complement of positive pairs according to the Mahalanobis distance.
- We propose to reconstruct the topological structure in a new topological space by constructing a topological skeleton and extending the topological skeleton to increase the separability of ID and OOD instances.
- We carry out extensive experiments on several benchmark datasets. Experimental results verify the effectiveness and robustness of TSL.

2 RELATED WORK

In this section, we mainly introduce OOD detection methods with unlabeled data.

2.1 OOD Detection with Unlabeled Data

Fine-tuning has been a popular paradigm for OOD detection, and among methods of fine-tuning, OOD detection by introducing unlabeled data attracts more and more attention. Fine-tuning based OOD detection methods with unlabeled data boost the performance of OOD detection by introducing unlabeled data. The unlabeled and test data are sampled from an identical distribution that contains both ID and OOD classes. Large-scale unlabeled data is convenient and inexpensive to obtain. ID and OOD signals embedded in unlabeled data can mitigate the concerns of limited ID labeled data and...
contribute to amplifying ID/OOD separability. The existing OOD detection methods with unlabeled data include UOOD [34], SCOOD [28], STEP [36], GBND [22], etc. UOOD [34] uses unlabeled data to maximize the discrepancy between the decision boundaries of the two classifiers after the public feature extractor and push OOD data outside of ID decision boundary. The key to SCOOD [28] is to select ID instances from unlabeled data by unsupervised clustering of ID labeled data and unlabeled data. STEP [36] obtains reliable features of ID labeled data and unlabeled data through a simple contrastive learning method, SimCLR. It then mines positive and negative pairs for contrast learning, thus keeping OOD instances away from ID instances and obtaining good OOD detection performance. GBND [22] constructs a gradient-based OOD detector, extracting OOD instances from unlabeled instances using the Mahalanobis distance. GBND obtains good OOD performance by using better and better OOD signals in the process of selecting OOD instances over and over again. [12] proposes a framework for constrained optimization on unlabeled wild data mixed by ID and OOD classes and applies ALM (Augmented Lagrangian Method) on a deep neural network to solve this constrained optimization problem. Although these methods using unlabeled data are closer to real-world situations than those using auxiliary OOD datasets and get more OOD signals than those generating virtual instances, these methods are still based on a strong assumption that ID labeled data is sufficient. This assumption is difficult to be satisfied in several real-world applications because of the high cost of labeling. Therefore, the new problem of out-of-distribution detection with limited in-distribution labeled data needs studying urgently.

3 METHOD

In this section, we elaborate on how our method, Topological Structure Learning (TSL), works. Firstly, we describe the problem statements in Section 3.1. Secondly, we extract reliable feature representations with SimCLR to construct the initial points for the topological space in Section 3.2. Then, to construct edges for the topological space and make preparation for the next topological structure reconstruction, we propose topological connection mining in Section 3.3. Next, to increase the separability of ID and OOD instances, we reconstruct the topological structure in a new topological space in Section 3.4. When reconstructing the topological structure, TSL performs topological skeleton maintenance in the new topological space in Section 3.4.1 and extends the whole topological structure based on topological skeleton with topological connections mined before in Section 3.4.2. What’s more, we summarize TSL’s optimizing objective in Section 3.5. Finally, we introduce the scoring function of TSL in Section 3.6. The overview of TSL can be seen in Fig. 2.

3.1 Problem Statement

In this part, first of all, we introduce common OOD detection. Then, we introduce OOD Detection with sufficient unlabeled data consisting of ID and OOD data. After that, we introduce the newly proposed weakly-supervised OOD detection, provide formal descriptions of the new scenario and introduce notations used throughout this paper.

OOD Detection. While training, the available information is a labeled set $D_L = \{(x_i, y_i)\}_{i=1}^n$ from the ID joint distribution $P(X, Y)$, where $x_i \in X$, $y_i \in Y$, $n$ denotes the number of ID labeled instances, $X$ denotes the feature space, and $Y$ denotes the label space. $Y = \{1, \cdots, k\}$, where $k$ denotes the number of ID classes. Let $D_{test}$ denotes the test set, which consists of ID test set $D_{in}^{test}$ from ID marginal distribution $P(X)$ and OOD test set $D_{out}^{test}$ from OOD marginal distribution $Q(X)$, $P(X) \cap Q(X) = \emptyset$. The goal of OOD detection is to obtain a decision function $G$ such that for a given test input $x \in D_{test}$,

$$G(x) = \begin{cases} 0 & \text{if } x \sim Q(X), \\ 1 & \text{if } x \sim P(X), \end{cases}$$

(1)

where $G(x) = 1$ means that $x$ belongs to ID data and $G(x) = 0$ means that $x$ belongs to OOD data.

OOD Detection with Sufficient Unlabeled Data. Different from the common OOD detection methods, in this setting, large-scale unlabeled data is available in addition to $D_L$. Let $D_U = \{x_i\}_{i=1}^m$ denotes unlabeled data set mixed by both ID and OOD unlabeled data, where $m$ denotes the number of unlabeled data. In $D_U$, we denote the set of data from $P(X)$ by $D_{in}^U$ and the set of data from $Q(X)$ by $D_{out}^U$.

Weakly-Supervised Out-of-Distribution Detection (WSOOD). The previous OOD detection methods assume that ID labeled data is sufficient, which is hard to satisfy in many real-life applications. However, obtaining unlabeled data is usually cheap. Then we define...
limited ID labeled set by \( D_L = \{ (x_i, y_i) \}_{i=1}^n \) and sufficient unlabeled data set by \( D_U = \{ x_i \}_{i=1}^m, n \ll m \).

The new setting we study is more complex than the previous settings. The biggest challenge is severely limited ID labeled data. Separating ID and OOD data from unlabeled data is highly dependent on a prior model, which is trained by labeled data only. Due to limited ID labeled data, the prior model tends to be not effective enough in our setting. Therefore, separating ID and OOD data from unlabeled data is challenging, which causes poor performance of OOD detection. At the same time, this new setting is more realistic than the previous settings. In many real-world applications, collecting large-scale ID labeled data is costly and time-consuming. In order to solve the new problem effectively, we propose Topological Structure Learning (TSL).

### 3.2 Topological Space Representation Learning

In the new setting, the number of labeled data is limited, which makes feature extraction more difficult. Representation learning by supervised learning is impracticable due to the limited number of labeled data. Large-scale unlabeled data is easy to collect. We use a self-supervised learning strategy to obtain reliable features based on all labeled and unlabeled data. Actually, contrastive learning can be considered as an extension of spectral clustering that can help extract features and learn the initial connections between instances [7]. Concretely, we employ a simple contrastive learning method called SimCLR [2] on labeled data in \( D_L \) and unlabeled data in \( D_U \). SimCLR randomly samples a minibatch of \( N \) examples and obtains \( 2N \) augmented examples by using two different augmentation strategies over the original \( N \) examples. Then, SimCLR defines the contrastive prediction task as follows,

\[
L = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k - 1, 2k) + \ell(2k, 2k - 1)],
\]

where \( 2k - 1 \) and \( 2k \) denote the two augmented examples from the same original example and are considered as a positive pair, similar to [3]. SimCLR does not sample negative examples explicitly but treats the other \( 2(N - 1) \) augmented examples within a minibatch as negative examples. \( \ell \) denotes contrast loss, which is defined by

\[
\ell_{i,j} = -\log \frac{\exp (\text{sim}(z_i, z_j) / \tau)}{\sum_{k=1}^{2N} I_{[k \neq i]} \exp (\text{sim}(z_i, z_k) / \tau)},
\]

where \( z_i \) and \( z_j \) are two different feature representations generated from an example \( x \) after the first three modules of SimCLR, \( \text{sim}(z_i, z_j) \) denotes cosine similarity of \( z_i \) and \( z_j \), and \( \tau \) denotes a temperature parameter.

### 3.3 Topological Connection Mining

To mine reliable connections of edges from limited ID labeled data and unlabeled data for the topological space and prepare for the next topological structure reconstruction, TSL uses all the reliable features extracted by SimCLR to mine the positive and negative pairs among data in \( D_L \cup D_U \). Although the existing work [36] solves the problem of limited weakly-supervised OOD detection by positive and negative pairs mining, [36] suffers from noisy negative pairs and unreliable positive pairs. Our approach greatly alleviates the problems by reliable positive and negative pair mining.

The existing work, STEP [36] defines positive pairs as

\[
P_p = \{ (x_i, x_j) \mid x_i \in D_L \cup D_U, x_j \in B_K(x_i) \},
\]

where \( x_i \) is sampled from \( D_L \cup D_U \), and \( x_j \) is sampled from \( B_K(x_i) \). \( B_K(x_i) \) denotes the set of \( K \) nearest neighbors of the example \( x_i \) which is defined by the KNN algorithm based on the Mahalanobis distance. Then for each instance, STEP considers all other instances except itself as its negative instances during the process of negative pairs mining by

\[
N_p = \{ (x_i, x_j) \mid i \neq j \},
\]

where \( x_i \) and \( x_j \) are sampled from \( D_L \cup D_U \).

![Figure 3: Mining of positive pairs.](image)

#### 3.3.1 Positive Pair Mining

The mined positive pairs are one-sided and rough in STEP. An example gives the same degree of trust to all its \( K \) nearest neighbors and considers all the \( K \) nearest neighbors as its positive instances. Such a connection of positive pairs is not reliable. Fig. 3 demonstrates that the neighbor connection of ‘a’ on the left is more intimate than that on the right. However, according to STEP, the connection of ‘a’ and ‘b’ is treated equally on both the left and right sides, which is problematic. Obviously, if two instances are each other’s positive instances, then such a connection of positive pairs should be more reliable.

TSL notes that the positive pairs can be further subdivided into positive pairs with different credibility for better mining of positive pairs. Then TSL defines close positive pairs and loose positive pairs according to reciprocal and non-reciprocal neighbors. More specifically, we divide the \( K \) nearest neighbors provided by the KNN algorithm according to Mahalanobis distance for each example into reciprocal neighbors and non-reciprocal neighbors. If two instances are both in each other’s \( K \) nearest neighbors, then we judge that the two instances are reciprocal neighbors with each other, which means that the two instances are close positive pairs, which is defined by

\[
P_c = \{ (x_i, x_j) \mid x_i \in B_K(x_j) \land x_j \in B_K(x_i) \},
\]

where \( x_i \) and \( x_j \) are randomly sampled from the feature space. The process of mining close positive pairs is clearly shown in the left diagram of Fig. 3. On the contrary, if example ‘a’ is one of \( K \) nearest neighbors of example ‘b’, but example ‘b’ is not one of \( K \) nearest neighbors of example ‘a’, we conclude that ‘a’ and ‘b’ are non-reciprocal neighbors with each other, which means the two instances are loose positive pairs, which is defined by

\[
P_l = \{ (x_i, x_j) \mid x_i \in B_K(x_j) \oplus x_j \in B_K(x_i) \},
\]
where \( x_i \) and \( x_j \) are randomly sampled from the feature space. The right diagram of Fig. 3 presents how TSL mines loose positive pairs. TSL subdivides positive pair connection by defining close positive pairs and loose positive pairs:

\[
\mathcal{P}_p = \mathcal{P}_l + \mathcal{P}_c.
\]  

3.3.2 Negative Pair Mining. The negative pairs screening has apparent drawbacks in STEP. For example, STEP considers two examples from the same class as a negative pair, which causes ineffective class distribution learning. To obtain reliable negative pairs, TSL defines a more targeted negative pairs instead of defining any two instances as negative pairs. More specifically, for each example \( x_i \), TSL progressively sorts all the remaining instances according to their Mahalanobis distance to \( x_i \). Then, TSL regards the instances ranked after the \( \beta \times K \) as its negative instances. The negative example set of \( x_i \) can be expressed as

\[
\mathcal{N}B_{\beta \times K}(x_i) = \left\{ x_j \mid x_j \notin B_{\beta \times K}(x_i) \right\},
\]

where \( B_{\beta \times K}(x_i) \) denotes the \( \beta \times K \) closest neighbors of \( x_i \), \( x_i \) and each of \( \mathcal{N}B_{\beta \times K}(x_i) \) form a negative pair. The set of negative pairs can be denoted by

\[
\mathcal{N}P_{\beta} = \left\{ (x_i, x_j) \mid x_i \in (D_L \cup D_U), x_j \notin \mathcal{N}B_{\beta \times K}(x_i) \right\},
\]  

where \( x_j \) is randomly sampled from \( D_L \cup D_U \), and \( x_i \) is randomly sampled from \( \mathcal{N}B_{\beta \times K}(x_i) \). Fig. 4 illustrates how TSL mines negative pairs well. Such a negative pair mining method can alleviate the problem of defining positive pairs and instances from the same ID category as negative pairs to some extent.

3.4 Topological Structure Reconstruction

To increase the separability of ID and OOD instances, TSL reconstructs the topological structure in a new topological space by training the projector \( P \). The role of \( P \) is to project the features to a new feature space. When constructing the topological structure, TSL performs two steps. The first is the maintenance of the topological skeleton. The second is to extend and reconstruct the whole topological structure on the basis of the topological skeleton.

3.4.1 Topological Skeleton Maintenance. In the new topological space, TSL reduces the intra-class Mahalanobis distance of limited ID labeled data to maintain the topological skeleton. Therefore, in the new topological space, these instances form \( k \) small clusters according to the ground truth labels, which further influences the distribution of all instances and makes the decision boundary between ID and OOD clearer. TSL considers any two instances of the same class from \( D_L \) as a positive pair, performs metric learning of positive pairs by training the projector \( P \), and assigns a smaller coefficient than that of close positive pairs. TSL conducts \( L_a \) to aggregate ID intra-class labeled data. \( L_a \) is defined by

\[
L_a = \sum_{(x_p, x_q) \in \mathcal{P}_a} \max \left( 0, \|P_{x_p} - P_{x_q}\|_2^2 - \lambda_1 MD(x_p, x_q) \right),
\]  

where \( \lambda_1 \) is a coefficient with the constraint of \( \lambda_1 < \lambda_2 \), and \( \mathcal{P}_a \) is ID labeled positive set. \( \mathcal{P}_a \) is defined by

\[
\mathcal{P}_a = \left\{ (x_i, x_j) \mid x_i, x_j \in D_L, y_i = y_j, x_i \neq x_j \right\},
\]

where \( \mathcal{P}_a \) denotes Intra-class positive pairs in \( D_L \).

Compared to TSL, the existing method STEP does not fully use the valuable ID labeled data. STEP only uses the labeled instances when measuring known class centers and when measuring the covariance of the Mahalanobis distance, which causes a tremendous waste of the information carried by the ID labeled instances, especially considering the high cost of obtaining those labeled data.

3.4.2 Topological Skeleton Extension. In the new topological space, based on the topological skeleton, TSL extends the topological structure with the positive and negative pairs mined before. Specifically, in the new feature space, TSL adjusts the distances of close positive and loose positive pairs while trying to separate negative pairs.

For positive pairs, TSL gives different confidence levels to close positive and loose positive pairs by imposing different coefficients when maintaining topological structure dynamically. Concretely, TSL imposes a lower coefficient on close positive pairs because they are more credible neighbors to each other. Close positive pairs ought to be closer to each other than loose positive pairs. Conversely, TSL imposes a higher coefficient on loose positive pairs because the connection of loose positive pairs is farther than close positive pairs. TSL conducts \( L_c \) to adjust the positive pairs.

\[
L_c = \sum_{(x_i, x_j) \in \mathcal{P}_c} \max \left( 0, \|P_{x_i} - P_{x_j}\|_2^2 - \lambda_2 MD(x_i, x_j) \right),
\]  

where \( \lambda_2 \) is the coefficient for close positive pairs, \( P_{x_i} \) and \( P_{x_j} \) denote the projected features in the new feature space, and \( MD(x_i, x_j) \) denotes Mahalanobis distance between \( x_i \) and \( x_j \), defined by

\[
MD(x_i, x_j) = \sqrt{(x_i - x_j)^\top \hat{\Sigma}^{-1} (x_i - x_j)},
\]  

where \( \hat{\Sigma} \) denotes the covariance matrix of features of \( D_L \). And \( L_l \) is defined by

\[
L_l = \sum_{(x_i, x_j) \in \mathcal{P}_l} \max \left( 0, \|P_{x_i} - P_{x_j}\|_2^2 - \lambda_3 MD(x_i, x_j) \right),
\]  

where \( \lambda_3 \) is the coefficient for loose positive pairs, and \( \lambda_2 < \lambda_3 \). More details about \( \lambda_2, \lambda_3 \) will be introduced in the experiment setup.

For negative pairs, TSL constructs \( L_f \) as follows,

\[
L_f = \sum_{(x_i, x_j) \in \mathcal{N}P_{\beta}} \max \left( 0, M - \|P_{x_i} - P_{x_j}\|_2 \right),
\]  

where \( M \) denotes the margin. The larger \( M \) is, the farther the negative pairs are.
3.5 Optimizing Objective

With Eqs. (11), (13), (15) and (16), we state the overall loss $\mathcal{L}$ as

$$\mathcal{L} = L_{\alpha} + L_{c} + L_{l} + L_{f}.$$  

(17)

Then we train the projector $P$ for projecting the original topological space to the new topological space by minimizing $\mathcal{L}$.

3.6 Scoring Function Designing

During inference, we use the output of projector $P$ for OOD detection. In particular, given a test input $x$, the OOD uncertainty score is given by

$$\text{MS}(x^*) = \min_{c \in \{1, \ldots, K\}} \left\| Px^* - P\mu_c \right\|_2,$$

where $K$ denotes the number of ID classes, and $\mu_c$ denotes $c$-th class centers of ID labeled data.

Setting a threshold $\gamma$ for $\text{MS}(\cdot)$ can help model distinguish between ID and OOD. With the role of score function $\text{MS}(\cdot)$ and threshold $\gamma$, we can redefine the decision function $G$:

$$G_{\gamma}(x) = \begin{cases} 1 & \text{MS}(x) \geq \gamma, \\ 0 & \text{MS}(x) < \gamma. \end{cases}$$  

(19)

The principle of choosing $\gamma$ is usually to allow the decision function $G$ to classify the vast majority of ID instances correctly (e.g., 95%).

4 EXPERIMENTS

4.1 Experimental Setup

In-distribution Dataset. Following [36], we choose CIFAR-10 and CIFAR-100 [13] as the ID datasets. Both CIFAR-10 and CIFAR-100 include 50,000 training images and 10,000 testing images. The image size of the two data sets is $32 \times 32$. For CIFAR-10, there are ten classes in CIFAR-10. In the training phase, we randomly select 25 images from each class of CIFAR-10 to form $D_t$, and then place the rest of the images from the CIFAR-10’s training set into $D_u$. For CIFAR-100, there are 100 classes. In the training phase, we randomly select four images from each class of CIFAR-100 to $D_t$ and then put the rest into $D_u$. In the testing phase, we select 9,000 images from the testing set of CIFAR-10 or CIFAR-100 as $D_{\text{test}}$.

Out-of-distribution Dataset. Similar to [36], we use TinyImageNet-crop (TINc), TinyImageNet-resize (TINr), LSUN-crop (LSUNC) and LSUN-resize (LSUNr) as OOD datasets. TINc and TINr are two variants of Tiny ImageNet (TIN) [4]. TIN consists of 10,000 images from 200 classes, and it is a subset of ImageNet. TIN is changed into TINc or TINr by the operation of randomly cropping or downsampling each image to $32 \times 32$. LSUNC and LSUNr are two variants of the Large-scale Scene Understanding (LSUN) dataset [33]. LSUN consists of 10,000 images from 10 classes. LSUN is changed into LSUNC or LSUNr by the same operation like TIN. We put the remaining data in OOD dataset as $D_u$ and $D_{\text{test}}$. Following STEP, we set $D_{\text{test}}$ and $D_{\text{test}}$ identical.

Evaluation Metrics. Our experiments use the following five metrics in OOD detection for comparison: AUROC, FPR95, Detection Error, AUPR-In, and AUPR-Out.

Comparison Methods. To verify the effectiveness of TSL, we compare our method with some classical and advanced OOD detection or PU learning method: ADOA [35], ODIN [15], MAH [14], UOOD [34] and STEP [36].

Implementation Details. In our experiments, we choose DenseNet-BC [11] as the backbone of SimCLR.

4.2 OOD Detection Performance

Table 1 shows the OOD detection performance of TSL and baseline methods. The results show that PU learning methods, like ADOA, post-hoc-based methods, like ODIN and MAH, and fine-tuning-based methods with unlabeled data, like UOOD, fail to solve weakly-supervised OOD detection. PU learning and our problem have different assumptions about the proportion of positive instances in unlabeled data, so the performance of ADOA is not good. Post-hoc-based methods rely entirely on ID labeled data, so their performance is bound to drop when facing a severe lack of ID labeled data. For UOOD, due to the severe scarcity of ID labeled data, the empirical risk of treating all unlabeled instances as OOD during training is no longer affordable. Therefore, the OOD detection performance of UOOD drops dramatically.

STEP is the state-of-the-art OOD detection method when ID labeled data is limited. Surprisingly, TSL performs better than SETP on almost all datasets. Especially on difficult tasks, such as CIFAR-10 & TINr, CIFAR-100 & TINr, CIFAR-10 & LSUNC, and CIFAR-100 & LSUNr, TSL outperforms STEP by a large margin. We use the symbol & to connect the ID dataset and the OOD dataset in a task. For example, TSL reduces the FPR95 and Detection Error by 15.37% and 6.49% compared to STEP on CIFAR-10 & TINr, respectively. Through Fig. 5, we can also observe that TSL separates ID and OOD.
Table 3: Performance comparison of STEP and TSL when generalizing to unseen OOD data.

| Metrics | D^\text{In}\_\text{test} | D^\text{Out}\_\text{test} | D^\text{Un}\_\text{test} | STEP | TSL |
|---------|----------------|----------------|----------------|-----|-----|
| CIFAR-10 TINc | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINr | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINc | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINr | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINc | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINr | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINc | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINr | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINc | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINr | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINc | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINr | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINc | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINr | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINc | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINr | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINc | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINr | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINc | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINr | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINc | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |
| CIFAR-10 TINr | 0.84 | 0.32 | 0.37 | 0.84 | 0.32 |

4.3 Generalization of TSL

4.3.1 Generalizing to Unseen OOD Data. In the previous experimental setup, D^\text{out}\_\text{test} and D^\text{un}\_\text{test} are totally identical, which follows the setting of STEP. However, such an assumption is difficult to meet in the real world. For example, in autonomous driving, we cannot provide the model with all the OOD instances encountered in the future during the training phase. To explore the generalization ability of TSL when detecting unseen OOD data, we conduct two expanded experiments. In the first experiment, we randomly divide OOD data into D^\text{out}\_\text{test} and D^\text{un}\_\text{test} according to the ratio of 8:2. Following this experimental setup, we conduct experiments on CIFAR-10 & TINr and CIFAR-100 & TINr. By Table 2, we observe that compared to STEP, TSL reduces the average FPR95 by 24.15% and 9.33% on CIFAR-10 & TINr and CIFAR-100 & TINr, respectively. The experiment results show that TSL is far ahead of STEP in terms of OOD generalization ability on unseen OOD data of the same classes. In the second experiment, D^\text{out}\_\text{test} is no longer the same as D^\text{un}\_\text{test}. For example, we consider TINc as D^\text{un}\_\text{test} and TINr as D^\text{out}\_\text{test}. From the results in Table 3, we can observe that OOD generalization capability of TSL is significantly better than STEP. And we also observe that the task difficulty of generalizing TINc to TINr is lower than generalizing TINc to TINr. Whether the generalization task is easy or difficult, our proposed TSL outperforms STEP comprehensively, which verifies the generalization ability of TSL.

4.4 Ablation Study

We perform ablation experiments to verify the effectiveness of each of the three critical modules of TSL. As introduced in Section 3, TSL has three effective modules to improve the performance of OOD detection compared with STEP: positive pairs mining (PPM), negative pairs mining (NPM), and topological skeleton maintenance (TSM) module. We perform ablation experiments on these three modules on CIFAR-10 & TINr, CIFAR-10 & LSUNr, CIFAR-100 & TINr, and CIFAR-100 & LSUNr, respectively, to show their contribution to TSL’s performance. We add the three modules to STEP in turn to verify the effectiveness of each module. Meanwhile, to demonstrate the effectiveness of each module after stacking, we fix the random seed during the ablation experiment. From Table 5, we observe that the model’s performance on each combination of datasets constantly improves when adding PPM, NPM, and TSM in turn.
**Table 5**: TSL’s module ablation studies on different combinations of ID and OOD dataset.

| ID                       | OOD                       | PPM | NPM | TSPL  | AUROC | Detection Error | AUROC-In T | AUROC-Out T |
|--------------------------|---------------------------|-----|-----|-------|-------|-----------------|-------------|-------------|
| CIFAR-10 TIN             | ✓                         | -   | -   | ✓     | 98.36 | 28.44           | 97.77       | 98.00       |
|                          | ✓                         | ✓   | ✓   | ✓     | 98.87 | 28.38           | 97.33       | 97.31       |
|                          | ✓                         | ✓   | ✓   | ✓     | 98.95 | 28.38           | 97.77       | 98.04       |
|                          | ✓                         | ✓   | ✓   | ✓     | 99.14 | 28.84           | 98.72       | 99.27       |
| CIFAR-10 LSUN            | ✓                         | -   | -   | ✓     | 98.36 | 28.44           | 97.77       | 98.00       |
|                          | ✓                         | ✓   | ✓   | ✓     | 98.95 | 28.38           | 97.33       | 97.31       |
|                          | ✓                         | ✓   | ✓   | ✓     | 98.95 | 28.38           | 97.77       | 98.04       |
|                          | ✓                         | ✓   | ✓   | ✓     | 99.14 | 28.84           | 98.72       | 99.27       |

4.5 Sensitivity of Hyperparameters

**Figure 6**: An analysis of AUROC at different $K$.

4.5.1 Analysis of $K$. $K$ is the number of neighbors for the KNN algorithm when mining positive pairs. We perform experiments to compare the performance of TSL on different ID and OOD dataset combinations for different $K$. From Fig. 6, we can observe that TSL is insensitive to $K$ from 2 to 18.

4.5.2 Analysis of $M$. $M$ is a coefficient responsible for regulating the distance between negative pairs when topological structure learning. Fig. 7(a) shows that TSL is insensitive to margin when $M$ is larger than three.

4.5.3 Analysis of $\lambda_1$, $\lambda_2$, and $\lambda_3$. TSL has hyperparameters $\lambda_1$, $\lambda_2$, and $\lambda_3$. $\lambda_1$ regulates the closeness of instances from the same class in $D_T$ for topological skeleton maintenance with ID labeled data. $\lambda_2$ and $\lambda_3$ are critical hyperparameters in topological skeleton extension for close positive pairs and loose positive pairs, respectively. We conduct experiments to analyze them. First, Fig. 7 (b) shows that AUROC basically shows a decreasing trend as $\lambda_1$ increases, and TSL performs best when $\lambda_1$ takes the value of 0.1. So a smaller $\lambda_1$ helps improve TSL’s performance. Second, from Table 6(a), we observe that AUROC tends to level off after $\lambda_2 > 0.5$. Third, Table 6(b) shows how AUROC changes with different values of $\lambda_3$ on CIFAR-100 & TINr. We observe that as $\lambda_3$ increases, AUROC first rises and then falls, and AUROC is maximum when $\lambda_3$ is 6.

4.5.4 Analysis of $\beta$. As can be seen from Table 6(c), TSL maintains good robustness when $\beta$ goes from 1 to 4,500 on all CIFAR-100 & TINr. However, after $\beta$ is greater than 4,500, the performance of TSL starts to drop sharply as $\beta$ increases.

5 CONCLUSION

In this paper, we explored a new problem, weakly-supervised OOD detection, and analyzed why limited ID labeled instances brought challenges. To solve this new problem, we proposed a novel OOD detection method called TSL. TSL constructed the initial points for the topological space by extracting reliable features with SimCLR. In the initial topological space, TSL mined topological connections, including close positive pairs, loose positive pairs, and negative pairs, to construct several different kinds of edges with different credibility. Moreover, TSL reconstructs the topological structure in a new topological space with two steps, topological skeleton maintenance and topological skeleton extension, to increase the separability of ID and OOD instances. Empirical studies showed that TSL achieved stable performance gain across different OOD datasets and different tasks, which verified the validity and robustness of TSL. Beyond this work, this problem is more relevant to real-world applications, so it is worth studying in the future.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (62176139), the Major Basic Research Project of Natural Science Foundation of Shandong Province (ZR2021ZD15).

REFERENCES

[1] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria de la Iglesia-Vaya. 2020. Padchest: A large chest x-ray image dataset with multi-label annotated reports. Medical image analysis 66 (2020), 101797.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In International conference on machine learning. PMLR, 1597–1607.

[3] Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering. In Proceedings of the 23rd ACM
