






# Rethinking Unsupervised Outlier Detection via Multiple Thresholding

Zhonghang Liu<sup>1</sup> , Panzhong Lu<sup>2</sup> , Guoyang Xie<sup>3,4</sup> , Zhichao Lu<sup>4</sup> , and Wen-Yan Lin<sup>1</sup> 

<sup>1</sup> Singapore Management University

<sup>2</sup> Westlake University

<sup>3</sup> Contemporary Ampere Technology Co., Limited

<sup>4</sup> City University of Hong Kong

zhliu.2020@phdcs.smu.edu.sg

lupanzhong@westlake.edu.cn

guoyang.xie@ieee.org

luzhichaocn@gmail.com

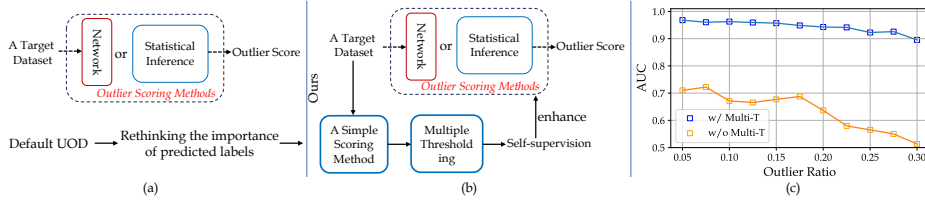
daniellin@smu.edu.sg

**Abstract.** In the realm of unsupervised image outlier detection, assigning outlier scores holds greater significance than its subsequent task: thresholding for predicting labels. This is because determining the optimal threshold on non-separable outlier score functions is an ill-posed problem. However, the lack of predicted labels not only hinders some real applications of current outlier detectors but also causes these methods not to be enhanced by leveraging the dataset’s self-supervision. To advance existing scoring methods, we propose a multiple thresholding (Multi-T) module. It generates two thresholds that isolate inliers and outliers from the unlabelled target dataset, whereas outliers are employed to obtain better feature representation while inliers provide an uncontaminated manifold. Extensive experiments verify that Multi-T can significantly improve proposed outlier scoring methods. Moreover, Multi-T contributes to a naive distance-based method being state-of-the-art. Code is available at: <https://github.com/zhliu-uod/Multi-T>.

**Keywords:** Unsupervised outlier detection, multiple thresholding, outlier scoring

## 1 Introduction

Which is more important for the unsupervised outlier detection (UOD), *outlier score* or *label*? Currently, the mainstream UOD approaches [23, 28, 31, 56] focus on the first, i.e., learning a discriminative outlier score function. However, those recent outlier detectors are usually complex and non-iterative since the absence of predicted labels limits these methods to be further advanced with the dataset’s self-supervision. Such that, this work tends to label the unlabelled tar-



**Fig. 1:** Unlike the default UOD paradigm (a) concerned about learning a discriminative score function, our perspective (b) is to explore the dataset’s prior knowledge via thresholding, to advance the previously proposed method. (c) illustrates the significant improvement of DeepSVDD [50] with Multi-T module.

get dataset, thereby leveraging the dataset’s prior knowledge to enhance simple outlier scoring methods<sup>1</sup>, achieving state-of-the-art (SOTA) results efficiently.

To understand the importance of the target dataset’s self-supervision, we first decompose the predominant image outlier scoring approaches [9, 32, 50, 51] into two continuous stages: inlier/normal manifold learning and distance/similarity inference. There is no doubt that identifying inliers will contribute to learning an uncontaminated inlier manifold (e.g. the hypersphere of DeepSVDD [50]). Additionally, motivated by Shell Theory [32, 33] in the high-dimensional space, image feature representation for distance computation can be significantly improved via outliers-based Shell Normalization [32]. Such that it is critical to obtain the thresholds that separate both inliers and outliers from target datasets.

To align with our above objective, we present a Multi-T module, which follows a *multiple* thresholding mechanism. Compared with conventional *single* thresholding methods [19, 24, 48, 49], Multi-T generates two distinct thresholds that separably isolate outliers and inliers from the target dataset, which is capable of dealing with the inevitable inseparability of the initial given score distribution. This allows us to directly implement Multi-T on a simple outlier score function.

By understanding the impact of the outlier ratio, Multi-T involves two stages: **Uncontaminated inliers.** We employ an iterative two-step process (i) identifying a noisy inlier distribution through the analysis of sorted initial score function, i.e., Ergodic-set normalized [34] distance to the mean of target dataset; (ii) subsequently filtering outliers with 3-sigma rule [42]. When converged, we can identify uncontaminated inliers, thereby training an inlier/normal manifold.

**Adaptive outliers.** The uncontaminated inliers identification process also outputs a series of outlier threshold candidates. Subsequently, we compare the ranking-index similarity between Ergodic-set Normalization [34] and Shell Normalization [32] on the initial outlier score function. The structural consistency and contrastive properties of the two normalization procedures motivate us to implicitly estimate a rough outlier ratio, thereby detecting the adaptive outliers.

The Multi-T module is training-free, highly efficient and grants the adaptiveness to choose thresholds suited to downstream implementations. As evaluated

<sup>1</sup> Outlier scoring methods refers to the classical outlier detectors.

with comprehensive experimental settings, Multi-T helps to boost significant performance (efficacy and efficiency) improvement. For example, the AUC score of DeepSVDD [50] is improved from 0.622 to 0.925 on STL-10 dataset [15]. Moreover, the naive distance-to-the-target dataset’s mean [32] metric integrated with Multi-T achieves SOTA results with only 1.2 second consumption for 10,000 ResNet-50 [21] samples that are orders of magnitude faster than baselines. Our primary contributions are summarized below:

- We introduce a novel perspective for UOD about enhancing previously proposed scoring methods via thresholding on a simple initial outlier score function.
- We present a multiple thresholding (Multi-T) module that generates two different thresholds to separate both inliers and outliers from the target dataset.
- The efficacy of integrated scoring methods can be significantly improved without the external complexity increase, as evaluated in extensive experiments.

## 2 Related Work

### 2.1 Unsupervised Outlier Detection

Classic UOD task aims to assign an outlier score/likelihood to an image sample. The recent models can be divided into inlier-manifold learning and outlier exposure. Inlier-manifold learning assumes that inliers are the majority. Deep learning-based outlier detectors, as illustrated in studies [14, 28, 30, 38, 56, 58, 62, 63], typically focus on reducing the dimensionality of image data by projecting it into a latent and discriminative space. Additionally, various statistics-based methods model the inlier manifold using discrimination-based [32, 34] or density-based approaches [11, 26, 31, 40]. In applications where outliers are not the minority [13, 57], different approaches are required. Some outlier exposure methods [22, 36] employ out-of-distribution (OOD) data to train networks for detecting unseen OOD samples. For example, Shell-Renormalization [32] employs predicted outlier candidates to iteratively refine feature representation. It is particularly useful in scenarios where classic manifold learners, assuming outliers as a minority, may not be effective. However, these two paradigms are separable, this work will attempt to design a thresholding framework that generates both inliers and outliers, leveraging the benefits of both. inlier-manifold learning and outlier exposure.

### 2.2 Thresholding

Some traditional outlier detection methods [32, 35, 52], provide both the outlier score and its corresponding threshold concurrently. In this context, the threshold manifests as a hypersphere that accepts the inliers while rejecting those outliers. Unquestionably, discerning it presents its challenges in high-dimensional space [34]. In response, we venture into an alternate perspective. Some other threshold detectors are mostly based on statistical analysis that can be utilized for any given 1- $d$  outlier score function [28, 31, 32]. The representative

works involve Kernel-based [43, 48]; Curve-based [17]; Normality-based: [5, 10]; Filtering-based [19, 55]; Statistical-based [1, 3, 4, 6–8, 11, 12, 16, 24, 37, 49] and Transformation-based [25, 46]. In practice, the default thresholding follows a *single* thresholding paradigm is not realistic since the perfect discriminativeness of the outlier score function is usually not assumed. Therefore, this work introduces a new perspective: *multiple* thresholding.

### 3 Overview

#### 3.1 Problem Definition

Before introducing our method, we first formally define the problem. The target dataset  $\mathbb{Z} = \mathbb{Z}_{in} \cup \mathbb{Z}_{out}$  involves  $n$  unlabelled images, here  $\mathbb{Z}_{in}$  and  $\mathbb{Z}_{out}$  represent inliers (normal data) and outliers (anomalous data), respectively. The outlier (contamination) ratio  $\gamma \in (0, 1)$  is denoted as  $\frac{\#\mathbb{Z}_{out}}{\#\mathbb{Z}_{in} + \#\mathbb{Z}_{out}}$ . Notably, this setup differs from the related semi-supervised outlier detection task by eliminating the need for a prior train/test split for the target dataset. Aligning with previous works [28, 32, 34], the raw images are extracted to feature representation  $\mathbf{X} = \mathbf{X}_{in} \cup \mathbf{X}_{out} = \{\mathbf{x}_i\}_{i=1}^n$  with pixel representation [32, 34] or pre-trained neural networks (e.g., ResNet [21] and CLIP [45]). The main objective of UOD is to create an outlier score function  $F(\cdot)$  to evaluate the likelihood of an image feature  $\mathbf{x}_i \in \mathbf{X}$  being outliers:

$$Y(\mathbf{x}_i) = \begin{cases} 0, & F(\mathbf{x}_i) < \phi \\ 1, & F(\mathbf{x}_i) \geq \phi, \end{cases} \quad (1)$$

where  $Y = 0$  (*inlier*)/ $1$  (*outlier*) refers to predicted labels and  $\phi$  is the threshold (decision boundary). In this study, we not only center on evaluating the ranking accuracy of  $F(\cdot)$  [28, 32, 34, 50], but also measure the efficacy of the threshold  $\phi$ .

#### 3.2 Rethinking UOD

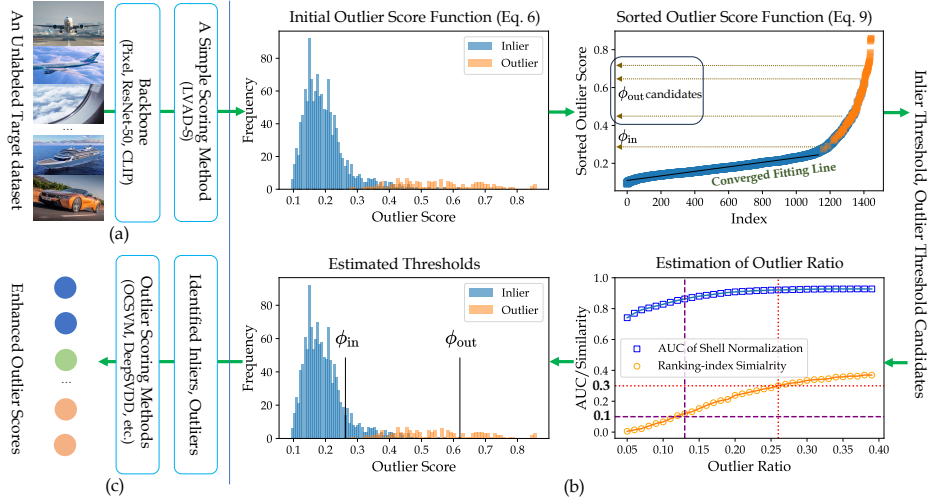
As there is no prior train/test split for the target dataset, a classic outlier scoring method  $M(\cdot)$  predicts the outlier score function  $F(\mathbf{X}) = \{F(\mathbf{x}_i)\}_{i=1}^n$  follows the below process:

$$F(\mathbf{X}) = M.\text{fit}(\mathbf{X}).\text{predict}(\mathbf{X}), \quad (2)$$

where  $\text{fit}(\cdot)$  and  $\text{predict}(\cdot)$  refer to fitting the target dataset and predicting outlier scores, respectively. Motivated by previous research [32, 33], Shell normalization S-norm( $\cdot$ ) illustrated below is an *ideal* feature de-noising/refining paradigm.

$$\text{S-norm}(\mathbf{x}_i, \mathbf{v}_S) = \frac{\mathbf{x}_i - \mathbf{v}_S}{\|\mathbf{x}_i - \mathbf{v}_S\|_2}, \mathbf{v}_S = \left[ \frac{1}{n} \sum_{i=1}^n \mathbf{X}_{out}[i][1], \dots, \frac{1}{n} \sum_{i=1}^n \mathbf{X}_{out}[i][d] \right], \quad (3)$$

where  $\|\cdot\|_2$  refers to  $\ell_2$ -norm and  $d$  is the feature dimension. Thus, our first stage is to identify outlier candidates  $\mathbf{X}'_{out}$ . Subsequently, as how some one-class learning-based methods (e.g., OCSVM [51], DeepSVDD [50]) usually perform, we secondly predict the inliers  $\mathbf{X}'_{in}$  to fit an inlier/normal manifold. Thus, our objective is to identify both  $\mathbf{X}'_{in}$  and  $\mathbf{X}'_{out}$ .



**Fig. 2:** The overall paradigm of adopting the multi-thresholds learning (Multi-T) module to advance the existing outlier scoring methods. (a) The preparation consists of feature extraction and an initial outlier score function. (b) Visualization of our Multi-T module. (c) Integrating the predicted inliers and outliers with the previously proposed outlier detectors and obtaining an enhanced outlier score function.

## 4 Method

Our solution is displayed as below subsections. Sec. 4.1: we introduce the initial outlier score function; Sec. 4.2: we present the Multi-T module, to generate two thresholds for separating inliers and outliers from the target dataset; Sec. 4.3: we leverage predicted inliers and outliers to advance previously proposed methods.

### 4.1 Initial Outlier Score Function

In our approach, we employ LVAD [34], one of the UOD SOTAs, as the initial outlier score function. LVAD assumes each image feature  $\mathbf{x}_i \in \mathbf{X}$  comes from one of  $T$  high-dimensional generative processes. Thus, the outlier score becomes a sum of the weighted distance of the given instance,  $\mathbf{x}_i$ , arising from each generative process  $\{w_t, \mathbf{m}_t \mid t \in \{1, \dots, T\}\}$ , which can be simplified as:

$$F_{\text{LVAD}}(\mathbf{x}_i) = \sum_{t=1}^T w_t \cdot \text{Dist}(\text{E-norm}(\mathbf{x}_i), \text{E-norm}(\mathbf{m}_t)), \quad (4)$$

where  $\text{Dist}(\cdot)$  refers to the  $\ell_2$ -norm metric (Euclidean distance),  $w_t$  is the weight of the  $t$ -th generative process,  $\mathbf{m}_t$  is the  $t$ -th generative process's centroid and  $\text{E-norm}(\cdot)$  refers to the Ergodic-set normalization, an effective and outlier ratio

**Algorithm 1: MTL**


---

```

1 Input: Initial Outlier Score Function:  $F_{\text{init}}(\mathbf{X})$ , Outlier Scoring Method:  $M(\cdot)$ 
2 Output: Advanced Outlier Score Function:  $F_{\text{M+Multi-T}}(\mathbf{X})$ 
3 // Stage 1: Identifying Uncontaminated Inliers
4 for  $b$  in iterations do
5   if not converged then
6      $\hat{F}_{\text{init}}(\mathbf{X}) = \text{sort}(F_{\text{init}}(\mathbf{X}))$  // sort-transform(Eq.9)
7      $I^b = \{\hat{F}_{\text{init}}(\mathbf{x}_i) | i < \max_i \{g(a_i) > \hat{F}_{\text{init}}(\mathbf{x}_i)\}\}$  // linear-fit(Eq.10)
8      $\phi_{\text{out}}^b = \text{mean}(I^b) + 3 \cdot \text{std}(I^b)$  // outlier threshold (Eq.11)
9      $I^{b+1} = I^b \cdot \text{remove}(\{F_{\text{init}}(\mathbf{x}_i) | F_{\text{init}}(\mathbf{x}_i) > \phi_{\text{out}}^b\})$  // filtering(Eq.11)
10    $b = b + 1$ 
11 Converged  $b : b^*$ 
12  $\phi_{\text{in}} = \max(I^{b^*})$  // inlier threshold(Eq.12)
13  $\mathbf{X}'_{\text{in}} = \{\mathbf{x}_i | F_{\text{init}}(\mathbf{x}_i) \leq \phi_{\text{in}}\}$  // predicted inliers(Eq.13)
14 // Stage 2: Identifying Adaptive Outliers
15  $\rho \leftarrow \text{similarity}(F_{\text{S}}(\mathbf{X}), F_{\text{E}}(\mathbf{X}))$  // outlier ratio estimation(Eq.15,16)
16 if  $\rho > 0.3$  then
17    $\phi_{\text{out}} = \phi_{\text{out}}^*$ 
18 else
19   if  $0.1 \leq \rho \leq 0.3$  then
20      $\phi_{\text{out}} = \phi_{\text{out}}^1$ 
21   else
22      $\phi_{\text{out}} = \phi_{\text{out}}^0$ 
23  $\mathbf{X}'_{\text{out}} = \{\mathbf{x}_i | F_{\text{init}}(\mathbf{x}_i) > \phi_{\text{out}}\}$  // predicted outliers(Eq.20)
24 // Stage 3: Adopting Multi-T to Outlier Scoring Method
25 return  $M \cdot \text{fit}(\{\text{S-norm}(\mathbf{x}_i, \mathbf{v}'_{\text{S}}) | \mathbf{x}_i \in \mathbf{X}'_{\text{in}}\}) \cdot \text{predict}(\{\text{S-norm}(\mathbf{x}_i, \mathbf{v}'_{\text{S}}) | \mathbf{x}_i \in \mathbf{X}\})$ 

```

---

invariant normalization procedure, illustrated as follows:

$$\text{E-norm}(\mathbf{x}_i, \mathbf{v}_{\text{E}}) = \frac{\mathbf{x}_i - \mathbf{v}_{\text{E}}}{\|\mathbf{x}_i - \mathbf{v}_{\text{E}}\|_2}, \mathbf{v}_{\text{E}} = \left[ \frac{1}{n \cdot d} \sum_{i=1}^n \sum_{j=1}^d \mathbf{x}_{i,j}, \dots, \frac{1}{n \cdot d} \sum_{i=1}^n \sum_{j=1}^d \mathbf{x}_{i,j} \right], \quad (5)$$

where  $\mathbf{v}_{\text{E}}$  is its reference vector and  $j$  is the dimension index. To gain higher efficiency and maintain efficacy, we set  $T = 1$ , such that the initial outlier score function (LVAD-S) is formulated as:

$$F_{\text{init}}(\mathbf{X}) = F_{\text{LVAD-S}}(\mathbf{X}) = \{\text{Dist}(\text{E-norm}(\mathbf{x}_i, \mathbf{v}_{\text{E}}), \text{E-norm}(\mathbf{m}_{\mathbf{X}}, \mathbf{v}_{\text{E}}))\}_{i=1}^n, \quad (6)$$

$$\mathbf{m}_{\mathbf{X}} = \left[ \frac{1}{n} \sum_{i=1}^n \mathbf{x}_{i,1}, \dots, \frac{1}{n} \sum_{i=1}^n \mathbf{x}_{i,d} \right],$$

where  $\mathbf{m}_{\mathbf{X}}$  is the mean of target dataset's features. Despite its simplicity, Eq. 6 is still a reliable outlier scoring method [34].

## 4.2 Multi-T: Multiple Thresholding

**Motivation.** In the field of UOD, the conventional thresholding paradigm is to learn a *single* threshold  $\phi$  on the initial outlier score function  $F_{\text{init}}(\mathbf{X})$ . However, it ignores the impact of the outlier ratio  $\gamma$ . As the mean of target dataset  $\mathbf{m}_\mathbf{X}$  will be shifted to outliers when  $\gamma$  becomes higher, i.e., mean-shift problem [47], there will be an inevitable overlap between inlier and outlier score distributions. So we define the distribution of the initial outlier score function  $D$  as follows:

$$D = F_{\text{init}}(\mathbf{X}) = F_{\text{init}}(\mathbf{X}_{\text{in}}) \cup F_{\text{init}}(\mathbf{X}_{\text{out}}) = I \cup A \cup O, \quad (7)$$

where  $I$  and  $O$  refer to the uncontaminated inlier and outlier score distributions,  $A$  refers to the overlap. Such that we are concerned about *multi*-thresholding instead of the conventional way that explicitly finding a *single* threshold between  $F_{\text{init}}(\mathbf{X}_{\text{in}})$  and  $F_{\text{init}}(\mathbf{X}_{\text{out}})$ . Specifically, our objective is identifying the inlier threshold  $\phi_{\text{in}}$  and outlier threshold  $\phi_{\text{out}}$  that isolate  $I$  and  $O$  from  $D$ .

**Identifying uncontaminated inliers.** We identify  $I$  by understanding the initial outlier score function (Eq. 6), which computes the normalized  $\ell_2$ -norm of each instance  $\mathbf{x}_i \in \mathbf{X}$  relative to the mean of  $\mathbf{X}$ . Inspired by Shell Theory [32],  $I$  satisfies a Gaussian-like distribution, the classic statistical estimation of  $I$  is:

$$I = \{F_{\text{init}}(\mathbf{x}_i) | F_{\text{init}}(\mathbf{x}_i) < \text{mean}(D) + k \cdot \text{std}(D)\}, \quad (8)$$

where  $\text{mean}(D)$  and  $\text{std}(D)$  refer to the mean and standard deviation,  $k$  is the hyper-parameter of the " $k$ -sigma" rule. In practice, optimizing  $k$  without supervision is challenging. Thus, we convert to a non-parametric way by utilizing an ascending sort projection  $\text{sort}(\cdot)$  to  $F_{\text{init}}(\mathbf{X})$ , which projects  $D$  into a 2- $d$  space ( $x$ -axis: instance index,  $y$ -axis: sorted outlier score), the sorted outlier score is:

$$\hat{F}_{\text{init}}(\mathbf{X}) = \text{sort}(\hat{F}_{\text{init}}(\mathbf{X})), \text{ i.e., } \hat{F}_{\text{init}}[i+1] > \hat{F}_{\text{init}}[i]. \quad (9)$$

Intuitively,  $\hat{I} = \text{sort}(I)$  can be fitted with a naive linear regressior  $g(a) = \beta^\top \cdot a = \beta_0 + \beta_1 \cdot a$ , as shown in Fig. 2 (b). So we identify the potential inliers as:

$$I = \left\{ \hat{F}_{\text{init}}(\mathbf{x}_i) | i < \max_i \left\{ g(a_i) > \hat{F}_{\text{init}}(\mathbf{x}_i) \right\} \right\}, \text{ s.t. } \min_{\beta} \|a^\top \beta - \hat{F}_{\text{init}}(\mathbf{X})\|_2, \quad (10)$$

where  $\beta$  is the coefficients. Inevitably, this fitting process will be shifted with outliers, which is addressed together with the following outliers' identification.

**Identifying adaptive outliers.** In statistics, 3-sigma rule [42] declares variables  $C$  has a large probability within three standard deviations from the mean if  $C$  follows a Gaussian-like distribution. So we employ it with two aspects:

(i) iteratively filter outliers during the phase of inliers identification:

$$\phi_{\text{out}}^b = \text{mean}(I^b) + 3 \cdot \text{std}(I^b), I^{b+1} = I^b \cdot \text{remove}(\{F_{\text{init}}(\mathbf{x}_i) | F_{\text{init}}(\mathbf{x}_i) > \phi_{\text{out}}^b\}), \quad (11)$$

where  $b$  refers to the  $b$ -th iteration. When converged, the inlier threshold is:

$$\phi_{\text{in}} = \max(I^{b*}), \quad (12)$$

where  $I^{b*}$  refers to the converged inlier prediction and we can identify inliers as:

$$\mathbf{X}'_{\text{in}} = \{\mathbf{x}_i | F_{\text{init}}(\mathbf{x}_i) \leq \phi_{\text{in}}\}. \quad (13)$$

(ii) During the above phase, we estimate the outlier threshold candidates:

$$\phi_{\text{out}}^1 = \text{mean}(I^1) + 3 \cdot \text{std}(I^1), \phi_{\text{out}}^* = \text{mean}(I^{b*}) + 3 \cdot \text{std}(I^{b*}), \quad (14)$$

where  $\phi_{\text{out}}^1, \phi_{\text{out}}^*$  refer to the outlier threshold candidates at the first and final iterations, respectively, i.e.,  $\phi_{\text{out}}^1 > \phi_{\text{out}}^*$ . To obtain a  $\gamma$ -adaptive outlier threshold, we consider the impact and relationship of two mentioned normalization procedures: Shell Normalization S-norm( $\cdot$ ) (Eq. 3) and Ergodic-set Normalization E-norm( $\cdot$ ) (Eq. 5). Both them follow a similar normalization formula (denominator:  $\ell_2$ -norm), the difference is their reference vectors  $\mathbf{v}'_S$  and  $\mathbf{v}_E$ , contrast as:

$$\begin{aligned} \mathbf{v}'_S &= \left[ \frac{1}{n} \sum_{i=1}^n \mathbf{X}'_{\text{out}}[i][1], \dots, \frac{1}{n} \sum_{i=1}^n \mathbf{X}'_{\text{out}}[i][d] \right], \\ \mathbf{v}_E &= \left[ \frac{1}{n \cdot d} \sum_{i=1}^n \sum_{j=1}^d \mathbf{x}_{i,j}, \dots, \frac{1}{n \cdot d} \sum_{i=1}^n \sum_{j=1}^d \mathbf{x}_{i,j} \right]. \end{aligned} \quad (15)$$

Obviously, Shell normalization is strongly subjective to the efficacy of outlier prediction  $\mathbf{X}'_{\text{out}}$  while Ergodic-set Normalization is independent with  $\gamma$  that is a sub-optimal but stable operation. To compare these two normalization procedures, we first employ them to Eq. 6, and obtain two outlier score functions:

$$\begin{aligned} F_S(\mathbf{X}) &= \{\text{Dist}(\text{S-norm}(\mathbf{x}_i, \mathbf{v}'_S), \text{S-norm}(\mathbf{m}_X, \mathbf{v}'_S))\}_{i=1}^n, \\ F_E(\mathbf{X}) &= \{\text{Dist}(\text{E-norm}(\mathbf{x}_i, \mathbf{v}_E), \text{E-norm}(\mathbf{m}_X, \mathbf{v}_E))\}_{i=1}^n, \end{aligned} \quad (16)$$

where  $\mathbf{v}'_S$  comes from  $\mathbf{X}'_{\text{out}}$  obtained by MAD [6] (Shell Normalization's default thresholding method that is only effective on high- $\gamma$ ). Subsequently, two outlier score functions  $F_S(\mathbf{X})$  and  $F_E(\mathbf{X})$  are arranged in ascending order, and we denote the ranked-index lists as  $\mathcal{R}_S$  and  $\mathcal{R}_E$ . The outlier ratio  $\gamma$  can be approximately estimated with the similarity between  $\mathcal{R}_S$  and  $\mathcal{R}_E$  since  $F_S(\mathbf{X})$  and  $F_E(\mathbf{X})$  share a consistent structure (distance-to-the-mean), making  $\mathcal{R}_S$  and  $\mathcal{R}_E$  comparable. Besides,  $F_E(\mathbf{X})$  serves as a reliably effective baseline. When  $\gamma$  is low, S-norm( $\mathbf{X}, \mathbf{v}'_S$ ) tends to under-perform, lead the similarity between  $\mathcal{R}_S$  and  $\mathcal{R}_E$  becomes low. In contrast, with a high  $\gamma$ , S-norm( $\mathbf{X}, \mathbf{v}'_S$ ) merely refines the ranking of a small number of outliers, indicative of high similarity. The similarity  $\rho \in [-1, 1]$  is computed with the Pearson correlation coefficient:

$$\rho = \frac{\text{cov}(\mathcal{R}_S, \mathcal{R}_E)}{\text{std}(\mathcal{R}_S) \cdot \text{std}(\mathcal{R}_E)}, \quad (17)$$

where  $\text{cov}(\mathcal{R}_S, \mathcal{R}_E)$  and  $\text{std}(\mathcal{R}_S)/\text{std}(\mathcal{R}_E)$  are the covariance and standard deviation of the rank variables. If the similarity is relatively low ( $< 0.1$ ), we compute the 3-sigma of the entire outlier score distribution  $D$  as the outlier threshold:

$$\phi_{\text{out}}^0 = \text{mean}(D) + 3 \cdot \text{std}(D). \quad (18)$$



It is a conservative estimator for outliers since the above linear fitting might suffer from some challenges at low- $\gamma$  since there is no explicit outlier score distribution. If the similarity is large ( $> 0.3$ ), we choose the relatively smooth outlier threshold  $\phi_{\text{out}}^*$ , while  $\phi_{\text{out}}^1$  is used for middle- $\gamma$  cases. The outlier threshold is described as:

$$\phi_{\text{out}} = \begin{cases} \phi_{\text{out}}^* & , \text{ if } \rho > 0.3 \\ \phi_{\text{out}}^1 & , \text{ if } 0.1 \leq \rho \leq 0.3 \\ \phi_{\text{out}}^0 & , \text{ otherwise} \end{cases} \quad (19)$$

Therefore, we can subsequently identify outlier ratio invariant outliers as:

$$\mathbf{X}'_{\text{out}} = \{\mathbf{x}_i | F_{\text{init}}(\mathbf{x}_i) > \phi_{\text{out}}\}. \quad (20)$$

### 4.3 Implementation

Firstly, we can directly employ Multi-T with the distance-to-the-mean paradigm:

$$\begin{aligned} F_{\text{Multi-T}}(\mathbf{X}) &= \{\text{Dist}(\text{S-norm}(\mathbf{x}_i, \mathbf{v}'_S), \text{S-norm}(\mathbf{m}_{\mathbf{X}'_{\text{in}}}, \mathbf{v}'_S))\}_{i=1}^n, \\ \mathbf{v}'_S &= \left[ \frac{1}{n} \sum_{i=1}^n \mathbf{X}'_{\text{out}}[i][1], \dots, \frac{1}{n} \sum_{i=1}^n \mathbf{X}'_{\text{out}}[i][d] \right], \\ \mathbf{m}_{\mathbf{X}'_{\text{in}}} &= \left[ \frac{1}{n} \sum_{i=1}^n \mathbf{X}'_{\text{in}}[i][1], \dots, \frac{1}{n} \sum_{i=1}^n \mathbf{X}'_{\text{in}}[i][d] \right]. \end{aligned} \quad (21)$$

Secondly, Multi-T can be integrated with UOD methods  $M(\cdot)$ , illustrated as:

$$\begin{aligned} F_{M+\text{Multi-T}}(\mathbf{X}) &= M.\text{fit}(\{\text{S-norm}(\mathbf{x}_i, \mathbf{v}'_S) | \mathbf{x}_i \in \mathbf{X}'_{\text{in}}\}) \\ &\quad .\text{predict}(\{\text{S-norm}(\mathbf{x}_i, \mathbf{v}'_S) | \mathbf{x}_i \in \mathbf{X}\}). \end{aligned} \quad (22)$$

## 5 Experiments

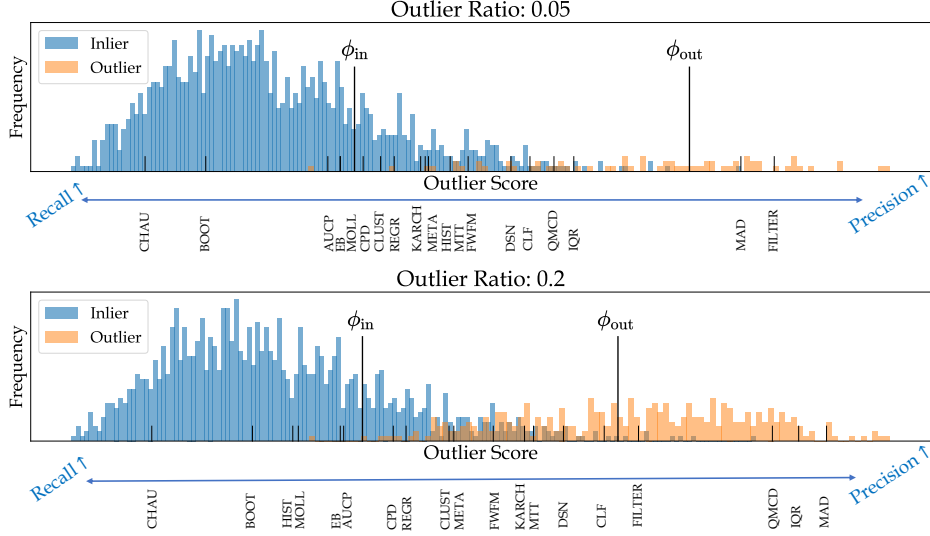
We cover our experiments for (i) comparing with other threshold learners on dataset splitting (Sec. 5.1); (ii) measuring the performance improvement of adopting the Multi-T to previously proposed outlier scoring methods (Sec. 5.2).

**Benchmarks and Feature Extraction.** Based on the existing research [32, 34], we use the raw pixel representation for grayscale datasets, including MNIST [29] and Fashion-MNIST [59]. For RGB datasets such as STL-10 [15], Internet [32], CIFAR-10 [27], MIT-Places-Small [61], we adopt two deep feature extractors: ImageNet pretrained [20] ResNet-50 [21] (ResNet)<sup>2</sup>, and CLIP [45].

**Target Dataset.** In our experiments, each class within a benchmark dataset is alternatively regarded as inliers, with instances from all other classes considered outliers. The target dataset includes all inliers along with a randomly selected subset of outliers. Results for each dataset are averaged across all classes. Additionally, for every round, we further average the results over a wide range of outlier ratios:  $[0.05, 0.1, 0.2, 0.3, 0.4]^3$ , to assess the method’s robustness/stability.

<sup>2</sup> Unless otherwise stated, ResNet-50 is the default feature extractor.

<sup>3</sup> Unless otherwise stated, the experimental results are averaged across  $[0.05, \dots, 0.4]$ .



**Fig. 3:** Qualitative results compared with SOTA threshold learners. The experiments are conducted on STL-10 (Inlier class: Monkey).

### 5.1 Threshold Learning

**Competing Methods.** We compare our proposed Multi-T with those effective thresholding (TL) methods involving Kernel-based: AUCP [48], FGD [43]; Curve-based: EB [17]; Normality-based: DSN [5], CHAU [10]; Filtering-based: FILTER [19], HIST [55]; Statistical-based: MTT [49], BOOT [37], QMCD [24], CLF [7], IQR [8], KARCH [1], MCST [16], GESD [4], REGR [3], MAD [6], CLUST [11], CPD [12] and Transformation-based: MOLL [25], YJ [46].

**Evaluation Metric.** The performance of general thresholding is measured with  $F_\beta$ -score, defined as follows:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}, \quad (23)$$

where  $\beta < 1$  tilts towards precision while  $\beta > 1$  prioritizes recall. In this work, we utilize  $F_{0.1}$ -score ( $F_{0.1}$ ) and  $F_{10}$ -score ( $F_{10}$ ) that measure the accuracy of predicted outliers and inliers. Besides,  $F_1$ -score ( $\beta = 1$ ), a harmonic mean of the precision and recall, is not practical for this task since the complete separability between inliers' and outliers' score distributions cannot be guaranteed.

**Main Results.** Across a series of competing threshold learners, while one method may achieve a commendable  $F_{0.1}$ -score of 0.976, the corresponding  $F_{10}$ -score is mere 0.452, and vice versa ( $F_{10}$ -score: 0.979;  $F_{0.1}$ -score: 0.447). By contrast, our solution maintains its efficacy across various evaluation criteria ( $F_{0.1}$ -score: 0.917,  $F_{10}$ -score: 0.980). It exhibits remarkable stability across a diverse spectrum of outlier ratios, as illustrated in Fig. 3. To further investigate the benefits of our technique, Tab. 1 categorizes two groups, each representing a different

**Table 1:** Average  $F_\beta$ -score of thresholding methods on STL-10. "+GT Norm" refers to the ideal initial outlier score function. "Highest  $F_\beta$ " ( $\beta$ : 0.1, 10) refers to the method that garners the highest  $F_\beta$ -score across all baseline models. The better average result is highlighted in **bold**.

Outlier Score Fun.	Method (TL)	ResNet-50			CLIP		
		$F_{0.1}$	$F_{10}$	Avg.	$F_{0.1}$	$F_{10}$	Avg.
LVAD-S [34]	Highest $F_{0.1}$	0.911	0.454	0.682	0.936	0.401	0.669
	Highest $F_{10}$	0.382	0.967	0.674	0.357	0.954	0.655
	Multi-T(Ours)	0.840	0.869	<b>0.855</b>	0.860	0.866	<b>0.863</b>
+GT Norm [32]	Highest $F_{0.1}$	0.976	0.452	0.714	0.977	0.448	0.713
	Highest $F_{10}$	0.447	0.979	0.713	0.691	0.986	0.839
	Multi-T(Ours)	0.917	0.980	<b>0.949</b>	0.857	0.984	<b>0.920</b>

outlier score function. Such categorization is pivotal as threshold methods ought to accommodate various foundational bases.

**Table 2:** Average AUC results compared with SOTA outlier scoring methods (outlier detectors). **Blue** and **Orange** indicates the best and second-best results, respectively.

Method (OS)	STL-10		CIFAR-10		CIFAR-100		MIT-Places		MNIST
	ResNet	CLIP	ResNet	CLIP	ResNet	CLIP	ResNet	CLIP	Pixel
IF [35]	0.836	0.943	0.780	0.891	0.790	0.866	0.687	0.868	0.776
LOF [11]	0.628	0.626	0.673	0.621	0.839	0.849	0.556	0.520	0.754
RSRAE [28]	<b>0.962</b>	0.938	0.862	0.879	0.914	0.885	<b>0.874</b>	0.876	0.793
ECOD [31]	0.907	<b>0.981</b>	<b>0.873</b>	<b>0.935</b>	0.873	<b>0.918</b>	0.777	<b>0.943</b>	0.734
LUNAR [18]	0.776	0.821	0.767	0.774	0.838	0.899	0.643	0.871	0.797
Shell-Re. [32]	0.862	0.838	0.860	0.813	0.835	0.813	0.826	0.914	0.776
LVAD [34]	0.954	0.968	0.860	0.917	<b>0.921</b>	0.917	0.844	0.919	<b>0.867</b>
Multi-T	<b>0.968</b>	<b>0.989</b>	<b>0.895</b>	<b>0.957</b>	<b>0.938</b>	<b>0.956</b>	<b>0.867</b>	<b>0.974</b>	<b>0.897</b>
DeepSVDD [50]	0.622	0.597	0.560	0.509	0.563	0.581	0.583	0.549	0.513
+Multi-T	0.925	0.921	0.769	0.819	0.835	0.826	0.755	0.832	0.732
Improve.	48.7%	54.3%	37.3%	60.9%	48.3%	42.2%	29.5%	51.6%	37.9%
OCSVM [51]	0.927	0.921	0.826	0.850	0.879	0.850	0.826	0.871	0.831
+Multi-T	0.957	0.965	0.859	0.916	0.916	0.899	0.846	0.924	0.863
Improve.	3.24%	4.78%	4.00%	7.76%	4.21%	5.76%	2.42%	6.08%	3.85%

## 5.2 Outlier Scoring

**Competing Methods.** We compare two themes of baseline outlier scoring (OS) methods: (i) statistical-based methods: IF [35], OCSVM [51], DeepSVDD [51], ECOD [31], LUNAR [28], RSRAE [28], Shell-Re. [32], LVAD [34]. (ii) deep-learning-based models: GOAD [9], ICL [53], REPEN [39], NeuTraL [44], SLAD [60]. For a fair comparison, we apply Ergodic-set normalization [34] if it improves the performance of the baseline algorithms, such as IF [35], OCSVM [51].

**Evaluation Metric.** The performance of outlier scoring (ranking accuracy) is primarily assessed using the Area Under the Receiver Operating Characteristic curve (AUC). This metric provides a thorough assessment of ranking accuracy.

**Main Results.** Tab. 2 shows that Multi-T can be seamlessly integrated with two classic outlier scoring methods: DeepSVDD [51], and OCSVM [51] while exhibiting significant improvements in both efficacy and stability across a diverse range of outlier ratios and various benchmarks. In most of our experiments, Multi-T itself achieves SOTA results. Notably, even in the case of non-aligned and low-resolution datasets like CIFAR-10 [27] and CIFAR-100 [27], which are known to pose challenges [41]. It surpasses the current SOTA AUC scores by margins of 7.76% and 5.76% for CIFAR-10 and CIFAR-100, respectively. Additionally, we observe a logical and significant enhancement in performance with improved feature representation, e.g., for the MIT-Places [61] dataset, the AUC improves from 0.867 using ResNet-50 [21] to 0.974 with CLIP [45].

**Table 3:** Average AUC results, for more outlier scoring methods with(w/) or without(w/o) our Multi-T module, conducted on STL-10.

Feature	IF [35]		ECOD [11]		ABOD [26]		PCA [54]		GMM [2]	
	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/
ResNet	0.836	0.899	0.907	0.919	0.665	0.883	0.865	0.945	0.859	0.952
CLIP	0.943	0.983	0.981	0.984	0.715	0.909	0.984	0.994	0.892	0.962
Avg.	0.890	0.941	0.944	0.951	0.690	0.896	0.925	0.970	0.876	0.957
Feature	GOAD [9]		ICL [53]		REPEN [39]		NeuTraL [44]		SLAD [60]	
	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/
ResNet	0.952	0.962	0.934	0.957	0.877	0.889	0.854	0.950	0.941	0.962
CLIP	0.961	0.989	0.951	0.982	0.879	0.930	0.851	0.971	0.945	0.985
Avg.	0.957	0.975	0.943	0.970	0.878	0.909	0.853	0.961	0.943	0.973

In Tab. 3, we present the results of Multi-T integrated with more methods involving both statistical and deep models, which shows its broad applications. Moreover, our solution excels not just in detection accuracy but also in running speed since Multi-T compresses the size of fitting data, as illustrated in Tab. 4, 5. In Fig. 4, we compare with the most related work Shell-Re. [32], which

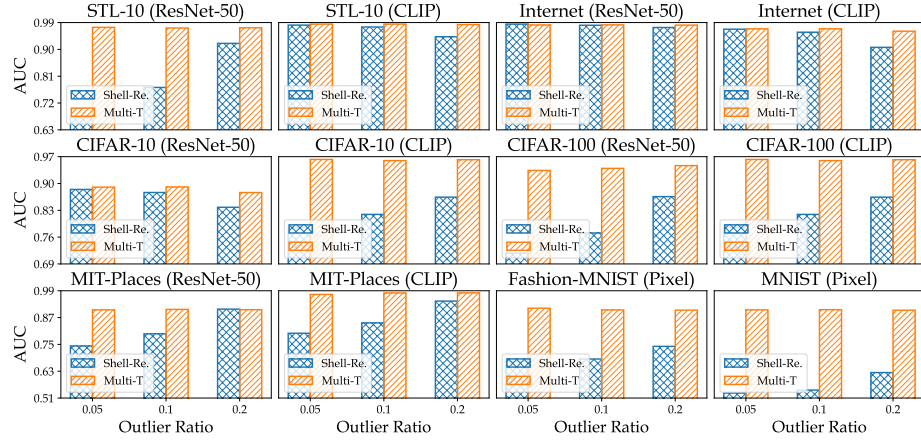
has a built-in Robust-Least-Square (RLS) thresholding procedure with MAD [6]. Apparently, our method is capable of estimating multiple thresholds that facilitate a clear demarcation of diverse scenarios (outlier ratio, dataset domain, and feature representation).

**Table 4:** Efficiency comparison for outlier scoring. Timing is measured with 10,000 samples, GPU: NVIDIA RTX 3080.

Method (OS)	Device	Time ( $\downarrow$ )
LVAD [34]	CPU	645.2s
RSRAE [28]	GPU	121.6s
Multi-T	CPU	<b>1.2s</b>
OCSVM [51]	CPU	154.8s
+Multi-T	CPU	105.4s
GOAD [9]	GPU	268.1s
+Multi-T	GPU	211.7s

**Table 5:** Efficiency comparison for thresholding. All thresholding methods are conducted on CPU.

Method (TL)	Time ( $\downarrow$ )
CPD [12]	3.61s
FWFM [19]	1.26s
DSN [5]	3.79s
CLUST [11]	7.36s
AUCP [48]	1.26s
Multi-T	<b>0.97s</b>



**Fig. 4:** AUC results of Multi-T compared with Shell-Re. [32] over various outlier ratios.

**Discussions.** We conclude the feasibility of adopting Multi-T to outlier scoring methods into three aspects: (i) both normalization with *outliers* and manifold learning with *inliers* are necessary for most of outlier scoring methods; (ii) Despite the initial outlier score distribution (Eq. 6) is usually not perfectly separable, it provides a reliable baseline. (iii) Multi-T is able to isolate the overlap region and predict both uncontaminated inliers.

**Table 6:** Ablation study for DeepSVDD and LVAD-S with Multi-T ( $\gamma$ : 0.2).

DeepSVDD	AUC	LVAD-S	AUC
+Multi-T	<b>0.950</b>	+Multi-T	<b>0.971</b>
w/o In	0.643	w/o In	<b>0.965</b>
w/o Out	<b>0.939</b>	w/o Out	0.951

**Table 7:** Comparison with  $k$ -sigma.

Metric	Method	Score
$F_{0.1}$	3-sigma	0.406
	Multi-T	<b>0.840</b>
$F_{10}$	1-sigma	0.806
	Multi-T	<b>0.869</b>

### 5.3 Ablation Study

The experiments thus far have established the effectiveness of Multi-T. However, there remains a concern regarding how sensitive the performance improvement is to Multi-T's two primary components: adaptive outliers (Out) and uncontaminated inliers (In), which refer to the normalization and manifold learning procedures, respectively. We select two representative outlier scoring models: DeepSVDD [50] and LVAD-S [34], whereas DeepSVDD follows a widely-used outlier detection mechanism, i.e., learning the normality (hyper-sphere), and identifying inliers is of great significance. Without predicted inliers, the AUC result decreases from 0.950 to 0.643 on STL-10, shown in Tab. 6. Additionally, since the normalization procedure can be considered as a distance de-noising procedure [33], Multi-T will contribute to those methods with distance computation, e.g., the result of LVAD-S is improved from 0.951 to **0.971** on STL-10 and 0.838 to **0.882** on CIFAR-10, which verify our prior assumption that the identification of both inliers and outliers is of great significance, which enhances the reliability of multiple thresholding perspective. Moreover, Tab. 7 indicates that our thresholding process markedly surpasses the classical  $k$ -sigma rule [42].

### 5.4 Limitation

Our method is tied to the ranking accuracy (separability) of the initial outlier score function. Based on most related works, we consider UOD as an *one-class* learning task (only one inlier/normal class in the target dataset). However we find in some specific cases, there might exist multi-normal classes, e.g., the digit 5 class of the MNIST dataset. In that case, the efficacy of our method might be decreased. Additionally, this statistical-based method has limitations in very small-scale datasets since the "three-sigma" rule will be less effective.

## 6 Conclusion

This work introduces a novel perspective for UOD about advancing existing outlier detectors (scoring methods) via thresholding. To this end, we propose a multiple thresholding (Multi-T) module, to label the unlabelled target dataset. Comprehensive experiments verify that the Multi-T can significantly improve both the efficacy and efficiency of previously proposed outlier scoring methods.

## Acknowledgement

We would like to thank the reviewers for their valuable comments and suggestions.

## References

1. Afsari, B.: Riemannian  $L^p$  center of mass: existence, uniqueness, and convexity. *Proceedings of the American Mathematical Society* pp. 655–673 (2011)
2. Aggarwal, C.C.: *Outlier analysis* second edition (2016)
3. Aggarwal, C.C., Aggarwal, C.C.: *An introduction to outlier analysis*. Springer (2017)
4. Alrawashdeh, M.J.: An adjusted grubbs’ and generalized extreme studentized deviation. *Demonstratio Mathematica* pp. 548–557 (2021)
5. Amagata, D., Onizuka, M., Hara, T.: Fast and exact outlier detection in metric spaces: a proximity graph-based approach. In: *Proceedings of the 2021 International Conference on Management of Data*. pp. 36–48 (2021)
6. Archana, N., Pawar, S.: Periodicity detection of outlier sequences using constraint based pattern tree with mad. *International Journal of Advanced Studies in Computers, Science and Engineering* p. 34 (2015)
7. Barbado, A., Corcho, Ó., Benjamins, R.: Rule extraction in unsupervised anomaly detection for model explainability: Application to one-class svm. *Expert Systems with Applications* p. 116100 (2022)
8. Bardet, J.M., Dimby, S.F.: A new non-parametric detector of univariate outliers for distributions with unbounded support. *Extremes* pp. 751–775 (2017)
9. Bergman, L., Hoshen, Y.: Classification-based anomaly detection for general data. *arXiv preprint arXiv:2005.02359* (2020)
10. Bol’shev, L., Ubaidullaeva, M.: Chauvenet’s test in the classical theory of errors. *Theory of Probability & Its Applications* pp. 683–692 (1975)
11. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*. pp. 93–104 (2000)
12. Van den Burg, G.J., Williams, C.K.: An evaluation of change point detection algorithms. *arXiv preprint arXiv:2003.06222* (2020)
13. Camposeco, F., Sattler, T., Cohen, A., Geiger, A., Pollefeys, M.: Toroidal constraints for two-point localization under high outlier ratios. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 4545–4553 (2017)
14. Chalapathy, R., Menon, A.K., Chawla, S.: Robust, deep and inductive anomaly detection. In: *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. pp. 36–51 (2017)
15. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*. pp. 215–223 (2011)
16. Coin, D.: Testing normality in the presence of outliers. *Statistical Methods and Applications* pp. 3–12 (2008)
17. Friendly, M., Monette, G., Fox, J.: Elliptical insights: understanding statistical methods through elliptical geometry. *Statistical Science* pp. 1–39 (2013)

18. Goodge, A., Hooi, B., Ng, S.K., Ng, W.S.: Lunar: Unifying local outlier detection methods via graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 6737–6745 (2022)
19. Hashemi, N., German, E.V., Ramirez, J.P., Ruths, J.: Filtering approaches for dealing with noise in anomaly detection. In: 2019 IEEE 58th Conference on Decision and Control. pp. 5356–5361 (2019)
20. He, K., Girshick, R., Dollár, P.: Rethinking imagenet pre-training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4918–4927 (2019)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
22. Hendrycks, D., Mazeika, M., Dietterich, T.: Deep anomaly detection with outlier exposure. arXiv preprint arXiv:1812.04606 (2018)
23. Huyan, N., Quan, D., Zhang, X., Liang, X., Chanussot, J., Jiao, L.: Unsupervised outlier detection using memory and contrastive learning. IEEE Transactions on Image Processing pp. 6440–6454 (2022)
24. Iouchtchenko, D., Raymond, N., Roy, P.N., Nooijen, M.: Deterministic and quasi-random sampling of optimized gaussian mixture distributions for vibronic monte carlo. arXiv preprint arXiv:1912.11594 (2019)
25. Keyzer, M.A., Sonneveld, B.: Using the mollifier method to characterize datasets and models: the case of the universal soil loss equation. ITC Journal pp. 263–272 (1997)
26. Kriegel, H.P., Schubert, M., Zimek, A.: Angle-based outlier detection in high-dimensional data. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 444–452 (2008)
27. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
28. Lai, C.H., Zou, D., Lerman, G.: Robust subspace recovery layer for unsupervised anomaly detection. arXiv preprint arXiv:1904.00152 (2019)
29. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010)
30. Li, T., Wang, Z., Liu, S., Lin, W.Y.: Deep unsupervised anomaly detection. In: Proceedings of IEEE Winter Conference on Applications of Computer Vision. pp. 3636–3645 (2021)
31. Li, Z., Zhao, Y., Hu, X., Botta, N., Ionescu, C., Chen, G.H.: Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. arXiv preprint arXiv:2201.00382 (2022)
32. Lin, D., Liu, S., Li, H., Cheung, N.M., Ren, C., Matsushita, Y.: Shell theory: A statistical model of reality. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
33. Lin, W.Y., Liu, S., Dai, B.T., Li, H.: Distance based image classification: A solution to generative classification’s conundrum? International Journal of Computer Vision pp. 177–198 (2023)
34. Lin, W.Y., Liu, Z., Liu, S.: Locally varying distance transform for unsupervised visual anomaly detection. In: European Conference on Computer Vision. pp. 354–371. Springer (2022)
35. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: Proceedings of International Conference on Data Mining. pp. 413–422 (2008)
36. Liznerski, P., Ruff, L., Vandermeulen, R.A., Franks, B.J., Müller, K.R., Kloft, M.: Exposing outlier exposure: What can be learned from few, one, and zero outlier images. arXiv preprint arXiv:2205.11474 (2022)



37. Martin, M.A., Roberts, S.: An evaluation of bootstrap methods for outlier detection in least squares regression. *Journal of Applied Statistics* pp. 703–720 (2006)
38. Masci, J., Meier, U., Cireřan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: *Proceedings of International Conference on Artificial Neural Networks*. pp. 52–59 (2011)
39. Pang, G., Cao, L., Chen, L., Liu, H.: Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. pp. 2041–2050 (2018)
40. Parzen, E.: On estimation of a probability density function and mode. *The annals of mathematical statistics* pp. 1065–1076 (1962)
41. Perera, P., Nallapati, R., Xiang, B.: Ogan: One-class novelty detection using gans with constrained latent representations. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2898–2906 (2019)
42. Pukelsheim, F.: The three sigma rule. *The American Statistician* pp. 88–91 (1994)
43. Qi, Z., Jiang, D., Chen, X.: Iterative gradient descent for outlier detection. *International Journal of Wavelets, Multiresolution and Information Processing* p. 2150004 (2021)
44. Qiu, C., Pfrommer, T., Kloft, M., Mandt, S., Rudolph, M.: Neural transformation learning for deep anomaly detection beyond images. In: *International Conference on Machine Learning*. pp. 8703–8714 (2021)
45. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: *Proceedings of International Conference on Machine Learning*. pp. 8748–8763 (2021)
46. Raymaekers, J., Rousseeuw, P.J.: Transforming variables to central normality. *Machine Learning* pp. 1–23 (2021)
47. Reiss, T., Hoshen, Y.: Mean-shifted contrastive loss for anomaly detection. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. pp. 2155–2162 (2023)
48. Ren, K., Yang, H., Zhao, Y., Chen, W., Xue, M., Miao, H., Huang, S., Liu, J.: A robust auc maximization framework with simultaneous outlier detection and feature selection for positive-unlabeled classification. *IEEE Transactions on Neural Networks and Learning Systems* pp. 3072–3083 (2018)
49. Rengasamy, D., Rothwell, B.C., Figueredo, G.P.: Towards a more reliable interpretation of machine learning outputs for safety-critical systems using feature importance fusion. *Applied Sciences* p. 11854 (2021)
50. Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: *Proceedings of International Conference on Machine Learning*. pp. 4393–4402 (2018)
51. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. *Neural Computation* **13**(7), 1443–1471 (2001)
52. Schölkopf, B., Smola, A.J., Bach, F., et al.: *Learning with kernels: support vector machines, regularization, optimization, and beyond* (2002)
53. Shenkar, T., Wolf, L.: Anomaly detection for tabular data with internal contrastive learning. In: *International Conference on Learning Representations* (2021)
54. Shyu, M.L., Chen, S.C., Sarinapakorn, K., Chang, L.: A novel anomaly detection scheme based on principal component classifier. In: *Proceedings of the IEEE foundations and new directions of data mining workshop*. pp. 172–179 (2003)

55. Thanammal, K., Jayasudha, J., Vijayalakshmi, R., Arumugaperumal, S.: Effective histogram thresholding techniques for natural images using segmentation. *Journal of Image and Graphics* pp. 113–116 (2014)
56. Wang, S., Zeng, Y., Liu, X., Zhu, E., Yin, J., Xu, C., Kloft, M.: Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network. *Proceedings of Advances in Neural Information Processing Systems* (2019)
57. Wang, S., Zhu, E., Hu, X., Liu, X., Liu, Q., Yin, J., Wang, F.: Robustness can be cheap: A highly efficient approach to discover outliers under high outlier ratios. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. pp. 5313–5320 (2019)
58. Xia, Y., Cao, X., Wen, F., Hua, G., Sun, J.: Learning discriminative reconstructions for unsupervised outlier removal. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 1511–1519 (2015)
59. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017)
60. Xu, H., Wang, Y., Wei, J., Jian, S., Li, Y., Liu, N.: Fascinating supervisory signals and where to find them: Deep anomaly detection with scale learning. *arXiv preprint arXiv:2305.16114* (2023)
61. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp. 1452–1464 (2017)
62. Zhou, C., Paffenroth, R.C.: Anomaly detection with robust deep autoencoders. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 665–674 (2017)
63. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: *Proceedings of International Conference on Learning Representations* (2018)