# FlexUOD: The Answer to Real-world Unsupervised Image Outlier Detection

Zhonghang Liu
SMU, Singapore
zhliu.2020@phdcs.smu.edu.sg

Kun Zhou
SmartMore Corporation, China
zhoukun303808@gmail.com

Changshuo Wang
NTU, Singapore
wangchangshuo1@gmail.com

Wen-Yan Lin
SMU, Singapore
linwenyan.daniel@gmail.com

Jiangbo Lu
SmartMore Corporation, China
jiangbo.lu@gmail.com

## Abstract

*How many outliers are within an unlabeled and contaminated dataset? Despite a series of unsupervised outlier detection (UOD) approaches have been proposed, they cannot correctly answer this critical question, resulting in their performance instability across various real-world (varying contamination factor) scenarios. To address this problem, we propose FlexUOD, with a novel contamination factor estimation perspective. FlexUOD not only achieves its remarkable robustness but also is a general and plug-and-play framework, which can significantly improve the performance of existing UOD methods. Extensive experiments demonstrate that FlexUOD achieves state-of-the-art results as well as high efficacy on diverse evaluation benchmarks.*

## 1. Introduction

UOD for images plays an important role in improving data quality [29] and supports a wide range of visual applications [62]. It aims to identify unusual samples within an unlabeled dataset. Despite significant progress that has been achieved, existing approaches often exhibit instability across different scenarios (Fig. 1). This work attributes the major instability to the dependence on the crucial but unknown contamination factor $\gamma$[1], and proposes a solution that can be implemented into real-world scenarios, eliminating the dependency of $\gamma$.

We first clarify the $\gamma$-instability: when $\gamma$ is low, the contaminated dataset is primarily composed of inliers that can be effectively addressed by inlier/normal-manifold learning methods [36, 51, 65]. In contrast, the learned inlier manifolds undergo substantial shifts when $\gamma$ is high, rendering these models ineffective. Nevertheless, a higher proportion of outliers benefits certain outlier-based techniques [34, 56].

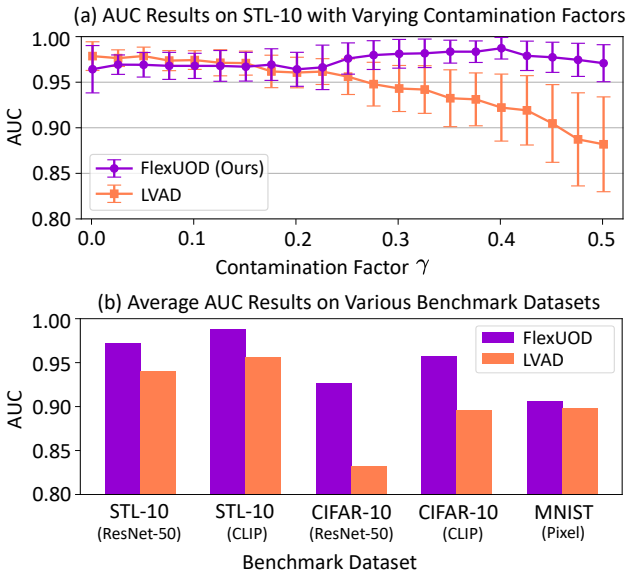


Figure 1. Comparison between our FlexUOD framework and LVAD [36]. (a) demonstrates that our solution maintains stability across varying contamination factors, while (b) highlights its consistent performance across different benchmarks.

Based on the above analysis, it is challenging for a single method to solve varying $\gamma$-scenarios. Thus, **estimating $\gamma$** is our major objective. Currently, the mainstream $\gamma$-estimators [44, 45] are threshold learning-based, focusing on determining a boundary on a series of given outlier scores. However, we argue that without prior knowledge of the outlier scores, e.g., ranking accuracy and distribution, learning an optimal threshold becomes an ill-posed problem. To address the issues arising from the inconsistency between $\gamma$-estimation and outlier scoring, this study introduces a novel perspective.

We propose FlexUOD, a framework that can flexibly

[1]This term [44] is also known as the outlier ratio/percentage.

choose the proper UOD method with varying contamination factors. The outlier score function of FlexUOD can be denoted as:

$$f_{our} \leftarrow \{f_{low}, f_{high}, \gamma\}, \gamma \leftarrow \mathrm{Comp}(f_{low}, f_{high}), \quad (1)$$

where $f_{low}$ and $f_{high}$ refers to appropriate methods for low-$\gamma$ and high-$\gamma$ scenarios, respectively. Instead of independently designing $\{f_{low}, f_{high}\}$ and estimating $\gamma$, our $f_{low}$ and $f_{high}$ are structurally consistent, thereby leveraging their comparison $\mathrm{Comp}(f_{low}, f_{high})$ as a natural prior for estimating $\gamma$.

To assess the performance of our proposed framework, we subject it to rigorous evaluation across diverse benchmarks (classical UOD, industrial-AD, medical-AD, tabular-AD), while spanning a wide range of contamination factors. Despite its remarkable efficiency, the experimental results demonstrate high efficacy and stability, in comparison to existing methods, as shown in Fig. 1. The main contributions of this paper can be summarised as:

- Our research underscores the significance of considering the contamination factor as a pivotal factor in UOD tasks, an aspect often overlooked in previous studies.
- We introduce a novel framework about contamination factor estimation, named FlexUOD which drops out complicated processes/tricks and is plug-and-play. Code is available at: https://github.com/zhliu-uod/FlexUOD.

## 2. Related Work

### 2.1. Unsupervised Outlier Detection

The goal of UOD is to assign an outlier score to each sample. Existing methods can be broadly categorized into: inlier-manifold learning and outlier exposure. Deep conventional outlier detectors, as discussed in [28, 32, 57, 65, 66], primarily focus on modeling the inlier manifold by reconstructing each image sample within a low-dimensional latent space, where inliers are assumed to exhibit lower reconstruction errors. Additionally, statistics-based methods model the inlier manifold using either discrimination-based [34, 36] or density-based [26] approaches. In scenarios where outliers are not the minority [10, 58], different methods, such as outlier exposure [22, 34, 41], become necessary. However, the unknown contamination factor in practice complicates maintaining stability across various scenarios. Thus, contamination factor estimation is essential.

### 2.2. Contamination Factor Estimation

Currently, contamination factor (outlier ratio) estimation is typically achieved through threshold learning, which involves determining a decision boundary based on a series of outlier scores. Representative approaches include kernel-based methods [45], curve-based methods [17], normality-based approaches [5], filtering-based methods [20, 53], statistical methods [1, 2, 5, 8, 24, 42, 63], and transformation-based methods [9, 25, 54]. However, since conventional threshold learning is independent of the outlier detectors, and the discriminative power of the given outlier score function is not guaranteed, threshold-based contamination factor estimation tends to be less effective in practice. To overcome these limitations, this work introduces an outlier detector-aware contamination factor estimator.

## 3. Preliminaries

**Problem Definition.** Given $n$ unlabeled image samples, we define the target dataset as $\mathbf{I} = \mathbf{I}_{in} \cup \mathbf{I}_{out} = \{I_i\}_{i=1}^{n}$, where $\mathbf{I}_{in}$ and $\mathbf{I}_{out}$ are inliers and outliers, respectively. The contamination factor $\gamma \in (0, 1)$ is denoted as $\frac{\#\mathbf{I}_{out}}{\#\mathbf{I}_{in}+\#\mathbf{I}_{out}}$. Remarkably, this setup differs from the semi-supervised outlier detection/one-class learning task [55] by **not** involving a train-test split of the target dataset. Typically, images are transformed to features $\mathbf{X} = \mathbf{X}_{in} \cup \mathbf{X}_{out} = \{x_i\}_{i=1}^{n}$, where $i$ is the feature index and $n$ refers to the size of target dataset.
**Major Objective.** UOD aims to generate an outlier score function $f(\cdot)$ to assess the outlier likelihood of each $x_i$:

$$y(x_i) = \begin{cases} 1, & \text{if } f(x_i) \geq \tau \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where $y = 1$ (outlier) and $y = 0$ (inlier) refer to the predicted labels and $\tau$ is the threshold, i.e., decision boundary. In this study, we not only measure the ranking accuracy of $f(\cdot)$, aligning with the related works [28, 34, 36, 50], as well as the classification accuracy with respect to $\tau$.

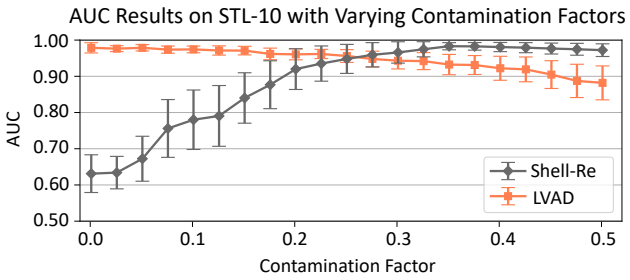


Figure 2. Comparison between two representative UOD methods, Shell-Re (TPAMI-21) [34] and LVAD (ECCV-22) [36], which shows distinct performance on low-$\gamma$ and high-$\gamma$ scenarios.

## 4. Method: FlexUOD

### 4.1. Overview

Currently, the mainstream UOD approaches follow a single (method)-fits-all (scenarios) paradigm, i.e., a single outlier score function is learned to predict outlier scores across different scenarios. However, based on the above analysis, we argue that this paradigm suffers from performance instability when contamination factors vary, as shown in Fig. 2.

Such that, the ideal outlier score function is denoted as:

$$f_{ideal} = \begin{cases} f_{low}\ , & \text{if } \gamma \text{ is low} \\ f_{high}\ , & \text{if } \gamma \text{ is high} \\ \widehat{f}_{low} + \widehat{f}_{high}\ , & \text{otherwise,} \end{cases} \tag{3}$$

where $\widehat{f}.$ refers to the min-max normalized outlier score function. Thus, this study involves three following stages: (**i**) Proposing $f_{low}$; (**ii**) Proposing $f_{high}$; (**iii**) Estimating $\gamma$.

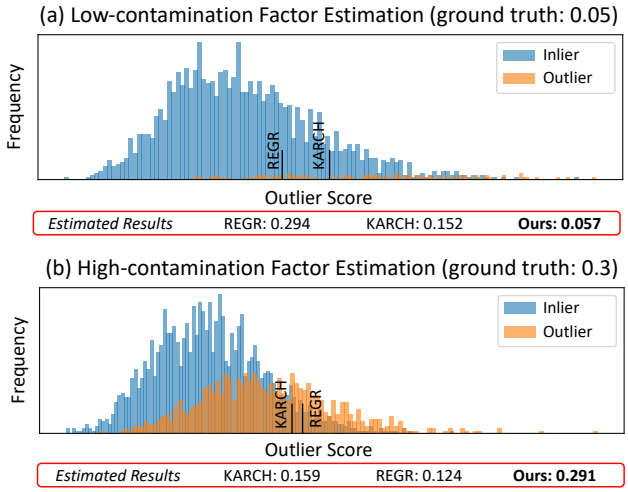


Figure 3. Examples of indiscriminative distributions illustrate the challenges of estimating the contamination factor through threshold learning methods, e.g., KARCH [1] and REGR [2]. In contrast, our proposed estimator maintains the contamination factor estimation efficacy on low-$\gamma$ and high-$\gamma$ scenarios.

**Motivation.** Recently, contamination factor estimators ($\gamma$-estimator) are usually threshold learning-based [40, 44], i.e., determining the decision boundary $\tau$ on a series of given outlier scores $\{f(x_i)\}_{i=1}^{n}$, i.e., $\gamma = |\{x_i | f(x_i) \geq \tau\}|/n$. In this phase, proposing $\{f_{low}, f_{high}\}$ and estimating $\gamma$ are treated as independent problems, where the $\gamma$-estimator is disconnected from the given outlier score function. However, we argue that this leads to an inconsistency problem. First, some $\gamma$-estimators rely on intrinsic assumptions about the inlier distribution [40] (e.g., Gaussian or Chi-squared distributions), which may not hold for all outlier score functions. Moreover, if the outlier score function is not discriminative enough, reliable $\gamma$-estimation becomes difficult, e.g., Fig. 3.

To address the inconsistency problem between outlier score function $f$ and $\gamma$-estimator, we propose FlexUOD, with an outlier score function-**consistent** $\gamma$-estimator. Specifically, we first propose structurally consistent $f_{low}$ and $f_{high}$, allowing us to estimate $\gamma$ by a natural comparison between $f_{low}$ and $f_{high}$. In Fig. 3, our solution can accurately estimate $\gamma$ on both low and high $\gamma$-scenarios.

## 4.2. Proposing $f_{low}$

### 4.2.1. Basic Structure

As the input raw images will be transformed into features, i.e., high-dimensional vectors. We propose $f_{low}$ by understanding the high-dimensional space. Different from the intuitive understanding applicable to 2-D or 3-D spaces, the high-dimensional data typically resides within a hypersphere characterized by its dimension $d$, and radius $r$ [35].

**Property 1.** *When $\gamma$ is low, the radius of the hypersphere becomes a natural discriminative outlier score function.*

For all inliers $\mathbf{X}_{in}$ within a $d$-dimensional hyper-sphere, as its volume is defined as: $V_d(r) = (\pi^{d/2} r^d)/\Gamma(1 + d/2)$, where $\Gamma(\cdot)$ is the Gamma function. If we shrink its radius $r$ with a small value $\epsilon$ ($\epsilon > 0$), we will have:

$$\lim_{d\to\infty} \frac{V_d((1-\epsilon)r)}{V_d(r)} = \lim_{d\to\infty} (1-\epsilon)^d \leq \lim_{d\to\infty} e^{-\epsilon d} = 0. \tag{4}$$

If $\epsilon$ is fixed and $d \to \infty$, the volume inside the hypersphere rapidly tends to zero, meaning that almost all of the volume is concentrated on the surface of the hypersphere, often referred to as a shell [34]. This suggests that the distance from each sample $x_i$ to the centroid of inliers $m_{\mathbf{X}_{in}}$ could serve as a naturally discriminative outlier score. Notably, when $\gamma$ is low, $m_{\mathbf{X}}$ is almost certainly equal to $m_{\mathbf{X}_{in}}$. Thus, the formulation of the basic $f_{low}$ can be denoted as:

$$f_{low\text{-}basic}(x_i) = ||x_i - m_{\mathbf{X}}||_2, \tag{5}$$

where $m_{\mathbf{X}} = [m_{\mathbf{X},1}, \cdots, m_{\mathbf{X},j}, \cdots, m_{\mathbf{X},d}]$ refers to the mean of all features $\mathbf{X}$ and $||\cdot||_2$ is the Euclidean distance ($\ell_2$-norm). Despite the simplicity of Eq. 5, Fig. 4 shows its unexpected efficacy while maintaining efficiency. Next, we will discuss how it can be further advanced.

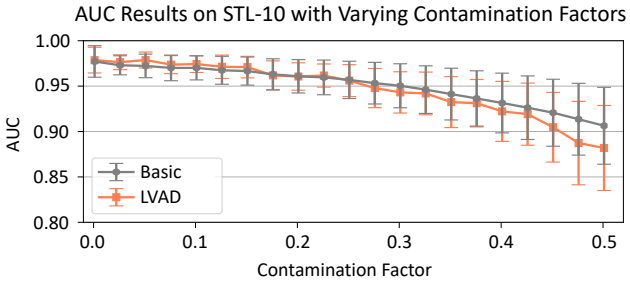


Figure 4. Comparison between our $f_{low\text{-}basic}$ (Eq. 5) and LVAD. Despite its simplicity, it exhibits competitive overall performance.

### 4.2.2. Distance Metric

**Motivation.** For UOD and related tasks, Euclidean distance (Cosine Similarity) is widely used as metrics [28, 34, 37]. However, we argue that these symmetric metrics sometimes lack separability. For instance, consider two data points $x_1$ and $x_2$, where $x_1$ is an inlier and $x_2$ is an outlier.

Even if $||x_1 - x_2||_2$ is relatively large, the points may be symmetrically positioned around the mean $m_{\mathbf{X}}$, such that $||x_1 - m_{\mathbf{X}}||_2 \approx ||x_2 - m_{\mathbf{X}}||_2$. In such cases, Euclidean distance could erroneously assign similar outlier scores to both $x_1$ and $x_2$, leading to a decrease in discriminativeness.

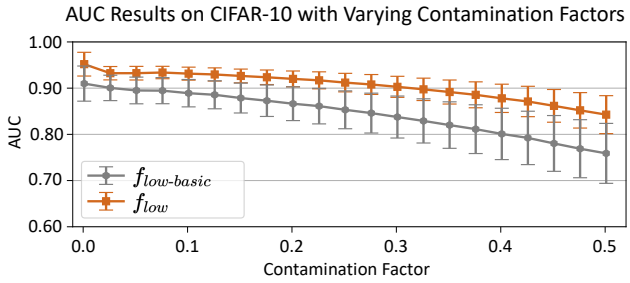


Figure 5. The comparison between $f_{low}$ and $f_{low\text{-}basic}$.

**Bray-Curtis Distance.** To overcome the limitation posed by symmetric metrics, we adopt an asymmetric metric, the Bray-Curtis distance [12] to advance the $f_{low\text{-}basic}$.

$$\text{Bray-Curtis}(x_i) = \frac{||x_i - m_{\mathbf{X}}||_1}{||x_i + m_{\mathbf{X}}||_1}, \tag{6}$$

where $||\cdot||_1$ refers to Manhattan distance ($\ell_1$-norm).
**Therefore**, our outlier score function for low contamination factors $f_{low}(\cdot)$ is defined as follows:

$$\begin{aligned} f_{low}(x_i) = \omega * \text{Bray-Curtis}(x_i) + \\ (1-\omega) * f_{low\text{-}basic}(x_i), \end{aligned} \tag{7}$$

where $\omega = 1/0$, whose decision is based on our analysis for Bray-Curtis distance, as shown in the supplementary materials.

### 4.3. Proposing $f_{high}$

A previous research work for high-dimensional space named Shell Theory [34] proves that the outlier-based normalization $\text{norm}(x_i, \mathbf{X}_{out}) = (x_i - m_{\mathbf{X}_{out}})/||x_i - m_{\mathbf{X}_{out}}||_2$ can significantly improve the discriminativeness of feature representation. Hence, when $\gamma$ is high, obtaining enough outliers becomes relatively easier. Specifically, we initially take $f_{low}$ as a reliable outlier score foundation. Subsequently, we identify potential outlier candidates $\mathbf{X}'_{out}$, using its default outlier filtering method MAD (Median Absolute Deviation) [49], i.e., $\mathbf{X}'_{out} = \{x_i | f_{low}(x_i) > \text{Median}(\{f_{low}(x_i)\}_{i=1}^n) + k^2 \cdot \text{MAD}(\{f_{low}(x_i)\}_{i=1}^n)\}$.
**Therefore**, our outlier score function for high contamination factors $f_{high}(\cdot)$ is denoted as follows:

$$f_{high}(x_i) = f_{low}(\text{norm}(x_i, \mathbf{X}'_{out}), \text{norm}(m, \mathbf{X}'_{out})). \tag{8}$$

[2] We set $k = 1$ to constrain MAD only works on high-$\gamma$ scenarios, thereby ensuring $f_{high}$ only performs well on high-$\gamma$ scenarios.

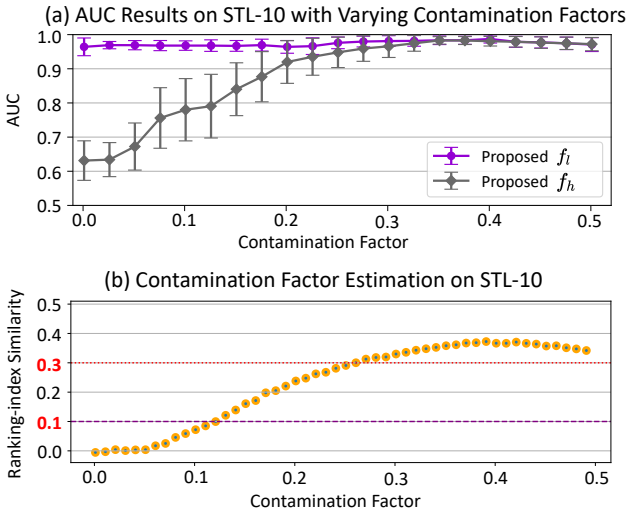


Figure 6. (a) illustrates the distinct performance differences between $f_{low}$ (Eq. 7) and $f_{high}$ (Eq. 8). (b) depicts how the efficacy (AUC score) of $f_{high}$ correlates with the ranking-index similarity between $\{f_{low}(x_i)\}_{i=1}^n$ and $\{f_{high}(x_i)\}_{i=1}^n$.

### 4.4. Contamination Factor Estimation

Inspired by the specific property analysis of $f_{high}$, its ranking instance index of $\{f_{high}(x_i)\}_{i=1}^n$ performs distinctly with low and high contamination factors. In contrast, the above proposed $f_{low}$ (Eq. 7) is more stable and can be considered as a reliable baseline, Such that, we can estimate $\gamma$ in a heuristic way, demonstrated as follows.

**Property 2.** *The contamination factor $\gamma$ can be approximately estimated by measuring the ranking-index similarity between $\{f_{low}(x_i)\}_{i=1}^n$ and $\{f_{high}(x_i)\}_{i=1}^n$.*

When $\gamma$ is low, the ranking accuracy (i.e., AUC score) for $\{f_{low}(x_i)\}_{i=1}^n$ and $\{f_{high}(x_i)\}_{i=1}^n$ are distinct; when $\gamma$ is high, as $f_{high}$ is effective and $\{f_{low}(x_i)\}_{i=1}^n$ maintains remarkable efficacy, the AUC score for $\{f_{low}(x_i)\}_{i=1}^n$ and $\{f_{high}(x_i)\}_{i=1}^n$ are both relatively high, as shown in Fig. 6 (a). Moreover, since $f_{low}$ and $f_{high}$ is structurally consistent, $\gamma$ can be measured by the ranking-index similarity $\phi$ between $\{f_{low}(x_i)\}_{i=1}^n$ and $\{f_{high}(x_i)\}_{i=1}^n$. We use Spearman's rank correlation coefficient [16, 40] to estimate $\phi$.

$$\begin{aligned} \phi = |\texttt{Spearman}(\text{rank}(\{f_{low}(x_i)\}_{i=1}^n), \\ \text{rank}(\{f_{high}(x_i)\}_{i=1}^n)|, \end{aligned} \tag{9}$$

where $\text{rank}(\cdot)$ refers to the operation that transforms outlier scores to an ascending-order ranking index list. As shown in Fig. 6 (b), the ranking-index similarity $\phi$ ($y$-axis) is quasi-**linearly correlated** with the contamination factor $\gamma$ ($x$-axis).

### 4.5. Summary

In our proposed FlexUOD framework (illustrated in Fig. 7), we fix the lower and upper boundary of $\phi$ are 0.1 and 0.3,

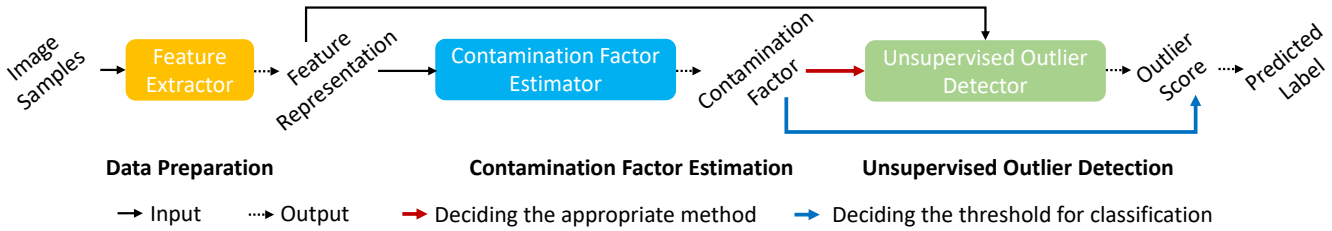


Figure 7. An overview of our framework. It has two stages: contamination factor estimation and appropriate outlier detector selection.

respectively (the ablation study is shown in Tab. 9). For each $x_i$, its outlier score $f_{ours}(x_i)$ is defined as follows:

$$f_{ours}(x_i) = \begin{cases} f_{low}(x_i)\,, & \text{if } \phi < 0.1 \\ f_{high}(x_i)\,, & \text{if } \phi > 0.3 \\ \widehat{f}_{low}(x_i) + \widehat{f}_{high}(x_i)\,, & \text{otherwise.} \end{cases} \tag{10}$$

Moreover, since it is usually insufficient to merely quantify anomalousness for real applications, FlexUOD is also able to determine the threshold $\tau$. Based on the above analysis, we conclude that: $\tau \not\Rightarrow \gamma, \gamma \Rightarrow \tau$, i.e., although threshold learning for contamination factor estimation is an ill-posed problem, we can determine the threshold $\tau_{ours}$ with our estimated approximate contamination factor $|\phi|$: $\tau_{ours} = \text{sort}(\{f_{ours}(x_i)\}_{i=1}^{n})[n \cdot (1 - |\phi|)]$, thereby predicting the labels of each $x_i$:

$$y_{ours}(x_i) = \begin{cases} 1, & \text{if } f_{ours}(x_i) \geq \tau_{ours}; \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Notably, as most of the proposed methods suffer from the efficacy decrease with high outlier ratios while the estimated contamination factor is model-agnostic, FlexUOD can be seamlessly integrated with any low-ratio effective method, i.e., replace $f_{low}$ with the existing methods such as OCSVM [51], LVAD [36] and ICL [52], as evaluated in Fig. 8.

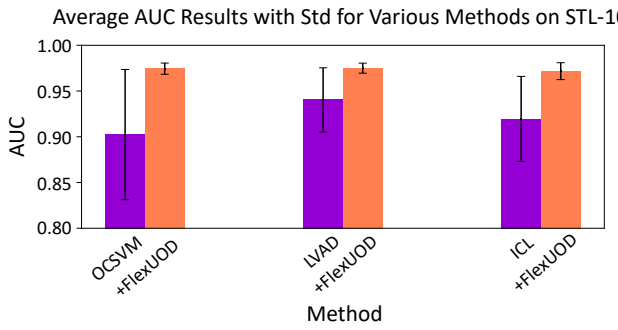


Figure 8. AUC results from integrating FlexUOD with existing methods, including OCSVM [51], LVAD [36], and ICL [52]. Notably, FlexUOD significantly improves their performance.

# 5. Experiments

## 5.1. Settings

**Evaluated Datasets.** As our desired UOD method should be applied to real-world scenarios, the evaluated datasets involve both **classic UOD benchmarks**: STL-10 [13], Internet [34], CIFAR-10 [27], CIFAR-100 [27], MIT-Places-Small [64], MNIST [30], Fashion-MNIST [59] and **UOD applications** (Anomaly Detection, i.e., AD), which involves (i) Industrial defect detection dataset: MVTec-AD [7]; (ii) Medical-AD datasets: HIS, BrainMRI, LiverCT and RESC, sourced from BMAD benchmark [4]; (iii) Tabular-AD datasets sourced from ADBench [19] involves multiple categories such as finance, Web and healthcare.

Building upon related research [34, 36, 40], we leverage the raw pixel features for MNIST and Fashion-MNIST. For other image datasets such as STL-10, CIFAR-10, and MVTec-AD, we adopt ImageNet pre-trained ResNet-50 [21] and CLIP [47]. For medical AD datasets, we adopt ResNet-18 [15]. Besides, for UOD datasets, each class within a certain evaluated benchmark dataset is alternately treated as inliers, and instances from the other classes are considered outliers. The target dataset comprises all inliers and a portion of randomly selected outliers. The reported results for each benchmark dataset are averaged across all classes. For every class, we further average the results over a broad range of outlier ratios: $[0.01, 0.1, 0.2, 0.3, 0.4, 0.5]$. For AD datasets, we combine its default training and test sets within each category together to be the target dataset.

**Baselines.** We first compare with two themes of UOD methods: (i) statistical-based methods: OC-SVM [51], IF [39], Shell-Re [34], ECOD [33], LVAD [36], Multi-T [40]. (ii) deep learning-based methods: INNE [3], Deep SVDD [50], RSRAE [28], LUNAR [28], GOAD [6], NeuTraL [46], ICL [52], SLAD [61] and DIF [60]. Besides, we compare with a series of threshold learning methods involving kernel-based: FGD [45]; curve-based EB [17];filtering-based: HIST [53], FILTER [20]; statistical-based: KARCH [1], REGR [2], QMCD [24], CLF [5], DECOMP [8], META [63]; transformation-based: MOLL [25], CLUST [9], CPD [54].

**Evaluation Metrics.** We evaluate our proposed method on two aspects: ranking accuracy of the generated outlier

Table 1. Average results over a wide range of outlier ratios ($[0.01, \cdots, 0.5]$) with SOTAs on a series of RGB datasets. **Red** and **Blue** indicates the best and second-best results, respectively.

| Feature | Method | Venue | STL-10 [13] | | | Internet [34] | | | CIFAR-10 [27] | | | CIFAR-100 [27] | | | MIT-Places-Small [64] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | AUC | PR-I | PR-O | AUC | PR-I | PR-O | AUC | PR-I | PR-O | AUC | PR-I | PR-O | AUC | PR-I | PR-O |
| ResNet-50 | OC-SVM [51] | Neu. Com.-01 | 0.902 | 0.934 | 0.771 | 0.935 | 0.955 | 0.844 | 0.802 | 0.878 | 0.561 | 0.858 | 0.910 | 0.662 | 0.805 | 0.885 | 0.574 |
| | IF [39] | ICDM-08 | 0.818 | 0.902 | 0.557 | 0.848 | 0.920 | 0.595 | 0.760 | 0.861 | 0.498 | 0.784 | 0.873 | 0.532 | 0.671 | 0.826 | 0.391 |
| | INNE [3] | Com. Int.-18 | 0.846 | 0.922 | 0.553 | 0.862 | 0.935 | 0.581 | 0.787 | 0.885 | 0.471 | 0.867 | 0.928 | 0.629 | 0.761 | 0.873 | 0.450 |
| | Deep SVDD [50] | ICML-18 | 0.601 | 0.777 | 0.378 | 0.695 | 0.821 | 0.484 | 0.539 | 0.762 | 0.293 | 0.567 | 0.771 | 0.356 | 0.584 | 0.777 | 0.344 |
| | RSRAE [28] | ICLR-19 | 0.891 | 0.931 | 0.740 | 0.935 | 0.960 | 0.821 | 0.778 | 0.865 | 0.532 | 0.893 | 0.944 | 0.696 | 0.735 | 0.843 | 0.497 |
| | GOAD [6] | ICLR-20 | 0.942 | 0.967 | 0.819 | **0.973** | **0.986** | **0.888** | 0.835 | 0.904 | 0.603 | 0.897 | 0.941 | 0.709 | 0.833 | 0.905 | 0.608 |
| | RCA [38] | IJCAI-21 | 0.497 | 0.748 | 0.252 | 0.502 | 0.749 | 0.253 | 0.499 | 0.749 | 0.252 | 0.494 | 0.748 | 0.252 | 0.490 | 0.748 | 0.252 |
| | Shell-Re [34] | TPAMI-21 | 0.870 | **0.980** | 0.669 | 0.916 | **0.986** | 0.701 | **0.860** | **0.958** | 0.640 | 0.846 | **0.959** | 0.592 | 0.759 | **0.896** | 0.541 |
| | NeuTraL [46] | ICML-21 | 0.839 | 0.904 | 0.648 | 0.899 | 0.947 | 0.750 | 0.744 | 0.854 | 0.479 | 0.793 | 0.906 | 0.550 | 0.718 | 0.830 | 0.443 |
| | ICL [52] | ICLR-22 | 0.920 | 0.952 | 0.767 | 0.957 | 0.979 | 0.837 | 0.827 | 0.897 | 0.583 | 0.904 | 0.948 | 0.732 | 0.795 | 0.882 | 0.527 |
| | LUNAR [18] | AAAI-22 | 0.759 | 0.854 | 0.494 | 0.770 | 0.872 | 0.505 | 0.748 | 0.847 | 0.481 | 0.825 | 0.881 | 0.608 | 0.655 | 0.792 | 0.378 |
| | ECOD [33] | TKDE-22 | 0.892 | 0.944 | 0.677 | 0.889 | 0.947 | 0.654 | 0.856 | 0.917 | 0.630 | 0.906 | 0.949 | 0.734 | 0.765 | 0.865 | 0.476 |
| | LVAD [36] | ECCV-22 | 0.940 | 0.966 | 0.816 | 0.958 | 0.978 | 0.860 | 0.832 | 0.904 | 0.600 | 0.910 | 0.950 | 0.737 | 0.820 | **0.896** | 0.600 |
| | SLAD [61] | ICML-23 | 0.920 | 0.956 | 0.740 | 0.951 | 0.973 | 0.817 | 0.841 | 0.911 | 0.592 | 0.904 | 0.946 | 0.720 | 0.818 | 0.894 | 0.566 |
| | DeepIF [60] | TKDE-23 | 0.807 | 0.896 | 0.543 | 0.828 | 0.910 | 0.577 | 0.749 | 0.865 | 0.466 | 0.774 | 0.886 | 0.495 | 0.700 | 0.838 | 0.396 |
| | Multi-T [40] | ECCV-24 | **0.948** | 0.971 | **0.820** | 0.971 | 0.985 | 0.869 | 0.859 | 0.920 | **0.643** | **0.920** | 0.956 | **0.762** | **0.821** | **0.896** | **0.609** |
| | FlexUOD (Ours) | – | **0.972** | **0.989** | **0.869** | **0.978** | **0.990** | **0.883** | **0.927** | **0.964** | **0.766** | **0.941** | **0.972** | **0.813** | **0.877** | **0.931** | **0.680** |
| CLIP | OC-SVM [51] | Neu. Com.-01 | 0.896 | 0.925 | 0.780 | 0.870 | 0.920 | 0.648 | 0.846 | 0.903 | 0.630 | 0.832 | 0.893 | 0.634 | 0.842 | 0.901 | 0.557 |
| | IF [39] | ICDM-08 | 0.930 | 0.955 | 0.812 | 0.901 | 0.944 | 0.680 | 0.878 | 0.928 | 0.667 | 0.854 | 0.913 | 0.640 | 0.852 | 0.917 | 0.583 |
| | INNE [3] | Com. Int.-18 | 0.859 | 0.926 | 0.580 | 0.797 | 0.903 | 0.430 | 0.839 | 0.917 | 0.521 | 0.850 | 0.911 | 0.639 | 0.798 | 0.908 | 0.399 |
| | Deep SVDD [50] | ICML-18 | 0.585 | 0.791 | 0.319 | 0.653 | 0.838 | 0.332 | 0.527 | 0.760 | 0.289 | 0.583 | 0.808 | 0.267 | 0.495 | 0.765 | 0.291 |
| | RSRAE [28] | ICLR-19 | 0.915 | 0.947 | 0.795 | 0.897 | 0.939 | 0.679 | 0.855 | 0.914 | 0.641 | 0.885 | 0.931 | 0.707 | 0.821 | 0.898 | 0.493 |
| | GOAD [6] | ICLR-20 | 0.949 | 0.968 | 0.856 | 0.918 | 0.954 | 0.718 | 0.890 | 0.936 | 0.695 | 0.876 | 0.926 | 0.687 | 0.909 | 0.948 | 0.673 |
| | RCA [38] | IJCAI-21 | 0.957 | 0.974 | 0.866 | 0.932 | 0.964 | 0.746 | 0.893 | 0.937 | 0.702 | 0.876 | 0.922 | 0.695 | 0.914 | 0.951 | 0.681 |
| | Shell-Re [34] | TPAMI-21 | 0.861 | 0.981 | 0.648 | 0.876 | **0.986** | 0.670 | 0.874 | **0.969** | 0.605 | 0.821 | 0.952 | 0.552 | 0.885 | **0.965** | 0.666 |
| | NeuTraL [46] | ICML-21 | 0.818 | 0.870 | 0.590 | 0.854 | 0.926 | 0.557 | 0.751 | 0.846 | 0.474 | 0.837 | 0.905 | 0.605 | 0.705 | 0.824 | 0.431 |
| | ICL [52] | ICLR-22 | 0.938 | 0.964 | 0.807 | 0.917 | 0.957 | 0.703 | 0.890 | 0.940 | 0.659 | **0.918** | **0.954** | 0.758 | 0.869 | 0.931 | 0.565 |
| | LUNAR [18] | AAAI-22 | 0.802 | 0.861 | 0.643 | 0.781 | 0.861 | 0.547 | 0.785 | 0.863 | 0.532 | 0.894 | 0.927 | **0.772** | 0.741 | 0.841 | 0.437 |
| | ECOD [33] | TKDE-22 | **0.972** | **0.982** | **0.912** | **0.961** | 0.977 | **0.879** | **0.921** | 0.951 | **0.768** | 0.905 | 0.939 | 0.758 | **0.927** | 0.960 | **0.724** |
| | LVAD [36] | ECCV-22 | 0.956 | 0.974 | 0.868 | 0.889 | 0.933 | 0.678 | 0.896 | 0.938 | 0.712 | 0.903 | 0.940 | 0.753 | 0.899 | 0.946 | 0.629 |
| | SLAD [61] | ICML-23 | 0.925 | 0.950 | 0.808 | 0.888 | 0.936 | 0.641 | 0.882 | 0.930 | 0.674 | 0.895 | 0.938 | 0.729 | 0.875 | 0.935 | 0.570 |
| | DeepIF [60] | TKDE-23 | 0.910 | 0.946 | 0.751 | 0.903 | 0.946 | 0.701 | 0.853 | 0.915 | 0.621 | 0.842 | 0.911 | 0.616 | 0.830 | 0.911 | 0.528 |
| | Multi-T [40] | ECCV-24 | 0.966 | 0.980 | 0.883 | 0.941 | 0.970 | 0.760 | 0.917 | 0.952 | 0.741 | 0.913 | 0.951 | 0.741 | 0.897 | 0.945 | 0.661 |
| | FlexUOD (Ours) | – | **0.988** | **0.995** | **0.928** | **0.985** | **0.994** | **0.916** | **0.957** | **0.981** | **0.808** | **0.954** | **0.980** | **0.827** | **0.979** | **0.991** | **0.862** |
| Average (ResNet-50, CLIP) | OC-SVM [51] | Neu. Com.-01 | 0.899 | 0.930 | 0.776 | 0.903 | 0.937 | 0.746 | 0.824 | 0.891 | 0.595 | 0.845 | 0.901 | 0.648 | 0.824 | 0.893 | 0.565 |
| | IF [39] | ICDM-08 | 0.874 | 0.928 | 0.685 | 0.874 | 0.932 | 0.638 | 0.819 | 0.895 | 0.583 | 0.819 | 0.893 | 0.586 | 0.762 | 0.872 | 0.487 |
| | INNE [3] | Com. Int.-18 | 0.852 | 0.924 | 0.567 | 0.829 | 0.919 | 0.505 | 0.813 | 0.901 | 0.496 | 0.859 | 0.919 | 0.634 | 0.780 | 0.890 | 0.425 |
| | Deep SVDD [50] | ICML-18 | 0.593 | 0.784 | 0.349 | 0.674 | 0.829 | 0.408 | 0.533 | 0.761 | 0.291 | 0.575 | 0.790 | 0.311 | 0.539 | 0.771 | 0.317 |
| | RSRAE [28] | ICLR-19 | 0.903 | 0.939 | 0.767 | 0.916 | 0.950 | 0.750 | 0.816 | 0.890 | 0.587 | 0.889 | 0.938 | 0.702 | 0.778 | 0.870 | 0.495 |
| | GOAD [6] | ICLR-20 | 0.946 | 0.968 | 0.838 | 0.945 | 0.970 | 0.803 | 0.862 | 0.920 | 0.649 | 0.886 | 0.934 | 0.698 | **0.871** | 0.926 | **0.640** |
| | RCA [38] | IJCAI-21 | 0.727 | 0.861 | 0.559 | 0.717 | 0.857 | 0.499 | 0.696 | 0.843 | 0.477 | 0.685 | 0.835 | 0.473 | 0.702 | 0.850 | 0.466 |
| | Shell-Re [34] | TPAMI-21 | 0.866 | **0.981** | 0.659 | 0.896 | **0.986** | 0.686 | 0.867 | **0.964** | 0.623 | 0.834 | **0.956** | 0.572 | 0.822 | **0.931** | 0.604 |
| | NeuTraL [46] | ICML-21 | 0.828 | 0.887 | 0.619 | 0.877 | 0.937 | 0.653 | 0.748 | 0.850 | 0.476 | 0.815 | 0.905 | 0.577 | 0.711 | 0.827 | 0.437 |
| | ICL [52] | ICLR-22 | 0.929 | 0.958 | 0.787 | 0.937 | 0.968 | 0.770 | 0.859 | 0.918 | 0.621 | 0.911 | 0.951 | 0.745 | 0.832 | 0.906 | 0.546 |
| | LUNAR [18] | AAAI-22 | 0.781 | 0.857 | 0.568 | 0.776 | 0.867 | 0.526 | 0.766 | 0.855 | 0.507 | 0.859 | 0.904 | 0.690 | 0.698 | 0.817 | 0.408 |
| | ECOD [33] | TKDE-22 | 0.932 | 0.963 | 0.795 | 0.925 | 0.962 | 0.766 | **0.888** | 0.934 | **0.699** | 0.905 | 0.944 | **0.746** | 0.846 | 0.912 | 0.600 |
| | LVAD [36] | ECCV-22 | 0.948 | 0.970 | 0.842 | 0.923 | 0.956 | 0.769 | 0.864 | 0.921 | 0.656 | 0.907 | 0.945 | 0.745 | 0.860 | 0.921 | 0.615 |
| | Multi-T [40] | ECCV-24 | **0.957** | 0.976 | **0.851** | **0.956** | 0.978 | **0.814** | **0.888** | 0.936 | 0.692 | **0.917** | 0.954 | **0.751** | 0.859 | 0.920 | 0.635 |
| | FlexUAD | – | **0.980** | **0.992** | **0.899** | **0.981** | **0.992** | **0.900** | **0.942** | **0.973** | **0.787** | **0.947** | **0.976** | **0.820** | **0.928** | **0.961** | **0.771** |

scores and classification accuracy of the predicted threshold. Specifically, the ranking accuracy is primarily assessed using the Area Under the Receiver Operating Characteristic Curve (AUC). This metric offers a comprehensive evaluation of ranking accuracy, interpreted as the likelihood of an outlier receiving a higher outlier score [14]. We also employ AUPR-In (PR-I) and AUPR-Out (PR-O), which indicate the nuanced ranking accuracy for inliers and outliers, providing a more detailed assessment. Additionally, the performance of the predicted threshold is measured with $F_1$-score, a harmonic mean of the precision and recall.

## 5.2. Main Results

**Evaluation on Benchmark Datasets.** Tab. 1 shows the substantial improvements in both efficacy and stability achieved by our proposed FlexUOD framework across a diverse range of contamination factors and various benchmark datasets. Our FlexUOD method attains SOTA results in almost all experiments. Notably, even with non-aligned, low-resolution datasets such as CIFAR-10 and CIFAR-100, which are known to present challenges [43], our method surpasses the current SOTA AUC scores by margins of 6% and 3%, respectively. Furthermore, we observe a significant enhancement in performance with improved feature representation. For example, for the MIT-Places dataset, the AUC score improves from 0.877 using the ResNet-50 feature to 0.979 with CLIP. Tab. 2 underscores the promising performance observed on the MNIST and Fashion-MNIST datasets. This is particularly noteworthy considering that our framework operates solely on raw pixel features, em-

Table 2. Average results on Gray-scale datasets.

| Method | MNIST [30] | | | Fashion-MNIST [59] | | |
|---|---|---|---|---|---|---|
| | AUC | PR-I | PR-O | AUC | PR-I | PR-O |
| OCSVM | 0.815 | 0.880 | 0.564 | 0.828 | 0.886 | 0.636 |
| IF | 0.769 | 0.889 | 0.425 | 0.853 | 0.921 | 0.606 |
| INNE | 0.872 | 0.930 | 0.621 | 0.796 | 0.894 | 0.456 |
| D-SVDD | 0.561 | 0.770 | 0.299 | 0.568 | 0.777 | 0.313 |
| RSRAE | 0.871 | 0.917 | 0.667 | 0.781 | 0.889 | 0.523 |
| GOAD | 0.866 | 0.927 | 0.614 | 0.871 | 0.925 | **0.682** |
| Shell-Re | 0.709 | 0.925 | 0.432 | 0.769 | 0.921 | 0.477 |
| NeuTraL | 0.766 | 0.849 | 0.504 | 0.787 | 0.874 | 0.526 |
| ICL | 0.865 | 0.918 | 0.662 | 0.812 | 0.888 | 0.603 |
| LUNAR | 0.761 | 0.815 | 0.581 | 0.719 | 0.827 | 0.451 |
| ECOD | 0.722 | 0.841 | 0.436 | 0.801 | 0.896 | 0.536 |
| LVAD | **0.898** | **0.944** | **0.670** | 0.849 | 0.916 | 0.646 |
| SLAD | 0.845 | 0.909 | 0.592 | 0.839 | 0.909 | 0.604 |
| DeepIF | 0.811 | 0.898 | 0.527 | 0.806 | 0.888 | 0.613 |
| Multi-T | 0.868 | 0.932 | 0.609 | **0.874** | **0.931** | 0.663 |
| FlexUOD | **0.906** | **0.959** | **0.667** | **0.873** | **0.931** | **0.682** |

ploying a simpler approach compared to deep learning solutions, e.g., GOAD [6].

Table 3. AUC results on MVTec-AD dataset. The best results are highlighted in bold.

| Type | Category | ResNet-50 | | | CLIP | | |
|---|---|---|---|---|---|---|---|
| | | LVAD | Shell-Re | FlexUOD | LVAD | Shell-Re | FlexUOD |
| Textures | Carpet | 0.872 | 0.895 | **0.983** | 0.877 | 0.964 | **0.983** |
| | Grid | **0.516** | 0.497 | 0.442 | 0.662 | 0.587 | **0.824** |
| | Leather | 0.997 | 0.997 | **1.000** | **1.000** | 0.998 | **1.000** |
| | Tile | 0.977 | 0.984 | **0.987** | 0.940 | 0.925 | **0.990** |
| | Wood | 0.972 | 0.616 | **0.984** | 0.976 | 0.794 | **0.991** |
| Objects | Bottle | 0.941 | 0.909 | **0.972** | 0.816 | 0.801 | **0.992** |
| | Cable | **0.795** | 0.599 | 0.792 | 0.659 | 0.362 | **0.733** |
| | Capsule | **0.679** | 0.583 | 0.628 | 0.546 | 0.647 | **0.656** |
| | Hazelnut | **0.829** | 0.628 | 0.693 | 0.797 | 0.694 | **0.846** |
| | Metal Nut | **0.798** | 0.585 | 0.717 | 0.703 | 0.677 | **0.836** |
| | Pill | 0.687 | 0.613 | **0.688** | **0.681** | 0.646 | 0.676 |
| | Screw | **0.578** | 0.510 | 0.516 | 0.488 | **0.599** | 0.583 |
| | Toothbrush | 0.699 | 0.609 | **0.891** | 0.714 | 0.663 | **0.776** |
| | Transistor | 0.761 | 0.729 | **0.843** | **0.735** | 0.644 | 0.711 |
| | Zipper | 0.889 | 0.859 | **0.955** | 0.748 | 0.840 | **0.909** |
| | Average | 0.799 | 0.707 | **0.806** | 0.756 | 0.723 | **0.834** |
| | P75 (75th percentile) | 0.915 | 0.877 | **0.977** | 0.846 | 0.820 | **0.987** |

Table 4. AUC results on medical-AD datasets.

| Method | HIS | BrainMRI | LiverCT | RESC |
|---|---|---|---|---|
| Shell-Re | 0.443 | 0.671 | **0.732** | 0.818 |
| LVAD | 0.663 | 0.632 | 0.640 | 0.787 |
| FlexUOD | **0.708** | **0.746** | 0.711 | **0.948** |

Table 5. AUC results on tabular-AD datasets.

| Method | fraud | http | Lymphography | pendigits | shuttle | WBC | WDBC | WINE |
|---|---|---|---|---|---|---|---|---|
| Shell-Re | 0.607 | 0.568 | 0.415 | 0.541 | 0.803 | **0.975** | 0.400 | 0.853 |
| LVAD | 0.954 | **0.994** | 0.981 | 0.882 | 0.879 | 0.837 | **0.954** | 0.876 |
| FlexUOD | **0.955** | **0.994** | **0.985** | **0.919** | **0.987** | 0.914 | 0.943 | **0.921** |

**Evaluation for Anomaly Detection.** Industrial defect detection datasets (e.g., MVTec-AD) are usually patch/pixel-level contaminated instead of semantic-level, i.e., outlier detection. To maintain the evaluation consistency, we still use the default feature and FlexUOD statistically outperforms the LVAD and Shell-Re, as shown in Tab. 3. As defect detection is previously utilized for semi-supervised/few-shot scenarios [23, 31, 48], the unsupervised method is usually not the best solution, especially for some challenging categories. However, we still obtain high efficacy on "Carpet" (AUC: 0.983), "Leather" (1.000), "Tile" (0.990), "Wood" (0.991) and "bottle" (0.992). Additionally, Tab. 4 and 5 show the superiority of FlexUOD that can be generalized to medical images and even some non-image, i.e., tabular datasets.

Table 6. $F_1$-score comparison with threshold learning methods.

| Method | STL-10 | Internet | CIFAR-10 | CIFAR-100 |
|---|---|---|---|---|
| CLF [5] | 0.638 | 0.624 | 0.479 | 0.514 |
| CPD [54] | 0.613 | 0.618 | 0.560 | 0.602 |
| DECOMP [8] | 0.607 | 0.615 | 0.553 | 0.591 |
| EB [17] | 0.504 | 0.505 | 0.470 | 0.500 |
| HIST [53] | 0.560 | 0.572 | 0.582 | 0.586 |
| FILTER [20] | 0.611 | 0.538 | 0.599 | 0.583 |
| FGD [45] | 0.652 | 0.659 | 0.519 | 0.522 |
| MOLL [25] | 0.423 | 0.419 | 0.410 | 0.434 |
| META [63] | 0.617 | 0.515 | 0.551 | 0.508 |
| KARCH [1] | **0.660** | **0.671** | 0.585 | 0.611 |
| QMCD [24] | 0.136 | 0.129 | 0.181 | 0.166 |
| REGR [2] | 0.583 | 0.628 | **0.601** | **0.624** |
| FlexUOD | **0.687** | **0.683** | **0.645** | **0.653** |

**Evaluation for Threshold Learning.** Traditional UOD is usually considered an outlier score ranking task, and its main objective is to have a discriminative outlier score function. Nevertheless, without the threshold (decision boundary), the real applications of UOD methods will be largely limited. Interestingly, although FlexUOD does not involve an explicit threshold learning procedure, our estimated contamination factor can be an effective threshold, as evaluated in Tab. 6.

Table 7. Timing is measured with ResNet-50 features. CPU refers to the Apple M2 Pro chip while GPU refers to one NVIDIA A100.

| Method | Device | # Features | |
|---|---|---|---|
| | | 1,000 (sec.) | 10,000 (sec.) |
| GOAD | GPU | 14.06 | 157.1 |
| ICL | GPU | 18.48 | 180.4 |
| SLAD | GPU | 15.75 | 108.5 |
| OCSVM | CPU | 1.293 | 20.99 |
| ECOD | CPU | 1.283 | 4.747 |
| LVAD | CPU | 8.057 | 101.9 |
| Multi-T | CPU | 0.167 | 0.472 |
| FlexUOD | CPU | **0.118** | **0.404** |

**Discussions.** To be deemed a desirable real-world unsupervised image outlier detector, the method should demonstrate both effectiveness/stability and efficiency across different benchmark datasets (as evaluated in Tab. 1, 2, 3), various feature representations (as evaluated in Tab. 1, 2) and a wide range of contamination factors (as evaluated in Tab. 1, 2). Notably, many existing methods exhibit domain-specific (e.g., RSRAE), feature-specific (e.g., IF, ECOD) or contamination factor-specific (e.g., OCSVM, Shell-Re, LVAD, ICL, etc.) characteristics, while FlexUOD appears

Figure 9. The outlier score progressively decreases from left to right and from top to bottom. Notably, "unusual" airplane images, such as those depicting passengers, cockpits, and windows, register higher FlexUOD outlier scores than typical airplane images.

to strike a good balance across diverse performance scenarios. Despite lacking widely used feature refinement procedures such as self-supervised learning, FlexUOD consistently outperforms existing methods. In Tab. 7, our solution not only excels in accuracy but also speed. More importantly, Fig. 8 demonstrates that our framework can significantly enhance the stability of existing methods.

Table 8. Ablation study for our FlexUOD framework.

| Dataset | Baseline (Eq. 5) | +Bray-Curtis (Eq. 7) | FlexUOD (Eq. 10) |
|---|---|---|---|
| STL-10 | 0.950 | 0.963 (+**1.4**%) | 0.980 (+**3.2**%) |
| CIFAR-10 | 0.870 | 0.919 (+**5.6**%) | 0.942 (+**8.3**%) |
| CIFAR-100 | 0.891 | 0.933 (+**4.7**%) | 0.947 (+**6.3**%) |
| MNIST | 0.861 | 0.877 (+**1.9**%) | 0.906 (+**5.2**%) |

### 5.3. Ablation Study

As there remain some concerns regarding the extent to which components and parameters contributed to the FlexUOD framework, we conduct ablation studies on four benchmark datasets, and AUC results are averaged on ResNet-50 and CLIP features excluding the MNIST dataset. Compared with the basic approach (Eq. 5), our proposed two stages (Sec. 4.2.2 and 4.3) improve the performance by 3.2%-8.3%, as shown in Tab. 8. Besides, Tab. 9 shows our default $\phi$ values (Eq. 9) are both effective and stable.

### 5.4. Qualitative Result

The outliers identified in our numerical evaluations represent instances that deviate from the inlier/normal distribution. Equally crucial is our method's capability to address in-distribution outliers, such as passengers or cockpit images within the *airplane* class. Fig. 9 shows the ranking of internet-crawled airplane images, sorted by their outlier scores. These findings intuitively align with expectations and underscore the potential value of our proposed unsupervised outlier detector in tasks like data cleaning and organization.

Table 9. Average AUC results are reported with different $\phi$-pairs (Eq. 9) of FlexUOD. $\{0.1, 0.3\}$ is the default parameter, which shows an overall stability.

| Dataset | $\{0.1, 0.3\}$ | $\{0.1, 0.2\}$ | $\{0.1, 0.4\}$ | $\{0.05, 0.1\}$ | $\{0.05, 0.2\}$ |
|---|---|---|---|---|---|
| STL-10 | 0.980 | 0.979 | 0.980 | 0.975 | 0.978 |
| CIFAR-10 | 0.942 | 0.939 | 0.946 | 0.934 | 0.939 |
| CIFAR-100 | 0.947 | 0.942 | 0.951 | 0.933 | 0.940 |
| MNIST | 0.906 | 0.899 | 0.899 | 0.887 | 0.895 |
| Dataset | $\{0.05, 0.3\}$ | $\{0.05, 0.4\}$ | $\{0.2, 0.3\}$ | $\{0.2, 0.4\}$ | $\{0.3, 0.4\}$ |
| STL-10 | 0.980 | 0.980 | 0.979 | 0.979 | 0.977 |
| CIFAR-10 | 0.943 | 0.945 | 0.941 | 0.944 | 0.935 |
| CIFAR-100 | 0.945 | 0.948 | 0.948 | 0.951 | 0.948 |
| MNIST | 0.898 | 0.896 | 0.900 | 0.898 | 0.897 |

## 6. Limitation

As FlexUOD is built upon the analysis of entire image data, it might become less effective on some anomaly segmentation/localization tasks [11], which is our future direction.

## 7. Conclusion

In this paper, we address the instability issue in UOD tasks by introducing the FlexUOD framework. It is designed to be aware of the contamination factor, enabling it to automatically select the appropriate method based on the estimated contamination factor. Our evaluation demonstrates that FlexUOD offers high efficacy and stability across diverse experimental setups. Additionally, FlexUOD is distinguished by its training-free approach and lightweight architecture, which facilitate easy integration with existing methods and practical deployment in real-world visual systems.

## Acknowledgements

This work is partially supported by the Shenzhen Science and Technology Program KQTD20210811090149095 and also the Pearl River Talent Recruitment Program 2019QN01X226. Besides, we would like to thank the reviewers for their valuable comments and suggestions.

## References


[1] Bijan Afsari. Riemannian $l_p$ center of mass: existence, uniqueness, and convexity. *PAMS*, pages 655–673, 2011. 2, 3, 5, 7

[2] Charu C Aggarwal and Charu C Aggarwal. *An introduction to outlier analysis*. 2017. 2, 3, 5, 7

[3] Tharindu R Bandaragoda, Kai Ming Ting, David Albrecht, Fei Tony Liu, Ye Zhu, and Jonathan R Wells. Isolation-based anomaly detection using nearest-neighbor ensembles. *Comput. Intell.*, pages 968–998, 2018. 5, 6

[4] Jinan Bao, Hanshi Sun, Hanqiu Deng, Yinsheng He, Zhaoxiang Zhang, and Xingyu Li. Bmad: Benchmarks for medical anomaly detection. In *CVPR*, pages 4042–4053, 2024. 5

[5] Alberto Barbado, Óscar Corcho, and Richard Benjamins. Rule extraction in unsupervised anomaly detection for model explainability: Application to oneclass svm. *Epert Syst. Appl.*, page 116100, 2022. 2, 5, 7

[6] Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. *arXiv preprint arXiv:2005.02359*, 2020. 5, 6, 7

[7] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *CVPR*, pages 9592–9600, 2019. 5

[8] Graciela Boente, Ana M Pires, and Isabel M Rodrigues. Influence functions and outlier detection under the common principal components model: A robust approach. *Biometrika*, pages 861–875, 2002. 2, 5, 7

[9] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *SIGMOD*, pages 93–104, 2000. 2, 5

[10] Federico Camposeco, Torsten Sattler, Andrea Cohen, Andreas Geiger, and Marc Pollefeys. Toroidal constraints for two-point localization under high outlier ratios. In *CVPR*, pages 4545–4553, 2017. 2

[11] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, and Matthias Rottmann. Segmentmeifyoucan: A benchmark for anomaly segmentation. *arXiv preprint arXiv:2104.14812*, 2021. 8

[12] K Robert Clarke, Paul J Somerfield, and M Gee Chapman. On resemblance measures for ecological studies, including taxonomic dissimilarities and a zero-adjusted bray–curtis coefficient for denuded assemblages. *J. Exp. Mar. Biol. Ecol.*, pages 55–80, 2006. 4

[13] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, pages 215–223, 2011. 5, 6

[14] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *ICML*, pages 233–240, 2006. 6

[15] Tharindu Fernando, Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Deep learning for medical anomaly detection–a survey. *CSUR*, 2021. 5

[16] Edgar C Fieller and Egon S Pearson. Tests for rank correlation coefficients: Ii. *Biometrika*, pages 29–40, 1961. 4

[17] Michael Friendly, Georges Monette, and John Fox. Elliptical insights: understanding statistical methods through elliptical geometry. *STAT SCI*, pages 1–39, 2013. 2, 5, 7

[18] Adam Goodge, Bryan Hooi, See-Kiong Ng, and Wee Siong Ng. Lunar: Unifying local outlier detection methods via graph neural networks. In *AAAI*, pages 6737–6745, 2022. 6

[19] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. *NeurIPS*, pages 32142–32159, 2022. 5

[20] Navid Hashemi, Eduardo Verdugo German, Jonatan Pena Ramirez, and Justin Ruths. Filtering approaches for dealing with noise in anomaly detection. In *CDC*, pages 5356–5361, 2019. 2, 5, 7

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[22] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018. 2

[23] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. Registration based few-shot anomaly detection. In *ECCV*, pages 303–319, 2022. 7

[24] Dmitri Iouchtchenko, Neil Raymond, Pierre-Nicholas Roy, and Marcel Nooijen. Deterministic and quasi-random sampling of optimized gaussian mixture distributions for vibronic monte carlo. *arXiv preprint arXiv:1912.11594*, 2019. 2, 5, 7

[25] Michiel A Keyzer and BGJS Sonneveld. Using the mollifier method to characterize datasets and models: the case of the universal soil loss equation. *ITC Journal*, pages 263–272, 1997. 2, 5, 7

[26] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. Angle-based outlier detection in high-dimensional data. In *KDD*, pages 444–452, 2008. 2

[27] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5, 6

[28] Chieh-Hsin Lai, Dongmian Zou, and Gilad Lerman. Robust subspace recovery layer for unsupervised anomaly detection. *arXiv preprint arXiv:1904.00152*, 2019. 2, 3, 5, 6

[29] Stefan Larson, Anish Mahendran, Andrew Lee, Jonathan K Kummerfeld, Parker Hill, Michael A Laurenzano, Johann Hauswald, Lingjia Tang, and Jason Mars. Outlier detection for improved data quality and diversity in dialog systems. *arXiv preprint arXiv:1904.03122*, 2019. 1

[30] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. 5, 7

[31] Hanxi Li, Jianfei Hu, Bo Li, Hao Chen, Yongbin Zheng, and Chunhua Shen. Target before shooting: Accurate anomaly detection and localization under one millisecond via cascade patch retrieval. *arXiv preprint arXiv:2308.06748*, 2023. 7

[32] Tangqing Li, Zheng Wang, Siying Liu, and Wen-Yan Lin. Deep unsupervised anomaly detection. In *WACV*, pages 3636–3645, 2021. 2

[33] Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George H Chen. Ecod: Unsupervised outlier

detection using empirical cumulative distribution functions. *arXiv preprint arXiv:2201.00382*, 2022. 5, 6

[34] Daniel Lin, Siying Liu, Hongdong Li, Ngai-Man Cheung, Changhao Ren, and Yasuyuki Matsushita. Shell theory: A statistical model of reality. *TPAMI*, 2021. 1, 2, 3, 4, 5, 6

[35] Wen-Yan Lin, Siying Liu, Jian-Huang Lai, and Yasuyuki Matsushita. Dimensionality's blessing: Clustering images by underlying distribution. In *CVPR*, pages 5784–5793, 2018. 3

[36] Wen-Yan Lin, Zhonghang Liu, and Siying Liu. Locally varying distance transform for unsupervised visual anomaly detection. In *ECCV*, pages 354–371, 2022. 1, 2, 5, 6

[37] Wen-Yan Lin, Siying Liu, Bing Tian Dai, and Hongdong Li. Distance based image classification: A solution to generative classification's conundrum? *IJCV*, pages 177–198, 2023. 3

[38] Boyang Liu, Ding Wang, Kaixiang Lin, Pang-Ning Tan, and Jiayu Zhou. Rca: A deep collaborative autoencoder approach for anomaly detection. In *IJCAI*, page 1505, 2021. 6

[39] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *ICDM*, pages 413–422, 2008. 5, 6

[40] Zhonghang Liu, Panzhong Lu, Guoyang Xie, Zhichao Lu, and Wen-Yan Lin. Rethinking unsupervised outlier detection via multiple thresholding. In *European Conference on Computer Vision*, pages 258–275, 2025. 3, 4, 5, 6

[41] Philipp Liznerski, Lukas Ruff, Robert A Vandermeulen, Billy Joe Franks, Klaus-Robert Müller, and Marius Kloft. Exposing outlier exposure: What can be learned from few, one, and zero outlier images. *arXiv preprint arXiv:2205.11474*, 2022. 2

[42] Michael A Martin and Steven Roberts. An evaluation of bootstrap methods for outlier detection in least squares regression. *J. Appl. Stat*, pages 703–720, 2006. 2

[43] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *CVPR*, pages 2898–2906, 2019. 6

[44] Lorenzo Perini, Paul-Christian Bürkner, and Arto Klami. Estimating the contamination factor's distribution in unsupervised anomaly detection. In *ICML*, pages 27668–27679, 2023. 1, 3

[45] Zhuang Qi, Dazhi Jiang, and Xiaming Chen. Iterative gradient descent for outlier detection. *IJWMIP*, page 2150004, 2021. 1, 2, 5, 7

[46] Chen Qiu, Timo Pfrommer, Marius Kloft, Stephan Mandt, and Maja Rudolph. Neural transformation learning for deep anomaly detection beyond images. In *ICML*, pages 8703–8714, 2021. 5, 6

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 5

[48] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *CVPR*, pages 14318–14328, 2022. 7

[49] Peter J Rousseeuw and Christophe Croux. Alternatives to the median absolute deviation. *JASA*, pages 1273–1283, 1993. 4

[50] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *ICML*, pages 4393–4402, 2018. 2, 5, 6

[51] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, pages 1443–1471, 2001. 1, 5, 6

[52] Tom Shenkar and Lior Wolf. Anomaly detection for tabular data with internal contrastive learning. In *ICLR*, 2021. 5, 6

[53] KK Thanammal, JS Jayasudha, RR Vijayalakshmi, and S Arumugaperumal. Effective histogram thresholding techniques for natural images using segmentation. *JOIG*, pages 113–116, 2014. 2, 5, 7

[54] Gerrit JJ Van den Burg and Christopher KI Williams. An evaluation of change point detection algorithms. *arXiv preprint arXiv:2003.06222*, 2020. 2, 5, 7

[55] Miryam Elizabeth Villa-Pérez, Miguel A Alvarez-Carmona, Octavio Loyola-González, Miguel Angel Medina-Pérez, Juan Carlos Velazco-Rossell, and Kim-Kwang Raymond Choo. Semi-supervised anomaly detection algorithms: A comparative summary and future research directions. *KBS*, page 106878, 2021. 2

[56] Gaoang Wang, Yibing Zhan, Xinchao Wang, Mingli Song, and Klara Nahrstedt. Hierarchical semi-supervised contrastive learning for contamination-resistant anomaly detection. In *ECCV*, pages 110–128, 2022. 1

[57] Siqi Wang, Yijie Zeng, Xinwang Liu, En Zhu, Jianping Yin, Chuanfu Xu, and Marius Kloft. Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network. *NIPS*, 2019. 2

[58] Siqi Wang, En Zhu, Xiping Hu, Xinwang Liu, Qiang Liu, Jianping Yin, and Fei Wang. Robustness can be cheap: A highly efficient approach to discover outliers under high outlier ratios. In *AAAI*, pages 5313–5320, 2019. 2

[59] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking mach learn algorithms. 2017. 5, 7

[60] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *TKDE*, 2023. 5, 6

[61] Hongzuo Xu, Yijie Wang, Juhui Wei, Songlei Jian, Yizhou Li, and Ning Liu. Fascinating supervisory signals and where to find them: Deep anomaly detection with scale learning. In *ICML*, pages 38655–38673, 2023. 5, 6

[62] Mohamad Zamini and Seyed Mohammad Hossein Hasheminejad. A comprehensive survey of anomaly detection in banking, wireless sensor networks, social networks, and healthcare. *Intelligent Decision Technologies*, pages 229–270, 2019. 1

[63] Yue Zhao, Ryan A Rossi, and Leman Akoglu. Automating outlier detection via meta-learning. *arXiv preprint arXiv:2009.10606*, 2020. 2, 5, 7

[64] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, pages 1452–1464, 2017. 5, 6

[65] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *KDD*, pages 665–674, 2017. 1, 2

[66] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *ICLR*, 2018. 2