E^3Outlier: a Self-Supervised Framework for Unsupervised Deep Outlier Detection

Siqi Wang, Yijie Zeng, Guang Yu, Zhen Cheng, Xinwang Liu, *Senior Member, IEEE*, Sihang Zhou, En Zhu, Marius Kloft, Jianping Yin, and Qing Liao, *Member, IEEE*

Abstract—Existing unsupervised outlier detection (OD) solutions face a grave challenge with surging visual data like images. Although deep neural networks (DNNs) prove successful for visual data, deep OD remains difficult due to OD's unsupervised nature. This paper proposes a novel framework named *E^3Outlier* that can perform **e**ffective and **e**nd-to-**e**nd deep outlier removal. Its core idea is to introduce *self-supervision* into deep OD. Specifically, our major solution is to adopt a discriminative learning paradigm that creates multiple pseudo classes from given unlabeled data by various data operations, which enables us to apply prevalent discriminative DNNs (e.g., ResNet) to the unsupervised OD problem. Then, with theoretical and empirical demonstration, we argue that inlier priority, a property that encourages DNN to prioritize inliers during self-supervised learning, makes it possible to perform end-to-end OD. Meanwhile, unlike frequently-used outlierness measures (e.g., density, proximity) in previous OD methods, we explore network uncertainty and validate it as a highly effective outlierness measure, while two practical score refinement strategies are also designed to improve OD performance. Finally, in addition to the discriminative learning paradigm above, we also explore the solutions that exploit other learning paradigms (i.e., generative learning and contrastive learning) to introduce self-supervision for *E^3Outlier*. Such extendibility not only brings further performance gain on relatively difficult datasets, but also enables *E^3Outlier* to be applied to other OD applications like video abnormal event detection. Extensive experiments demonstrate that *E^3Outlier* can considerably outperform state-of-the-art counterparts by 10%-30% AUROC. Demo codes are available at https://github.com/demonzyj56/E3Outlier.

Index Terms—Deep neural networks, outlier detection, self-supervised learning, unsupervised learning

1 Introduction

In realms like machine learning and data science, outliers, which are also called novelties, anomalies, deviants, exceptions, irregularities, etc. [1], have a pervasive existence. Outlier detection (OD), which may also be referred as unsupervised anomaly/outlier detection, is a long-standing problem that draws continuous attention from the research community. To provide a clear and strict formulation of OD problem, this paper follows the definition used in the recent OD survey paper [2]: Given a set of data instances, OD is an unsupervised task that aims to identify those instances that deviate significantly from the rest of data. Thus, outliers are discerned from given unlabeled data by a *transductive* learning setup. OD is of great importance in practice: First, as data labeling is usually expensive and time-consuming, it is often required to deal with massive unlabeled data. As a result, OD has been a frequently-encountered unsupervised task when handling prevalent unlabeled data. Second, even for supervised/semi-supervised tasks, OD plays a vital role in the data cleansing stage (e.g., removing wrongly-labeled data or noise when building a data set), which is the foundation for obtaining high-quality models. OD enjoys a variety of real-world applications, such as financial fraud detection [3], emerging topic detection [4], computer-aided medical diagnosis [5], motion trajectory analysis [6], etc. Since the only prior knowledge is that outliers have rare occurrence when compared with inliers, no supervision information is available for OD here. Due to its unsupervised nature, OD is usually addressed by exploiting some intrinsic properties of data, e.g., density, proximity, cluster membership, etc. A more detailed review of classic OD is given in Section 2.1. In particular, we distinguish OD in this paper from the *(semi-supervised) anomaly detection* or *one-class classification* [7], which builds a normality model from a pure set of labeled normal data and detects deviants in a separated test set by an *inductive* learning setup. To avoid any confusion, a detailed clarification of terms is also provided in Sec. 6 of the supplementary material, which can be found on the Computer

- *Siqi Wang, Guang Yu, Zhen Cheng, Xinwang Liu, and En Zhu are with the College of Computer Science and Technology, National University of Defense Technology (NUDT), Changsha 410073, China.*
  E-mail: 405976789@qq.com, {guangyu, chengzhen16, xinwangliu, enzhu}@nudt.edu.cn.
- *Yijie Zeng is with Nanyang Technological University, Singapore 639798. E-mail: yzeng004@e.ntu.edu.sg.*
- *Sihang Zhou is with the College of Intelligent Science and Technology, NUDT, Changsha 410073, China. E-mail: sihangjoe@gmail.com.*
- *Marius Kloft is with the Department of Computer Science, TU Kaiserslautern, 67663 Kaiserslautern, Germany. E-mail: kloft@cs.tu-kl.de.*
- *Jianping Yin is with the Dongguan University of Technology, Dongguan 523808, China. E-mail: jpyin@dgut.edu.cn.*
- *Qing Liao is with the School of Computer Science and Technology, Harbin Institue of Technology (Shenzhen), Harbin, Heilongjiang 150001, China. E-mail: liaoqing@hit.edu.cn.*

Manuscript received 29 June 2020; revised 23 August 2021; accepted 9 April 2022. Date of publication 6 July 2022; date of current version 3 February 2023. This work was supported in part by the National Natural Science Foundation of China under Grants 61922088 and 62006236, in part by NUDT Research Project under Grant ZK20-10, and in part by HPCL Autonomous Project under Grant 202101-15.
(Corresponding author: Xinwang Liu.)
Recommended for acceptance by T.M. Hospedales.
This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2022.3188763, provided by the authors.
Digital Object Identifier no. 10.1109/TPAMI.2022.3188763

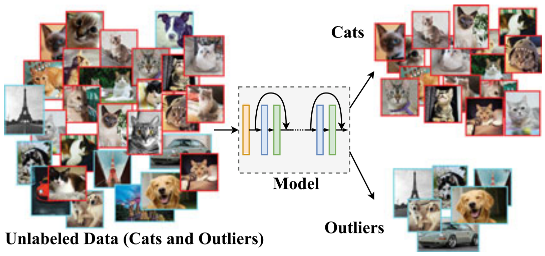


Fig. 1. An example of deep outlier image removal task.

Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3188763, so as to differentiate OD here from other relevant but different realms like (semi-supervised) anomaly detection and out-of-distribution detection.

With the widespread use of photographic equipment (e.g., cameras, smart phones), visual data like images and videos have undergone an explosive growth in these years. In this context, a marriage of OD and visual data is pretty natural, and it gives birth to many novel applications, such as the refinement of web image search results [8], [9] and video abnormal event detection [10], [11]. Among various forms of visual data, images have constantly played a fundamental role in all sorts of visual analysis. Therefore, this paper will focus on OD for image data, i.e., the image outlier removal task. For an intuitive illustration, we show an example that aims to remove outliers from images of cats (inliers) in Fig. 1. Compared with frequently-seen tabular data (or vectorized data), image data exhibit evidently different characteristics: They possess a variety of high-level spatial structures that are endowed with rich semantics, and low-level details (i.e., image pixels) alone are much less meaningful to perception. As a consequence, a direct application of those classic OD methods to image data usually leads to poor performance, and proper image representations will be a prerequisite for successful outlier removal. As a simple solution, some works [8], [12] extract the image representations by hand-crafted feature descriptors (e.g., SIFT [13], sparsity-constrained linear coding [14]), and then feed the extracted feature vectors into a classic OD method. However, such solutions bring about complex feature engineering issues, and they often suffer from sub-optimal image representations and poor transferability. To this end, an emerging trend is to learn good representations automatically via deep neural networks (DNNs) during the learning process, so as to realize a certain goal like image classification or segmentation. Such an end-to-end deep learning paradigm has achieved remarkable success in computer vision, especially with discriminative DNNs for supervised learning tasks [15]. However, although introducing DNNs for deep outlier removal seems to be pretty straightforward, a both *effective* and *end-to-end* DNN based OD solution still requires exploration. The major impediment to developing such a solution lies in the unsupervised nature of the OD task, i.e., the absence of data labels results in a lack of supervision signal. Consequently, as several recent surveys point out [2], [16], [17], [18], auto-encoder (AE) still plays a dominant role in deep OD, while other widely-used DNNs like discriminative ResNet [19] are not directly applicable for deep OD without any given labels.

To bridge those gaps in deep OD, we propose the first self-supervised framework termed E^3*Outlier*, which aims to realize both *effective* and *end-to-end* deep outlier removal. Specifically, our core idea is to remedy the label absence in OD by introducing self-supervision. To this end, our major solution is to create multiple pseudo classes from given unlabeled data by imposing certain data operations like rotation and patch re-arranging. With labels of those pseudo classes, powerful discriminative DNNs that have been thoroughly studied can be exploited in OD and enable more effective representation learning. Second, in order to further conduct end-to-end OD, we unveil a property named "inlier priority": Even though inliers and outliers are indiscriminately fed into the DNN during self-supervised learning, the DNN tends to prioritize inliers' loss reduction. We provide both theoretical and empirical demonstration to this property. Third, instead of commonly-used outlierness measure (e.g., density and proximity), we point out that the DNN uncertainty in self-supervised learning can be leveraged to design highly effective outlier scores. Meanwhile, inspired by the inlier priority and network uncertainty, we develop two practical strategies and fuse them into a score refinement stage to yield performance enhancement. Finally, in addition to the aforementioned discriminative learning paradigm, we further design the solution to leverage generative/contrastive learning paradigm to perform self-supervised learning for the proposed E^3*Outlier* framework. With the extendibility to different learning paradigms, E^3*Outlier* is not only able to be flexibly applied to other OD applications like video abnormal event detection, but also yield further performance gain on relatively difficult datasets. Our main contributions can be summarized below:

- We for the first time design a self-supervised learning framework for DNN based OD. It not only eases the lack of supervision, but also enables discriminative DNNs to be directly applied to the deep OD problem.
- We unveil a property named inlier priority during self-supervised learning, and theoretical and empirical demonstration are presented to justify this property. It lays the foundation to perform end-to-end OD with the proposed E^3*Outlier* framework.
- We point out that the uncertainty of discriminative DNN can be exploited as a novel outlierness measure in deep OD, and develop several highly effective uncertainty based outlier scores for end-to-end OD. Moreover, we propose joint score refinement with two practical strategies to boost the OD performance.
- We further design solutions that incorporates generative learning and contrastive learning paradigm into the E^3*Outlier* framework to provide self-supervision, which endows the proposed framework with more flexibility and better OD performance.

An earlier version of this paper is reported in [20], and this paper is mainly extended in terms of the following aspects: (1) This paper explicitly points out that DNN uncertainty can be used as a new outlierness measure, and intuitively unveils the connection among OD, self-supervised learning and network uncertainty. Compared with this paper, [20] just reported empirical comparison of different outlier scores and did not provide in-depth analysis into the underlying principle of score design. (2) We design several practical strategies to conduct outlier score refinement, which enables the model to achieve consistent performance enhancement against the performance reported in [20] on all benchmark datasets. (3) Unlike [20] that only exploited discriminative learning paradigm for deep OD, this paper further validates the applicability of generative learning or contrastive learning paradigm to E^3*Outlier* . (4) Apart from the image outlier removal task in [20], this paper shows that the proposed E^3*Outlier* framework is also able to achieve superior performance in other deep OD application like unsupervised video abnormal event detection.

2 Related Work

2.1 Shallow Model Based Outlier Detection

A vast number of shallow methods have been proposed to handle OD, and they usually fall into the following categories: (1) Proximity based methods, which measure the outlierness of a datum by its relation to its neighboring data. Early methods of this type simply assume the data density to be homogeneous, and define some intuitive quantities as outlier scores, such as the distance to the k-th neareast neighbors (k-nn) [21] and the number of neighbors within a predefined radius [22]. To this end, Local Outlier Factor (LoF) [23] is the first work that considers local outliers using the average ratio of one datum's neighbor's local reachability density to its own reachability density, which inspires numerous subsequent works, e.g., Connectivity-based Outlier Factor (CoF) [24] which considers the degree of connectivity among data when computing outlier scores, while Local Outlier Probability (LoOP) [25] estimates the probability of being an outlier by assuming a half-Gaussian distribution on a datum's distance to its k-nn. As computing k-nn can be time-consuming, recent works [26], [27] propose to leverage subsampling and achieve linear time complexity. (2) Statistics based methods, which view data endowed with low likelihood as outliers. The likelihood can be estimated by several statistical models, including parametric and non-parametric statistical models. As to parametric models, the most representative model is Gaussian Mixture Model (GMM) [28], and recently a more robust GMM based OD approach is proposed by Tang *et al.* [29] by incorporating subspace learning. Meanwhile, as to non-parametric models, kernel density estimation (KDE) [30] is frequently used for OD, while its recent variants like [31], [32], [33] are developed to improve its efficiency of OD. (3) Clustering based methods, which view data that do not belong to any major data cluster as outliers. For example, Jiang *et al.* [34] perform OD by a modified k-means algorithm and calculating a minimal spanning tree with cluster centers. He *et al.* [35] use clustering to devise CBLOF, which quantitatively distinguishes clusters with different sizes. To avoid specifying the number of clusters, a recent work by Yan *et al.* [36] proposes to leverage Gibbs Sampling of Dirichlet Process Multinomial Mixture (GSDPMM) for OD. Chenaghlou *et al.* [37] extends the clustering based OD to online streaming data by considering evolving of clusters. (4) Projection based methods, which project the input data into a new space to manifest outlierness. Concretely, data can be projected into a low-dimensional embedding by dimension reduction techniques like principal component analysis (PCA) [38] or neural networks like autoencoder networks [39], and outliers are viewed to be those data that are poorly recovered from the embeddings. In particular, Liu *et al.* [40] propose Isolation Forest (IF), which projects input data into the tree nodes of random binary trees, and then discriminate outliers by the depth of tree nodes. IF proves to be a both effective and efficient OD method, while recent works by Hariri [41] propose to further improve IF by using random hyperplane cut. Besides, projection techniques like local sensitivity hashing [42] and random projection [43] are also used to reduce complexity of OD models. A more comprehensive review on shallow OD methods can be found in recent survey papers [2], [16], [17], [18]

2.2 DNN Based Outlier Detection

As a newly-emerging topic, DNN based OD is highly challenging as it requires to learn suitable data representations for OD. To our best knowledge, only few attempts have been made in the literature. A straightforward idea is to exploit a two-stage solution, which performs representation learning by DNNs first, and then feeds learned features into a separated module that is implemented by some classic OD model (reviewed in [44]). However, such two-stage approaches may suffer from the incompatibility between learned features and the OD module, which can lead to sub-optimal performance. By contrast, state-of-the-art methods usually conduct a joint learning of data representations and outlier scores, and we review each existing solution to our best knowledge below: Xia *et al.* [9] design a new loss function that encourages a better separation of inliers and outliers by minimizing intra-class variance for multi-layer AE, and propose an adaptive thresholding technique to discriminate outliers; Zhai *et al.* [45] connect an energy based model with a regularized AE, and develop an energy based score for OD; Zhou *et al.* [46] utilize a combination of deep AE and Robust Principal Component Analysis (RPCA), which decomposes the matrix of unlabeled data into a low-rank part and a sparse part to represent inliers and outliers respectively, while Chalapathy *et al.* [47] also adopt a similar idea; Chen *et al.* [39] propose to generate a set of AEs that possess randomly varied connectivity architecture to perform OD, while adaptive sampling is leveraged to make the approach more efficient and effective. Inspired by Gaussian Mixture Model (GMM), Zong *et al.* [48] focus on developing an end-to-end OD solution that embeds a GMM density estimation network into the deep AE, and both components are optimized simultaneously; Unlike other methods that rely on AEs, Pang *et al.* [49] propose a ranking-model based framework named RAMODO, which can be readily incorporated into random distance based OD approach to perform efficient OD with tabular data; Liu *et al.* [50] convert OD into a binary classification problem via generative adversarial networks (GANs) [51], which are modified to generate simulated outliers; The most recent work [52] exploits the latent low-dimensional subspace structure in data by adding a Robust Subspace Recovery (RSR) regularizer into AE, and two variants, RSRAE and RSRAE+, are proposed for deep outlier removal. As several recent surveys point out [2], [17], [18], AE still plays a center role in existing deep OD solutions due to its unsupervised nature, which motivates us to develop $E^3Outlier$.

2.3 Self-Supervised Learning and Network Uncertainty

Self-supervised learning, which is also known as surrogate supervision [53] based learning or pseudo supervision [54] based learning, enjoys a swift growth of popularity in recent research. Its core idea is to construct additional supervision signals from given data by introducing a pretext task. The learning targets of pretext task can be obtained by numerous ways, such as clustering [55], geometric transformations [56], [57], masking [58], image patch permutation [59], time sequence shuffling [60], contrastive learning [61], etc. As a highly effective pre-training technique or auxiliary task to improve the performance of high-level downstream tasks, self-supervised learning has been explored in many application scenarios, such as image classification, segmentation, object detection, action recognition [62] and anomaly detection [57]. To our best knowledge, this is the first work that connects self-supervised learning to unsupervised OD.

DNN's uncertainty reflects its confidence to a certain prediction, which usually makes it a concept for inductive learning. Several methods have been proposed to quantify network uncertainty, such as Bayesian Neural Networks (BNN) [63], Monte Carlo dropout (MC-Dropout) [64], model ensemble [65], maximum softmax probability [66], information entropy [67], etc. Despite that network uncertainty has drawn increasing attention, its application is typically limited to knowing whether DNN makes trustworthy predictions or detecting the dataset shift. In this paper, we for the first time discuss network uncertainty under a transductive setup, and demonstrate that it can serve as a fairly effective outlierness measure for DNN based OD.

3 The Proposed Framework

3.1 Problem Formulation

Suppose that the data space spanned by all images is denoted by $\mathcal{X}$. DNN based OD deals with a completely unlabeled image data collection $X \subseteq \mathcal{X}$ that is contaminated by outlier images. In other words, X consists of an inlier set X_{in} and an outlier set X_{out}, while $X = X_{in} \cup X_{out}$ and $X_{in} \cap X_{out} = \emptyset$. By the definition of outliers [68], image data of the inlier set are from the same underlying distribution that shares close semantics, but outliers originate from different distributions. Given any image $\mathbf{x} \in \mathcal{X}$, DNN based OD intends to build a scoring model $S(\cdot)$, which takes raw $\mathbf{x}$ as the input and does not perform any prior feature extraction. The goal of $S(\cdot)$ is to output $S(\mathbf{x}) = 1$ for any inlier $\mathbf{x} \in X_{in}$, while $S(\mathbf{x}) = 0$ for any outlier $\mathbf{x} \in X_{out}$. In practice, a larger output $S(\mathbf{x})$ signifies a lower likelihood to be an outlier for $\mathbf{x}$. Besides, within the domain of DNN based OD, end-to-end or deep OD refers to the case where both representation learning and OD can be carried out by the same DNN, and no separated classic OD method is involved. In this paper, the proposed *E^3Outlier* framework aims to achieve both effective and end-to-end OD.

3.2 Discriminative *E^3Outlier*

3.2.1 Motivation

As reviewed in Section 2.2, it is noted that AE based solutions play a center role in the deep OD task due to its unsupervised setup. Specifically, deep AE based solutions typically perform unsupervised representation learning by learning to reconstruct the inputs, which is realized by training the deep AE to reduce pixel-wise reconstruction errors like mean square errors (MSE). However, recent researches like [69], [70] demonstrate that such a pixel-wise reconstruction tends to overemphasize low-level image details, which are of very limited interest to human perception. By contrast, semantics of high-level image structures are ignored, but they are actually pivotal to DNN based OD. Another emerging type of generative DNNs is GANs. Despite of fruitful progress, it is still challenging to integrate them into OD [71]: First, it is actually difficult to generate sufficient realistic image outliers, as potential image outliers are infinite and generating high-quality image outliers by GANs is still an open topic; Second, efficient representation learning with GANs is neither straightforward nor easy. By comparison, the supervised discriminative learning paradigm is still the most effective way to learn image semantics and capture high-level structures so far. As a result, these reasons above motivate us to introduce *self-supervision*, so as to enable the use of discriminative learning paradigm in OD.

3.2.2 Self-Supervised Discriminative Network (SSD)

The availablity of supervision signals is the key to introduce discriminative DNNs like ResNet [19] and Wide ResNet (WRN) [72] to OD. As image classification is the most fundamental task in supervised learning, creating several pseudo classes from given unlabeled data is a natural idea. Instead of generating a pseudo outlier class like [50], which is a straightforward but difficult task, we propose to build self-supervision by exerting some frequently-seen data operations on given images. Those new data produced by a certain operation are viewed as one pseudo class. Afterwards, we can readily realize representation learning with a discriminative DNN by training it to classify those created pseudo classes. As the discriminative DNN is guided by self-supervision, we term it *self-supervised discriminative network (SSD)* here. Formally, supposing a set of K operations $\mathcal{O} = \{O(\cdot|y)\}_{y=1}^{K}$ is designed to create pseudo classes, we impose the y-th operation $O(\cdot|y)$ on an unlabeled image $\mathbf{x}$ (regardless of an inlier or outlier) and produce a new image $\mathbf{x}^{(y)} = O(\mathbf{x}|y)$. In this way, we can create the y-th pseudo class $X^{(y)} = \{\mathbf{x}^{(y)}|\mathbf{x} \in X\}$, with the pseudo label y assigned to all data in this class. Then, given all data $X' = \{X^{(1)}, \ldots, X^{(K)}\}$ and their label set Y, an SSD with a K-node Softmax layer is trained to perform classification. Like the standard classification process, the SSD is supposed to classify a datum $\mathbf{x}^{(y')}$ into the y'-th pseudo class. The probability vector of $\mathbf{x}^{(y')}$ output by SSD's Softmax layer is denoted as $\mathbf{P}(\mathbf{x}^{(y')}|\boldsymbol{\theta}) = [P^{(y)}(\mathbf{x}^{(y')}|\boldsymbol{\theta})]_{y=1}^{K}$, where $P^{(y)}(\cdot)$ and $\boldsymbol{\theta}$ indicate the probability from the y-th node of Softmax layer and DNN's learnable parameters respectively. To train the SSD, we can minimize the following objective function:

$$\mathcal{L}_{DSS} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta}) \tag{1}$$

where $\mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta})$ represents the loss incurred by $\mathbf{x}_i$ in X during the self-supervised learning. When the standard cross-

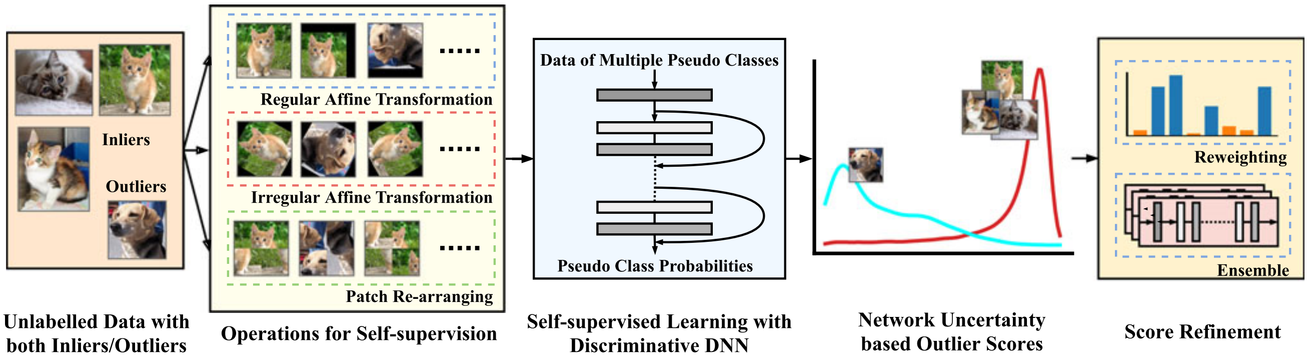

Inliers
Outliers
Unlabelled Data with both Inliers/Outliers
Regular Affine Transformation
Irregular Affine Transformation
Patch Re-arranging
Operations for Self-supervision
Data of Multiple Pseudo Classes
Pseudo Class Probabilities
Self-supervised Learning with Discriminative DNN
Network Uncertainty based Outlier Scores
Reweighting
Ensemble
Score Refinement

Fig. 2. Overview of the proposed discriminative E^3*Outlier* for deep OD.: Given unlabeled image data polluted by outliers, three operation sets are first imposed on images to create multiple pseudo classes and provide self-supervision. Then, a discriminative DNN is trained to perform the self-supervised learning, i.e., learning to classify those created pseudo classes. Next, the outlierness of each image is measured by the proposed network uncertainty based outlier score. Finally, the joint score refinement with re-weighting and ensemble strategy can be used to further boost the OD performance of E^3*Outlier*.

entropy loss is used, $\mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta})$ takes the form below:

$$\mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta}) = -\frac{1}{K}\sum_{y=1}^{K}\log\left(P^{(y)}(\mathbf{x}_i^{(y)}|\boldsymbol{\theta})\right) \tag{2}$$

Another key to SSD is the design of data operation. We introduce three sets of operations: Regular affine operation set $\mathcal{O}_{RA}$, irregular affine operation set $\mathcal{O}_{IA}$ and patch re-arranging operation set $\mathcal{O}_{PR}$. The general intuition behind those operations is to force DNN to capture the semantics of high-level structures in an image when it is required to fulfill such a classification task. For example, to recognize what type of rotation is imposed on the original image, the DNN must learn to localize salient object in images and recognize the orientation of its high-level parts, such as the head and legs of a human. Due to the page limit, we illustrate the details of data operation design in Section 1 of the supplementary material, available online. Because of the prevalence of discriminative DNNs, creating pseudo classes by data operations is an intuitive and convenient way to provide self-supervision for deep OD. The overview of discriminative E^3*Outlier* is presented in Fig. 2. We will show other learning paradigms are also applicable to the proposed E^3*Outlier* in later chapters.

3.2.3 Comparison Between SSD and AE

To verify whether SSD can learn better image representations, we conduct a simple experiment that compares SSD with Convolutional AE (CAE). We select WRN-28-10 [72] as SSD and adopt the CAE architecture in [57], which has a close depth to the SSD. Then, we extract the outputs of SSD's penultimate layer as learned representations, while the outputs of CAE's intermediate layer are extracted for comparison (note that they share the same dimension). With the protocol described in Section 4.1 to evaluate the OD performance on image datasets, learned representations of SSD and CAE are both fed into an Isolation Forest (IF) model with the same parameterization to conduct OD. The comparison is shown in Fig. 3: On those image benchmarks, learned representations of SSD are always able to improve IF's OD performance, which justifies SSD's effectiveness.

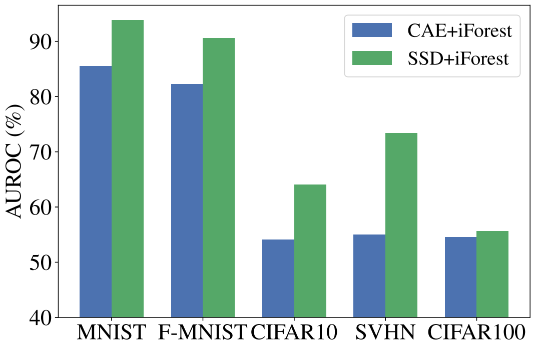


Fig. 3. Comparison of learned image representations.

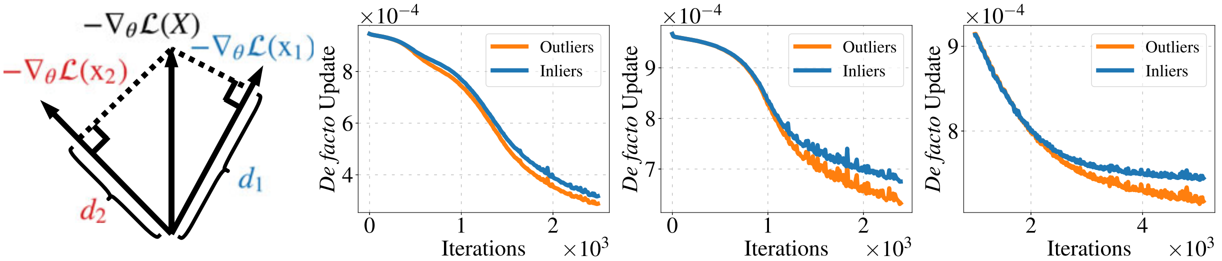
3.3 Inlier Priority: Foundation of End-to-End OD

3.3.1 Motivation

Although the proposed SSD achieves more effective representation learning than CAE, there are still some problems: First, without using a specialized OD network like [48], the proposed paradigm actually learns a pre-text task (i.e., classification) instead of OD, so by now we cannot draw OD results directly from SSD alone; Second, although we can resort to a classic OD model like we did in Section 3.2.3, such a two-stage solution can be sub-optimal as learned representations and the OD model are not jointly optimized. In fact, the OD performance of SSD+IF solution in Section 3.2.3 indeed has room for improvement (60%-70% AUROC) on relatively difficult benchmarks, i.e., CIFAR10/SVHN/CIFAR100. Therefore, an end-to-end solution is favorable for deep OD. However, for the proposed SSD, data operations are equally imposed on both inliers and outliers to create a pseudo class, and they are indiscriminately fed into DNN for training. Thus, it is still not sure whether inliers and outliers will behave differently during the self-supervised learning. This motivates us to explore this issue below from both theoretical and empirical view.

3.3.2 The Theoretical View

First of all, we approach this issue from a theoretical view. Since the theoretical analysis of DNNs remains particularly difficult, we consider a simplified case that is analyzable: We choose a feed-forward network with a single hidden layer and sigmoid activation to be SSD. Suppose that the hidden layer and Softmax layer have $(L+1)$ and K nodes respectively. Parameters of the simple SSD is randomly



(a) *De facto* update (b) MNIST ("3") (c) Fashion-MNIST ("bag") (d) CIFAR10 ("horse")

Fig. 4. An illustration of *de facto* update and the average *de facto* update of inliers/outliers during the network training. The class used as inliers is in brackets.

initialized by an i.i.d uniform distribution on $[-1, 1]$. Since neural networks are usually optimized by gradient descent, the influence of inliers and outliers imposed on the SSD can be reflected by the gradients that they back-propagate to update the network parameters. Hence, we analyze gradients w.r.t the weights associated with the c-th class ($1 \leq c \leq K$) between the hidden layer (it is also the penultimate layer in this case) and the final Softmax layer, $\mathbf{w}_c = [w_{s,c}]_{s=1}^{(L+1)}$ ($w_{L+1,c}$ is the bias), which are directly responsible for making SSD's predictions. We discuss the case of iniers (X_{in}) first: For the cross-entropy loss $\mathcal{L}$ that is used in our case, suppose that only those data yielded by imposing the c-th operation on X_{in} are used to update $\mathbf{w}_c$, i.e., $X_{in}^{(c)} = \{\mathbf{x}^{(c)} = O(\mathbf{x}|c)|\mathbf{x} \in X_{in}\}$. The gradient vector incurred by $X_{in}^{(c)}$ is denoted by $\nabla_{\mathbf{w}_c}\mathcal{L} = [\nabla_{w_{s,c}}\mathcal{L}]_{s=1}^{(L+1)}$, and each element of $\nabla_{w_{s,c}}\mathcal{L}$ is given by:

$$\nabla_{w_{s,c}}\mathcal{L} = \sum_{i=1}^{N_{in}} \nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i) = \sum_{i=1}^{N_{in}} (P^{(c)}(\mathbf{x}_i) - 1)h^{(s)}(\mathbf{x}_i) \tag{3}$$

where $N_{in} = |X_{in}^{(c)}| = |X_{in}|$ is the number of inliers. For $\mathbf{x}_i \in X_{in}^{(c)}$, $P^{(c)}(\mathbf{x}_i)$ is the output of c-th node in the Softmax layer, and $h^{(s)}(\mathbf{x}_i)$ is the output of s-th node in the penultimate layer. To quantify inliers' influence on a randomly initialized SSD, a direct indicator can be the expectation of inliers' gradient magnitude to update $\mathbf{w}_c$, $E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2)$. Thus, our goal is to obtain:

$$E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2) = E\left(\sum_{s=1}^{L+1} (\nabla_{w_{s,c}}\mathcal{L})^2\right) = \sum_{s=1}^{L+1} E\Big((\nabla_{w_{s,c}}\mathcal{L})^2\Big) \tag{4}$$

By addition in (3), computing (4) requires the term below:

$$\begin{aligned} E\Big((\nabla_{w_{s,c}}\mathcal{L})^2\Big) &= E\left(\left(\sum_{i=1}^{N_{in}} \nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i)\right)^2\right) \\ &= \sum_{i=1}^{N_{in}} \sum_{j=1}^{N_{in}} E(\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i)\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_j)) \end{aligned} \tag{5}$$

To compute (5), in our case we can resort to the second-order Taylor series expansion to derive the approximation below (detailed in Section 2 of the supplementary material, available online):

$$\begin{aligned} &E(\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i)\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_j)) \approx \\ &h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)\left[\frac{(K-1)^2}{K^2} + \frac{K-1}{3\ K^3}\sum_{t=1}^{L+1} h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j)\right] \end{aligned} \tag{6}$$

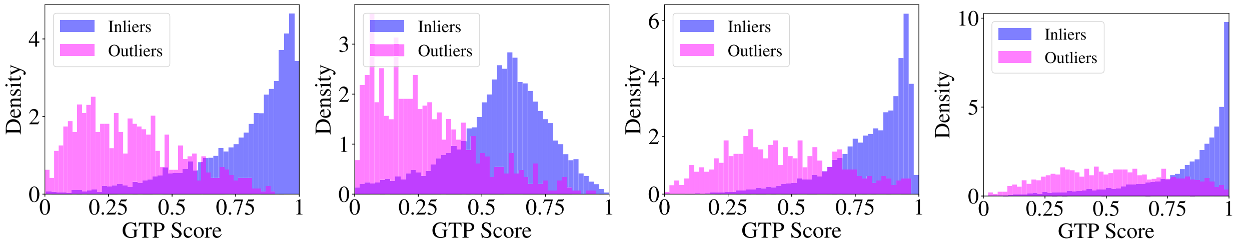
There remains to calculate $h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j)$ in (6). In this case, [73, Lemma 3.b] has proved that the expectation of $h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)$ w.r.t the randomly initialized weights between the input and hidden layer satisfies $E(h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)) \approx \frac{1}{4}$ and $E(h^{(s)}(\mathbf{x}_i)^2\ h^{(s)}(\mathbf{x}_j)^2) \approx \frac{1}{16}$. Thus, by definition of $||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2$ in (4) and (5), we yield:

$$\begin{aligned} E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2) &\approx N_{in}^2\left[(L+1)\left(\frac{(K-1)^2}{4\ K^2} + \frac{(K-1)(L+1)}{48\ K^3}\right)\right] \\ &\triangleq N_{in}^2 \cdot Q \end{aligned} \tag{7}$$

Since L, K above are both fixed, Q is a constant. As a result, (7) shows that for the self-supervised learning of SSD, $E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2)$ is roughly proportional to N_{in}^2. Likewise, we can also derive that the expectation of outliers' gradient magnitude is $E^{(out)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2) \approx N_{out}^2 \cdot Q$. Since $N_{in} \gg N_{out}$ is an indispensable premise for the OD task, we have $E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2) \gg E^{(out)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2)$, which leads to an interesting conclusion: Although inliers and outliers are equally used for the self-supervised learning of SSD, the gradients contributed by inliers are much more important than outliers. Since those back-propagated gradients are used to train SSD, the theoretical analysis leads to an underlying property: *SSD is inclined to prioritize inliers during self-supervised learning*, which is named *inlier priority* in this paper. Such a property implies that inliers and outliers behave differently in self-supervised learning, which makes it possible to establish an end-to-end OD solution. Since it is intractable to compute $E(h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j))$ for more complex SSD, we will further validate inlier priority by empirical validations in the next section.

3.3.3 Empirical Validations

To further validate the property of inlier priority empirically, we propose to calculate a more direct indicator named "*de facto* update" for inliers and outliers respectively: In addition to gradient magnitude that we have considered in previous theoretical analysis, another important attribute of gradient vectors is gradient direction. As illustrated by Fig. 4a, consider $\mathbf{x}_i$ from a batch of data X (we slightly abuse the notation of X here). The negative gradient $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)$ is the fastest network updating direction to reduce $\mathbf{x}_i$'s loss. However, the network weights $\boldsymbol{\theta}$ are actually updated by the averaged negative gradient of the entire batch X, $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(X) = -\frac{1}{N}\sum_i \nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)$. Thus, the actual updating direction at each iteration is usually different from the best updating direction for each individual



(a) MNIST ("3") (b) Fashion-MNIST ("bag") (c) CIFAR10 ("horse") (d) SVHN ("3")

Fig. 5. Normalized histograms of inliers/outliers' $S_{gtp}(\mathbf{x})$. The class used as inliers is in brackets.

datum. To measure the actual gradient magnitude that $\mathbf{x}_i$ obtains along its best direction for loss reduction from $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(X)$, we introduce the concept *de facto* update, which is computed by projecting $\nabla_{\boldsymbol{\theta}}\mathcal{L}(X)$ onto the direction of $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)$: $d_i = \nabla_{\boldsymbol{\theta}}\mathcal{L}(X) \cdot \frac{\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)}{||\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)||}$ [9]. For example, as shown in Fig. 4a, the *de facto* update d_1 and d_2 reflect how much effort the network will devote to reduce the training loss of $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively. *De facto* update can be viewed as an even more direct indicator of data's priority during training. In our case, we still take the gradients w.r.t. the weights between SSD's penultimate and softmax layer as an example. Under the setup in Section 4.1, we calculate the average *de facto* update for inliers and outliers respectively, and visualize typical results of *de facto* update on several image benchmarks in Figs. 4b, 4c, and 4d: As can be seen from the results, despite being close at the beginning, the average *de facto* update of inliers becomes evidently higher than outliers as the training continues, which justifies that SSD will bias towards inliers' best updating directions.

3.3.4 Baseline Outlier Score and Additional Remarks

Having illustrated inlier priority both theoretically and empirically, it can be expected that inliers are likely to achieve better training performance than outliers on a SSD after the self-supervised learning. In other words, SSD will prioritize reducing inliers' loss, which suggests that it is possible to discriminate outliers directly by each datum's loss value after training. To be more specific, for an image $\mathbf{x}^{(y)}$, we note that the calculation of its cross entropy loss only depends on its ground truth class probability $P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})$ that corresponds to its pseudo class label y. Thus, we propose Ground Truth Probability (GTP) score $S_{gtp}(\mathbf{x})$ that averages $P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})$ for all K operations to measure outlierness:

$$S_{gtp}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K}\mathbf{1}_y^{\top}\cdot \mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) = \frac{1}{K}\sum_{y=1}^{K}P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) \qquad (8)$$

where $\mathbf{1}_y$ denotes the one-hot vector with the y-th element to be 1. To validate whether GTP score is a plausible way to measure outlierness, we calculate the $S_{gtp}(\mathbf{x})$ on image benchmarks and visualize the accumulated histograms for inliers and outliers respectively (note that histograms are normalized for better visualization). Representative results are shown in Figs. 5a, 5b, 5c, and 5d, and the score distributions of inliers and outliers are observed to be readily separable. Thus, GTP score can be a feasible baseline score for end-to-end OD. In addition, we would also like to point out the relation between inlier priority and representation learning: In deep OD task like outlier image removal, the difference between outliers and inliers lie in their semantics, e.g., high-level structure and appearance. To encourage the semantic similarity within inliers and maximize the semantic difference between inliers and outliers, it is necessary to learn good representations with rich semantics in the first place. Thus, a learning task that can yield semantically meaningful representations is the foundation for inliers to be semantically similar and joint their efforts into a priority against outliers.

3.4 Network Uncertainty as an Outlierness Measure

3.4.1 Motivation

SSD+GTP score provides a baseline end-to-end OD solution. However, it is imperfect and still has room for improvement, especially considering that the proposed self-supervised learning is not as precise as the classic supervised learning with human annotations: The data operation sometimes may not be able to transform the original image into an actual new one, e.g., a digit "8" is still itself after flipping is performed. Therefore, labels assigned to pseudo classes can be inaccurate. Since the calculation of GTP score in (8) relies on the pseudo class label y, such inaccurate labeling may undermine the GTP score's effectiveness to discriminate outliers. Motivated by this problem, we intend to design a new outlierness measure that is independent of pseudo class labels, so as to exploit the possibility to further improve end-to-end OD performance. Besides, when compared with other outlierness measures like density or proximity, uncertainty is usually directly optimized during the training of DNN, while other measures are not an explicit goal of the optimization. Therefore, we believe that network uncertainty can be a more direct indicator of inlier priority than other traditional measures. To this end, network uncertainty comes into our sight, since it is exactly an orthogonal attribute to DNN's classification accuracy [74]. As previous works basically discuss this concept in the context of DNN's prediction confidence, it is interesting to explore whether network uncertainty can be used for end-to-end OD.

3.4.2 A Demonstration Experiment

We carry out a simple demonstration experiment to shed light on this issue. For visualization, we generate 2D data with different degree of outlierness (detailed in Section 3 in supplementary material, available online): The generated data (dots in Fig. 6) exhibit a larger dispersion as their coordinate on x-axis, x_i, gets more distant from the origin of

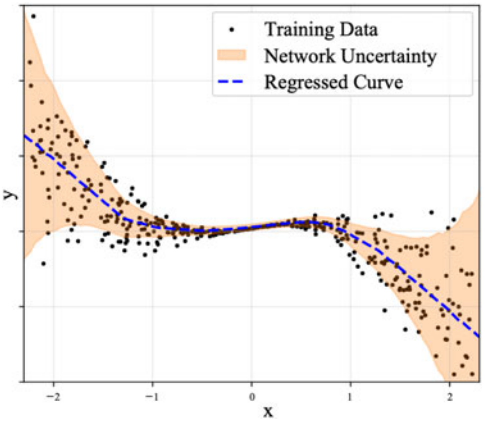


Fig. 6. The uncertainty of a regression network.

x-axis, which enables data on two ends to show larger outlierness. To calculate network uncertainty, we introduce a regression task that predicts y-axis coordinate y_i by corresponding x_i. Note that the regression task can be viewed as a self-supervised learning task, since we actually intend to infer the missing coordinate y_i by the incomplete data $\tilde{\mathbf{x}}_i = [x_i]$ like the masking mechanism [58]. The regression task is performed by training a simple neural network with the generated 2D data, and we estimate the uncertainty of neural network by the popular MC-Dropout method [64]. As it is shown in Fig. 6, it is easy to discover that the network uncertainty (highlighted orange region) is positively correlated to the outlierness of data. In other words, the experiment demonstrates some interesting connections among network uncertainty, OD and self-supervised learning: *The uncertainty of a neural network, which is trained to accomplish a self-supervised learning task (not OD itself), actually serves as a fairly effective way to measure data's outlierness.* Besides, it is also worth noting that network uncertainty is not relevant to the label y_i. This facilitates it to be more robust to label noises in self-supervised learning, just as we discussed in Section 3.4.1.

3.4.3 Network Uncertainty Based Outlier Scores

As reviewed in Section 2.3, the uncertainty of DNN can be estimated by several ways, which can be categorized into Bayesian methods and non-Bayesian methods. Since Bayesian methods are usually more complicated and require more modifications to DNN itself, we focus on non-Bayesian methods when designing outlier scores. The following network uncertainty based scores are designed: **(1)** Maximum Probability (MP) score $S_{mp}(\mathbf{x})$. $S_{mp}(\mathbf{x})$ utilizes the maximum probability (i.e., prediction probability) output by the Softmax layer of SSD, which has proved to be a simple but strong baseline for uncertainty estimation [66], [67]:

$$S_{mp}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K} \max \mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) = \frac{1}{K}\sum_{y=1}^{K} \max_t P^{(t)}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) \quad (9)$$

(2) MC-Dropout (MCD) score $S_{mcd}(\mathbf{x})$. MC-Dropout keeps the dropout layers functional during inference, and calculates the first and second-order moment of DNN's outputs by several forward passes [64]. Since the maximum output probability and variance in DNN's outputs are both able to reflect DNN's uncertainty, we devise $S_{mcd}(\mathbf{x})$ as follows, so as to adapt it to OD task ($Mean(\cdot)$ and $Var(\cdot)$ refers to the mean and variance of multiple forward passes):

$$S_{mcd}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K} -Var(\max \mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta})) + Mean(\max \mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta})) \quad (10)$$

(3) Negative Entropy (NE) based score $S_{ne}(\mathbf{x})$. Information entropy (i.e., Shannon entropy) has constantly been used for measuring information and uncertainty embedded in data. Thus, we design $S_{ne}(\mathbf{x})$ to be computing the negative entropy of SSD's output probability distribution $\mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta})$:

$$S_{ne}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K}\sum_{t=1}^{K} P^{(t)}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) \log\left(P^{(t)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})\right) \quad (11)$$

In addition to scores above, other network uncertainty based scores can also be explored. Our later evaluations show that network uncertainty based scores typically work better than the baseline outlier score S_{gtp}.

3.5 Score Refinement of Discriminative E^3 *Outlier*

3.5.1 Motivation

Although components presented above have constituted a fully-functional end-to-end OD solution, it is still possible to improve discriminative E^3*Outlier*'s performance. As we have demonstrated how inlier priority and network uncertainty enable end-to-end OD, they should also be considered as the origin for performance improvement. Intuitively, a better OD performance essentially suggests that the priority of inliers is magnified, while it can also be accomplished by better uncertainty estimation. Inspired by such instincts, we propose two types of strategies to refine outlier scores.

3.5.2 Re-Weighting Strategy

Our first instinct is to make SSD further prioritize inliers during training. Nevertheless, it is noted that inliers and outliers are indiscriminately fed into SSD at the very beginning of training, i.e., inliers and outliers are equally weighted by 1. Having revealed the role of inlier priority in OD, it is undoubted that this default initialization is not optimal: We can assign inliers with larger weights right before the beginning of SSD's training, which justifies the introduction of a re-weighting scheme. Since given data are completely unlabeled in OD, how and when to re-weight those unlabeled data for OD are key issues that we have to answer. As to how to re-weight, our solution is to utilize scores yielded by the proposed outlierness measure as weights, which have already achieved far better OD performance than existing methods. To be more specific, we can normalize scores into non-negative weights $w_1, \cdots w_N$ that satisfy $\sum_{i=1}^{N} w_i = 1$, and modify the objective function in (1) into the form below:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} w_i \mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta}) \quad (12)$$

As for when to re-weight, since scores are only accessible after self-supervised learning begins, we can perform re-weighting during or after SSD's training. Accordingly, we propose *online* re-weighting and *reboot* re-weighting strategy: Online re-weighting strategy will update the weights at the end of every epoch, and only one SSD is trained. By contrast, reboot re-weighting trains two SSD models: The first SSD is trained by a standard procedure, while the scores yielded by the first SSD are used as fixed weights to train the second SSD. The full algorithms are detailed in Algorithm 1 and Algorithm 2 in Section 4 of supplementary material, available online. Our evaluations show that both algorithms can improve E^3*Outlier*'s performance.

3.5.3 Ensemble Strategy

In addition to the re-weighting strategy, another instinct is to improve uncertainty estimation for better OD performance. Since a generic strategy that can be easily embedded into the model is always preferred, we introduce the ensemble strategy into the score refinement stage. Ensemble is a widely-used technique in machine learning that combines multiple models into a stronger one. It is shown to be a powerful tool to improve the predictive performance [75], and recent works also demonstrate that an ensemble of DNNs can be highly efficient for producing good model uncertainty estimates [65], [67]. Specifically, we first create multiple SSD models $M_1, \ldots, M_e$ in a certain way, where $e > 1$ is the number of SSD models. For example, we can initialize SSD models with different random seeds, or adopt several different network architectures as different SSD models. After self-supervised learning, we simply average the outputs of different SSD models by $\bar{\mathbf{P}}(\mathbf{x}_i^{(y)}|\boldsymbol{\theta}) = \frac{1}{e}\sum_{j=1}^{e}\mathbf{P}_j(\mathbf{x}_i^{(y)}|\boldsymbol{\theta})$, where $\mathbf{P}_j(\mathbf{x}_i^{(y)}|\boldsymbol{\theta})$ is the outputs of j_{th} SSD model. Afterwards, we can calculate any network uncertainty based score with $\bar{\mathbf{P}}(\mathbf{x}_i^{(y)}|\boldsymbol{\theta})$. Note that the ensemble process can be readily paralleled for potential acceleration. Our later empirical evaluations show that such simple ensemble technique almost consistently improves the OD performance when compared with the case where a single SSD model is used.

3.5.4 Joint Score Refinement

Two aforementioned strategies are both able to yield better outlier scores, but it should be noted that they actually refine outlier scores from different views: The re-weighting strategy strengthens the inlier priority during self-supervised learning, while the ensemble strategy aims to improve the estimation of network uncertainty. In other words, two strategies exploit non-overlapping facets for score refinement. Thus, using a joint strategy of the re-weighting and ensemble to achieve even better OD performance is natural. In this paper, we devise the final score refinement stage by combining the reboot re-weighting strategy with the ensemble strategy (shown in Algorithm 3 in Section 4 of the supplementary material, available online). Note that this is not the only form to combine re-weighting and ensemble, e.g., combining online re-weighting with the ensemble is also possible.

3.6 Other Learning Paradigms for E^3*Outlier*

In previous sections, we have demonstrated the way to leverage discriminative self-supervised learning to perform deep OD. As the way to introduce self-supervision is not limited to the discriminative learning paradigm, it is natural for us to explore other learning paradigms for E^3*Outlier* , which brings two benefits: First, more available learning paradigms enable E^3*Outlier* to be more flexible when dealing with different application scenarios. Second, emerging self-supervised learning paradigms like contrastive learning also facilitate E^3*Outlier* to further exploit its potential for deep OD. Thus, this section will detail our solution to apply generative and contrastive learning paradigms to E^3*Outlier*.

3.6.1 Generative E^3Outlier

Generative learning paradigm is not new, because AE based reconstruction is exactly the most frequently-used method in existing deep OD solutions so far. However, as illustrated in Section 3.2.3, existing generative solutions often perform unsatisfactorily. As self-supervision is shown to be surprisingly effective in discriminative E^3*Outlier* , it is instinctive for us to explore *whether self-supervision can also improve the performance of generative deep OD* . Specifically, our solution is to add richer self-supervision information into the generation process to avoid simple reconstruction of the inputs. Inspired by the fact that data operations can provide rich self-supervision signal in SSD, we propose the generative self-supervised learning (GSS) paradigm below: Consider a data operation set with K_g operations $\mathcal{O}_g = \{O_g(\cdot|y)\}_{y=1}^{K_g}$. The data operations in $\mathcal{O}_g$ can be defined by various ways, such as certain transformations or fetching a specific part or modality of the input data. Then, we draw two different operations $O_g(\cdot|y_1)$ and $O_g(\cdot|y_2)$ from $\mathcal{O}_g$. Given an input data $\mathbf{x}$, two operations are required to satisfy:

$$O_g(\mathbf{x}|y_1) \neq O_g(\mathbf{x}|y_2), \qquad y_1 \neq y_2 \tag{13}$$

Then, generative DNNs $\mathcal{G}$ (e.g., AEs, UNets [76] or GANs) are trained to generate $O_g(\mathbf{x}|y_2)$ by taking $O_g(\mathbf{x}|y_1)$ as input, which is equivalent to minimizing the objective below:

$$\mathcal{L}_{GSS}(y_1, y_2) = \frac{1}{N}\sum_{i=1}^{N}||\mathcal{G}(O_g(\mathbf{x}_i|y_1)) - O_g(\mathbf{x}_i|y_2)||_2^2 \tag{14}$$

It is easy to note that when Eq. (13) is not satisfied, Eq. (14) will degrade into plain reconstruction. When $\mathcal{G}$ has been trained, one can simply obtain an outlier score of $\mathbf{x}$ based on the MSE loss of generation:

$$S_g(\mathbf{x}|y_1, y_2) = -||\mathcal{G}(O_g(\mathbf{x}|y_1)) - O_g(\mathbf{x}|y_2)||_2^2 \tag{15}$$

Since there exist different ways to select operations, it is natural to train the model and compute final outlier score by a combination of different y_1, y_2 configurations:

$$\begin{aligned} \mathcal{L}_{GSS} &= \sum_{y_1}\sum_{y_2}\mathcal{L}_{GSS}(y_1, y_2), \\ S_g(\mathbf{x}) &= \sum_{y_1}\sum_{y_2} S_g(\mathbf{x}|y_1, y_2) \end{aligned} \tag{16}$$

Compared with the plain reconstruction adopted by AE based deep OD methods, the key to our generative E^3*Outlier* is to make DNN generate a different datum obtained by a non-identical operation, which makes the learning task more challenging for DNNs. This not only avoids the DNN to simply memorize the low-level details, but also encourages the DNN to consider high-level semantics by learning the correlations of two different data, which can be viewed as valuable self-supervision information. Our later evaluations show that generative E^3*Outlier* can produce tangible performance improvement when it shares the same generative DNN with other reconstruction based deep OD solutions. More importantly, generative E^3*Outlier* can be readily applied to some important scenarios where the input data can be decomposed into multiple views or modalities. For example, video data are usually considered from the view of both appearance and motion. In those cases, the correspondence between different data views/ modalities is valuable self-supervision signal in itself, and generative E^3*Outlier* provides a convenient and straightforward way to exploit such semantics. As a demonstration, we will show how to design a new unsupervised video abnormal event detection solution by generative E^3*Outlier* in Section 4.3.2.

3.6.2 Contrastive E^3 Outlier

It is easy to notice that the performance of current deep OD solutions, including the proposed discriminative E^3*Outlier* , suffers from evidently inferior performance on colored image datasets (e.g., CIFAR10) when compared with comparatively simple gray-scale image datasets (e.g., MNIST). Meanwhile, we also note that color based operations (e.g., color jittering and RGB-to-gray transformation) play an important role in many vision tasks. To further exploit color information and enhance the capability to handle more ubiquitous colored images in practical applications, we leverage the emerging contrastive learning paradigm, which is shown to be highly effective in unsupervised representation learning of real-world colored images, to provide self-supervision in deep OD and design contrastive E^3*Outlier* . The core idea of contrastive learning is to learn meaningful representations by making DNNs compare a pair of data drawn from the unlabeled dataset. We choose one of the most representative contrastive learning method, SimCLR [77], as the foundation for the proposed contrastive E^3*Outlier* . Specifically, a contrastive loss for a datum $\mathbf{x}$ is defined as follows:

$$\begin{aligned}&\mathcal{L}_{cl}(\mathbf{x}, X^+, X^-)\\ &= -\frac{1}{|X^+|}\log\frac{\sum_{\mathbf{x}'\in X^+}\exp(sim(z(\mathbf{x}), z(\mathbf{x}'))/\tau)}{\sum_{\mathbf{x}'\in X^+\cup X^-}\exp(sim(z(\mathbf{x}), z(\mathbf{x}'))/\tau)}\end{aligned} \tag{17}$$

where X^+/X^- denote the set with data that can form a positive/negative pair with $\mathbf{x}$, and $sim(\cdot,\cdot)$ is a similarity measure like cosine similarity. $|\cdot|$ is the cardinality of the set, and $z(\mathbf{x})$ is the projection yielded by feeding DNN's learned representation $f(\mathbf{x})$ into a projection layer $g(\cdot)$: $z(\mathbf{x}) = g(f(\mathbf{x}))$. τ is a hyperparameter. Next, the issue is to construct positive and negative data pairs to enable the calculation of Eq. (17). To this end, we introduce a random augmentation set $\mathcal{A}$, which contains augmentation operations that is composed of color jittering, RGB-to-gray transformation and image crop with random parameterization. Each time two independent random augmentation A_1 and A_2 are drawn from $\mathcal{A}$. After that, the data pair of augmented data $A_1(\mathbf{x})$ and $A_2(\mathbf{x})$ are viewed as a positive pair, while any other pair is viewed as negative. The goal of contrastive loss defined in Eq. (17) is to yield similar representations for a positive data pair, and make representations of a negative pair dissimilar. Given a mini-batch data set B drawn from the unlabeled dataset, SimCLR defined the following training objective to perform contrastive learning:

$$\begin{aligned}\mathcal{L}_{scl}(B, A_1, A_2) =& \frac{1}{2|B|}\sum_{i=1}^{|B|}(\mathcal{L}_{cl}(A_1(\mathbf{x}_i), \{A_2(\mathbf{x}_i)\}, \hat{B}_{-i})\\ &+ \mathcal{L}_{cl}(A_2(\mathbf{x}_i), \{A_1(\mathbf{x}_i)\}, \hat{B}_{-i}))\end{aligned} \tag{18}$$

where we define $\hat{B}_{-i} = \{A_1(\mathbf{x}_j)\}_{j\neq i} \cup \{A_2(\mathbf{x}_j)\}_{j\neq i}$. Some recent works [77], [78] point out that some data operations (e.g., 90 ° rotation) can be used to generate negative pairs as they produce very different data from the original one. This is also verified in discriminative E^3*Outlier*, since those data operations are often likely to produce pseudo classes that are readily separable. Following such an observation, we collect an operation set $\mathcal{O}_c = \{O_c(\cdot|y)\}_{y=1}^{K_c}$ with K_c operations (including one identity transformation), and expand the mini-batch B into $B' = O_c(B|1) \cup \cdots \cup O_c(B|K_c)$, where the data set $O_c(B|y) = \{O_c(\mathbf{x}|y)|\mathbf{x} \in B\}$. Since B' can be viewed as a data set with K_c pseudo classes and discriminative E^3*Outlier* works well in deep OD, we substitute B by B' into Eq. (18) for training, and make DNN learn to classify those pseudo classes by an additional discriminative module and the cross-entropy loss $\mathcal{L}_{cls}(B')$, so as to produce more meaningful representations. In this way, the contrastive self-supervised learning (CSS) of E^3*Outlier* can be performed by the joint loss below:

$$\mathcal{L}_{CSS} = \lambda \cdot \mathcal{L}_{scl}(B,' A_1, A_2) + \mathcal{L}_{cls}(B') \tag{19}$$

where λ is a weight. After training, we design a simple but effective outlier score based on inner product of projected representations: For the datum $\mathbf{x}_i^{(y)} = O_c(\mathbf{x}_i|y)$ obtained by imposing the y-th operation in $\mathcal{O}_c$ on $\mathbf{x}_i$, its outlier score $S_c(\mathbf{x}_i^{(y)})$ is given by the second largest inner product:

$$S_c(\mathbf{x}_i^{(y)}) = \frac{1}{Z_{scl}^{(y)}}\max_{j\neq j_{max}} z^\top(\mathbf{x}_i^{(y)}) \cdot z(\mathbf{x}_j^{(y)}) \tag{20}$$

where $Z_{scl}^{(y)}$ is the normalization term computed as follows:

$$Z_{scl}^{(y)} = \frac{1}{N}\sum_{i=1}^{N}||z(\mathbf{x}_i^{(y)})|| \tag{21}$$

In Eq. (20), the score actually computes the inner product between the projected representations of $\mathbf{x}_i^{(y)}$ and all data yielded by operation $O_c(\cdot|y)$, so as to measure how similar $\mathbf{x}_i^{(y)}$ is to data in the collection. With multiple operations in $\mathcal{O}_c$, the final outlier score can be computed by:

$$S_c(\mathbf{x}_i) = \sum_{y=1}^{K_c} S_c(\mathbf{x}_i^{(y)}) \tag{22}$$

Just like that contrastive learning paradigm significantly improves the performance of self-supervised learning, our later empirical evaluations show that contrastive E^3*Outlier* also advances the deep OD performance by a notable margin on those colored datasets that are relatively difficult for previous generative and discriminative E^3*Outlier*. As a summary, by designing generative learning and contrastive learning based solutions, we enable E^3*Outlier* to be a more flexible and stronger deep OD framework.

4 Experiments

4.1 Experimental Settings

4.1.1 Benchmark Datasets and Evaluation

To validate the effectiveness of the proposed framework, we conduct extensive experiments on five frequently-used public image benchmarks: MNIST (MST) [79], Fashion-MNIST (FMST) [80], CIFAR10 (C10) [81], SVHN (SH) [82], CIFAR100 (C100) [81]. We follow the standard procedure, which is shared by previous image outlier removal works like [8], [9], [46], to construct a mixed image set with outliers: Given a standard image benchmark, all images from a class with one common semantic concept (e.g., "horse," "bag") are retrieved as inliers, while outliers are randomly sampled from the rest of classes by an outlier ratio ρ. We vary ρ from 5% to 25% by a step of 5%. The assigned inlier/outlier labels are strictly unknown to OD methods and only used for evaluation. Each class of a benchmark is used as inliers in turn, and the performance on all classes is averaged as the overall OD performance on this benchmark dataset. Since all images are viewed as unlabeled in OD, we do not use the split of train/test set and merge them for experiments. Note that for CIFAR100 dataset, we uses 20 superclasses instead of the original 100 classes to ensure that the constructed mixed image set contains sufficient data for DNN's training, and it can also test the OD performance when inliers have multiple subclasses (each superclass in CIFAR100 contains 5 classes). All experiments are repeated for 5 times with different random seeds, so as to yield the average results. Raw pixels are directly used as inputs with their intensity normalized into $[-1, 1]$. As for evaluation, we adopt the commonly-used Area under the Receiver Operating Characteristic curve (AUROC) and Area under the Precision-Recall curve (AUPR) as threshold-independent metrics [83].

4.1.2 Compared Methods

We extensively compare generative E^3*Outlier* (E^3 Out. (G)), discriminative E^3*Outlier* (E^3 Out. (D)) and contrastive E^3*Outlier* (E^3 Out. (C)) with baselines and existing state-of-the-art DNN based OD methods in literature: **(1)** Convolutional Auto-Encoder (CAE) [84]. CAE is the most prevalent DNN type to deal with image data in many unsupervised learning tasks. Here it serves as an end-to-end baseline, which directly uses CAE's reconstruction loss to perform deep outlier removal. **(2)** CAE+ Isolation Forest (CAE+IF). IF [40] is a classic OD method with wide popularity, so we combine it with CAE as the baseline of two-stage OD approaches. Specifically, CAE+IF feeds CAE's learned representations from its intermediate hidden layer into IF to perform OD. **(3)** SSD+IF. It shares E^3*Outlier*'s SSD part but feeds SSD's learned representations into an IF model to perform OD. SSD+IF serves as a two-stage baseline to compare against the proposed end-to-end E^3*Outlier*. **(4)** Discriminative Reconstruction based Auto-Encoder (DRAE) [9]. DRAE discriminates outliers by thresholding CAE's reconstruction loss with a self-adaptive scheme, which is in turn integrated into the loss function to refine the outlier removal performance. **(5)** Deep Structured Energy based Models (DSEBM) [45]. DSEBM uses an energy based function and score matching technique to estimate the probability that a datum fits the data distribution. **(6)** Robust Deep Auto-Encoder (RDAE) [46]. RDAE synthesizes CAE and RPCA, and it iteratively decomposes unlabeled data into a low-rank part and a sparse error part for outlier removal. **(7)** Deep Auto-encoding Gaussian Mixture Model (DAGMM) [48]. DAGMM embeds a GMM parameter estimation network into CAE, which realizes end-to-end OD by performing representation learning and fitting a GMM simultaneously. **(8)** Multiple-Objective Generative Adversarial Active Learning (MOGAAL) [50]. MOGAAL attempts to generate pseudo outliers that are distributed around given unlabeled data with modified GANs and active learning, so as to transform OD into a supervised binary classification problem. **(9)** Robust Subspace Recovery based AE (RSRAE) [52]. RSRAE is the latest method that improves OD performance by learning to recover the underlying data manifold in a subspace while performing AE's reconstruction. For RSRAE, the reconstruction loss and RSR loss are optimized in a separated manner. In addition to deep solutions, we also include the following baseline solutions for a more comprehensive comparison: **(10)** Two-stage solutions based on pre-trained DNN and the classic OD model. DNN models pre-trained on large-scale generic datasets prove to be an effective tool for feature extraction. Thus, to design a two-stage solution, we use a ResNet50 model pre-trained on ImageNet dataset as feature extractor, and the extracted features are then fed into a classic OD model. IF and the classic Local Outlier Factor (LoF) are exploited here. Due to page limit, implementation details are provided in Section 5 of the supplementary material, available online.

4.2 Experimental Results

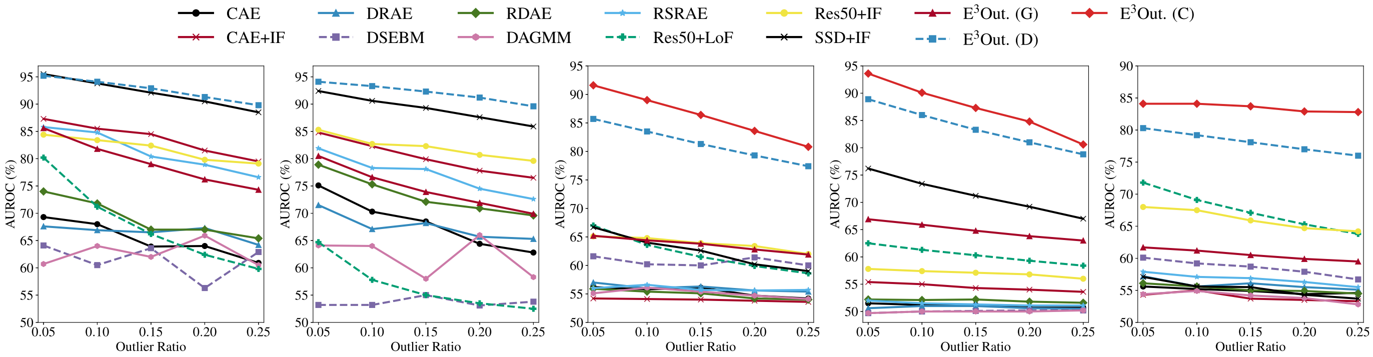
4.2.1 Raw OD Performance Comparison

Due to the space limit, we report numerical results under $\rho = 10\%$ and 20% in Table 1, while the AUROC comparison under different outlier ratios are shown in Fig. 7. From those results, we can obtain the following observations: **(1)** First of all, the proposed E^3*Outlier* framework possesses an evident advantage against existing state-of-the-art DNN based OD methods and baselines in terms of all evaluation metrics. Taking discriminative E^3*Outlier* as an example, it outperforms the best performer among state-of-the-art DNN based OD methods and baselines by a considerable 8%-20% AUROC on different benchmark datasets. In particular, it has realized a performance leap on CIFAR10, SVHN and CIFAR100, which are generally acknowledged to be challenging benchmarks for unsupervised learning tasks like deep outlier removal or clustering. Meanwhile, generative E^3*Outlier* uses the same CAE architecture to achieve consistently better performance than other CAE based deep OD methods CAE/DRAE/DSEBM/DAGMM/RSRAE on CIFAR10/SVHN/CIFAR100 (note that CAE+IF and RDAE are two-stage CAE based OD solutions that contains a classic learning module). On simpler MNIST/Fashion-MNIST,

TABLE 1
OD Performance Comparison (In %) in Terms of AUROC (Area Under ROC Curve, Shorted as ROC), AUPR-In (Area Under PR Curve With Inliers to Be the Positive Class, Shorted as PR-I) and AUPR-Out (Area Under PR Curve With Outliers to Be the Positive Class, Shorted as PR-O). Each Benchmark Shows the Case Where $\rho = 10\%$ and $\rho = 20\%$. Note That Contrastive E^3*Outlier* is Only Used for Benchmark Datasets With Colored Images (CIFAR10/SVHN/CIFAR100), and the Raw Performance Without Score Refinement is Compared for Fairness. The Best Performer is Shown in Bold Font

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
$\rho = 10\%$															
CAE	68.0	92.0	32.9	70.3	94.3	29.3	55.8	91.0	14.4	51.2	90.3	10.6	55.2	91.0	14.5
CAE+IF	85.5	97.8	49.0	82.3	97.2	40.3	54.1	90.2	13.7	55.0	91.4	11.9	54.5	90.7	13.8
DRAE	66.9	93.0	30.5	67.1	93.9	25.5	56.0	90.7	14.7	51.0	90.3	10.5	55.6	90.9	15.0
DSEBM	60.5	91.6	23.0	53.2	88.9	19.7	60.2	92.3	14.7	50.0	90.0	10.1	59.2	92.2	16.2
RDAE	71.8	93.1	35.8	75.3	95.8	31.7	55.4	90.7	14.0	52.1	90.6	10.8	55.6	90.9	15.0
DAGMM	64.0	92.9	26.6	64.0	92.7	30.3	56.1	91.3	15.6	50.0	90.0	19.3	54.9	91.1	14.2
MOGAAL	30.9	78.8	15.2	22.8	74.8	14.8	56.2	91.1	13.6	49.0	89.7	9.8	53.2	90.4	12.6
RSRAE	84.8	97.4	45.4	78.3	96.2	37.0	56.6	91.4	14.0	51.5	90.3	10.6	57.1	91.6	14.1
Res50+LoF	71.2	94.6	26.6	57.8	91.1	16.9	63.6	93.6	17.4	61.3	93.2	14.0	69.1	94.8	22.2
Res50+IF	83.4	97.5	43.3	82.7	97.3	43.8	64.8	93.8	17.9	57.4	92.0	12.8	67.5	94.3	21.0
SSD+IF	93.8	99.2	**68.7**	90.6	98.5	68.6	64.0	93.5	18.3	73.4	95.9	22.0	55.6	91.5	13.0
E^3 Out. (G)	81.8	95.1	50.9	76.6	96.4	32.5	64.4	93.2	18.6	65.9	94.4	15.7	61.2	92.6	17.8
E^3 Out. (D)	**94.1**	**99.3**	67.5	**93.3**	**99.0**	**75.9**	83.5	97.5	43.4	86.0	98.0	36.7	79.2	96.8	33.3
E^3 Out. (C)	-	-	-	-	-	-	**89.0**	**98.5**	**53.2**	**90.1**	**98.5**	**51.3**	**84.1**	**97.8**	**38.0**
$\rho = 20\%$															
CAE	64.0	82.7	40.7	64.4	85.3	36.8	54.7	81.6	25.5	50.7	80.2	20.7	54.4	81.7	25.6
CAE+IF	81.5	93.6	57.2	77.8	92.2	49.0	53.8	80.7	25.3	54.0	82.0	22.4	53.5	80.9	25.1
DRAE	67.3	86.6	42.5	65.7	86.9	36.6	55.6	81.7	26.8	50.6	80.4	20.5	55.5	81.8	27.0
DSEBM	56.3	81.2	32.3	53.1	79.6	31.7	61.4	85.2	27.8	50.2	80.0	20.2	57.9	83.7	27.8
RDAE	67.0	84.2	43.2	70.9	89.2	41.4	54.2	81.0	25.7	51.8	80.9	21.1	54.9	81.5	26.5
DAGMM	65.9	86.4	41.3	66.0	86.7	43.5	54.7	81.8	26.3	50.0	79.9	29.6	53.8	81.5	24.7
MOGAAL	37.8	70.6	28.0	34.0	66.6	28.3	55.7	82.0	25.0	49.6	79.8	19.8	53.1	80.9	24.4
RSRAE	78.9	91.3	53.0	74.5	90.4	46.3	55.6	82.1	25.8	51.1	80.3	21.0	56.3	82.7	25.2
Res50+LoF	62.4	84.9	31.0	53.5	80.3	24.9	59.9	84.9	27.9	59.3	85.0	25.2	65.3	87.5	32.6
Res50+IF	79.8	93.6	52.1	80.7	93.5	55.0	63.4	86.6	30.4	56.8	83.3	24.2	64.7	87.1	32.4
SSD+IF	90.5	97.3	71.0	87.6	95.6	71.4	60.2	85.0	28.3	69.2	89.5	33.7	54.3	82.1	23.4
E^3 Out. (G)	76.2	87.4	55.4	71.9	90.4	41.6	62.8	85.3	30.7	63.8	87.4	27.7	59.9	84.3	29.9
E^3 Out. (D)	**91.3**	**97.6**	**72.3**	**91.2**	**97.1**	**78.9**	79.3	93.1	52.7	81.0	93.4	47.0	77.0	92.4	46.5
E^3 Out. (C)	-	-	-	-	-	-	**83.6**	**94.8**	**59.0**	**84.8**	**94.9**	**57.6**	**82.9**	**95.1**	**53.0**

it is only inferior to the latest RSRAE and still outperforms other CAE based deep OD methods. Such improvement further justifies the effectiveness of introducing richer self-supervision information, and in later sections we show that generative E^3*Outlier* also enables us to flexibly handle other deep OD applications. Next, the proposed contrastive E^3*Outlier* is able to produce a significant performance gain (about 4%-6% AUROC) on colored datasets (CIFAR10/SVHN/CIFAR100) that are relatively difficult for its discriminative and generative counterparts, and it suggests that the potential of E^3*Outlier* can be further exploited by introducing more advanced self-supervised learning paradigms. Thus, the above observations have justified E^3*Outlier* as a highly effective framework for DNN based OD. **(2)** Second, we notice that the baseline OD solutions that combine the classic OD model and features extracted from pre-



(a) MNIST (b) Fashion-MNIST (c) CIFAR10 (d) SVHN (e) CIFAR100

Fig. 7. AUROC comparison of OD methods under different outlier ratios.

TABLE 2
Performance of Discriminative E^3*Outlier* (In %) Before and After Joint Score Refinement (JSR) in Terms of Area Under ROC Curve, PR Curve With Inliers to Be the Positive Class (PR-I) and PR Curve With Outliers to Be the Positive Class (PR-O)

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
$\rho = 10\%$															
E^3 Out.	94.1	99.3	67.5	93.3	**99.0**	75.9	83.5	97.5	43.4	86.0	98.0	36.7	79.2	96.8	33.3
E^3 Out.+JSR	**94.9**	**99.4**	**71.0**	**93.5**	**99.0**	**77.2**	**84.7**	**97.7**	**45.7**	**87.1**	**98.2**	**37.7**	**81.3**	**97.2**	**37.0**
$\rho = 20\%$															
E^3 Out.	91.3	97.6	72.3	91.2	97.1	78.9	79.3	93.1	52.7	81.0	93.4	47.0	77.0	92.4	46.5
E^3 Out.+JSR	**92.9**	**98.1**	**76.3**	**92.1**	**97.4**	**81.9**	**80.3**	**93.5**	**54.5**	**82.0**	**94.2**	**47.9**	**79.1**	**93.1**	**49.9**

Each benchmark shows the case where $\rho = 10\%$ and $\rho = 20\%$ due to the space limit.

trained ResNet50 model (Res50+LoF and Res50+IF) can indeed produce better performance than previous end-to-end OD solutions in many cases, which verifies the importance of the good representation. However, there is still a large performance gap between such two-stage solutions and the proposed deep OD framework, especially discriminative and contrastive E^3*Outlier*. Thus, it further demonstrates the effectiveness of the proposed deep OD framework. **(3)** Third, it is interesting to note that two-stage OD approaches can be more effective than previous end-to-end OD approaches. Specifically, the two-stage counterpart of discriminative E^3*Outlier* SSD+IF achieves fairly close performance to discriminative E^3*Outlier* on relatively simple gray-scale image datasets (MNIST/Fashion-MNIST). Meanwhile, CAE based end-to-end OD solutions (DRAE/DSEBM/DAGMM/RSRAE) cannot constantly outperform their two-stage counterparts (CAE+IF/RDAE), and CAE+IF even performs much better than some CAE based end-to-end solutions on MNIST/Fashion-MNIST. Nevertheless, as shown in Figs. 7a, 7b, 7c, 7d, and 7e, the proposed discriminative E^3*Outlier* almost defeats its two-stage baseline SSD +IF in all experiments, and it suffers from evidently worse performance (i.e., over 10% AUROC loss) on difficult datasets like CIFAR10/SVHN/CIFAR100. **(4)** Among existing end-to-end OD methods, we notice that although recent end-to-end DNN based OD methods (RSRAE) are indeed making progress on relatively simple benchmarks like MNIST and Fashion-MNIST, their performance on difficult datasets like CIFAR10 is still as unsatisfactory as previous counterparts. Besides, MOGAAL performs poorly in almost all cases, which suggests that generating proper pseudo outliers are s till very difficult for deep OD by now.

4.2.2 Score Refinement

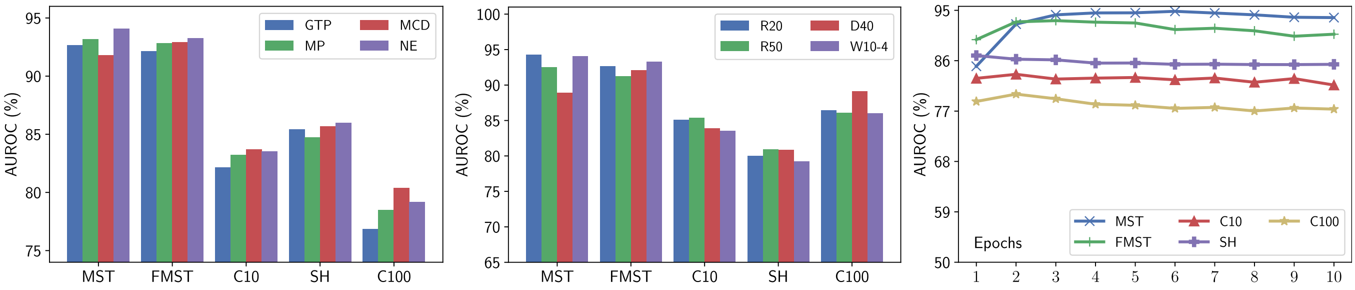
In this section, we validate the effectiveness of score refinement for discriminative E^3*Outlier*. As shown in Table 2, JSR enables consistent performance improvement under different outlier ratios and all evaluation metrics. To show the effect of each score refinement strategy, we further compare the OD performance of five cases in terms of AUROC: Baseline using no score refinement (BAS), using the online re-weighting strategy only (ORW), with the reboot re-weighting strategy only (RRW), using the ensemble strategy only (ENS) and using the joint score refinement (JSR), under $\rho = 10\%$ with default NE score for discriminative E^3*Outlier*. We report the results in Table 3, from which the following facts are drawn: First, when compared with the baseline (BAS), score refinement strategies are able to produce performance gain on all benchmarks by up to 2.1% AUROC gain. The improvement tends to be more tangible on comparatively difficult benchmarks like CIFAR100. Besides, under other outlier ratios, using score refinement also produces stable performance improvement (1% to 2% AUROC) on difficult benchmarks. Second, RRW tends to be slightly better than ORW, while ORW enjoys lower computational cost. Finally, the joint score refinement (JSR) with both reboot re-weighting and ensemble is typically better than a single score refinement strategy, except for the case Fashion-MNIST where JSR performs comparably to other refinement strategies. We also discuss the parameters in score refinement in Section 4 of supplementary material, available online.

4.2.3 Discussion

In this section, we discuss several key factors in E^3*Outlier*. Similarly, we conduct experiments under $\rho = 10\%$ to show the general trends. We investigate the following factors of discriminative E^3 *Outlier:* **(1)** Outlier scores: We compare four different outlier scores for discriminative E^3*Outlier*, i.e., GTP/MP/MCD/NE. As shown by Fig. 8a, uncertainty based scores (MP/MCD/NE) basically prevail over the baseline GTP score, which validates the advantages of exploring network uncertainty as outlierness measure for E^3*Outlier*. Among uncertainty based outlier scores, MCD and NE are prone to outperform the simplest MP. Although MCD achieves the best performance on some benchmarks, it requires multiple forward passes and tends to be less efficient than NE. By contrast, NE consistently outperforms the baseline by a notable margin, and it realizes a good trade-off between performance and efficiency. **(2)** The network architecture of SSD: With other settings fixed, we

TABLE 3
Comparison of Score Refinement Strategies (In %)

Config.	MST	FMST	C10	SH	C100
BAS	94.1	93.3	83.5	86.0	79.2
BAS+ORW	94.4	93.6	84.1	86.7	80.3
BAS+RRW	94.6	93.6	84.4	86.5	80.5
BAS+ENS	94.3	93.4	84.1	86.7	80.7
BAS+JSR	94.9	93.5	84.7	87.1	81.3



(a) Different outlier scores. (b) Different network architectures. (c) Performance with different training epochs.

Fig. 8. Different factors' influence on E^3*Outlier*'s performance under $\rho = 10\%$.

additionally explore ResNet20/ResNet50 [19] and DenseNet40 [85] as the backbone architecture for SSD (shown in Fig. 8b). Despite of some differences, those frequently-used architectures basically perform satisfactorily. Interestingly, we note that a more complex architecture (ResNet50/DenseNet40) tends to be more effective on relatively complex datasets (CIFAR10, SVHN and CIFAR100), but its performance is inferior on simpler datasets. **(3)** Training epochs (see Fig. 8c): We measure the OD performance when the SSD is trained by different epoch numbers to evaluate its impact on self-supervised learning. In general, we notice that the OD performance is inclined to be improved at the initial stage of training (less than $\lceil \frac{250}{K} \rceil$ training epochs) and then reach a plateau. No drastic performance changes are observed as the training epochs continue to increase. **(4)** Pseudo label design. Since the operation set is often constructed by a composite of multiple types of transformations, it is natural to consider a multi-label way to assign pseudo labels. To explore its possibility, we assign each transformed datum with 5 labels based on the performed transformations: Simple rotation label (4 classes in total), translation label ($3 \times 3 = 9$ classes in total), irregular rotation label (8+1=9 classes in total), flip label (2 classes in total) and patch re-arranging label (23+1=24 classes in total). The DNN is equipped with 5 classification heads to predict 5 labels, while the outlier score is computed by averaging the outlier scores yielded by 5 heads. We report the performance of such a multi-label setup in Table 4, and the results suggest that it can yield slightly better performance on most benchmark datasets. Thus, it is possible to explore a more effective design of pseudo labels for E^3*Outlier*. For generative and contrastive E^3*Outlier*, we investigate two major factors: **(1)** Backbone architecture for generative E^3*Outlier*. In fact, one can explore different backbone architecture to implement the generative DNN $\mathcal{G}$ for generative E^3*Outlier*, and we test UNet as an example. As shown in Table 5, the results suggest that UNet is also able to yield fairly satisfactory OD performance. We notice that UNet's performance is better than CAE on Fashion-MNIST/CIFAR10/SVHN, while CAE tends to be better on MNIST. **(2)** Classification loss $\mathcal{L}_{cls}$ for contrastive E^3*Outlier*. It is noted that the loss of classification $\mathcal{L}_{cls}$ when training the DNN model of contrastive E^3*Outlier*, and we also discuss the case where only the contrastive loss $\mathcal{L}_{scl}$ is applied. Interestingly, contrastive E^3*Outlier* without $\mathcal{L}_{cls}$ yields significantly worse performance on CIFAR10/CIFAR100 (77.3%/76.6% AUROC under $\rho = 10\%$), but the performance is better on SVHN (91.7% AUROC under $\rho = 10\%$). The reason is that the performance on "0" class of SVHN suffers from a drastic degradation when classification is performed, as "0" is still a "0" aften a rotation of 90, 180 or 270 degrees. Thus, the classification task is completely invalid in this case.

4.3 E^3*Outlier* Based Video Abnormal Event Detection

4.3.1 Unsupervised Video Abnormal Event Detection

Inspired by E^3*Outlier*'s success with images, it is natural to explore E^3*Outlier* for other type of visual data, e.g., videos. To this end, unsupervised video abnormal event detection (UVAD) [10] is exactly an application of deep OD to videos. UVAD is an emerging task that aims to detect those unusual events that divert from other frequently-encountered routine in completely unlabeled video sequences. As it does not require labeling and enumerating normal video events to construct a training set, UVAD is more challenging than semi-supervised VAD (SSVAD) that has been thoroughly studied [86]. Most existing UVAD solutions approach UVAD by change detection and its variants [10], [87], [88], while the recent work [89] also proposes a different solution that first initializes the detection results based on IF and pre-trained DNNs, and then refines the detection iteratively. However, existing UVAD solutions typically perform unsatisfactorily.

4.3.2 *Design of* E^3Outlier *Based UVAD Solution*

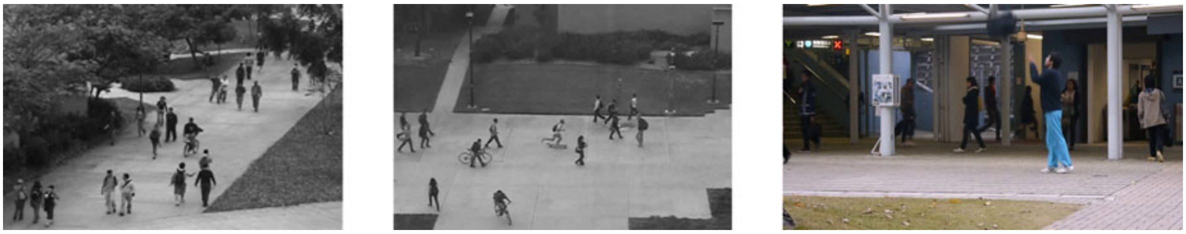
Before we tailor the E^3*Outlier* for UVAD, we notice two important differences between UVAD and previous outlier

TABLE 4
Performance Comparison (In %) of Discriminative E^3*Outlier* With Single-Label (SL) and Multi-Label (ML) Learning

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
E^3 Out. (SL)	94.1	99.3	67.5	**93.3**	**99.0**	**75.9**	83.5	97.5	43.4	86.0	98.0	36.7	79.2	96.8	33.3
E^3 Out. (ML)	**95.4**	**99.5**	**71.1**	92.7	98.9	72.9	**84.1**	**97.6**	**45.1**	**86.9**	**98.1**	**38.5**	**80.0**	**97.0**	**34.9**

TABLE 5
Performance Comparison (In %) of Different DNN Models for Generative E^3*Outlier*

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
CAE	**81.8**	**95.1**	**50.9**	76.6	96.4	32.5	64.4	93.2	18.6	65.9	94.4	15.7	61.2	**92.6**	**17.8**
UNet	79.6	94.2	50.8	**78.5**	**96.7**	**37.7**	**67.3**	**93.6**	**22.3**	**68.7**	**94.9**	**18.6**	**61.3**	**92.6**	16.8



(a) A person riding in the crowd.   (b) A skater and a riding person.   (c) A student throwing his backpack.

Fig. 9. Examples of abnormal events on UCSDped1, UCSDped2, and Avenue datasets (walking pedestrians are normal).

image removal task: First, despite that discriminative and contrastive E^3*Outlier* are shown to be highly effective in detecting outlier images by appearance information (e.g., structure and texture), normal and abnormal video events are often conducted by the same type of subjects in UVAD (For example, humans in Fig. 9). In other words, appearance differences are less important to UVAD. Second, unlike static images, videos are described by both appearance and motion information. As motion is the key to detecting many abnormal events, optical flow maps of video frames are often computed to describe the motion in videos. Therefore, both raw video frames and optical flow maps are supposed to be exploited for providing self-supervision. Due to those differences, we naturally turn to generative E^3*Outlier* to connect both appearance and motion view. Based on generative E^3*Outlier*, the designed UVAD solution is presented below:

First of all, we follow our previous SSVAD work [90] to extract and represent video events: Foreground objects in each video frame are first localized by a series of regions of interest (RoIs). Then, 5 rectangular patches are extracted from current and 4 neighboring frames by the location of each RoI. Afterwards, they are normalized into 32×32 and stacked into a $5 \times 32 \times 32$ spatio-temporal cube (STC) $\mathbf{x} = [p_1; \cdots ; p_5]$, where p_i is a normalized patch ($i = 1, \ldots, 5$). Note that a STC $\mathbf{x}$ serves as the basic representation of a video event, because it not only describes the foreground object but also contains its motion in a time interval. To apply generative E^3*Outlier*, we then design the operation $O(\cdot|y_1)$ and $O(\cdot|y_2)$ like [90]: Given an input STC, $O(\cdot|y_1)$ is defined by $O(\mathbf{x}|y_1) = [p_1; p_2; p_4; p_5]$, which means deleting the middle patch in the STC $\mathbf{x}$. Meanwhile, we devise two types of $O(\cdot|y_2)$: (1) $O(\mathbf{x}|y_2) = p_3$, which suggests fetching the middle patch of $\mathbf{x}$. (2) $O(\mathbf{x}|y_2) = OF(p_3)$, which means transforming p_3 into its corresponding optical flow map. In this way, we actually define a self-supervised learning task that aims to infer p_3 and its optical flow map based on $\mathbf{x}$'s remaining patches p_1, p_2, p_4, p_5. We simple use CAE to carry out this generative task. As described in Section 3.6.1, we can train the models by the objective in Eq. (14) and score each STC by Eq. (15). The scores yielded by two types of $O(\cdot|y_2)$ operations are normalized and then summed to obtain the final score of each STC. The minimum of all STCs' scores on a frame is viewed as the frame score. More details are provided in Section 5 of supplementary material, available online.

4.3.3 Performance Evaluation and Comparison

To evaluate the performance of our UVAD solution, we conduct experiments on three most commonly-used VAD benchmark datasets: UCSDped1 [91], UCSDped2 [91] and Avenue [92]. Following the standard practice in VAD, we compute frame-level AUC [91] as the quantitative performance measure, and compare our method with latest state-of-the-art UVAD approaches: Shuffled change detection (SCD) [10], Unmasking (UM) [87], Multiple Classifier Two Sample Test (MC2ST) [88], and Deep Ordinal Regression (DOR) [89]. The results are displayed in Table 6, and we can discover that the proposed E^3*Outlier* based UVAD solution outperforms existing UVAD solutions by by a 4% to 10% frame-level AUROC, which justifies E^3*Outlier* as a flexible and effective solution to different OD applications. Besides, unlike SCD, UM and MC2ST that require feature extraction based on hand-crafted descriptors, the proposed E^3*Outlier* based solution achieves end-to-end UVAD, while it also leads the other deep UVAD solution DOR by a huge margin.

TABLE 6
Performance Comparison of State-of-The-Art UVAD Methods With Our E^3*Outlier* Based UVAD Solution in Terms of Frame-Level AUC ("-" Indicates Unreported Performance)

	UCSDped1	UCSDped2	Avenue
SCD [10]	59.6%	63.0%	78.3%
UM [87]	68.4%	82.2%	80.6%
MC2ST [88]	71.8%	87.5%	84.4%
DOR [89]	71.7%	83.2%	-
E ^{3} Out.	**79.5%**	**92.6%**	**89.2%**

5 Conclusion

In this paper, we propose a self-supervised deep OD framework named E^3*Outlier*. E^3*Outlier* for the first time leverages discriminative self-supervised learning for deep OD, which facilitates more effective representation learning from raw images. Then we demonstrate inlier priority, a property that lays the foundation for end-to-end OD, by both theory and empirical validations. Afterwards, we illustrate how the network uncertainty of discriminative DNNs can be utilized as a new outlierness measure, and present three specific outlier scores that can outperform the baseline. Then, the joint score refinement that fuses two types of strategies can be used to further boost OD performance. Finally, we demonstrate the applicability of E^3*Outlier* to different learning paradigms and other deep OD applications.

Acknowledgments

S. Wang and Y. Zeng contributed equally to this work.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Sur.*, vol. 41, no. 3, 2009, Art. no. 15.

[2] A. Boukerche, L. Zheng, and O. Alfandi, "Outlier detection: Methods, models, and classification," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–37, 2020.

[3] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Inf. Sci.*, vol. 479, pp. 448–455, 2019.

[4] H. Soleimani and D. J. Miller, "ATD: Anomalous topic discovery in high dimensional discrete data," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2267–2280, Sep. 2016.

[5] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, 2017, pp. 146–157.

[6] J. Mao, T. Wang, C. Jin, and A. Zhou, "Feature grouping-based outlier detection upon streaming trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2696–2709, Dec. 2017.

[7] D. Tax, "One-class classification," PhD thesis, Delft Univ. Technol., 2001.

[8] W. Liu, G. Hua, and J. R. Smith, "Unsupervised one-class learning for automatic outlier removal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3826–3833.

[9] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1511–1519.

[10] A. Del Giorno, J. A. Bagnell, and M. Hebert, "A discriminative framework for anomaly detection in large videos," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 334–349.

[11] S. Wang, Y. Zeng, Q. Liu, C. Zhu, E. Zhu, and J. Yin, "Detecting abnormality without knowing normality: A two-stage approach for unsupervised video abnormal event detection," in *Proc. ACM Multimedia Conf. Multimedia Conf.*, 2018, pp. 636–644.

[12] S. Wang *et al.*, "Robustness can be cheap: A highly efficient approach to discover outliers under high outlier ratios," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5313–5320.

[13] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, 1999, pp. 1150–1157.

[14] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3360–3367.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015, Art. no. 436.

[16] C. C. Aggarwal, "An introduction to outlier analysis," in *Outlier Analysis*. Berlin, Germany: Springer, 2017, pp. 1–34.

[17] H. Wang, M. J. Bah, and M. Hammad, "Progress in outlier detection techniques: A survey," *IEEE Access*, vol. 7, pp. 107 964–108 000, 2019.

[18] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, 2021.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[20] S. Wang *et al.*, "Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 5962–5975.

[21] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *ACM Sigmod Rec.*, vol. 29, no. 2, pp. 427–438, 2000.

[22] E. M. Knox and R. T. Ng, "Algorithms for mining distancebased outliers in large datasets," in *Proc. Int. Conf. Very Large Data Bases*, 1998, pp. 392–403.

[23] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *ACM Sigmod Rec.*, vol. 29, no. 2, pp. 93–104, 2000.

[24] J. Tang, Z. Chen, A. W. C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, 2002, pp. 535–548.

[25] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: Local outlier probabilities," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1649–1652.

[26] T. R. Bandaragoda, K. M. Ting, D. Albrecht, F. T. Liu, and J. R. Wells, "Efficient anomaly detection by isolation using nearest neighbour ensemble," in *Proc. IEEE Int. Conf. Data Mining Workshop*, 2014, pp. 698–705.

[27] G. Pang, K. M. Ting, and D. Albrecht, "LeSiNN: Detecting anomalies by identifying least similar nearest neighbours," in *Proc. IEEE Int. Conf. Data Mining Workshop*, 2015, pp. 623–630.

[28] X. Yang, L. J. Latecki, and D. Pokrajac, "Outlier detection with globally optimal exemplar-based GMM," in *Proc. SIAM Int. Conf. Data Mining*, 2009, pp. 145–154.

[29] X.-M. Tang, R.-X. Yuan, and J. Chen, "Outlier detection in energy disaggregation using subspace learning and gaussian mixture model," *Int. J. Control Automat.*, vol. 8, no. 8, pp. 161–170, 2015.

[30] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier detection with kernel density functions," in *Proc. Int. Workshop Mach. Learn. Data Mining Pattern Recognit.*, 2007, pp. 61–75.

[31] A. P. Boedihardjo, C.-T. Lu, and F. Chen, "Fast adaptive kernel density estimator for data streams," *Knowl. Inf. Syst.*, vol. 42, no. 2, pp. 285–317, 2015.

[32] L. Zhang, J. Lin, and R. Karim, "Adaptive kernel density-based anomaly detection for nonlinear systems," *Knowl.-Based Syst.*, vol. 139, pp. 50–63, 2018.

[33] X. Qin, L. Cao, E. A. Rundensteiner, and S. Madden, "Scalable kernel density estimation-based local outlier detection over large data streams," in *Proc. Annu. Int. Conf. Extending Database Technol.*, 2019, pp. 421–432.

[34] M.-F. Jiang, S.-S. Tseng, and C.-M. Su, "Two-phase clustering process for outliers detection," *Pattern Recognit. Lett.*, vol. 22, no. 6–7, pp. 691–700, 2001.

[35] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognit. Lett.*, vol. 24, no. 9–10, pp. 1641–1650, 2003.

[36] J. Yin and J. Wang, "A model-based approach for text clustering with outlier detection," in *Proc. IEEE 32nd Int. Conf. Data Eng.*, 2016, pp. 625–636.

[37] M. Chenaghlou, M. Moshtaghi, C. Leckie, and M. Salehi, "Online clustering for evolving data streams with online anomaly detection," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, 2018, pp. 508–521.

[38] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Proc. Int. Conf. Artif. Neural Netw.*, 1997, pp. 583–588.

[39] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proc. SIAM Int. Conf. Data Mining*, 2017, pp. 90–98.

[40] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, 2008, pp. 413–422.

[41] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4, pp. 1479–1489, Apr. 2019.

[42] Y. Wang, S. Parthasarathy, and S. Tatikonda, "Locality sensitive outlier detection: A ranking driven approach," in *Proc. IEEE 27th Int. Conf. Data Eng.*, 2011, pp. 410–421.

[43] T. Pevnỳ, "LODA: Lightweight on-line detector of anomalies," *Mach. Learn.*, vol. 102, no. 2, pp. 275–304, 2016.

[44] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*.

[45] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang, "Deep structured energy based models for anomaly detection," *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1100–1109.

[46] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 665–674.

[47] R. Chalapathy, A. K. Menon, and S. Chawla, "Robust, deep and inductive anomaly detection," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2017, pp. 36–51.

[48] B. Zong *et al.*, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Representations*, 2018.

[49] G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2041–2050.

[50] Y. Liu *et al.*, "Generative adversarial active learning for unsupervised outlier detection," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 8, pp. 1517–1528, Aug. 2019.

[51] I. J. Goodfellow *et al.*, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 53–65.

[52] C.-H. Lai, D. Zou, and G. Lerman, "Robust subspace recovery layer for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Representations*, 2020.

[53] N. Tajbakhsh *et al.*, "Surrogate supervision for medical image analysis: Effective deep learning from limited quantities of labeled data," in *Proc. IEEE 16th Int. Symp. Biomed. Imag.*, 2019, pp. 1251–1255.

[54] D. Zhang, J. Han, and Y. Zhang, "Supervision by fusion: Towards unsupervised learning of deep salient object detector," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4048–4056.

[55] X. Ji, J. F. Henriques, and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2018, pp. 9864–9873.

[56] S. Gidaris, S. Praveer and, N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Representations*, 2018.

[57] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9758–9769.

[58] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, " BERT: Pretraining of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[59] R. Santa Cruz, B. Fernando, A. Cherian, and S. Gould, "Visual permutation learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 3100–3114, Dec. 2018.

[60] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: Unsupervised learning using temporal order verification," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 527–544.

[61] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3733–3742.

[62] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2019.

[63] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7047–7058.

[64] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[65] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6402–6413.

[66] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[67] J. Snoek *et al.*, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13 969–13 980.

[68] D. M. Hawkins, *Identification of Outliers*. Berlin, Germany: Springer, vol. 11, 1980.

[69] A. B. L. Larsen, S. K. SΦnderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1558–1566.

[70] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 658–666.

[71] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," 2020, *arXiv:2001.06937*.

[72] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 87.1–87.12.

[73] R. Anand, K. G. Mehrotra, C. K. Mohan, and S. Ranka, "An improved algorithm for neural network classification of imbalanced training sets," *IEEE Trans. Neural Netw.*, vol. 4, no. 6, pp. 962–969, Jun. 1993.

[74] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1321–1330.

[75] T. G. Dietterich, "Ensemble methods in machine learning," in *International Workshop on Multiple Classifier Systems*, Berlin, Germany: Springer, 2000, pp. 1–15.

[76] O. Ronneberger, P. Fischer, and T. Brox,*U-Net: Convolutional Networks for Biomedical Image Segmentation*. Berlin, Germany: Springer, 2015.

[77] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[78] J. Tack, S. Mo, J. Jeong, and J. Shin, "CSI: Novelty detection via contrastive learning on distributionally shifted instances," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 11 839–11 852.

[79] Y. LeCun *et al.*, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[80] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms," 2047, *arXiv:1708.07747*.

[81] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[82] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. Int. Conf. Neural Inf. Process. Syst. Workshop Deep Learn. Unsupervised Feature Learn.*, 2011.

[83] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 233–240.

[84] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. Int. Conf. Artif. Neural Netw.*, 2011, pp. 52–59.

[85] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[86] B. Ramachandra, M. Jones, and R. R. Vatsavai, "A survey of single-scene video anomaly detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2293–2312, May 2020.

[87] R. Tudor Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, "Unmasking the abnormal events in video," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2895–2903.

[88] Y. Liu, C.-L. Li, and B. Póczos, "Classifier two sample test for video anomaly detections," in *Proc. Brit. Mach. Vis. Conf.*, 2018, Art. no. 71.

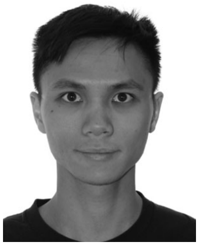
[89] G. Pang, C. Yan, C. Shen, A. V. D. Hengel, and X. Bai, "Self-trained deep ordinal regression for end-to-end video anomaly detection," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12 170–12 179.

[90] G. Yu *et al.*, "Cloze test helps: Effective video anomaly detection via learning to complete video events," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 583–591.

[91] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1975–1981.

[92] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 FPS in matlab," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2720–2727.

Siqi Wang is currently an assistant research Professor in the College of Computer, NUDT. His main research include outlier/anomaly detection and unsupervised learning. His works have been published on leading conferences and journals, such as NeurIPS, AAAI, IJCAI, ACM MM, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, and *IEEE Transactions on Image Processing*. He serves as a PC member and reviewer for top-tier conference like NeurIPS and AAAI and several prestigious journals.

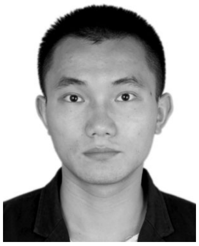


Yijie Zeng received the BSc in computational mathematics from the University of Science and Technology of China, in 2015, and the PhD degree in the School of Electrical and Electronic Engineering from Nanyang Technological University, Singapore, in 2020. His research interests include machine learning, computer vision, and pattern recognition.

Guang Yu received the bachelor's degree in computer science and technology from Sichuan University, Chengdu, China, in 2018. He is currently working toward the PhD degree with the College of Computer, National University of Defense Technology, Changsha, China. His main research interests include anomaly/outlier detection and self-supervised/unsupervised learning.

Zhen Cheng is currently working toward the PhD degree with the National University of Defense Technology (NUDT), China. His current research interests include transfer learning, outlier detection, and deep neural networks.

Xinwang Liu (Senior Member, IEEE) received the PhD degree from the National University of Defense Technology (NUDT), China. He is currently a professor of School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. He has published 60+ peer-reviewed papers, including those in highly regarded journals and conferences such as TPAMI, TKDE, TIP, TNNLS, TMM, TIFS, NeurIPS, ICCV, CVPR, AAAI, IJCAI, etc.



Sihang Zhou received the PhD degree from National University of Defense Technology (NUDT), China. He is currently a lecturer with the College of Intelligence Science and Technology, NUDT. His current research interests include machine learning and medical image analysis. He has published 20+ peer-reviewed papers, including *IEEE Transactions on Image Processing*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Medical Imaging*, *Information Fusion*, *Medical Image Analysis*, AAAI, and MICCAI.

En Zhu received the PhD degree from the National University of Defense Technology (NUDT), China. He is currently a professor with the School of Computer Science, NUDT, China. His main research interests are pattern recognition, image processing, machine vision and machine learning. He has published 60+ peer-reviewed papers, including *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Neural Networks and Learning Systems*, *Pattern Recognition*, AAAI, IJCAI, etc. He was awarded China National Excellence Doctoral Dissertation.

Marius Kloft received the PhD from TU Berlin and UC Berkeley. He is currently a professor of computer science with TU Kaiserslautern and an adjunct faculty member of the University of Southern California. Previously, he was a junior professor with HU Berlin and a joint postdoctoral fellow with the Courant Institute of Mathematical Sciences and Memorial Sloan-Kettering Cancer Center, New York.

Jianping Yin received the PhD degree from National University of Defense Technology (NUDT), China. He is currently the distinguished professor with the Dongguan University of Technology. His research interests include pattern recognition and machine learning. He has published 150+ peer-reviewed papers, including *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Neural Networks and Learning Systems*, PR, AAAI, IJCAI, etc. He was awarded China National Excellence Doctoral Dissertation' Supervisor and National Excellence Teacher. He served on the Technical Program Committees of 30+ international conferences and workshops.

Qing Liao (Member, IEEE) received the PhD degree in computer science and engineering, in 2016 supervised by prof. Qian Zhang from the Department of Computer Science and Engineering of the Hong Kong University of Science and Technology. She is currently an assistant professor with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. Her research interests include artificial intelligence and bioinformatics.