IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 10, 2021, accepted March 9, 2021, date of publication March 12, 2021, date of current version March 25, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3065838*

# Unsupervised Outlier Detection via Transformation Invariant Autoencoder

**ZHEN CHENG, EN ZHU, SIQI WANG, PEI ZHANG, AND WANG LI**
School of Computer, National University of Defense Technology, Changsha 410073, China

Corresponding authors: En Zhu (enzhu@nudt.edu.cn) and Siqi Wang (wangsiqi10c@nudt.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0204301, in part by the National Natural Science Foundation of China under Grant 62006236, in part by the Hunan Provincial Natural Science Foundation under Grant 2020JJ5673, and in part by the National University of Defense Technology (NUDT) Research Project under Grant ZK20-10.

**ABSTRACT** Autoencoder based methods are the majority of deep unsupervised outlier detection methods. However, these methods perform not well on complex image datasets and suffer from the noise introduced by outliers, especially when the outlier ratio is high. In this paper, we propose a framework named Transformation Invariant AutoEncoder (TIAE), which can achieve stable and high performance on unsupervised outlier detection. First, instead of using a conventional autoencoder, we propose a transformation invariant autoencoder to do better representation learning for complex image datasets. Next, to mitigate the negative effect of noise introduced by outliers and stabilize the network training, we select the most confident inliers likely examples in each epoch as the training set by incorporating adaptive self-paced learning in our TIAE framework. Extensive evaluations show that TIAE significantly advances unsupervised outlier detection performance by up to 10% AUROC against other autoencoder based methods on five image datasets.

**INDEX TERMS** Deep Learning, unsupervised outlier detection, autoencoder, transformation invariant autoencoder.

## I. INTRODUCTION

Outlier detection refers to finding patterns in data that do not conform to expected normal behavior [1], [2]. Instances in these patterns are often referred to as outliers, anomalies, faults, defects, novelty, or errors in different contexts of literature. Outlier detection has a wide range of applications in many different domains such as financial fraud detection [3], cybersecurity intrusion detection [4], [5], sensor network fault detection [6]–[8]. Many solutions have been proposed to tackle outlier detection. Labels indicate whether a chosen data example is an inlier or an outlier. Based on the availability of labels, outlier detection can be classified into three categories [1]. (1) Supervised outlier detection (SOD) involves training a supervised binary or multi-class classifier, using labels of both normal and anomalous data instances. (2) Semi-supervised outlier detection (SSOD) uses only normal data to separate outliers. The labels of normal samples are far easier to obtain than outliers, so solutions in this category are more widely adopted. (3) Detecting outliers based on intrinsic of the data instances, unsupervised outlier detection (UOD) handles unlabeled data, including both normal and anomalous data. Note that this classification criterion is not comprehensive. Weakly supervised outlier detection [9]–[11] is another promising area. We focus on unsupervised outlier detection in this paper as most data are unlabeled, and labeling is problematic or cost unacceptable.

Surge in image and video data in this data era has recently inspired many important unsupervised outlier detection applications in the computer vision field, e.g. the refinement of image query results and video abnormal event detection. With the advances in deep neural networks, deep learning-based outlier detection algorithms have become increasingly popular and show huge advantages compared with traditional methods such as principal component analysis (PCA) [12], support vector machine (SVM) [13] and isolation forest (IF) [14] in image/video outlier detection tasks. Autoencoders are the core of most unsupervised outlier detection models [15]–[18]. These models use autoencoder for reconstructing images and assume that inliers and outliers could result in significantly different latent embeddings, and thus differences in the corresponding reconstruction errors can be used to distinguish the two types of samples [19].

However, autoencoders are not good at handling datasets with more complex texture and structure information like SVHN, CIFAR-10. Experiment results from [20] show that

The associate editor coordinating the review of this manuscript and approving it for publication was Alicia Fornés.

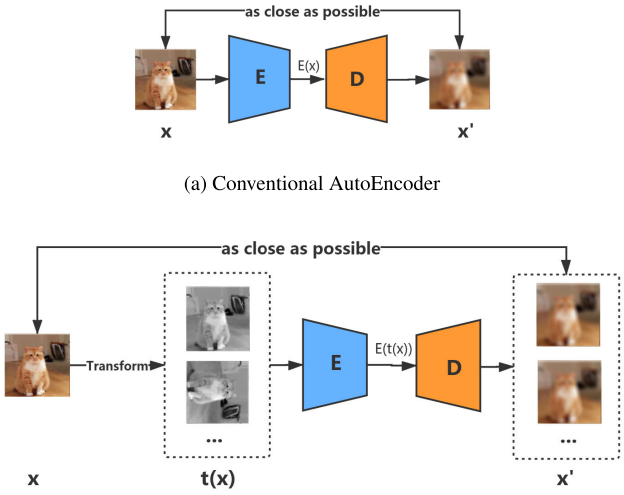
even a sophisticated deep convolutional autoencoder with isolation forest only performs slightly better than random guessing (AUROC = 50%). Applications of autoencoders to other unsupervised tasks (e.g., deep clustering) report similar results [21], [22]. The reason behind this is the use of mean square error (MSE) loss. Autoencoders typically use MSE as a supervise signal, focusing on low-level pixel features rather than high-level semantic features. The assumption of autoencoder based outlier detection may hold when the data is simple. As the data complexity grows, inliers and outliers share more low-level features learned by autoencoder, leading to similar reconstruction error for both inliers and outliers [23], [24], making the model fail to distinguish outliers from inliers.

To address this issue, some scholars attempt to introduce more efficient loss functions rather than the pixel-wise MSE loss. Sabokrou *et al.* [25] introduce adversarial training loss by adding a discriminator after autoencoders to classify whether it is original or reconstructed image. Zaheer *et al.* [26] propose a new adversarial training scheme. Instead of using reconstruction loss, they use a discriminator to distinguish between good and bad quality reconstructions. Akcay *et al.* [27] add another encoder after autoencoders and leverages an extra MSE loss between the two different embeddings. These attempts to alleviate the problems of autoencoders, but the improvements are limited or not suitable for unsupervised outlier detection.

Inspired by recent progress in unsupervised representation learning, especially contrastive learning [28]–[31], we propose Transformation Invariant Autoencoder to learn a better representation of data instead of finding a better loss function for autoencoder training. Fig. 1 provides a brief illustration of the proposed Transformation Invariant Autoencoder. The cat image in Fig. 1 is an unlabeled training example. During the training phase, we first apply transformations based on human priors to the original images and get a set of transformed images (grayscaled and rotated cat images in Fig. 1). Then we feed the transformed images to the TIAE. We optimize the TIAE by minimizing the restoration loss between the restored images and the original images. To alleviate the noise introduced by outliers during training TIAE, we also use the restoration loss to derive self-paced learning weight. During the testing phase, we feed test data to the trained TIAE and expect outliers and inliers leading to different restoration errors. Above all, we can distinguish outliers from inliers by restoration error. We call this pipeline the Transformation Invariant AutoEncoder for unsupervised outlier detection.

To validate the effectiveness of TIAE, we conduct extensive experiments on five popular benchmarks and compare them with other autoencoder based methods. Our experiment results show that TIAE outperforms these methods by a large margin (10% on average on CIFAR-10). We summarize our main contributions of this paper as follows:

1) To learn high-level semantic features instead of low-level features, we propose a simple but effective deep outlier detection framework named Transformation Invariant Autoencoder.
2) We derive an adaptive self-paced learning algorithm without extra hyper-parameters. By using adaptive self-paced learning, our model can mitigate the negative effect of outliers in the process of feature learning.
3) We conduct extensive experiments, and the results validate the effectiveness of our Transformation Invariant Autoencoder framework. The ablation study shows how adaptive self-paced learning affects the proposed unsupervised outlier detection method and provides possible ways to extend existing deep unsupervised outlier detection algorithms.




(a) Conventional AutoEncoder

(b) Transformation Invariant AutoEncoder (TIAE)

**FIGURE 1. An illustrative comparison between conventional autoencoder and transformation invariant autoencoder. In TIAE scenario, the encoder's inputs are transformed images (transformations are based on human priors). The decoder is forced to restore the transformed images to the original images.**

The rest of this paper is organized as follows. Section II outlines the related work of outlier detection. Section III presents the proposed Transformation Invariant AutoEncoder. Section IV shows the experiment results with evaluation. Section V) concludes the paper.

## II. RELATED WORK

Our proposed method falls into the category of deep unsupervised outlier detection and incorporates self-paced learning and representation learning. To facilitate the description of our method, we shall review the existing deep unsupervised outlier detection model, self-paced learning, and representation learning techniques in turn.

### A. DEEP UNSUPERVISED OUTLIER DETECTION

Deep unsupervised outlier detection represents a family of unsupervised outlier detection methods that adopt deep neural networks. Many deep methods have been proposed due to the success of deep learning. In this paper, we focus on unsupervised outlier detection on still image datasets.

Based on the type of network structure, the majority of existing deep unsupervised outlier detection methods can be divided into two categories: Autoencoder-based and self-supervised based methods. Autoencoders are the fundamental unsupervised deep architectures used in unsupervised outlier detection. Recently, self-supervised methods are showing promising results.

Autoencoder based deep unsupervised outlier detection has been extensively studied. These models use autoencoder for reconstructing images and assume that inlier and outlier could lead to significantly different latent embeddings, and thus we can leverage differences in the corresponding reconstruction errors to distinguish the two types of samples. Sakurada *et al.* [19] indicate that the latent embeddings in the hidden layer of autoencoders are distinguishable between inliers and outliers. Zhou and Paffenroth [32] propose a decoupled solution that combines a deep autoencoder with Robust PCA, which decomposes the inputs into a low-rank part from inliers and a sparse part from outliers. Xia *et al.* [33] use deep autoencoder directly and propose a model that estimates inliers by finding a threshold that maximizes the inter-class variance of autoencoder's reconstruction loss. A loss function is designed to encourage the separation of estimated inliers/outliers. Zong *et al.* [23] jointly optimize a deep autoencoder and an estimation network to perform simultaneous representation learning and density estimation for unsupervised outlier detection.

Self-supervised based methods for unsupervised outlier detection shows promising results recently. Golan and El-Yaniv [34] use several image geometric transformations and create a self-labeled dataset for transformation classification pretask, assuming that the pretask model cannot classify transformations of anomalous data properly. Wang *et al.* [20] introduce more self-label methods like patch rearranging and irregular affine transformations to strengthen supervision further.

### B. SELF-PACED LEARNING

Self-paced learning (SPL) [35] simulates the procedure of human learning: from easy to hard. Its core idea is to generally start with learning easier aspects of a task, then gradually consider more complex examples. This strategy of learning is deemed to be more effective. The critical problem is how to define ''easiness''. Depending on the current knowledge we have, the closer the answer we give gets to the correct answer, the easier the example (or problem) should be.

In machine learning problems, the value of loss function often serves as the measure of ''easiness''. A threshold $\lambda$ controls what examples should be used in the current step. Formally, given training examples $\mathcal{D} = \{f(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ and a learning model $f(\cdot)$ with parameters $\boldsymbol{w}$, the original machine learning problem is

$$\min_{\boldsymbol{w}} \quad \sum_{x \in \mathcal{D}} L(f_{\boldsymbol{w}(x_i)}, y_i)\,. \tag{1}$$

Then the objective of self-paced learning is

$$\begin{aligned} &\min_{\boldsymbol{u},\boldsymbol{w}} \sum_{x \in \mathcal{D}} v_i L(f_{\boldsymbol{w}(x_i)}, y_i) + g(\lambda, v_i) \\ &\text{s.t. } v_i \in [0, 1]\,, \end{aligned} \tag{2}$$

where $\boldsymbol{v} = [v_1, v_2, \ldots, v_n]^\top$ are weights of examples and $g(\lambda, v_i)$ is called self-paced regularization term. The $\boldsymbol{w}$ and $\boldsymbol{v}$ can be optimized using Alternative Search Strategy (ASS). Considering the simple hard-weighting self-paced learning where $g(\lambda, v_i) = -\lambda v_i$ and $v_i \in \{0, 1\}$, the new objective is

$$\begin{aligned} &\min_{\boldsymbol{u},\boldsymbol{w}} \sum_{x \in \mathcal{D}} v_i L(f_{\boldsymbol{w}(x_i)}, y_i) - \lambda v_i \\ &\text{s.t. } \ v_i \in \{0, 1\}\,. \end{aligned} \tag{3}$$

Given example weights $\boldsymbol{v}$, the minimization over $\boldsymbol{w}$ is a weighted loss minimization problem. When the model parameter $\boldsymbol{w}$ is fixed, the optimal $v_i$ has a closed-form solution

$$v_i = \begin{cases} 0 & \text{if } L_i < \lambda\,; \\ 1 & \text{otherwise}\,. \end{cases} \tag{4}$$

Self-paced learning has been successfully used in various applications, including co-saliency detection [36], mixture of regressions [37], person re-id [38], Object localization and segmentation in weakly labeled videos [39], category-specific 3D object shape models [40], weakly supervised object detection [41] and deep clustering [42]. Kumar *et al.* [35] demonstrate that self-paced learning algorithm outperforms the state-of-the-art methods for learning a latent structural SVM on four applications: object localization, noun phrase coreference, motif finding, and handwritten digit recognition. Han *et al.* [40] propose to use self-paced learning to alleviate data ambiguity under weak supervision of co-saliency detection, leading to a robust learning manner in complex scenarios. Experiments demonstrate the superiority of the proposed framework beyond the state-of-the-art methods. Huang *et al.* [38] propose a novel video-based person re-id method via self-paced weighting (SPW) and get the state-of-the-art performance on two public datasets. Guo *et al.* [42] incorporate self-paced learning and data augmentation into deep clustering autoencoder, outperforming the state-of-the-art methods on four image datasets.

Self-paced learning algorithms cannot avoid searching the best values for hyper-parameters, threshold $\lambda$, and step size $\delta$ that controls the amount of increasing $\lambda$ at each iteration. However, hyper-parameters are hard to set in the unsupervised scenario. This limits the application of self-paced learning in unsupervised outlier detection. Inspired by Guo *et al.* [42], we propose an adaptive self-paced learning variant that is hyper-parameter free for unsupervised outlier detection.

### C. TRANSFORMATION INVARIANT REPRESENTATION LEARNING

Transformation invariant representation learning is a special case of transformation equivariant representation learning,

which can be defined as

$$E(t(x)) = \rho E(x)\,, \tag{5}$$

where $E(\cdot)$ denotes representation learning model, $t(\cdot)$ denotes transformation and $\rho$ is a coefficient. In transformation invariant representation learning, the coefficient is the identity matrix, which means the representations learned from original samples and transformed samples are the same.

Learning transformation-equivariant representations can trace back to the seminal work on training capsule nets [43]–[45]. Recently, contrastive learning [28]–[31] as a novel unsupervised representation learning method, shows promising results on downstream tasks, which is exactly trying to learn transformation invariant representations.

Tadashi *et al.* propose a similar method to our TIAE to separate the input into transform invariant descriptor and transform parameters, which is efficient for extracting typical spatial subpatterns. Then they demonstrate the imitation of a human hand by a robot hand as an example of a regression-based on spatial subpatterns.

To our best knowledge, our proposed TIAE is the first method to connect transformation invariant representation learning with unsupervised outlier detection.

## III. THE PROPOSED TIAE FRAMEWORK

We first formulate the problem of unsupervised outlier detection in Section III-A. Then we give a brief introduction of transformation in Section III-B. In Section III-C, we introduce the basic model of Transformation Invariant Autoencoder. Furthermore, we incorporate an adaptive self-paced learning algorithm into the basic model in Section III-D.

### A. PROBLEM FORMULATION

We first formulate the problem of unsupervised outlier detection. Considering a data space $\mathcal{X}$ (in this context, the space of images), an unlabeled data collection $X = \{x_i \in \mathbb{R}^{C\times H\times W}\}_{i=1}^{N} \subseteq \mathcal{X}$, where $N$ denotes the total number of samples in $X$, C, H, and W denote the dimensions of image channels, height, and width. $X$ consists of an inlier set $X_{in}$ and an outlier set $X_{out}$, which originate from fundamentally different underlying distributions [46]. Our goal is to build a model $M(\cdot)$ for discriminating whether $x \in X_{in}$ or $x \in X_{out}$.

### B. TRANSFORMATIONS

In this section, we introduce the selection standards of image transformation used in the proposed framework. Transformations are widely used in deep learning literature, such as the data augmentation technique. Deep neural networks are easy to overfit the data, which can be solved by acquiring more training data. Data augmentation is an effective way of expanding training datasets. We consider several common augmentations here. One type of augmentation involves spatial/geometric transformation of data, such as cropping and resizing (with horizontal flipping), rotating, shifting. The other type of augmentation involves appearance transformation, such as color distortion (including color dropping, brightness, contrast, saturation), Gaussian blur, and Sobel filtering [31], [42].

To capture high-level features of training data and achieve effective outlier detection, the transformations we choose need to satisfy some conditions based on human priors and other literature results. First, transformation composition or transformation group is far better than single transformation [31]. Second, transformation should erase specific information, which is the key to differentiate inliers and outliers. The erased information of transformation should be much shared among inliers, and little shared among outliers [47].

Above all, in the TIAE framework, we recommend choosing multiple transformations based on dataset characteristics. After choosing appropriate transformations, we get a set of transformations $\mathcal{T} = \{t_i(\cdot) \mid i = 1, 2, \ldots, T\}$, where $t_i(\cdot)$ denotes the $i$th transformation and $T$ denotes the total number of transformations.

Based on results from [31], we choose color distortion and rotation in our experiments. Details are shown in Section IV-A.

### C. MODEL ARCHITECTURES

In this section, we present the Transformation Invariant Autoencoder (TIAE) framework in detail. TIAE is based on an encoder-decoder framework to capture high-level features by restoring the samples from transformed images. We stack a decoder network $h_{\boldsymbol{u}}(\cdot)$ on the top of the encoder network $f_{\boldsymbol{w}}(\cdot)$ to build an autoencoder.

Given an original image $x$ from the dataset $X$, we derive a transformed image $t_i(x)$ using transformation $t_i(\cdot)$ from the transformation set $\mathcal{T}$. The proposed TIAE takes the transformed images as the inputs and attempts to restore the original image $x$. Mathematically, given $x$, the restored image $\hat{x}$ is formulated as $\hat{x} = h_{\boldsymbol{u}}(f_{\boldsymbol{w}}(t_i(x)))$.

To train the TIAE for effective outlier detection, we use $\ell_2$ loss to measure the distance between restored images and targets (original images). We formulate the restoration loss can as

$$\mathcal{L}_{\text{restoration}} = \frac{1}{N}\frac{1}{T}\sum_{x\in X}\sum_{t\in\mathcal{T}} \|x - h_{\boldsymbol{u}}(f_{\boldsymbol{w}}(t(x)))\|_2^2\,, \tag{6}$$

where $\|\cdot\|_2$ denotes the $\ell_2$ norm. Our objective is to minimize the restoration loss and can be formulated as

$$\min_{\boldsymbol{u},\boldsymbol{w}} \quad \frac{1}{N}\frac{1}{T}\sum_{x\in X}\sum_{t\in\mathcal{T}} \|x - h_{\boldsymbol{u}}(f_{\boldsymbol{w}}(t(x)))\|_2^2\,. \tag{7}$$

As for the testing phase, we design a restoration error based score to distinguish whether a test sample is an inlier or an outlier. We notice that restoration errors vary a lot among different transformations, so transformation-wise normalization is necessary for score calculation. We choose $\ell_1$ loss to measure the distance between the restored image and target in this phase [47]. For each $t_i$ in the transformation set $\mathcal{T}$, we first calculate the expectation $\ell_1$ based restoration error of training data using the trained TIAE model. Then we use this global error to normalize restoration corresponding to each

transformation in the transformation set. Finally, we calculate the expectation of restoration errors across all the transformations, which we use as the outlier score. Let one specific test sample as $x_0$, we formulate the outlier score $S$ as

$$S(x_0) = \frac{1}{T}\sum_{i=1}^{T}\frac{\|x_0 - h_{\boldsymbol{u}}(f_{\boldsymbol{w}}(t_i(x_0)))\|_1}{\mathbb{E}_{x\sim X}\,\|x - h_{\boldsymbol{u}}(f_{\boldsymbol{w}}(t_i(x)))\|_1}\,. \tag{8}$$

### D. INCORPORATING ADAPTIVE SELF-PACED LEARNING

Unsupervised outlier detection is harder than semi-supervised learning because of the existence of outliers in training data. All autoencoder based models suffer from the noise introduced by outliers. With the training process going on, the model can remember enough information for constructing both inliers and outliers well, which leads to poor performance for outlier detection. The TIAE model proposed in Section III-C also has this problem.

To mitigate the negative effect of outliers, we incorporate self-paced learning to select the most confident examples (inliers most likely) gradually. By substituting (7) into (3), we get the new objective

$$\begin{aligned}&\min_{\boldsymbol{u},\boldsymbol{w},\boldsymbol{v}}\ \frac{1}{N}\frac{1}{T}\sum_{i=1}^{N}\sum_{j=1}^{T} v_i\left\|x_i - h_{\boldsymbol{u}}(f_{\boldsymbol{w}}(t_j(x_i)))\right\|_2^2 - \lambda v_i\\ &\text{s.t.}\ \ v_i \in [0,1]\,,\end{aligned} \tag{9}$$

where $\boldsymbol{v} = [v_1, v_2, \ldots, v_n]^\top$ are weights of training examples, and $\lambda$ is the age parameter which controls the number of selected examples.

Typical self-paced learning selects all examples into a training set at the end of model training. However, outliers in our problem are harmful to model performance. We shall prevent the self-paced learning algorithms from selecting outliers, even at the end of the training. Traditional self-paced learning introduces two additional hyper-parameters: the age parameter $\lambda$ for controlling the learning pace and step size $\delta$ for increasing $\lambda$ during training. A typical way is to set $\lambda$ to the median of losses at the beginning, then to increase it by a step size $\delta$ every several iterations. Different from the typical method, we propose to set $\lambda$ according to the statistics of outlier scores during training

$$\lambda = \mu(\boldsymbol{S}^k) + \frac{k}{K}\sigma(\boldsymbol{S}^k)\,, \tag{10}$$

where $\boldsymbol{S}^k$ denotes all scores at the $k$-th iteration, $K$ is the number of maximal iterations, $\mu(\cdot)$ and $\sigma(\cdot)$ are average and standard deviation of scores. As $K$ is determined by the learning model, the $\lambda$ now is adaptive to the losses of examples, not an independent hyper-parameter any more.

## IV. EXPERIMENTS

In this section, we extensively evaluate our approach and compare it with other autoencoder based unsupervised outlier detection methods. We also conduct an ablation study to explore the effect of each part of our TIAE framework. Our experiment codes and results can be verified at https://github.com/wogong/pt-tiae.

### A. EXPERIMENT SETUP

#### 1) UOD PERFORMANCE EVALUATION ON IMAGE BENCHMARKS

We follow the standard procedure from the previous image UOD literature [20], [32], [33], [48] to construct an image set with outliers: Given a standard image benchmark, all images from one class with the same semantic concept (e.g., ''airplane'') are retrieved as inliers, while outliers are randomly sampled from the rest of the classes by an outlier ratio $\rho$. We shift $\rho$ from 5% to 25% by a stage of 5%. The assigned inlier/outlier labels are unknown to UOD methods and only used for evaluation. We use each class of a benchmark as inliers in turn and report the overall UOD performance as the average performance on all classes. Every experiment is repeated five times to report 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 [49]. We evaluate the proposed approach on five public datasets, and briefly introduce them as follows:

- MNIST [50] is a well-known digit recognition dataset, consisting of 70,000 handwritten grayscale digit images with each in size of $28 \times 28$.
- Fashion-MNIST [51] is a more challenging dataset compared to MNIST, consisting of a training set of 70,000 examples. Each example is a $28 \times 28$ grayscale image, associated with a label from 10 classes.
- SVHN [52] is a real-world digit image dataset obtained from house numbers in Google Street View images, consisting of over 600,000 digit images. We use the training set of 73,257 digits in this paper.
- CIFAR-10 [53] is a natural image dataset. The objects in images come from objects in our daily life. It consists of 60,000 color images in size of $32 \times 32$, with 6,000 images per class.
- CIFAR-100 [53] is like the CIFAR-10, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. The 100 classes in the CIFAR-100 are grouped into 20 superclasses. Each image comes with a ''fine'' label (the class to which it belongs) and a ''coarse'' label (the superclass to which it belongs).

For RGB datasets, such as SVHN, CIFAR-10, and CIFAR-100, we use both graying and random rotation operations, together with some widely used standard data augmentations (flipping/mirroring/shifting). For grayscale datasets like MNIST and Fashion-MNIST, we only use rotation transformation without any data augmentation.

#### 2) IMPLEMENTATION DETAILS

Similar to previous image restoration method [56] and other autoencoder based outlier detection methods [47], [57], [58],

**TABLE 1.** AUROC/AUPR-in/AUPR-out (%) for UOD methods. The best performance is in bold.

| Dataset | $\rho$ | CAE [53] | DRAE [32] | RDAE [31] | DAGMM [22] | MemAE [23] | RSRAE [54] | Ours |
|---|---|---|---|---|---|---|---|---|
| AUROC (%) | | | | | | | | |
| MNIST | 10% | 67.19 ± 0.90 | 67.92 ± 1.43 | 71.76 ± 0.99 | 63.23 ± 2.15 | 69.13 ± 0.15 | 82.08 ± 3.09 | **85.18 ± 0.81** |
| | 20% | 63.21 ± 1.24 | 64.74 ± 2.62 | 67.00 ± 0.69 | 65.88 ± 2.89 | 65.56 ± 0.26 | **80.37 ± 2.04** | 79.41 ± 0.89 |
| F-MNIST | 10% | 69.04 ± 0.35 | 65.96 ± 1.72 | 75.34 ± 1.19 | 70.36 ± 3.18 | 72.10 ± 0.31 | 75.19 ± 2.78 | **86.77 ± 0.55** |
| | 20% | 65.88 ± 1.07 | 63.63 ± 1.07 | 70.94 ± 1.13 | 66.00 ± 4.96 | 67.88 ± 0.09 | 72.15 ± 2.55 | **81.23 ± 1.15** |
| SVHN | 10% | 51.18 ± 0.44 | 51.10 ± 0.27 | 51.44 ± 0.27 | 50.09 ± 0.29 | 58.29 ± 0.12 | 51.19 ± 0.44 | **68.78 ± 0.88** |
| | 20% | 50.97 ± 0.26 | 50.88 ± 0.18 | 51.67 ± 0.22 | 49.99 ± 0.24 | 56.55 ± 0.03 | 51.51 ± 0.40 | **67.98 ± 0.07** |
| CIFAR-10 | 10% | 55.86 ± 1.06 | 56.09 ± 0.21 | 54.02 ± 1.63 | 54.05 ± 0.92 | 55.69 ± 0.17 | 54.77 ± 1.84 | **71.18 ± 1.44** |
| | 20% | 54.78 ± 0.60 | 55.60 ± 0.09 | 52.46 ± 1.54 | 54.68 ± 0.70 | 54.68 ± 0.09 | 53.62 ± 1.98 | **66.61 ± 2.51** |
| CIFAR-100 | 10% | 55.78 ± 0.70 | 55.66 ± 0.49 | 53.62 ± 0.40 | 54.19 ± 1.37 | 55.03 ± 0.04 | 53.56 ± 0.48 | **65.01 ± 1.70** |
| | 20% | 54.95 ± 0.68 | 55.25 ± 0.30 | 52.86 ± 1.43 | 53.80 ± 0.61 | 54.30 ± 0.11 | 53.54 ± 1.13 | **62.55 ± 0.36** |
| AUPR-IN (%) | | | | | | | | |
| MNIST | 10% | 91.79 ± 0.29 | 92.81 ± 0.37 | 89.11 ± 1.28 | 92.89 ± 0.77 | 93.13 ± 0.00 | **96.55 ± 0.75** | 95.95 ± 0.16 |
| | 20% | 82.39 ± 0.56 | 84.92 ± 1.00 | 75.56 ± 2.58 | 86.42 ± 1.04 | 84.22 ± 0.11 | **92.15 ± 1.02** | 88.63 ± 0.64 |
| F-MNIST | 10% | 94.01 ± 0.14 | 93.51 ± 0.47 | 83.25 ± 1.22 | 92.70 ± 2.92 | 95.54 ± 0.08 | 95.29 ± 0.52 | **98.04 ± 0.11** |
| | 20% | 86.21 ± 0.60 | 85.87 ± 0.52 | 72.83 ± 2.64 | 86.66 ± 2.70 | 88.64 ± 0.07 | 89.74 ± 0.93 | **93.69 ± 0.55** |
| SVHN | 10% | 90.29 ± 0.09 | 90.48 ± 0.11 | 90.34 ± 0.08 | 90.00 ± 0.14 | 92.48 ± 0.04 | 90.37 ± 0.08 | **94.66 ± 0.23** |
| | 20% | 80.31 ± 0.13 | 80.43 ± 0.11 | 80.78 ± 0.17 | 79.90 ± 0.09 | 83.64 ± 0.03 | 80.47 ± 0.16 | **88.32 ± 0.07** |
| CIFAR-10 | 10% | 91.08 ± 0.30 | 90.81 ± 0.11 | 90.59 ± 0.51 | 91.28 ± 0.57 | 90.73 ± 0.04 | 90.89 ± 0.42 | **94.66 ± 0.28** |
| | 20% | 81.63 ± 0.24 | 81.68 ± 0.07 | 80.74 ± 0.82 | 81.76 ± 0.37 | 81.26 ± 0.03 | 81.05 ± 0.73 | **86.86 ± 0.99** |
| CIFAR-100 | 10% | 94.89 ± 0.04 | 90.93 ± 0.15 | 90.47 ± 0.13 | 91.12 ± 0.23 | 90.95 ± 0.01 | 90.72 ± 0.21 | **93.42 ± 0.35** |
| | 20% | 81.88 ± 0.34 | 81.70 ± 0.10 | 80.69 ± 0.78 | 81.45 ± 0.42 | 81.60 ± 0.06 | 81.39 ± 0.71 | **85.42 ± 0.01** |
| AUPR-OUT (%) | | | | | | | | |
| MNIST | 10% | 31.23 ± 1.01 | 33.03 ± 0.94 | 35.81 ± 0.77 | 20.63 ± 4.42 | 26.80 ± 0.02 | 46.84 ± 2.72 | **59.94 ± 2.27** |
| | 20% | 40.02 ± 1.03 | 39.89 ± 3.64 | 43.23 ± 0.75 | 33.48 ± 5.21 | 38.35 ± 0.29 | 55.62 ± 3.02 | **62.92 ± 0.29** |
| F-MNIST | 10% | 28.52 ± 0.66 | 24.16 ± 2.10 | 31.72 ± 1.39 | 35.44 ± 2.61 | 23.13 ± 0.20 | 33.91 ± 4.09 | **54.59 ± 0.73** |
| | 20% | 38.55 ± 1.57 | 34.21 ± 0.91 | 41.40 ± 0.88 | 42.04 ± 4.45 | 34.91 ± 0.08 | 40.41 ± 4.62 | **58.24 ± 1.26** |
| SVHN | 10% | 10.57 ± 0.17 | 10.54 ± 0.14 | 10.45 ± 0.14 | 19.79 ± 2.16 | 12.61 ± 0.08 | 10.45 ± 0.15 | **20.42 ± 0.97** |
| | 20% | 20.81 ± 0.04 | 20.70 ± 0.13 | 21.09 ± 0.05 | 30.74 ± 1.66 | 23.41 ± 0.01 | 21.29 ± 0.26 | **35.51 ± 0.52** |
| CIFAR-10 | 10% | 14.16 ± 0.74 | 14.61 ± 0.16 | 13.03 ± 0.57 | 13.69 ± 0.28 | 13.98 ± 0.06 | 13.04 ± 0.92 | **24.89 ± 2.32** |
| | 20% | 25.52 ± 0.48 | 26.66 ± 0.09 | 23.38 ± 1.22 | 25.62 ± 1.02 | 25.66 ± 0.12 | 24.53 ± 1.45 | **36.81 ± 1.92** |
| CIFAR-100 | 10% | 14.66 ± 0.35 | 14.75 ± 0.14 | 13.89 ± 0.43 | 13.62 ± 0.67 | 14.15 ± 0.09 | 12.36 ± 0.16 | **20.12 ± 2.22** |
| | 20% | 25.71 ± 0.45 | 26.60 ± 0.38 | 24.33 ± 0.66 | 24.26 ± 0.51 | 25.08 ± 0.08 | 23.70 ± 0.84 | **30.78 ± 1.97** |

Our TIAE adopt the U-Net [59] like structures. We use four blocks for the encoder and four blocks for the decoder. Each block has a max-pooling or an upsampling operation, following two $3 \times 3$ convolutional layers. We use upsampling instead of deconvolution for efficiency. The ability to recover image details for upsampling is limited, so we add skip-connection operations to pass input details from top layers to bottom layers, which improves the network's performance of image restoration.

Since we augment original data by $T$ times, we train TIAE for $800/T$ epochs with a batch size of 32. We use Stochastic Gradient Descent (SGD) optimizer with default settings in PyTorch for all datasets. We set the initial learning rate to 0.1 and drop the learning rate by half every $80/T$ epochs. We delay the incorporating of self-paced learning by ten epochs to get a better initial example weights.

As introduced in Section III-B, we choose color distortion and rotation in the experiments:

- *Color Distortion:* average each pixel value along the channel dimension of images.
- *Rotation:* rotate the original images by one of {0°, 90°, 180°, 270° }.

### 3) COMPARED METHODS

We compare our approach with existing state-of-the-art autoencoder based UOD methods: (1) Convolutional AutoEncoder (CAE) [54], CAE serves as a baseline for autoencoder based UOD methods. (2) Discriminative Reconstruction based AutoEncoder (DRAE) [33]. (3) Robust Deep AutoEncoder (RDAE) [32]. (4) Deep Autoencoding Gaussian Mixture Model (DAGMM) [23]. (5) Memory-augmented deep AutoEncoder (MemAE) [24]. (6) Robust Subspace Recovery based AutoEncoder (RSRAE) [55].

For MemAE, we use exactly the same autoencoder structure reported in the original paper. For CAE, DRAE, RDAE, DAGMM, and RSRAE, we use the same CAE architecture

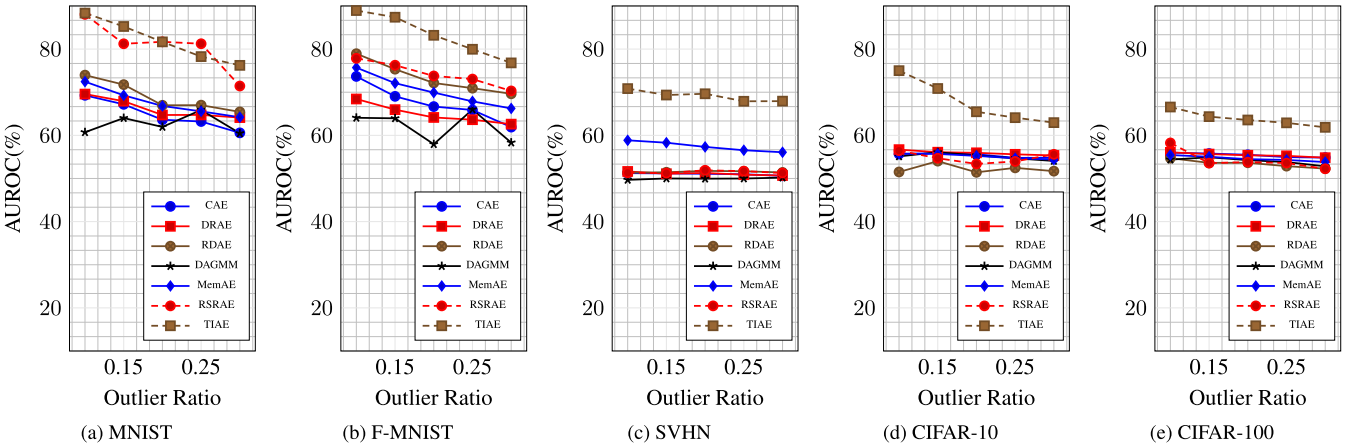



**FIGURE 2.** UOD performance (AUROC) comparison with varying $\rho$ from 5% to 25%.

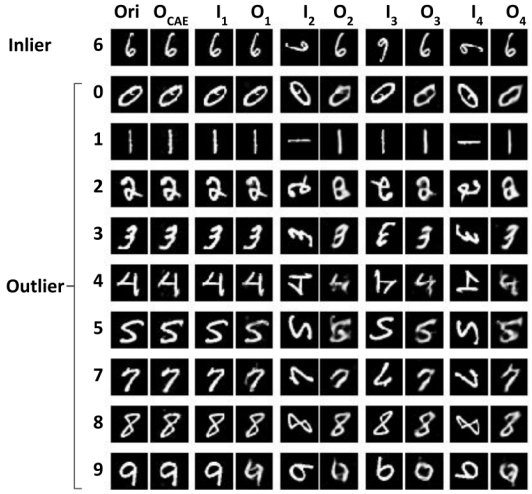



**FIGURE 3.** Visualization analysis comparing with CAE on MNIST. "Ori", "I" and "O" represent original images, transformed inputs, and outputs, respectively. Cases with outputs similar to "Ori" are considered inliers, otherwise outliers. All visualization results are based on the number "6" as inliers.

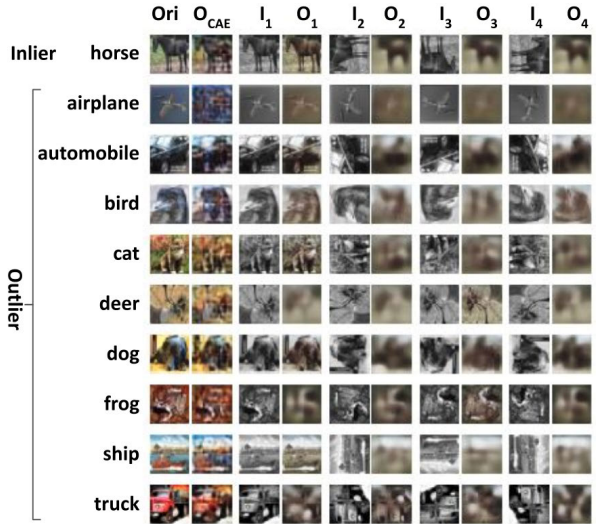



**FIGURE 4.** Visualization analysis comparing with CAE on CIFAR-10. "Ori", "I" and "O" represent original images, transformed inputs and outputs, respectively. Cases with outputs similar to "Ori" are considered as inliers, otherwise outliers. All visualization results are based on the class "horse" as inliers.

from [34] with a 4-layer encoder and 4-layer decoder. We do not use more complex CAE (e.g., CAE using skip connection or more layers) since they usually lower outliers' reconstruction error but do not contribute to CAE's UOD performance [20]. Our ablation study in Section IV-D also verifies this.

### B. UOD PERFORMANCE COMPARISON AND DISCUSSION

We report the numerical results on each benchmark under $\rho = 10\%$ and 20% in Table 1, and UOD performance by AUROC under $\rho$ from 5% to 25% is shown in Fig. 2. AUPR-in and AUPR-out in Table 1 denote the AUPR calculated when inliers and outliers are used as positive classes, respectively. To compare the performance for each individual image class, we also report the AUROC results for each class of the five benchmark datasets in Table 2. From these results, we have the following observations:

- On four of all five involved datasets with varying $\rho$ from 5% to 25%, experiment results present that the proposed TIAE framework outperforms existing state-of-the-art autoencoder based UOD methods. On MNIST, TIAE achieves comparable performance with RSRAE (<1% AUROC gap).
- As Table 2 shows, for each individual image class, we also obtain competitive performances, showing the effectiveness of TIAE for unsupervised outlier detection.
- For complex datasets like SVHN, CIFAR-10, and CIFAR-100, TIAE performances much better than compared methods (∼10% AUROC gain). As introduced in Section I, conventional autoencoder based methods are not good at handling datasets with more complex texture and structure information. Our proposed TIAE can handle more complex datasets compared with other autoencoder based methods and achieve a large performance gain.

**TABLE 2.** AUROC (%) for UOD methods when $\rho = 0.1$. The best performance is in bold.

| Dataset | Class name | CAE | DRAE | RDAE | DAGMM | MemAE | RSRAE | Ours |
|---|---|---|---|---|---|---|---|---|
| MNIST | 0 | 56.61 ± 3.80 | 52.92 ± 8.61 | 62.51 ± 5.83 | 42.85 ± 6.39 | 61.50 ± 0.91 | 85.18 ± 4.12 | 78.53 ± 3.81 |
| | 1 | 99.22 ± 0.17 | 96.99 ± 0.75 | 99.19 ± 0.08 | 84.89 ± 1.88 | 97.80 ± 0.24 | 97.50 ± 1.32 | 99.50 ± 0.09 |
| | 2 | 55.76 ± 2.97 | 54.70 ± 6.80 | 61.12 ± 1.81 | 60.34 ± 2.12 | 57.23 ± 0.72 | 74.10 ± 3.60 | 80.82 ± 2.38 |
| | 3 | 62.46 ± 3.68 | 62.45 ± 5.03 | 65.91 ± 1.84 | 61.38 ± 3.38 | 62.66 ± 0.95 | 60.84 ± 7.65 | 86.60 ± 2.54 |
| | 4 | 71.25 ± 4.07 | 75.59 ± 3.36 | 76.05 ± 1.20 | 59.13 ± 4.29 | 71.91 ± 0.01 | 82.68 ± 3.85 | 87.23 ± 0.91 |
| | 5 | 61.71 ± 1.57 | 56.48 ± 7.36 | 65.11 ± 2.43 | 55.44 ± 1.79 | 59.12 ± 0.25 | 78.30 ± 3.44 | 81.40 ± 0.52 |
| | 6 | 67.09 ± 3.88 | 68.86 ± 6.94 | 75.03 ± 1.94 | 64.90 ± 7.18 | 72.21 ± 0.76 | 91.31 ± 2.81 | 88.76 ± 0.38 |
| | 7 | 80.04 ± 2.51 | 84.39 ± 1.86 | 85.64 ± 1.11 | 74.55 ± 5.90 | 80.56 ± 1.33 | 90.57 ± 2.01 | 90.04 ± 1.00 |
| | 8 | 46.80 ± 2.82 | 53.08 ± 1.10 | 51.38 ± 2.30 | 59.72 ± 6.97 | 54.12 ± 0.00 | 74.30 ± 1.82 | 72.92 ± 1.38 |
| | 9 | 70.91 ± 2.21 | 73.77 ± 4.09 | 75.71 ± 2.60 | 69.08 ± 2.79 | 74.16 ± 0.27 | 85.99 ± 4.14 | 86.04 ± 0.69 |
| | *average* | 67.19 ± 0.90 | 67.92 ± 1.43 | 71.76 ± 0.99 | 63.23 ± 2.15 | 69.13 ± 0.15 | 82.08 ± 3.09 | **85.18 ± 0.81** |
| F-MNIST | t-shirt | 58.73 ± 6.16 | 59.47 ± 4.55 | 70.56 ± 4.18 | 72.98 ± 8.47 | 71.30 ± 0.63 | 81.09 ± 2.64 | 83.77 ± 1.72 |
| | trouser | 96.12 ± 0.34 | 79.04 ± 4.95 | 96.94 ± 0.36 | 77.49 ± 8.31 | 95.21 ± 0.05 | 81.83 ± 5.37 | 97.68 ± 0.47 |
| | pullover | 60.77 ± 5.44 | 57.75 ± 4.54 | 68.79 ± 4.73 | 55.61 ± 2.66 | 72.71 ± 0.55 | 70.36 ± 3.11 | 82.30 ± 1.69 |
| | dress | 74.85 ± 1.14 | 72.38 ± 5.68 | 79.12 ± 3.22 | 71.48 ± 5.28 | 75.96 ± 0.40 | 83.01 ± 3.96 | 88.60 ± 1.88 |
| | coat | 64.92 ± 5.45 | 61.88 ± 3.41 | 73.23 ± 1.59 | 50.69 ± 7.61 | 74.12 ± 1.23 | 69.03 ± 8.06 | 84.70 ± 1.20 |
| | sandal | 72.19 ± 4.76 | 72.94 ± 2.86 | 73.64 ± 1.33 | 90.52 ± 1.37 | 60.29 ± 1.73 | 77.20 ± 7.96 | 89.89 ± 0.43 |
| | shirt | 48.38 ± 2.15 | 54.36 ± 2.41 | 66.34 ± 3.00 | 54.03 ± 2.18 | 63.95 ± 0.15 | 62.81 ± 7.53 | 69.48 ± 1.88 |
| | sneaker | 89.35 ± 0.79 | 86.26 ± 4.30 | 91.15 ± 1.14 | 82.36 ± 5.20 | 82.08 ± 0.58 | 90.50 ± 2.51 | 97.00 ± 0.15 |
| | bag | 45.22 ± 2.97 | 44.26 ± 1.38 | 51.25 ± 2.93 | 60.09 ± 0.64 | 50.66 ± 0.84 | 61.60 ± 9.05 | 78.89 ± 1.02 |
| | ankle-boot | 79.92 ± 4.57 | 71.29 ± 3.47 | 82.35 ± 5.15 | 88.36 ± 1.93 | 74.71 ± 0.48 | 74.42 ± 5.07 | 95.39 ± 0.13 |
| | *average* | 69.04 ± 0.35 | 65.96 ± 1.72 | 75.34 ± 1.19 | 70.36 ± 3.18 | 72.10 ± 0.31 | 75.19 ± 2.78 | **86.77 ± 0.55** |
| SVHN | 0 | 50.29 ± 0.54 | 49.42 ± 1.12 | 50.07 ± 0.98 | 49.15 ± 0.69 | 58.27 ± 0.29 | 50.99 ± 1.20 | 52.95 ± 0.80 |
| | 1 | 57.10 ± 0.82 | 56.13 ± 1.29 | 57.16 ± 2.51 | 49.54 ± 1.25 | 63.83 ± 0.12 | 53.10 ± 0.61 | 63.95 ± 0.34 |
| | 2 | 50.93 ± 1.01 | 52.13 ± 0.91 | 51.21 ± 1.22 | 48.41 ± 0.49 | 58.09 ± 0.09 | 50.91 ± 1.23 | 75.32 ± 1.09 |
| | 3 | 49.72 ± 0.62 | 50.57 ± 0.57 | 49.73 ± 0.62 | 50.94 ± 0.30 | 55.91 ± 0.44 | 52.12 ± 0.97 | 65.73 ± 1.19 |
| | 4 | 52.19 ± 0.39 | 53.44 ± 1.59 | 53.99 ± 1.23 | 50.61 ± 1.02 | 60.28 ± 0.33 | 51.74 ± 1.10 | 78.59 ± 0.60 |
| | 5 | 49.01 ± 0.78 | 49.87 ± 1.08 | 49.83 ± 1.64 | 51.93 ± 0.75 | 54.89 ± 0.10 | 49.75 ± 0.73 | 68.26 ± 1.85 |
| | 6 | 49.43 ± 0.73 | 49.77 ± 0.84 | 49.80 ± 0.71 | 49.90 ± 0.89 | 56.00 ± 0.25 | 50.06 ± 1.68 | 69.11 ± 1.12 |
| | 7 | 52.75 ± 1.07 | 50.46 ± 1.03 | 53.69 ± 0.80 | 48.36 ± 0.84 | 60.79 ± 0.37 | 52.33 ± 1.82 | 74.93 ± 1.11 |
| | 8 | 50.15 ± 0.73 | 49.33 ± 1.15 | 49.27 ± 0.47 | 51.08 ± 0.84 | 56.55 ± 0.33 | 50.91 ± 0.83 | 66.60 ± 1.56 |
| | 9 | 50.25 ± 0.76 | 49.90 ± 1.19 | 49.67 ± 0.95 | 50.98 ± 0.80 | 58.33 ± 0.24 | 50.00 ± 2.18 | 72.39 ± 0.77 |
| | *average* | 51.18 ± 0.44 | 51.10 ± 0.27 | 51.44 ± 0.27 | 50.09 ± 0.29 | 58.29 ± 0.12 | 51.19 ± 0.44 | **68.78 ± 0.88** |
| CIFAR-10 | airplane | 69.75 ± 2.67 | 70.77 ± 1.34 | 68.11 ± 5.28 | 46.67 ± 0.73 | 65.87 ± 0.50 | 63.81 ± 6.62 | 70.23 ± 1.17 |
| | automobile | 37.59 ± 2.51 | 37.88 ± 1.39 | 41.08 ± 5.31 | 55.72 ± 5.87 | 36.31 ± 0.23 | 42.91 ± 7.25 | 68.13 ± 2.18 |
| | bird | 61.38 ± 1.26 | 65.68 ± 0.75 | 57.47 ± 5.51 | 47.46 ± 2.35 | 68.94 ± 0.29 | 60.09 ± 3.77 | 69.26 ± 1.80 |
| | cat | 58.75 ± 1.77 | 59.42 ± 0.64 | 56.05 ± 4.99 | 51.86 ± 1.43 | 53.91 ± 0.36 | 51.28 ± 3.02 | 61.70 ± 2.09 |
| | deer | 61.27 ± 3.29 | 61.72 ± 0.68 | 69.28 ± 1.70 | 51.15 ± 3.18 | 67.29 ± 0.36 | 65.64 ± 2.12 | 73.55 ± 1.70 |
| | dog | 58.07 ± 3.43 | 60.90 ± 0.33 | 49.34 ± 3.15 | 55.55 ± 4.51 | 53.83 ± 0.36 | 45.64 ± 1.43 | 71.75 ± 2.38 |
| | frog | 54.64 ± 5.24 | 43.28 ± 1.72 | 55.96 ± 0.13 | 61.74 ± 3.99 | 56.33 ± 0.38 | 64.29 ± 2.84 | 74.12 ± 2.18 |
| | horse | 47.88 ± 3.04 | 51.05 ± 1.02 | 49.93 ± 4.14 | 59.02 ± 2.78 | 49.83 ± 0.39 | 44.17 ± 1.36 | 74.54 ± 2.35 |
| | ship | 70.15 ± 3.78 | 72.01 ± 2.21 | 55.29 ± 8.68 | 46.10 ± 4.05 | 68.81 ± 0.59 | 64.14 ± 5.37 | 83.25 ± 1.86 |
| | truck | 39.13 ± 1.21 | 38.19 ± 1.53 | 37.65 ± 5.74 | 65.23 ± 4.25 | 35.78 ± 0.51 | 45.71 ± 7.22 | 65.22 ± 1.90 |
| | *average* | 55.86 ± 1.06 | 56.09 ± 0.21 | 54.02 ± 1.63 | 54.05 ± 0.92 | 55.69 ± 0.17 | 54.77 ± 1.84 | **71.18 ± 1.44** |
| CIFAR-100 | aquatic mammals | 63.84 ± 5.21 | 65.81 ± 1.47 | 60.59 ± 2.65 | 49.25 ± 2.73 | 65.39 ± 0.62 | 56.70 ± 4.05 | 63.96 ± 2.48 |
| | fish | 62.92 ± 2.08 | 64.90 ± 0.36 | 53.09 ± 3.30 | 47.98 ± 3.94 | 64.07 ± 0.28 | 54.21 ± 6.70 | 57.03 ± 3.36 |
| | flowers | 38.75 ± 6.75 | 34.45 ± 1.13 | 31.95 ± 3.03 | 65.44 ± 4.63 | 34.86 ± 0.64 | 60.38 ± 5.09 | 44.43 ± 2.12 |
| | food containers | 64.17 ± 1.36 | 62.81 ± 1.03 | 55.59 ± 4.44 | 45.40 ± 3.10 | 62.54 ± 0.50 | 59.39 ± 1.68 | 69.30 ± 1.47 |
| | fruit and vegetables | 48.70 ± 3.59 | 53.08 ± 1.66 | 40.60 ± 3.20 | 62.96 ± 4.99 | 50.22 ± 0.80 | 54.99 ± 4.61 | 58.61 ± 1.35 |
| | household electrical devices | 55.45 ± 1.69 | 54.08 ± 2.12 | 47.22 ± 4.72 | 46.60 ± 4.81 | 48.89 ± 0.36 | 49.15 ± 6.31 | 57.92 ± 2.21 |
| | household furniture | 62.82 ± 1.73 | 62.16 ± 1.27 | 54.17 ± 1.79 | 53.65 ± 4.32 | 57.81 ± 0.60 | 59.25 ± 3.97 | 69.46 ± 5.14 |
| | insects | 49.57 ± 2.66 | 46.89 ± 1.21 | 46.46 ± 2.15 | 51.70 ± 2.73 | 50.36 ± 0.35 | 52.91 ± 4.04 | 56.20 ± 1.42 |
| | large carnivores | 53.78 ± 5.59 | 51.59 ± 1.79 | 60.97 ± 1.84 | 59.07 ± 5.22 | 54.71 ± 0.56 | 53.08 ± 6.18 | 69.07 ± 1.82 |
| | large man-made outdoor things | 64.56 ± 3.68 | 65.94 ± 1.84 | 60.55 ± 5.40 | 57.65 ± 7.81 | 62.17 ± 0.15 | 54.30 ± 8.83 | 77.59 ± 2.74 |
| | large natural outdoor scenes | 79.83 ± 1.48 | 82.76 ± 1.04 | 75.45 ± 3.74 | 53.43 ± 8.55 | 79.11 ± 0.51 | 60.22 ± 8.13 | 79.10 ± 1.33 |
| | large omnivores and herbivores | 55.49 ± 1.95 | 55.24 ± 1.45 | 58.60 ± 1.86 | 58.95 ± 2.20 | 52.94 ± 0.72 | 51.38 ± 4.68 | 67.78 ± 1.71 |
| | medium-sized mammals | 57.01 ± 5.64 | 57.33 ± 1.35 | 60.10 ± 4.39 | 61.17 ± 4.60 | 54.94 ± 0.50 | 54.96 ± 4.57 | 68.49 ± 1.10 |
| | non-insect invertebrates | 50.29 ± 1.36 | 51.16 ± 1.06 | 53.54 ± 2.80 | 46.66 ± 2.16 | 54.72 ± 0.44 | 58.46 ± 2.30 | 57.97 ± 0.74 |
| | people | 47.52 ± 3.82 | 47.83 ± 1.94 | 45.11 ± 1.09 | 54.28 ± 3.18 | 42.45 ± 0.31 | 42.03 ± 0.83 | 61.55 ± 4.04 |
| | reptiles | 54.23 ± 1.44 | 53.90 ± 0.72 | 57.22 ± 0.98 | 51.70 ± 2.62 | 54.16 ± 0.37 | 56.79 ± 1.69 | 61.79 ± 1.40 |
| | small mammals | 58.86 ± 2.76 | 61.27 ± 1.28 | 64.12 ± 2.27 | 53.49 ± 2.02 | 61.97 ± 1.01 | 54.24 ± 3.13 | 69.66 ± 1.17 |
| | trees | 60.15 ± 3.91 | 58.21 ± 3.51 | 56.19 ± 3.88 | 59.70 ± 5.16 | 61.46 ± 1.11 | 44.08 ± 5.98 | 77.34 ± 2.02 |
| | vehicles 1 | 36.35 ± 3.00 | 34.60 ± 2.06 | 41.54 ± 4.05 | 53.78 ± 4.45 | 37.61 ± 0.88 | 43.97 ± 2.40 | 63.27 ± 1.66 |
| | vehicles 2 | 51.23 ± 2.73 | 49.14 ± 1.59 | 49.34 ± 2.49 | 50.85 ± 4.94 | 50.31 ± 0.16 | 50.72 ± 3.23 | 69.74 ± 2.26 |
| | *average* | 55.78 ± 0.70 | 55.66 ± 0.49 | 53.62 ± 0.40 | 54.19 ± 1.37 | 55.03 ± 0.04 | 53.56 ± 0.48 | **65.01 ± 1.70** |

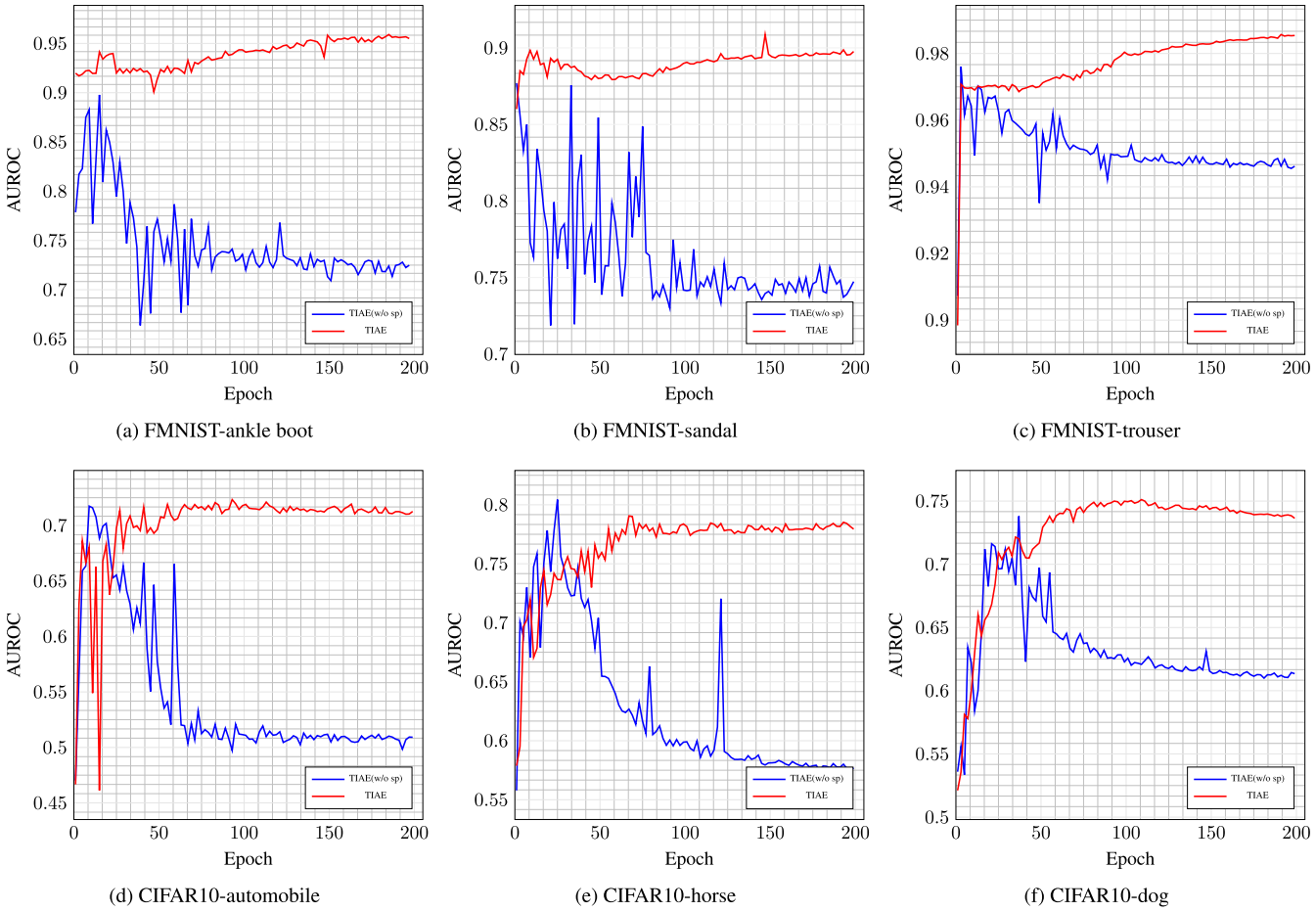



**FIGURE 5.** UOD performance (AUROC) in different training epochs. We plot the results of experiments on Fashion-MNIST and CIFAR10 with $\rho = 10\%$. With Fashion-MNIST, we show the results of the following classes as denoted in the caption: (a) ankle boot, (b) sandal, (c) trouser. With CIFAR10, we show the results of the following classes as denoted in the caption: (d) automobile, (e) horse, (f) dog. The proposed TIAE with the self-paced learning module achieves higher and more stable AUROC on all experiments compare with TIAE without the self-paced learning module. The results of other experiments show the same pattern.

The model stability of unsupervised outlier detection methods is essential. Validation during the training phase is impossible due to the lack of supervised labels. There is no way to obtain the best checkpoint for an unsupervised outlier detection model without validation. A stable model can make sure the performance of the final model is acceptable. The stability of model performance is mainly reflected in three aspects [47]: 1) Whether the model can reach convergence after acceptable training epochs in one training attempt. 2) Whether the model can reach a stable performance level in multiple training attempts using the same training configuration. 3) Whether the model can achieve good performance stably in various datasets and training configurations.

To assess the stability of our proposed TIAE model, we measure the UOD performance when the TIAE is being trained. Fig. 5 shows the AUROC in different training epochs. In general, the UOD performance is improved at the initial stage of training and then stabilizes as the training epochs continue to increase. Thus, through our TIAE, we can achieve a highly reliable model through acceptable training epochs in this task without validation.

### C. VISUALIZATION ANALYSIS

In this part, we conduct visualization analysis on MNIST and CIFAR-10 to demonstrate the effectiveness of TIAE for outlier detection. Fig. 3 shows the inputs and restoration/reconstruction outputs from both TIAE and CAE on MNIST during the testing period. Fig. 4 shows the inputs and restoration/reconstruction outputs from both TIAE and CAE on CIFAR-10 during the testing period.

The first row "Inlier" represents the inlier class. In both MNIST and CIFAR-10, we use the rest nine classes as outlier classes, corresponding to the "Outlier" rows. The first column "Ori" represents original images. "$O_{CAE}$" column means the reconstruction output from CAE. "$I_1, I_2, I_3, I_4$" mean the transformed images input to TIAE. "$O_1, O_2, O_3, O_4$" mean the restoration outputs from TIAE for corresponding transformed inputs. We force restoration outputs similar to original images but not transformed inputs. According to our score strategy, cases with outputs similar to "Ori" are considered as inliers, otherwise outliers.

The last row in Fig. 3 shows the restoration outputs of the number "9". All the four outputs are far different from

"Ori" and thus detected as an outlier. However, all the outputs from CAE are similar to the original images, which makes CAE less capable of distinguishing between inliers and outliers. We can get similar results on CIFAR-10 from Fig. 4. Besides, we can observe a more significant difference between the restoration outputs and original images in outliers on CIFAR-10 due to the color distortion. By comparing "Ori" and "$O_{CAE}$", we find that reconstruction outputs of CAE share more similar color patterns with original images, which is bad for outlier detection.

Above all, we conclude that our TIAE is effective for unsupervised outlier detection and work much better on complex datasets like CIFAR-10 compared with conventional autoencoder based methods.

### D. ABLATION STUDY

In this part, we perform an ablation study to analyze the contributions of two parts of the proposed TIAE framework: transformation invariant autoencoder and self-paced learning module. We conduct experiments on all five involved datasets. Table 3 shows experiment results. TIAE (w/o sp) denotes our proposed TIAE without the self-paced learning module, TIAE denotes our proposed TIAE with the self-paced learning module, CAE (unet) denotes the CAE method with the same backbone autoencoder with TIAE. To evaluate the contribution of the backbone network of CAE, we also copy CAE's UOD performance from Table 1 to Table 3.

**TABLE 3. AUROC (%) for UOD methods when $\rho = 10\%$. The best performance is in bold. TIAE (w/o sp) denotes our proposed TIAE without self-paced learning module, TIAE denotes our proposed TIAE with self-paced learning module. CAE (unet) denotes method CAE with U-Net like structure, the same as our TIAE.**

| Methods | MNIST | FMNIST | SVHN | CIFAR-10 | CIFAR-100 |
|---|---|---|---|---|---|
| CAE | 67.19 | 69.04 | 51.18 | 55.86 | 55.78 |
| CAE (unet) | 60.98 | 64.09 | 49.95 | 56.55 | 58.06 |
| TIAE (w/o sp) | 70.81 | 74.56 | 56.57 | 60.24 | 59.90 |
| TIAE | **85.18** | **86.77** | **68.78** | **71.18** | **65.01** |

When we use a more complex structure for the CAE method, UOD performance decreases instead of improving. A more complex structure of CAE contributes to lower reconstruction error but causes a lower UOD performance. By comparing the results of CAE (unet) and TIAE (w/o sp), we can verify the effectiveness of the transformation invariant autoencoder.

When we add the self-paced learning module, the performance (AUROC) improves on all five datasets. To further look into the mechanism of the self-paced learning module, we plot the AUROC in different epochs of TIAE with and without the self-paced learning module in Fig. 5. In the initial training phase, both TIAE and TIAE (w/o sp) reach a high AUROC value. With the training going on, the performance of TIAE (w/o sp) is decreasing, while the performance of TIAE is much more stable. This is because the autoencoder can catch features of both inliers and outliers with the training going on, making it challenging to distinguish inliers and outliers based on restoration error. The self-paced learning module can effectively filter out outliers during representation learning.

Based on the above analysis, The proposed TIAE with a self-paced learning module achieve higher and more stable AUROC on all experiments. Besides, our self-paced learning module can be easily incorporated into other reconstruction-based unsupervised outlier detection methods.

## V. CONCLUSION

In this paper, we propose a framework named Transformation Invariant Autoencoder (TIAE) for unsupervised outlier detection. By feeding transformed examples and trying to restore the original examples, the TIAE framework learns high-level semantic features instead of low-level features of conventional autoencoder based methods. To mitigate the negative effect of outliers during the representation learning phase, we incorporate self-paced learning to select inlier likely examples during training. We show that TIAE can achieve a promising performance gain compared to other autoencoder based unsupervised outlier detection methods. For future research, it is meaningful to explore more transformations, which are likely to increase performance further. Which transformation group is more suitable for representation learning and the downstream task is also worth further exploration. As an open framework, different network architectures, different transformations, and scoring strategies can also be explored for TIAE. Though this paper focus on image outlier detection, the TIAE can be easily applied to video outlier detection.

## REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Outlier detection: A survey," *ACM Comput. Surv.*, vol. 14, p. 15, Aug. 2007.
[2] H. Wang, M. J. Bah, and M. Hammad, "Progress in outlier detection techniques: A survey," *IEEE Access*, vol. 7, pp. 107964–108000, 2019.
[3] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Gener. Comput. Syst.*, vol. 55, pp. 278–288, Feb. 2016.
[4] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.
[5] A. J. Boddy, W. Hurst, M. Mackay, and A. E. Rhalibi, "Density-based outlier detection for safeguarding electronic patient record systems," *IEEE Access*, vol. 7, pp. 40285–40294, 2019.
[6] A. De Paola, S. Gaglio, G. L. Re, F. Milazzo, and M. Ortolani, "Adaptive distributed outlier detection for WSNs," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 902–913, May 2015.
[7] M. Zhang, X. Li, and L. Wang, "An adaptive outlier detection and processing approach towards time series sensor data," *IEEE Access*, vol. 7, pp. 175192–175212, 2019.
[8] M. Munoz-Organero, "Outlier detection in wearable sensor data for human activity recognition (HAR) based on DRNNs," *IEEE Access*, vol. 7, pp. 74422–74436, 2019.
[9] M. Z. Zaheer, A. Mahmood, M. Astrid, and S.-I. Lee, "CLAWS: Clustering assisted weakly supervised learning with normalcy suppression for anomalous event detection," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, 2020, pp. 358–376.
[10] M. Z. Zaheer, A. Mahmood, H. Shin, and S.-I. Lee, "A self-reasoning framework for anomaly detection using video-level labels," *IEEE Signal Process. Lett.*, vol. 27, pp. 1705–1709, 2020.

[11] Z. Zaheer, J.-H. Lee, M. Astrid, A. Mahmood, and S.-I. Lee, "Cleaning label noise with clusters for minimally supervised anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2020.

[12] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

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

[15] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*. [Online]. Available: http://arxiv.org/abs/1901.03407

[16] J. Dai, H. Song, G. Sheng, and X. Jiang, "Cleaning method for status monitoring data of power equipment based on stacked denoising autoencoders," *IEEE Access*, vol. 5, pp. 22863–22870, 2017.

[17] Z. Sun and H. Sun, "Stacked denoising autoencoder with density-grid based clustering method for detecting outlier of wind turbine components," *IEEE Access*, vol. 7, pp. 13078–13091, 2019.

[18] F. Wan, G. Guo, C. Zhang, Q. Guo, and J. Liu, "Outlier detection for monitoring data using stacked autoencoder," *IEEE Access*, vol. 7, pp. 173827–173837, 2019.

[19] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proc. MLSDA 2nd Workshop Mach. Learn. Sensory Data Anal. (ACM)*, 2014, p. 4.

[20] S. Wang, Y. Zeng, X. Liu, E. Zhu, J. Yin, C. Xu, and M. Kloft, "Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5962–5975.

[21] X. Peng, J. Feng, J. Lu, W.-Y. Yau, and Z. Yi, "Cascade subspace clustering," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2478–2484.

[22] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5879–5887.

[23] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent.*, Feb. 2018, pp. 1–19.

[24] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. Van Den Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1705–1714.

[25] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3379–3388.

[26] M. Zaigham Zaheer, J.-H. Lee, M. Astrid, and S.-I. Lee, "Old is gold: Redefining the adversarially learned one-class classifier training paradigm," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14183–14193.

[27] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Proc. Asian Conf. Comput. Vis.* Springer, 2018, pp. 622–637.

[28] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*. [Online]. Available: http://arxiv.org/abs/1807.03748

[29] A. L. Rezaabad and S. Vishwanath, "Learning representations by maximizing mutual information in variational autoencoders," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 15535–15545.

[30] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[31] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 1597–1607.

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

[33] 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. (ICCV)*, Dec. 2015, pp. 1511–1519.

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

[35] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1189–1197.

[36] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 865–878, May 2017.

[37] L. Han, D. Zhang, D. Huang, X. Chang, J. Ren, S. Luo, and J. Han, "Self-paced mixture of regressions," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1816–1822.

[38] W. Huang, C. Liang, Y. Yu, Z. Wang, W. Ruan, and R. Hu, "Video-based person re-identification via self paced weighting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–12.

[39] D. Zhang, J. Han, L. Yang, and D. Xu, "SPFTN: A joint learning framework for localizing and segmenting objects in weakly labeled videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 475–489, Feb. 2020.

[40] J. Han, Y. Yang, D. Zhang, D. Huang, D. Xu, and F. De La Torre, "Weakly-supervised learning of category-specific 3D object shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1423–1437, Apr. 2019.

[41] D. Zhang, J. Han, L. Zhao, and D. Meng, "Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework," *Int. J. Comput. Vis.*, vol. 127, no. 4, pp. 363–380, Apr. 2019.

[42] X. Guo, X. Liu, E. Zhu, X. Zhu, M. Li, X. Xu, and J. Yin, "Adaptive self-paced deep clustering with data augmentation," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 9, pp. 1680–1693, Sep. 2019.

[43] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Proc. Int. Conf. Artif. Neural Netw.* Espoo, Finland: Springer, 2011, pp. 44–51.

[44] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3859–3869.

[45] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–15.

[46] D. M. Hawkins, *Identification of Outliers*, vol. 11. Amsterdam, The Netherlands: Springer, 1980.

[47] Y. Fei, C. Huang, C. Jinkun, M. Li, Y. Zhang, and C. Lu, "Attribute restoration framework for anomaly detection," *IEEE Trans. Multimedia*, early access, Dec. 30, 2021, doi: 10.1109/TMM.2020.3046884.

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

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

[50] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Dec. 1998.

[51] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*. [Online]. Available: http://arxiv.org/abs/1708.07747

[52] 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. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011, p. 5.

[53] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

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

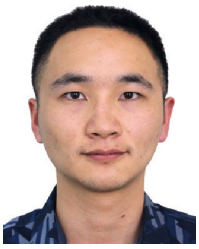
[55] C.-H. Lai, D. Zou, and G. Lerman, "Robust subspace recovery layer for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–28.

[56] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2802–2810.

[57] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Skip-GANomaly: Skip connected and adversarially trained encoder-decoder anomaly detection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[58] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[59] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Munich, Germany: Springer, 2015, pp. 234–241.

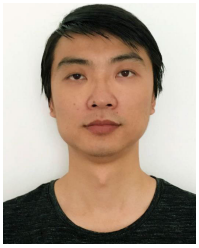


**ZHEN CHENG** is currently pursuing the Ph.D. degree with the National University of Defense Technology (NUDT), China. His current research interests include transfer learning, outlier detection, and deep neural networks.

**EN ZHU** received the Ph.D. degree from the National University of Defense Technology (NUDT), China. He received China National Excellence Doctoral Dissertation. He is currently a Professor with the School of Computer Science, NUDT. He has published over 60 peer-reviewed papers, including IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *PR*, AAAI, and IJCAI. His main research interests include pattern recognition, image processing, machine vision, and machine learning.

**SIQI WANG** received the B.S. and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), China. He is currently an Assistant Research Professor with the State Key Laboratory of High Performance Computing (HPCL), NUDT. He has published in leading conferences and journals, such as NeurIPS, AAAI, ACM MM, ICPR, *Pattern Recognition*, IEEE TRANSACTIONS ON CYBERNETICS, AND NEUROCOMPUTING. His main research interests include anomaly/outlier detection, pattern recognition, and unsupervised learning. He also serves as a reviewer for several international journals, including the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON AUTOMATION SCIENCE and ENGINEERING, *Artificial Intelligence Review*, and the *International Journal of Machine Learning and Cybernetics*.

**PEI ZHANG** received the bachelor's degree from Yunnan University, in 2018, and the master's degree in computer science from the National University of Defense Technology (NUDT), in 2020, where she is currently pursuing the Ph.D. degree. She has published several articles in highly regarded journals, such as IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE). Her current research interests include multi-view learning, incomplete multi-view clustering, and deep clustering.



**WANG LI** is currently pursuing the Ph.D. degree with the National University of Defense Technology (NUDT), China. His current research interests include deep clustering, graph neural networks, and deep neural networks.