Catching Both Gray and Black Swans: Open-set Supervised Anomaly Detection

Choubo Ding\textsuperscript{1\dagger}, Guansong Pang\textsuperscript{2\dagger}, Chunhua Shen\textsuperscript{3}

\textsuperscript{1} The University of Adelaide  \textsuperscript{2} Singapore Management University  \textsuperscript{3} Zhejiang University

Abstract

Despite most existing anomaly detection studies assume the availability of normal training samples only, a few labeled anomaly examples are often available in many real-world applications, such as defect samples identified during random quality inspection, lesion images confirmed by radiologists in daily medical screening, etc. These anomaly examples provide valuable knowledge about the application-specific abnormality, enabling significantly improved detection of similar anomalies in some recent models. However, those anomalies seen during training often do not illustrate every possible class of anomaly, rendering these models ineffective in generalizing to unseen anomaly classes. This paper tackles open-set supervised anomaly detection, in which we learn detection models using the anomaly examples with the objective to detect both seen anomalies (‘gray swans’) and unseen anomalies (‘black swans’). We propose a novel approach that learns disentangled representations of abnormalities illustrated by seen anomalies, pseudo anomalies, and latent residual anomalies (i.e., samples that have unusual residuals compared to the normal data in a latent space), with the last two abnormalities designed to detect unseen anomalies. Extensive experiments on nine real-world anomaly detection datasets show superior performance of our model in detecting seen and unseen anomalies under diverse settings. Code and data are available at: https://github.com/choubo/DRA

1. Introduction

Anomaly detection (AD) aims at identifying exceptional samples that do not conform to expected patterns [35]. It has broad applications in diverse domains, e.g., lesion detection in medical image analysis [48, 56, 70], inspecting micro-cracks/defects in industrial inspection [3, 4], crime/accident detection in video surveillance [11, 20, 51, 69], and unknown object detection in autonomous driving [10, 55]. Most of existing anomaly detection methods

\textsuperscript{*}Corresponding author: CS (e-mail: chunhua@me.com). This work was in part done when GP and CS were with The University of Adelaide.

\textsuperscript{†}First two authors contributed equally.

Figure 1. t-SNE visualization of features learned by SotA unsupervised (KDAD [46]) and supervised (DevNet [34, 36]) models, and our open-set supervised model (DRA) on the test data of two MVTec AD datasets, Leather and Tile. KDAD is trained with normal data only, learning less discriminative features than DevNet and DRA that are trained using ten samples from the seen anomaly classes, in addition to the normal data. DevNet is prone to overfitting the seen anomalies, failing to distinguish unseen anomalies from the normal data, while DRA effectively mitigates this issue.

\[2, 8, 11, 13, 32, 38, 41, 43, 45, 46, 48, 57–59, 68, 73\] are unsupervised, which assume the availability of normal training samples only, i.e., anomaly-free training data, because it is difficult, if not impossible, to collect large-scale anomaly data. However, a small number of (e.g., one to multiple) labeled anomaly examples are often available in many relevant real-world applications, such as some defect samples identified during random quality inspection, lesion images confirmed by radiologists in daily medical screening, etc. These anomaly examples provide valuable knowledge about application-specific abnormality [29, 34, 36, 44], but the unsupervised detectors are unable to utilize them. Due to the lack of knowledge about anomalies, the learned features in unsupervised models are not discriminative enough to distinguish anomalies (especially some challenging ones) from normal data, as illustrated by the results of KDAD [46], a recent state-of-the-art (SotA) unsupervised method, on two MVTec AD defect detection datasets [3] in Fig. 1.

In recent years, there have been some studies [29, 34, 36, 44] exploring a supervised detection paradigm that aims at exploiting those small, readily accessible anomaly data—rare but previously occurred exceptional cases/events, a.k.a.
anomalies from the normal data, as shown in Fig. 1. To train anomaly-informed detection models. The current methods in this line focus on fitting these anomaly examples using one-class metric learning with the anomalies as negative samples [29, 44] or one-sided anomaly-focused deviation loss [34, 36]. Despite the limited amount of the anomaly data, they achieve largely improved performance in detecting anomalies that are similar to the anomaly examples seen during training. However, these seen anomalies often do not illustrate every possible class of anomaly because i) anomalies per se are unknown and ii) the seen and unseen anomaly classes can differ largely from each other [35], e.g., the defective features of color stains are very different from that of folds and cuts in leather defect inspection. Consequently, these models can overfit the seen anomalies, failing to generalize to unseen/unknown anomaly classes—rare and previously unknown exceptional cases/events, a.k.a. black swans [54], as shown by the result of DevNet [34, 36] in Fig. 1 where DevNet improves over KDAD in detecting the seen anomalies but fails to discriminate unseen anomalies from normal samples. In fact, these supervised models can be biased by the given anomaly examples and become less effective in detecting unseen anomalies than unsupervised detectors (see DevNet vs. KDAD on the Tile dataset in Fig. 1).

To address this issue, this paper tackles open-set supervised anomaly detection, in which detection models are trained using the small anomaly examples in an open-set environment, i.e., the objective is to detect both seen anomalies (‘gray swans’) and unseen anomalies (‘black swans’). To this end, we propose a novel anomaly detection approach, termed DRA, that learns disentangled representations of abnormalities to enable the generalized detection. Particularly, we disentangle the unbounded abnormalities into three general categories: anomalies similar to the limited seen anomalies, anomalies that are similar to pseudo anomalies created from data augmentation or external data sources, and unseen anomalies that are detectable in some latent residual-based composite feature spaces. We further devise a multi-head network, with separate heads enforced to learn each type of these three disentangled abnormalities. In doing so, our model learns diversified abnormality representations rather than only the known abnormality, which can discriminate both seen and unseen anomalies from the normal data, as shown in Fig. 1.

In summary, we make the following main contributions:

• We introduce a latent residual-based abnormality learning module that learns abnormality upon the residuals between the intermediate feature maps of normal and abnormal samples. This helps learn discriminative composite features for the detection of hard anomalies (e.g., unseen anomalies) that cannot be detected in the original non-composite feature space.

• We propose a novel multi-head neural network-based model DRA to learn the disentangled abnormality representations, with each head dedicated to capturing one specific type of abnormality.

• We further introduce a latent residual-based abnormality learning module that learns abnormality upon the residuals between the intermediate feature maps of normal and abnormal samples. This helps learn discriminative composite features for the detection of hard anomalies (e.g., unseen anomalies) that cannot be detected in the original non-composite feature space.

• We perform comprehensive experiments on nine real-application datasets from industrial inspection, rover-based planetary exploration and medical image analysis. The results show that our model substantially outperforms five SotA competing models in diverse settings. The results also establish new baselines for future work in this important emerging direction.

2. Related Work

Unsupervised Approaches. Most existing anomaly detection methods, such as autoencoder-base methods [13, 18, 38, 71, 73], GAN-base methods [39, 45, 48, 68], self-supervised methods [2, 11, 12, 25, 50, 56, 60], and one-class classification methods [7, 8, 40, 43], assume that only normal data can be accessed during training. Although they do not have the risk of biasing towards the seen anomalies, they are difficult to distinguish anomalies from normal samples due to the lack of knowledge about true anomalies.

Supervised Approaches. A recently emerging direction focuses on supervised (or semi-supervised) anomaly detection that alleviates the lack of anomaly information by leveraging small anomaly examples to learn anomaly-informed models. This is achieved by one-class metric learning with the anomalies as negative samples [14, 29, 33, 44] or one-sided anomaly-focused deviation loss [34, 36, 70]. However, these models rely heavily on the seen anomalies and can overfit the known abnormality. A reinforcement learning approach is introduced in [37] to mitigate this overfitting issue, but it assumes the availability of large-scale unlabeled data and the presence of unseen anomalies in those data. Supervised anomaly detection is similar to imbalanced classification [6, 15, 30] in that they both detect rare classes with a few labeled examples. However, due to the unbounded nature and unknowingness of anomalies, anomaly detection is inherently an open-set task, while the imbalanced classification task is typically formulated as a closed-set problem.

Learning In- and Out-of-distribution. Out-of-distribution (OOD) detection [16, 17, 19, 28, 42, 67] and open-set recognition [1, 29, 47, 65, 72] are related tasks to ours. However, they aim at guaranteeing accurate multi-class inlier classification while detecting OOD/uncertain samples, whereas our task is focused on anomaly detection exclusively. Further, despite the use of pseudo anomalies
like outlier exposure [17, 19] shows effective performance, the current models in these two tasks are also assumed to be inaccessible to any true anomalous samples.

3. Proposed Approach

Problem Statement The studied problem, open-set supervised AD, can be formally stated as follows. Given a set of training samples \( \mathcal{X} = \{ x_i \}_{i=1}^{N+M} \), in which \( \mathcal{X}_n = \{ x_1, x_2, \ldots, x_N \} \) is the normal sample set and \( \mathcal{X}_s = \{ x_{N+1}, x_{N+2}, \ldots, x_{N+M} \} (M \ll N) \) is a very small set of annotated anomalies that provide some knowledge about true anomalies, and the \( M \) anomalies belong to the seen anomaly classes \( \mathcal{C} \), where \( \mathcal{C} = \{ c_i \}_{i=1}^{\mathcal{C}} \) denotes the set of all possible anomaly classes, and then the goal is to detect both seen and unseen anomaly classes by learning an anomaly scoring function \( g: \mathcal{X} \rightarrow \mathbb{R} \) that assigns larger anomaly scores to both seen and unseen anomalies than normal samples.

3.1. Overview of Our Approach

Our proposed approach DRA is designed to learn disentangled representations of diverse abnormalities to effectively detect both seen and unseen anomalies. The learned abnormality representations include the seen abnormality illustrated by the limited given anomaly examples, and the unseen abnormalities illustrated by pseudo anomalies and latent residual anomalies (i.e., samples that have unusual residuals compared to normal examples in a learned feature space). In doing so, DRA mitigates the issue of biasing towards seen anomalies and learns generalized detection models. The high-level overview of our proposed framework is provided in Fig. 2a, which is composed of three main modules, including seen, pseudo, and latent residual abnormality learning heads. The first two heads learn abnormality representations in a plain (non-composite) feature space, as shown in Fig. 2b, while the last head learns composite abnormality representations by looking into the deviation of the residual features of input samples to some reference (i.e., normal) images in a learned feature space, as shown in Fig. 2c. Particularly, given a feature extraction network \( f: \mathcal{X} \rightarrow \mathcal{M} \) for extracting the intermediate feature map \( \mathcal{M} \subset \mathcal{X} \subset \mathbb{R}^{c \times h \times w} \) from a training image \( x \in \mathcal{X} \subset \mathbb{R}^{c \times h \times w} \), and a set of abnormality learning heads \( \mathcal{J} = \{ g_i \}_{i=1}^{\mathcal{J}} \), where each head \( g: \mathcal{M} \rightarrow \mathbb{R} \) learns an anomaly score for one type of abnormality, then the overall objective of DRA can be given as follows:

\[
\arg\min_{\Theta} \sum_{i=1}^{\mathcal{J}} \ell_i \left( g_i (f(x; \Theta_f); \Theta_i), y_x \right),
\]

where \( \Theta \) contains all weight parameters, \( y_x \) denotes the supervision information of \( x \), and \( \ell_i \) denotes a loss function for one head. The feature extraction network \( f \) is jointly optimized by all the downstream abnormality learning heads, while these heads are independent from each other in learning the specific abnormality. Below we introduce each head in detail.

3.2. Learning Disentangled Abnormalities

**Abnormality Learning with Seen Anomalies.** Most real-world anomalies have only some subtle differences from normal images, sharing most of the common features with normal images. Patch-wise anomaly learning [4, 34, 59, 64] that learns anomaly scores for each small image patch has shown impressive performance in tackling this issue. Motivated by this, DRA utilizes a top-\( K \) multiple-instance-learning (MIL) -based method in [34] to effectively learn the seen abnormality. As shown in Fig. 2b, for the feature map \( \mathcal{M}_k \) of each input image \( x \), we generate pixel-wise vector representations \( \mathcal{D} = \{ \mathbf{d}_k \}_{k=1}^{h \times w} \), each of which corresponds to the feature vector of a small patch of the input image. These patch-wise representations are then mapped to learn the anomaly scores of the image patches by an anomaly classifier \( g_k: \mathcal{D} \rightarrow \mathbb{R} \). Since only selective image patches contain abnormal features, we utilize an op-
timization using top-K MIL to learn an anomaly score for an image based on the $K$ most anomalous image patches, with the loss function defined as follows:

$$
\ell_s(x, y_s) = \ell(g_p(M_x; \Theta_p), y_s),
$$

(2)

where $\ell$ is a binary classification loss function; $y_s = 1$ if $x$ is a seen anomaly, and $y_s = 0$ if $x$ is a normal sample; and

$$
g_p(M_x; \Theta_p) = \max_{\Psi_K(M_x) \in \mathcal{D}} \frac{1}{K} \sum_{d_i \in \Psi_K(M_x)} g_s(d_i; \Theta_s)
$$

(3)

where $\Psi_K(M_x)$ is a set of $K$ vectors that have the largest anomaly scores among all vectors in $M_x$. Abnormality Learning with Pseudo Anomalies. We further design a separate head to learn abnormalities that are different from the seen anomalies and simulate some possible classes of unseen anomaly. There are two effective methods to create such pseudo anomalies, including data augmentation-based methods [25,53] and outlier exposure [17,41]. Particularly, for the data augmentation-based method, we adapt the popular method CutMix [66] to generate pseudo anomalies $\tilde{x}$ from normal images $x_n$ for training, which is defined as follows:

$$
\tilde{x} = T \circ C(R \odot x_n) + (1 - T(R)) \odot x_n
$$

(4)

where $R \in \{0,1\}^{h \times w}$ denotes a binary mask of random rectangle, $1$ is an all-ones matrix, $\odot$ is element-wise multiplication, $T(\cdot)$ is a randomly translate transformation, and $C(\cdot)$ is a random color jitter. As shown in Fig. 2a, the pseudo abnormality learning uses the same architecture and anomaly scoring method as the seen abnormality learning to learn fine-grained pseudo abnormal features:

$$
\ell_p(x, y_s) = \ell(g_p(M_x; \Theta_p), y_s),
$$

(5)

where $y_s = 1$ if $x$ is a pseudo anomaly, i.e., $x = \tilde{x}$, and $y_s = 0$ if $x$ is a normal sample otherwise; and $g_p(M_x; \Theta_p)$ is exactly the same as $g_s$ in Eq. (3), but $g_p$ is trained in a separate head with different anomaly data and parameters from $g_s$ to learn the pseudo abnormality. As discussed in Secs. 4.1 and 4.6, the outlier exposure method [17] is used in anomaly detection on medical datasets. In such cases, the pseudo anomalies $\tilde{x}$ are samples randomly drawn from external data instead of creating from Eq. (4).

Abnormality Learning with Latent Residual Anomalies. Some anomalies, such as previously unknown anomalies that share no common abnormal features with the seen anomalies and have only small difference to the normal samples, are difficult to detect by using only the features of the anomalies themselves, but they can be easily detected in a high-order composite feature space provided that the composite features are more discriminative. As anomalies are characterized by their difference from normal data, we utilize the difference between the features of the anomalies and normal feature representations to learn such discriminative composite features. More specifically, we propose the latent residual abnormality learning that learns anomaly scores of samples based on their feature residuals comparing to the features of some reference images (normal images) in a learned feature space. As shown in Fig. 2c, to obtain the latent feature residuals, we first use a small set of images randomly drawn from the normal data as the reference data, and compute the mean of their feature maps to obtain the reference normal feature map:

$$
M_r = \frac{1}{N_r} \sum_{i=1}^{N_r} f(x_i; \Theta_f).
$$

(6)

where $x_{ri}$ is a reference normal image, and $N_r$ is a hyper-parameter that represents the size of the reference set. For a given training image $x$, we perform element-wise subtraction between its feature map $M_x$ and the reference normal feature map $M_r$ that is fixed for all training and testing samples, resulting in a residual feature map $M_{x \oplus r}$ for $x$:

$$
M_{x \oplus r} = M_r \ominus M_x,
$$

(7)

where $\oplus$ denotes element-wise subtraction. We then perform an anomaly classification upon these residual features:

$$
\ell_r(x, y_s) = \ell(g_r(M_{x \oplus r}; \Theta_r), y_s),
$$

(8)

where $y_s = 1$ if $x$ is a seen/pseudo anomaly, and $y_s = 0$ if $x$ is a normal sample otherwise. Again, $g_r$ uses exactly the same method to obtain the anomaly score as $g_s$ in Eq. 3, but it is trained in a separate head with the parameters $\Theta_r$ using different training inputs, i.e., residual feature map $M_{x \oplus r}$.

Since the $g_s$, $g_p$ and $g_r$ heads focus on learning the abnormality representations, the jointly learned feature map in $f$ does not well model the normal features. To address this issue, we add a separate normality learning head as follows:

$$
\ell_n(x, y_s) = \ell(g_n\left(\frac{1}{h \times w'} \sum_{i=1}^{h \times w'} d_i; \Theta_n\right), y_s),
$$

(9)

where $g_n : \mathcal{D} \rightarrow \mathbb{R}$ is a fully-connected binary anomaly classifier that discriminates normal samples from all seen and pseudo anomalies. Unlike abnormal features that are often fine-grained local features, normal features are holistic global features. Hence, $g_n$ does not use the top-$K$ MIL-based anomaly scoring as in other heads and learns holistic normal scores instead.

Training and Inference. During training, the feature mapping network $f$ is shared and jointly trained by all the four heads $g_s$, $g_p$, $g_r$, and $g_n$. These four heads are independent from each other, and so their parameters are not shared.
and independently optimized. A loss function called deviation loss [34, 36] is used to implement the loss function $\ell$ in all our heads by default, as it enables generally more stable and effective performance than other loss functions such as cross entropy loss or focal loss (see Appendix C.2). During inference, given a test image, we sum of all the scores from the abnormality learning heads ($g_1, g_2$ and $g_3$) and minus the score from the normality head $g_n$ to obtain its anomaly score.

4. Experiments

Datasets Many studies evaluate their models on synthetic anomaly detection datasets converted from popular image classification benchmarks, such as MNIST [24], Fashion-MNIST [63], CIFAR-10 [23], using one-vs-all or one-vs-one protocols. This conversion results in clearly disparate anomalies from normal samples. However, anomalies and normal samples in real-world applications, such as industrial defect inspection and lesion detection in medical images, typically have only subtle/small difference. Motivated by this, following [25, 34, 64], we focus on datasets with natural anomalies rather than one-vs-all/one-vs-one based synthetic anomalies. Particularly, nine diverse datasets with real anomalies are used in our experiments, including five industrial defect inspection datasets: MVTec AD [3], AITEX [49], SDD [52], ELPV [9] and Optical [62], in which we aim to inspect defective image samples; one planetary exploration dataset: Masticam [21] in which we aim to identify geologically-interesting/novel images taken by Mars exploration rovers; and three medical image datasets for detecting lesions on different organs: BrainMRI [46], HeadCT [46] and Hyper-Kvasir [5]. These datasets are popular benchmarks in the respective research domains and recently emerging as important benchmarks for anomaly detection [4, 18, 34, 46, 64] (see Appendix A for detailed introduction of these datasets).

4.1. Implementation Details

DRA uses ResNet-18 as the feature learning backbone. All its heads are jointly trained using 30 epochs, with 20 iterations per epoch and a batch size of 48. Adam is used for the parameter optimization using an initial learning rate $10^{-3}$ with a weight decay of $10^{-2}$. The top-$K$ MIL in DRA is the same as that in DevNet [34], i.e., $K$ in the top-$K$ MIL is set to 10% of the number of all scores per score map. $N_r = 5$ is used by default in the residual anomaly learning (see Sec. 4.6). The pseudo abnormality learning uses CutMix [66] to create pseudo anomalies samples on all datasets except the three medical datasets, on which DRA uses external data from another medical dataset LAG [26] as the pseudo anomaly source (see Sec. 4.6).

Our model DRA is compared to five recent and closely related state-of-the-art (SotA) methods, including MLEP [29], deviation network (DevNet) [34, 36], SAOE (combining data augmentation-based Synthetic Anomalies [25, 31, 53] with Outlier Exposure [17, 41]), unsupervised anomaly detector KDAD [46], and focal loss-driven classifier (FLOS) [27] (See Appendix C.1 for comparison with two other methods [44, 61]). MLEP and DevNet address the same open-set AD problem as ours. KDAD is a recent unsupervised AD method that works on normal training data only. It is commonly assumed that unsupervised detectors are more preferable than the supervised ones in detecting unseen anomalies, as the latter may bias towards the seen anomalies. Motivated by this, KDAD is used as a baseline. The implementation of DevNet and KDAD is taken from their authors. MLEP is adapted to the image task with the same setting as DRA. SAOE utilizes pseudo anomalies from both data augmentation-based and outlier exposure-based methods, outperforming the individuals that use one of these anomaly creation methods. FLOS is an imbalanced classifier trained with focal loss. For a fair comparison, all competing methods use the same network backbone (i.e., ResNet-18) as DRA except KDAD that requires its own special network architecture to perform training and inference. Further implementation details of DRA and its competing methods are provided in Appendix B.

4.2. Experiment Protocols

We use the following two experiment protocols:

**General setting** simulates a general scenario of open-set AD, where the given anomaly examples are a few samples randomly drawn from all possible anomaly classes in the test set per dataset. These sampled anomalies are then removed from the test data. This is to replicate real-world applications where we cannot determine which anomaly classes are known and how many anomaly classes the given anomaly examples span. Thus, the datasets can contain both seen and unseen anomaly classes, or only the seen anomaly classes, depending on the underlying complexity of the applications (e.g., the number of all possible anomaly classes).

**Hard setting** is designed to exclusively evaluate the performance of the models in detecting unseen anomaly classes, which is the very key challenge in open-set AD. To this end, the anomaly example sampling is limited to be drawn from one single anomaly class only, and all anomaly samples in this anomaly class are removed from the test set to ensure that the test set contains only unseen anomaly classes. Note that this setting is only applicable to datasets with no less than two anomaly classes.

As labeled anomalies are difficult to obtain due to their rareness and unknowingness, in both settings we use only very limited labeled anomalies, i.e., with the number of the given anomaly examples respectively fixed to one and ten. The popular performance metric, Area Under ROC Curve (AUC), is used. Each model yields an anomaly ranking,
Table 1. AUC results (mean±std) on nine real-world AD datasets under the general setting. The first 15 datasets are data subsets of MVTec AD whose results are the averaged results over these subsets. The supervised methods are trained using one or ten random anomaly examples, with the best results in red and the second-best in blue. KDAD is treated as a baseline. | is the number of anomaly classes.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Baseline</th>
<th>DevNet</th>
<th>FLOS</th>
<th>SAOE</th>
<th>MLEP</th>
<th>KDAD (Ours)</th>
<th>DRA (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carpet</td>
<td>0.72±0.02</td>
<td>0.77±0.01</td>
<td>0.76±0.01</td>
<td>0.71±0.01</td>
<td>0.80±0.01</td>
<td>0.80±0.01</td>
<td>1.00±0.01</td>
</tr>
<tr>
<td>Grid</td>
<td>0.74±0.02</td>
<td>0.78±0.02</td>
<td>0.79±0.02</td>
<td>0.83±0.02</td>
<td>0.97±0.02</td>
<td>0.97±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>Leather</td>
<td>0.94±0.02</td>
<td>0.96±0.02</td>
<td>0.96±0.02</td>
<td>0.98±0.02</td>
<td>0.99±0.02</td>
<td>0.99±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>Tile</td>
<td>0.91±0.02</td>
<td>0.93±0.02</td>
<td>0.93±0.02</td>
<td>0.93±0.02</td>
<td>0.97±0.02</td>
<td>0.97±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>Wood</td>
<td>0.94±0.02</td>
<td>0.92±0.02</td>
<td>0.94±0.02</td>
<td>0.94±0.02</td>
<td>0.94±0.02</td>
<td>0.94±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>Bottle</td>
<td>0.92±0.02</td>
<td>0.96±0.02</td>
<td>0.96±0.02</td>
<td>0.96±0.02</td>
<td>0.96±0.02</td>
<td>0.96±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>Capsule</td>
<td>0.77±0.02</td>
<td>0.80±0.02</td>
<td>0.80±0.02</td>
<td>0.80±0.02</td>
<td>0.80±0.02</td>
<td>0.80±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>Pill</td>
<td>0.82±0.02</td>
<td>0.76±0.02</td>
<td>0.73±0.02</td>
<td>0.65±0.02</td>
<td>0.82±0.02</td>
<td>0.82±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>Transistor</td>
<td>0.80±0.02</td>
<td>0.72±0.02</td>
<td>0.70±0.02</td>
<td>0.68±0.02</td>
<td>0.81±0.02</td>
<td>0.81±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>Zipper</td>
<td>0.92±0.02</td>
<td>0.92±0.02</td>
<td>0.92±0.02</td>
<td>0.92±0.02</td>
<td>0.92±0.02</td>
<td>0.92±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>Cable</td>
<td>0.80±0.02</td>
<td>0.73±0.02</td>
<td>0.79±0.02</td>
<td>0.81±0.02</td>
<td>0.81±0.02</td>
<td>0.81±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>Hazelnut</td>
<td>0.94±0.02</td>
<td>0.97±0.02</td>
<td>0.97±0.02</td>
<td>0.97±0.02</td>
<td>0.97±0.02</td>
<td>0.97±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>Metal_just</td>
<td>0.74±0.03</td>
<td>0.87±0.03</td>
<td>0.93±0.03</td>
<td>0.93±0.03</td>
<td>0.93±0.03</td>
<td>0.93±0.03</td>
<td>1.00±0.03</td>
</tr>
<tr>
<td>Screw</td>
<td>0.80±0.02</td>
<td>0.39±0.03</td>
<td>0.33±0.03</td>
<td>0.65±0.03</td>
<td>0.65±0.03</td>
<td>0.65±0.03</td>
<td>1.00±0.03</td>
</tr>
<tr>
<td>Toothbrush</td>
<td>0.86±0.02</td>
<td>0.73±0.03</td>
<td>0.73±0.03</td>
<td>0.68±0.03</td>
<td>0.68±0.03</td>
<td>0.68±0.03</td>
<td>1.00±0.03</td>
</tr>
<tr>
<td>MVTec AD</td>
<td>0.86±0.02</td>
<td>0.99±0.03</td>
<td>0.99±0.03</td>
<td>0.99±0.03</td>
<td>0.99±0.03</td>
<td>0.99±0.03</td>
<td>1.00±0.03</td>
</tr>
<tr>
<td>AITEX</td>
<td>0.57±0.02</td>
<td>0.59±0.02</td>
<td>0.53±0.02</td>
<td>0.67±0.02</td>
<td>0.64±0.02</td>
<td>0.64±0.02</td>
<td>0.84±0.02</td>
</tr>
<tr>
<td>ELPV</td>
<td>0.88±0.02</td>
<td>0.88±0.02</td>
<td>0.84±0.02</td>
<td>0.84±0.02</td>
<td>0.84±0.02</td>
<td>0.84±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>Optical</td>
<td>0.71±0.02</td>
<td>0.52±0.02</td>
<td>0.51±0.02</td>
<td>0.81±0.02</td>
<td>0.81±0.02</td>
<td>0.81±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>Mastic</td>
<td>0.64±0.02</td>
<td>0.59±0.02</td>
<td>0.54±0.02</td>
<td>0.62±0.02</td>
<td>0.62±0.02</td>
<td>0.62±0.02</td>
<td>1.00±0.02</td>
</tr>
<tr>
<td>BrainMRI</td>
<td>0.73±0.03</td>
<td>0.69±0.03</td>
<td>0.69±0.03</td>
<td>0.63±0.03</td>
<td>0.63±0.03</td>
<td>0.63±0.03</td>
<td>1.00±0.03</td>
</tr>
<tr>
<td>HeadCT</td>
<td>0.79±0.03</td>
<td>0.74±0.03</td>
<td>0.68±0.03</td>
<td>0.56±0.03</td>
<td>0.75±0.03</td>
<td>0.75±0.03</td>
<td>1.00±0.03</td>
</tr>
<tr>
<td>Hyper-Kvasir</td>
<td>0.80±0.02</td>
<td>0.65±0.03</td>
<td>0.68±0.03</td>
<td>0.49±0.03</td>
<td>0.45±0.03</td>
<td>0.45±0.03</td>
<td>1.00±0.03</td>
</tr>
</tbody>
</table>

and its AUC is calculated based on the ranking. All reported AUCs are averaged results over three independent runs.

4.3. Results under the General Setting

Tab. 1 shows the comparison results under the general setting protocol. Below we discuss the results in details.

**Application Domain Perspective.** Despite the datasets from diverse application domains, including industrial defect inspection, rover-based planetary exploration and medical image analysis, our model achieves the best AUC performance on nearly all of the datasets, i.e., eight (seven) out of nine datasets in the one-shot (ten-shot) setting, with the second-best results on the other datasets. On challenging datasets, such as MVTec AD, AITEX, Mastic and Hyper-Kvasir, where a larger number of possible anomaly classes is presented, our model obtains consistently better AUC results, increasing by up to 5% AUC.

**Sample Efficiency.** The reduction of training anomaly examples generally decreases the performance of all the supervised models. Compared to the competing detectors, our model shows better sample efficiency in that i) with reduced anomaly examples, our model has a much smaller decrease of AUC, i.e., an average of 15.1% AUC decrease across the nine datasets, which is much better than DevNet (22.3%), FLOS (21.6%), SAOE (19.7%), and MLEP (21.6%), and ii) our model trained with one anomaly example can largely outperform the strong competing methods trained with ten anomaly examples, such as DevNet, FLOS and MLEP on Optical, and SAOE and MLEP on Hyper-Kvasir.

**Comparison to Unsupervised Baseline.** Compared to the unsupervised model KDAD, our model and other supervised models demonstrate consistently better performance when using ten training anomaly examples (i.e., less open-set scenarios). In more open-set scenarios where only one anomaly example is used, our method is the only model that is still clearly better than KDAD on most datasets, even on challenging datasets which have many anomaly classes, such as MVTec AD, AITEX, and Mastic.

4.4. Results under the Hard Setting

The detection performance on six datasets applicable under the hard setting is presented in Tab. 2.

**Application Domain Perspective.** In both one-shot and ten-shot settings of the diverse application datasets, compared to the competing methods, our method is the best performer on most of the individual data subsets; At the dataset-level performance, our model achieves about 2%–10% mean AUC increase compared to the best contender on most of the six datasets, with close to the best performance on the other datasets. This shows substantially better generalizability of our model in detecting unseen anomaly classes than the other supervised detectors.

**Sample Efficiency.** Compared to one-shot scenarios to the ten-shot ones, our model, on average, has 5.5% AUC decrease at the dataset level, which is better than that of the competing methods: DevNet (9.8%), FLOS (7.1%), SAOE (7.8%), and MLEP (10%). More impressively, our model trained with one anomaly example outperforms the ten-shot competing models by a large margin on many of the individual data subsets as well as the overall datasets.

**Comparison to Unsupervised Baseline.** Current supervised AD models are often biased towards the seen anomaly class and fail to generalize to unseen anomaly classes, performing less effective than the unsupervised baseline.
Table 2: AUC results under the hard setting, where models are trained with one known anomaly class and tested to detect the rest of all other anomaly classes. Each data subset is named by the known anomaly class.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Data Subset</th>
<th>One Training Anomaly Examples</th>
<th>Ten Training Anomaly Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>KDAF</td>
<td>DevNet</td>
</tr>
<tr>
<td>Color</td>
<td>0.78±0.006</td>
<td>0.71±0.005</td>
<td>0.46±0.027</td>
</tr>
<tr>
<td>Cut</td>
<td>0.76±0.005</td>
<td>0.66±0.005</td>
<td>0.64±0.045</td>
</tr>
<tr>
<td>Hole</td>
<td>0.75±0.010</td>
<td>0.72±0.007</td>
<td>0.59±0.142</td>
</tr>
<tr>
<td>Metal</td>
<td>0.83±0.010</td>
<td>0.81±0.012</td>
<td>0.70±0.028</td>
</tr>
<tr>
<td>Thread</td>
<td>0.75±0.010</td>
<td>0.91±0.044</td>
<td>0.84±0.005</td>
</tr>
<tr>
<td>Metal</td>
<td>0.77±0.012</td>
<td>0.64±0.010</td>
<td>0.82±0.006</td>
</tr>
<tr>
<td>Metal_melt</td>
<td>0.75±0.014</td>
<td>0.90±0.025</td>
<td>0.82±0.015</td>
</tr>
<tr>
<td>Metal_melt</td>
<td>0.75±0.014</td>
<td>0.90±0.025</td>
<td>0.82±0.015</td>
</tr>
<tr>
<td>Metal_melt</td>
<td>0.75±0.014</td>
<td>0.90±0.025</td>
<td>0.82±0.015</td>
</tr>
</tbody>
</table>

KDAD has significantly improved generalizability and largely outperforms KDAD even under the one-shot scenarios. This generalizability is further supported by our preliminary results in cross-domain anomaly detection in Appendix C.3.

4.5. Ablation Study

Importance of Each Abnormality Learning. Tab. 3 shows the importance of each abnormality learning in DRA using ten training anomaly examples.

DRA1A is the variant of DRA that only learns the known anomaly using the seen anomalies. It performs comparably well to the two best contenders, DevNet and SAOE, but limiting to the known abnormality learning, it results in less effective results than the other variants, especially on the hard setting. DevRAA builds upon DRA1A with the addition of pseudo abnormality learning. Compared with DRA1A, the inclusion of the pseudo abnormality largely improves the performance on most datasets. On some datasets, such as Hyper-Kvasir on the hard setting, the performance of DevRAA drops, which may be due to the discrepancy between the pseudo and real anomalies.

DRA3Ar and DRA3An extend DevRAA with one additional head: DRA3Ar attempts to learn the latent residual abnormality, without the support of the holistic normality representations that DRA3An is specifically designed to learn. DRA3Ar and DRA3An further largely improves DRA2A, but both of them are much less effective than our full model DRA. This demonstrates that g_A and g_B need to be fused to effectively learn the latent residual abnormality.

Importance of Disentangled Abnormalities. Fig. 3 (Left) shows the results of non-disentangled, partially, and fully disentangled abnormality learning under the general setting, where the non-disentangled method is multi-classification with normal samples, seen and pseudo anomaly classes, the partially disentangled method is the variant of DRA that learns disentangled seen and pseudo abnormalities only, and the fully disentangled method is our model DRA. The results show that disentangled abnormality learning helps largely improve the detection performance across all application domains.

4.6. Sensitivity Analysis

Sensitivity w.r.t. the Source of Pseudo Anomalies. There are diverse ways to create pseudo anomalies, as shown in previous studies [17, 25, 41, 53] that focus on anomaly-free training data. We instead investigate the effect of these methods under our open-set supervised AD model DRA. We evaluate three data augmentation methods, including CutMix [66], CutPaste-Scar (CP-Scar) [25] and CutPaste-Mix (CP-Mix) that utilizes both CutMix and Cut-CP, and two outlier exposure methods that respectively
use samples from MVtec AD [3] and medical dataset LAG [26] as the pseudo anomalies. When using MVtec AD, we remove the classes that overlap with the training/test data; LAG does not have any overlapping with our datasets. Since pseudo anomalies are used mainly to enhance the generalization to unseen anomalies, we focus on the four hard setting datasets in our ablation study in Tab. 3.

The results are shown in Tab. 4, from which it is clear that the data augmentation-based pseudo anomaly creation methods are generally more stable and much better than the external data-based methods on non-medical datasets. On the other hand, the external data method is more effective on medical datasets, since the augmentation methods often fail to properly simulate the lesions. The LAG dataset provides more application-relevant features and enables DRA to achieve the best results on Hyper-Kvasir.

**Sensitivity w.r.t. the Reference Size in Latent Residual Abnormality Learning.** Our residual latent abnormality learning head requires to sample a fixed number $N_r$ of normal training images as reference data. We evaluate the sensitivity of our method using different $N_r$ and report the AUC results in Fig. 3 (Right). Using one reference image is generally sufficient to learn the residual anomalies. Increasing the reference size to five helps further improve the detection performance, but increasing the size to ten is not consistently helpful. $N_r = 5$ is generally recommended, which is the default setting in DRA in all our experiments.

5. Conclusions and Discussions

This paper proposes the framework of learning disentangled representations of abnormalities illustrated by seen anomalies, pseudo anomalies, and latent residual-based anomalies, and introduces the DRA model to effectively detect both seen and unseen anomalies. Our comprehensive results in Tabs. 1 and 2 justify that these three disentangled abnormality representations can complement each other in detecting the largely varying anomalies, substantially outperforming five SotA unsupervised and supervised anomaly detectors by a large margin, especially on the challenging cases, e.g., having only one training anomaly example, or detecting unseen anomalies.

The studied problem is largely under-explored, but it is very important in many relevant real-world applications. As shown by the results in Tabs. 1 and 2, there are still a number of major challenges requiring further investigation, e.g., generalization from smaller anomaly examples from fewer classes, of which our model and comprehensive results provide a good baseline and extensive benchmark results.
References


