Explicit Boundary Guided Semi-Push-Pull Contrastive Learning for Supervised Anomaly Detection

Xincheng Yao[1], Ruoqi Li[1], Jing Zhang[3], Jun Sun[1], Chongyang Zhang[1,2]*
[1]School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[3]Research Institute of Systems Engineering, Academy Military Science, Beijing, China
{i-Dover, nilponi, junsun, sunny_zhang}@sjtu.edu.cn[1], jingzhang@163.com.cn[3]

Abstract

Most anomaly detection (AD) models are learned using only normal samples in an unsupervised way, which may result in ambiguous decision boundary and insufficient discriminability. In fact, a few anomaly samples are often available in real-world applications, the valuable knowledge of known anomalies should also be effectively exploited. However, utilizing a few known anomalies during training may cause another issue that the model may be biased by those known anomalies and fail to generalize to unseen anomalies. In this paper, we tackle supervised anomaly detection, i.e., *we learn AD models using a few available anomalies with the objective to detect both the seen and unseen anomalies. We propose a novel explicit boundary guided semi-push-pull contrastive learning mechanism, which can enhance model's discriminability while mitigating the bias issue. Our approach is based on two core designs: First, we find an explicit and compact separating boundary as the guidance for further feature learning. As the boundary only relies on the normal feature distribution, the bias problem caused by a few known anomalies can be alleviated. Second, a boundary guided semi-push-pull loss is developed to only pull the normal features together while pushing the abnormal features apart from the separating boundary beyond a certain margin region. In this way, our model can form a more explicit and discriminative decision boundary to distinguish known and also unseen anomalies from normal samples more effectively. Code will be available at https://github.com/xcyao00/BGAD.*

1. Introduction

Anomaly detection (AD) has received widespread attention in diverse domains, such as industrial defect inspection [4,8,10,50] and medical lesion detection [12,38]. Most previous anomaly detection methods [1,3,5,6,8,10,15,30,33,47,50–52] are unsupervised and pay much attention to normal samples while inadvertently overlooking anomalies, because it is difficult to collect sufficient and all kinds of anomalies. However, learning only from normal samples may limit the discriminability of the AD models [12,24]. As illustrated in Figure 1(a), without anomalies, the decision boundaries are generally implicit and not discriminative enough. The *insufficient discriminability* issue is a common issue in unsupervised anomaly detection due to the lack of knowledge about anomalies. In fact, a few anomalies are usually available in real-world applications, which can be exploited effectively to address or alleviate this issue.

Recently, methods that can be called semi-supervised AD [27,34,36] or AD with outlier exposure [16,17] begin to focus on those available anomalies. These methods attempt to learn knowledge from anomalies by one-class classification with anomalies as negative samples [34,36] or by supervised binary classification [16,17] or by utilizing the deviation loss to optimize one anomaly scoring network [27]. They show a fact that the detection performance can be improved significantly even with a few anomalies. However, the known anomalies can't represent all kinds of anomalies. These methods may be biased by the known anomalies and fail to generalize to unseen anomalies (see Figure 5).

Therefore, to address the two above issues, we tackle supervised anomaly detection [12], in which a few known anomalies can be effectively exploited to train discriminative AD models with the objective to improve detection performance on the known anomalies and generalize well to unseen anomalies. Compared with unsupervised AD, supervised AD is more meaningful for real-world AD applications, because the detected anomalies can be used to further improve the discriminability and generalizability of the model. To this end, we propose a novel **B**oundary **G**uided **A**nomaly **D**etection (**BGAD**) model, which has two core

*Corresponding Author.

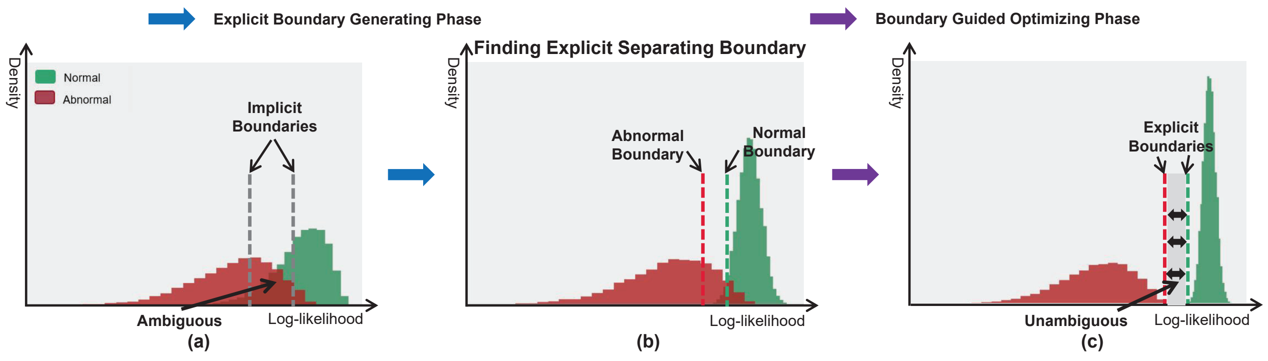


Figure 1. Conceptual illustration of our method. (a) In most unsupervised AD models, the anomaly score distribution usually has ambiguous regions, which makes it difficult to get one ideal decision boundary. *E.g.*, the left boundary will cause many false negatives, while the right boundary may induce many false positives. (b) With the normalized normal feature distribution, a pair of explicit and compact (close to the normal distribution) boundaries can be obtained easily. (c) With the proposed BG-SPP loss, boundary guided optimizing can be implemented to obtain an unambiguous anomaly score distribution: a significant gap between the normal and abnormal distributions.

designs as illustrated in Figure 1: explicit boundary generating and boundary guided optimizing.

• **Explicit Boundary Generating.** We first employ normalizing flow [14] to learn a normalized normal feature distribution, and obtain an explicit separating boundary, which is close to the normal feature distribution edge and controlled by a hyperparameter β (*i.e.*, the normal boundary in Figure 1(b)). The obtained explicit separating boundary only relies on the normal distribution and has no relation with the abnormal samples, thus the bias problem caused by the insufficient known anomalies can be mitigated.

• **Boundary Guided Optimizing.** After obtaining the explicit separating boundary, we then propose a boundary guided semi-push-pull (BG-SPP) loss to exploit anomalies for learning more discriminative features. With the BG-SPP loss, only the normal features whose log-likelihoods are smaller than the boundary are pulled together to form a more compact normal feature distribution (semi-pull); while the abnormal features whose log-likelihoods are larger than the boundary are pushed apart from the boundary beyond a certain margin region (semi-push).

In this way, our model can form a more explicit and discriminative separating boundary and also a reliable margin region for distinguishing anomalies more effectively (see Figure 1(c), 6). Furthermore, rarity is a critical problem of anomalies and may cause feature learning inefficient. We thus propose RandAugment-based Pseudo Anomaly Generation, which can simulate anomalies by creating local irregularities in normal samples, to tackle the rarity challenge.

In summary, we make the following main contributions:

1. We propose a novel Explicit Boundary Guided supervised AD modeling method, in which both normal and abnormal samples are exploited effectively by well-designed explicit boundary generating and boundary guided optimizing. With the proposed AD method, higher discriminability and lower bias risk can be achieved simultaneously.

2. To exploit a few known anomalies effectively, we propose a BG-SPP loss to pull together normal features while pushing abnormal features apart from the separating boundary, thus more discriminative features can be learned.

3. We achieve SOTA results on the widely-used MVTecAD benchmark, with the performance of 99.3% image-level AUROC and 99.2% pixel-level AUROC.

2. Related Work

Unsupervised Approaches. Most anomaly detection methods are unsupervised and only learn from normal samples, such as AutoEncoder [6, 26, 49], GAN [1, 29, 37, 39, 40, 53] and one-class-classification (OCC) based methods [35,42,46,50]. Recently, most superior methods utilize pre-trained deep models, such as DeepKNN [3], GaussianAD [31], SPADE [8], PaDiM [10] and PatchCore [32]. There are also some anomaly detection methods based on knowledge distillation [5, 38, 47], feature reconstruction [48], and normalizing flows [15, 33, 51, 55]. Our method is significantly different from these works [15, 33, 51, 55] in the following three aspects: (1) **New Motivation**: our work aims to learn knowledge from a few anomalies to address the insufficient discriminability issue and also mitigate the bias issue. (2) **Novel Method**: we only use the normalizing flow model as a basic likelihood estimation network, and are the first to propose Explicit Separating Boundary and BG-SPP loss to achieve higher discriminability and also lower bias. (3) **Different Task**: [55] focuses on out-of-distribution detection and [15, 33, 51] focus on unsupervised anomaly detection (AD), whereas our work focuses on supervised AD.

Supervised Approaches. Currently, a few existing works are similar to ours, *i.e.*, AD with outlier exposure [16, 17] and deep semi-supervised AD [24, 34, 36]. In [16], Hendrycks, *et al.* term random nature images from the large scale datasets that are likely not nominal as outlier exposure, and explore how to utilize such data to improve unsupervised AD. The method presented in [17] utilizes thousands of OE samples to achieve state-of-the-art results on standard image AD benchmarks. DeepSAD [36] is the first deep model utilizing a few anomalies by generalizing the unsupervised DeepSVDD [35] method to a semi-supervised AD setting. In [34], Ruff, *et al.* further modify the DeepSAD based on cross-entropy classification that concentrates nominal samples, this modification significantly improves the performance of DeepSAD. FCDD proposed in [24] extends the pseudo-Huber loss in [34] to construct a semi-supervised anomaly localization framework. Some works [27, 28] utilize the deviation loss to optimize an anomaly scoring network, in which the anomaly scores of normal samples are imposed to approximate scalar scores drawn from the prior while that of anomaly samples are enforced to have significant deviations from these normal scores. However, these methods simply push abnormal features apart from the normal patterns as far as possible, which may cause bias in the model for the known anomalies. The recent work DRA [12] is the most similar to ours, which also considers the model's generalization to unseen anomalies. The DRA model can learn disentangled representations of anomalies to enable generalizable detection.

3. Our Proposed Approach

Problem Statement. Different from the general unsupervised AD setting, the training set of supervised AD is composed of normal images and a few anomalies, denoted as $\mathcal{I}_{train} = \mathcal{I}^n \cup \mathcal{I}^a$, where $\mathcal{I}^n = \{I_i^n\}_{i=1}^{N_0}$ and $\mathcal{I}^a = \{I_j^a\}_{j=1}^{M_0} (M_0 \ll N_0)$ indicate the collection of normal samples and abnormal samples. The M_0 anomalies are randomly sampled from the seen anomaly classes $\mathcal{S}_s \subset \mathcal{S}$, where $\mathcal{S} = \mathcal{S}_s \cup \mathcal{S}_u$ means all the seen and unseen anomaly classes. The goal is to learn a model $m : \mathcal{I} \to \mathbb{R}$ that can assign larger anomaly scores to both seen and unseen anomalies than normal samples.

Overview. Figure 2 overviews our proposed method. The model consists of four parts: Feature Extractor $f : \mathcal{I} \to \mathcal{X}$, Conditional Normalizing Flow (CNFlow) $\varphi_\theta : \mathcal{X} \to \mathcal{Z}$, Explicit Boundary Generating and Boundary Guided Optimizing. We refer the features extracted by the feature extractor as input features for CNFlow, and denote these features as $\mathcal{X} = \mathcal{X}^n \cup \mathcal{X}^a$, where $\mathcal{X}^n = \{x_i^n\}_{i=1}^{N}$ and $\mathcal{X}^a = \{x_j^a\}_{j=1}^{M}$ are normal and abnormal features, respectively. The training procedure can be divided into two phases as shown in Figure 2: explicit boundary generating and boundary guided optimizing. In the testing procedure, the CNFlow can assign corresponding log-likelihoods for input features, and the log-likelihoods can be converted to anomaly scores (see Sec. 3.2).

3.1. Learning Normal Feature Distribution by Normalizing Flow

In order to find one anomaly-independent separating boundary, one simplified distribution of normal features should be learned firstly. Normalizing flow [13, 14] is employed to learn normal feature distribution in our method.

Conditional Normalizing Flow. Formally, we denote $\varphi_\theta : \mathcal{X} \in \mathbb{R}^d \to \mathcal{Z} \in \mathbb{R}^d$ as our normalizing flow. It is built as a composition of coupling layers [13] such that $\varphi_\theta = \varphi_L \circ \cdots \circ \varphi_2 \circ \varphi_1$, where θ is the trainable parameters and L is the total number of layers. Defining d-dimensional input and output features of normalizing flow as $y_0 = x \in \mathcal{X}$ and $y_L = z \in \mathcal{Z}$, the latents can be computed as $y_l = \varphi_l(y_{l-1})$, where $\{y_l\}_{l=1}^{L-1}$ are the intermediate outputs. The input distribution estimated by model $p_\theta(x)$ can be calculated according to the change of variables formula as follows [13, 14]:

$$\log p_\theta(x) = \log p_{\mathcal{Z}}(\varphi_\theta(x)) + \sum\nolimits_{l=1}^{L} \log\left|\det J_{\varphi_l}(y_{l-1})\right| \tag{1}$$

where $J_{\varphi_l}(y_{l-1}) = \frac{\partial \varphi_l(y_{l-1})}{\partial y_{l-1}}$ is the Jacobian matrix of the transformation φ_l at y_{l-1}, and det means determinant. Normalizing flow can approximate the feature distribution $p_{\mathcal{X}}$ with $p_\theta(x)$. The set of parameters θ is obtained by optimizing the log-likelihoods across the training distribution $p_{\mathcal{X}}$:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\, \mathbb{E}_{x \sim p_{\mathcal{X}}}[-\log p_\theta(x)] \tag{2}$$

The coupling layers in normalizing flow are usually implemented by fully connected layers, so the spatial position relationship will be destroyed because the 2D feature maps are flattened to 1D. To preserve the positional information, we follow [15] to add 2D-aware position embeddings.

Learning Normal Feature Distribution. We then employ normalizing flow to learn normal feature distribution by maximum likelihood optimization. The latent variable distribution $p_{\mathcal{Z}}(z), z \in \mathbb{R}^d$ can generally be assumed to obey the multivariate Gaussian distribution [15] as follows:

$$p_{\mathcal{Z}}(z) = (2\pi)^{-\frac{d}{2}} \det(\Sigma^{-\frac{1}{2}}) e^{-\frac{1}{2}(z-\mu)^T \Sigma^{-1}(z-\mu)} \tag{3}$$

where μ is the mean and Σ is the covariance. When training normal features, the latent variables for normal features can be assumed to obey $\mathcal{N}(0, \mathbf{I})$ for further simplicity. By replacing $p_{\mathcal{Z}}(z) = (2\pi)^{-\frac{d}{2}} e^{-\frac{1}{2}z^T z}$ in formula (1), the optimization objective in the formula (2) can be rewritten as:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\, \mathbb{E}_{x \sim p_{\mathcal{X}}}\Big[\frac{d}{2}\log(2\pi) + \frac{1}{2}\varphi_\theta(x)^T \varphi_\theta(x) - \sum\nolimits_{l=1}^{L} \log\left|\det J_{\varphi_l}(y_{l-1})\right|\Big] \tag{4}$$

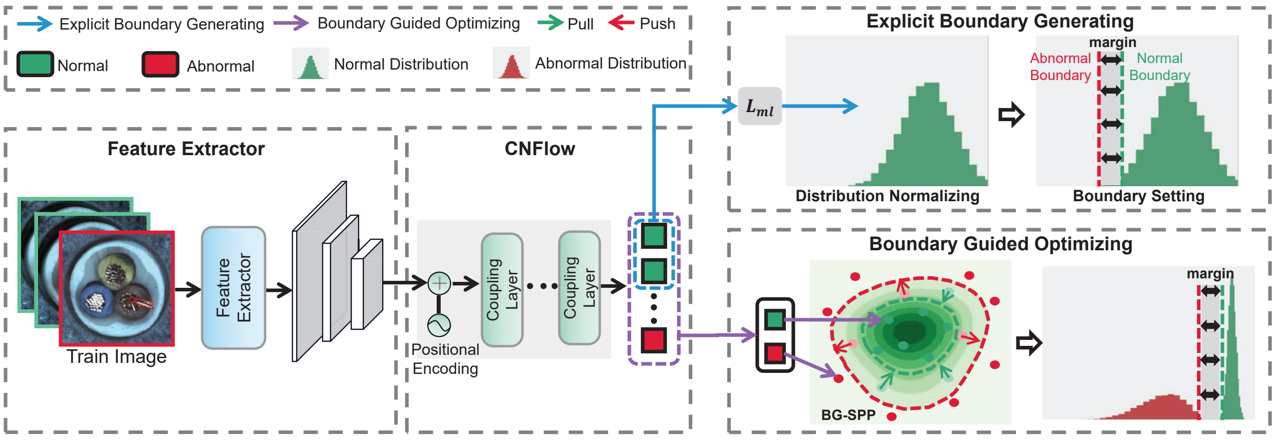


Figure 2. Model overview. The extracted feature maps are transformed into a latent space using a conditional normalizing flow (CNFlow), which is then used to generate an anomaly score for each feature. The training procedure can be divided into two phases: explicit boundary generating and boundary guided optimizing. In the first phase, only normal samples and ML loss (E.q.(5)) are utilized for model training to obtain a relatively stable normal log-likelihood distribution, and then one explicit separating boundary can be obtained based on the learned distribution. In the second phase, with the explicit separating boundary and the BG-SPP loss (E.q.(8)), both normal and abnormal samples are utilized for model training to learn more discriminative features.

The maximum likelihood loss function for learning normal feature distribution can be defined as:

$$\mathcal{L}_{ml} = \mathbb{E}_{x\in\mathcal{X}^n}\Big[\frac{d}{2}\log(2\pi) + \frac{1}{2}\varphi_\theta(x)^T\varphi_\theta(x) - \sum\nolimits_{l=1}^{L}\log\left|\det J_{\varphi_l}(y_{l-1})\right|\Big] \quad (5)$$

3.2. Finding an Explicit and Compact Separating Boundary

With the learned normal feature distribution, we can further find one explicit and compact separating boundary. However, due to the high dimensional characteristics of the features, we therefore consider finding the boundary from the anomaly score distribution. Since the loglikelihoods generated by the CNFlow can be equivalently converted to anomaly scores, we select the boundary on the log-likelihood distribution.

Anomaly Scoring. The advantage of normalizing flow is that we can estimate the exact log-likelihood $\log p(x)$ for each input feature x as follows:

$$\log p(x) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\varphi_\theta(x)^T\varphi_\theta(x) + \sum\nolimits_{l=1}^{L}\log\left|\det J_{\varphi_l}(y_{l-1})\right| \quad (6)$$

With the estimated log-likelihood $\log p(x)$, we can convert it to likelihood via exponential function. As we maximize log-likelihoods for normal features in E.q.(5), the likelihood can directly measure the normality. Thus, we can generate the anomaly score as follows:

$$s(x) = 1 - \exp(\log p(x)) \quad (7)$$

where the $s(x)$ means the anomaly score of x. Because exponential function is monotonic, the log-likelihood can be equivalently converted to the anomaly score. Thus, the separating boundary in log-likelihood distribution is equivalent to the boundary in anomaly score distribution.

Finding Explicit Separating Boundary. We then obtain the separating boundary based on log-likelihood distribution. We build the boundary through the following steps:

1. Building normal log-likelihood distribution. We can employ the log-likelihood estimation formulation in E.q.(6) to obtain all log-likelihoods of normal features $\mathcal{P}_n = \{\log p_i\}_{i=1}^N$. The $\mathcal{P}_n$ can be used to approximate the log-likelihood distribution of all normal features.

2. Finding explicit normal and abnormal boundary. How to find a suitable boundary is a dilemma. If we set the boundary too close to the distribution center, the samples in the tail of the normal distribution are more likely to be misclassified as abnormal. Meanwhile, if the boundary is far away from the distribution center, more anomalies would be determined as normal ones. Thus, we define a position hyperparameter β to control the distance from the boundary to the center. We select the β-th percentile (*e.g.*, $\beta = 1$) of sorted normal log-likelihood distribution as the normal boundary b_n, which also indicates the upper bound of the normal false positive rate is $\beta\%$. To make the feature learning more robust, we further introduce a margin hyperparameter τ (*e.g.*, $\tau = 0.1$) and define an abnormal boundary $b_a = b_n - \tau$ (see Figure 2). We provide hyperparameter sensitivity analysis for β and τ in App. Table 12, it shows that our model is not very sensitive to β and τ.

3.3. Learning More Discriminative Features by Boundary Guided Semi-Push-Pull

With the explicit normal and abnormal boundary, we propose a boundary guided semi-push-pull (BG-SPP) loss for more discriminative feature learning. Our BG-SPP loss can utilize the boundary b_n as the contrastive target (boundary guided), and only pull together normal features whose log-likelihoods are smaller than b_n (semi-pull) while pushing abnormal features whose log-likelihoods are larger than b_a apart from b_n at least beyond the margin τ (semi-push). The formulation of the BG-SPP loss is defined as:

$$\mathcal{L}_{bg-spp} = \sum_{i=1}^{N} |\min((\log p_i - b_n), 0)| + \sum_{j=1}^{M} |\max((\log p_j - b_n + \tau), 0)| \quad (8)$$

We define BG-SPP loss as ℓ_1 norm based formulation to encourage the sparse log-likelihood distribution in the margin region (b_a, b_n), because any log-likelihood $\log p_i$ fallen into the margin region (b_a, b_n) will increase the value of $\mathcal{L}_{bg-spp}$. Since the log-likelihoods can range from $(-\infty, 0]$, the large region makes it difficult to select the margin hyperparameter τ. Thus, we define a large enough normalizer α_n (*e.g.*, $\alpha_n = 10$) and employ it to normalize the log-likelihoods to the range $[-1, 0]$. We denote that the extremely small log-likelihoods (less than -1) can be excluded outside the BG-SPP loss in E.q.(8), as these log-likelihoods can be easily divided into anomalies. Therefore, minimizing the BG-SPP loss will encourage all log-likelihoods $\mathcal{P}$ to distribute in the regions $[-1, b_a]$ or $[b_n, 0]$. We further analyze the difference between our BG-SPP loss and the hinge loss in Appendix.

In the second training phase, the objective function is as follows:

$$\mathcal{L} = \mathcal{L}_{ml} + \lambda \mathcal{L}_{bg-spp} \quad (9)$$

3.4. Generalization Capability to Unseen Anomalies

Previous supervised AD methods are usually modeled as binary classification tasks regarding anomalies as positive samples. However, these models rely heavily on the known anomalies. Consequently, these models can overfit the known anomalies, failing to generalize to unseen anomalies. The serious bias issue can be mitigated by our method for three reasons: 1). The obtained explicit separating boundary only relies on the normal feature distribution and has no relation with the abnormal samples, this means that the final decision boundary mainly depends on the normal distribution rather than being affected greatly by the anomalies (see Table 4). 2). Our method still employs the normal distribution to determine anomalies, and our method can form a more compact and discriminative normal feature distribution (this is also conducive to detecting unseen anomalies), rather than the decision boundary between the normals and the known anomalies (see Figure 5). 3). The semi-push-pull mechanism in our method doesn't enforce the anomalies to deviate from the normal distribution as far as possible, which may lead to overfitting of the model for the known anomalies, but only pushes the anomalies outside the margin region (see ablations in Table 6).

3.5. RandAugment-based Pseudo Anomaly Generation

Anomalies are generally much less than normal samples, this may cause feature learning inefficient. We thus propose a RandAugment-based Pseudo Anomaly Generation (RPAG) strategy, which can simulate anomalies by randomly creating local irregularities, to improve the quantity and diversity of irregular patterns. The whole procedure is shown in Appendix (Figure 8) and summarized as follows:

Constructing Augmentation Sets. Adapted from RandAugment [9], we first select K available image transformations to construct an augmentation set $\mathcal{T} := \{T_1, \ldots, T_K | T_k : \mathcal{I} \rightarrow \mathcal{I}\}$: {Flip, Rotate, Transpose, Noise, Distortion, Brightness, Sharpness, Translate, Blur}.

Random Augmentation. We randomly select an augmentation subset $T_{RS} \subset \mathcal{T}$ containing S transformations to augment an abnormal sample: $(I^a)' = T_{RS}(I^a), I^a \in \mathcal{I}^a$.

Selecting Pasting Locations. Considering that anomalies generally only appear in object regions, thus to guarantee the semantics of simulated anomalies, we should limit the locations of simulated anomalies in the object regions. We adopt a foreground masking strategy, in which we can use grayscale binary thresholding algorithms to effectively locate the foreground objects $M_I = Binary(I^n), I^n \in \mathcal{I}^n$. Then, we can select a random pasting location $r_a = Rand(M_I = 1)$ from the object regions $(M_I = 1)$ to avoid anomalous regions generated in the background.

Cutting Anomalies. We cut the anomalous regions of the augmented abnormal sample: $\mathcal{R}_a = Cut((I^a)')$.

Pasting Anomalies. We paste the cropped anomalous regions back to the normal sample I^n at the selected location r_a to generate a simulated abnormal sample: $I_{sa} = Paste(I^n, \mathcal{R}_a, r_a)$.

In DRAEM [52] and NSA [41], the authors also attempt to utilize synthetic anomalies, we ablate our RPAG strategy with their strategies and show results in App. Table 9.

4. Experiments

4.1. Datasets and Metrics

Datasets. In this work, we focus on anomalies in real-world applications, such as industrial defect inspection and medical lesion detection. Specifically, we evaluate six real-world anomaly detection datasets, including four industrial defect inspection datasets: **MVTecAD** [4], **BTAD** [25], **AITEX** [43] and **ELPV** [11]; and two medical image datasets for detecting lesions on different organs: **BrainMRI** [38] and **HeadCT** [38]. A more detailed introduction to these datasets is provided in Appendix.

Evaluation Metrics. For evaluation, the standard metric in anomaly detection, AUROC, is used [4, 6, 40]. Image-level AUROC is used for anomaly detection and a pixel-level AUROC for evaluating anomaly localization. In order to weight ground-truth anomaly regions of various sizes equally, we also adopt the Per-Region-Overlap (PRO) curve metric proposed in [5].

4.2. Experimental Settings

Multi-Class Setting is designed to evaluate the performance of AD models in detecting the known anomaly classes. In this setting, the known anomalies are a few abnormal samples randomly drawn from existing anomaly classes in the test set. Then, we carefully exclude these added abnormal samples from the test set during testing.

One-Class Setting is designed to evaluate the generalizability of AD models in detecting unseen anomaly classes. In this setting, the known anomalies are randomly sampled only from one anomaly class, and all anomaly samples of this class are removed from the test set to ensure that the test set only contains unseen anomaly classes.

As anomalies are usually rare, our BGAD is trained with ten random abnormal samples per category by default (we also generate some pseudo anomalies by RPAG based on these known anomalies for training). After removing the known anomalies, the test set in our settings is different from the original test set. Thus, for a fair comparison, we re-run all the compared unsupervised and supervised AD methods with the publicly available implementations under the same experimental setup as our BGAD. Other implementation details can be found in Appendix.

4.3. Results under the Multi-Class Setting

MVTecAD. We compare our BGAD with the SOTA AD methods, including unsupervised (PaDiM [10], DRAEM [52], MSFD [47], PatchCore [32] and CFA [20]) and supervised methods (FCDD [24], DevNet [27] and DRA [12]). The detailed comparison results of all categories are shown in Table 1 and Table 2. We also implement a variant BGAD$^{w/o}$, which is optimized by the first part of the BG-SPP loss without anomalies. Compared with NFAD, our BGAD$^{w/o}$ can achieve better results, this shows that our boundary guiding mechanism is also beneficial to improve the AD performance under the unsupervised setting. Our BGAD reaches the best performance under all three evaluation metrics and can further surpass unsupervised baseline NFAD by 2.5% and 1.3% AUROC, and 3.0% PRO. The largest gain in PRO demonstrates that our BGAD is more suitable for anomaly localization to better locate the anomalous areas (see Figure 4). Compared with supervised AD methods, our method can also surpass these SOTA methods significantly. This shows that our boundary guiding mechanism can exploit a few known anomalies more effectively to learn a more discriminative AD model.

BTAD. We compare our BGAD with three baseline methods reported in [25]: AE-MSE, AE-SSIM, and VT-ADL. Following [25], we evaluate anomaly localization performance and report pixel-level AUROCs. The results are shown in Table 3. Our BGAD can achieve 98.6% mean pixel-level AUROC, which surpasses other methods by a large margin (8.6%) and surpasses unsupervised baseline NFAD by 0.8%. What's more, BGAD can achieve 82.4% PRO which surpasses unsupervised NFAD by 4.6%.

Other Datasets. Here, we compare our BGAD with six recent and closely related SOTA methods reported in [12]: unsupervised KDAD [47], and supervised DevNet [27], FLOS [22], SAOE [44], MLEP [23] and DRA [12]. Following [12], we evaluate anomaly detection performance and report image-level AUROCs. Same as [12], all models are trained with one known anomaly sample. The comparison results are shown in Figure 3. Our model can achieve the best AUROC performance on the two industrial defect inspection datasets (AITEX and ELPV), and comparable results with the SOTA methods on the two medical lesion detection datasets (BrainMRI and HeadCT).

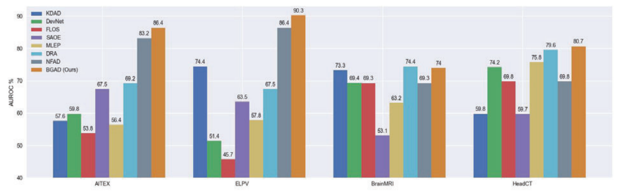


Figure 3. AUROC results on the AITEX, ELPV, BrainMRI, and HeadCT datasets.

4.4. Results under the One-Class Setting

The comparison results under the one-class setting are shown in Table 4, and more results are shown in Table 10 in Appendix.

Comparison to Supervised AD Methods. In both Table 4 and Table 10, compared to the competing methods, our method is the best performer on the diverse application datasets. On the mean image-level performance, our model achieves about 3.3%-12.9% mean AUROC increase compared to the best contender. This shows substantially better generalizability of our model in detecting unseen anomaly classes than the other supervised AD methods.

Comparison to Unsupervised Baseline. Since supervised AD methods are often biased by the seen anomaly

Table 1. AUROC and PRO results under the multi-class setting on the MVTecAD dataset. $\cdot/\cdot$ means pixel-level AUROC and PRO. The results of our model are averaged over three independent runs. ‡We implement NFAD as a baseline model, which has the same network structure as our BGAD but without explicit boundary guidance. $*$ We remove the added abnormal samples from the test set, and reproduce all the compared methods under the same experimental setup as our BGAD for a fair comparison.

	Category	Unsupervised AD Methods							Supervised AD Method
		DRAEM* [52]	PaDiM* [10]	MSFD* [47]	PatchCore* [32]	CFA* [20]	NFAD‡	BGAD$^{w/o}$ (Ours)	BGAD (Ours)
Textures	Carpet	0.954/0.947	0.983/0.946	0.990/0.958	0.985/0.959	0.989/0.943	0.994/0.983	0.994/0.982	**0.996**±0.0002/**0.989**±0.0004
	Grid	0.997/0.984	0.963/0.894	0.986/0.937	0.974/0.891	0.977/0.932	0.993/0.980	0.994/0.980	**0.995**±0.0002/**0.986**±0.0001
	Leather	0.992/0.981	0.984/0.966	0.978/0.924	0.992/0.974	0.991/0.958	0.997/0.994	0.997/0.994	**0.998**±0.0001/**0.994**±0.0003
	Tile	0.994/0.949	0.958/0.884	0.952/0.841	0.960/0.939	0.960/0.860	0.969/0.929	0.968/0.927	**0.994**±0.0077/**0.978**±0.0021
	Wood	0.962/0.935	0.963/0.891	0.953/0.925	0.968/0.857	0.948/0.882	0.969/0.957	0.970/0.957	**0.982**±0.0053/**0.970**±0.0007
Objects	Bottle	0.993/0.955	0.978/0.936	0.985/0.940	0.986/0.956	0.987/0.944	0.988/0.965	0.989/0.964	**0.994**±0.0009/**0.971**±0.0011
	Cable	0.961/0.910	0.979/0.973	0.972/0.922	0.986/**0.980**	**0.987**/0.931	0.975/0.944	0.980/0.968	0.986±0.0010/0.977±0.0030
	Capsule	0.869/0.901	0.980/0.924	0.979/0.878	0.990/0.946	0.989/0.943	0.989/0.952	0.992/0.959	**0.992**±0.0021/**0.964**±0.0033
	Hazelnut	0.997/0.985	0.980/0.951	0.982/0.968	0.988/0.924	0.986/0.953	0.984/0.976	0.985/0.976	**0.995**±0.0040/**0.982**±0.0028
	Metal nut	0.992/0.935	0.979/0.929	0.972/0.985	0.986/0.935	0.987/0.918	0.971/0.942	0.976/0.948	**0.996**±0.0003/**0.970**±0.0012
	Pill	0.979/0.959	0.978/0.957	0.971/0.929	0.983/0.947	0.986/0.965	0.976/0.978	0.980/0.980	**0.996**±0.0002/**0.988**±0.0005
	Screw	0.992/0.965	0.974/0.923	0.983/0.924	0.984/0.928	0.985/0.944	0.988/0.945	0.992/0.960	**0.993**±0.0003/**0.968**±0.0010
	Toothbrush	0.970/0.940	0.980/0.894	0.986/0.877	0.987/0.939	0.989/0.894	0.983/0.904	0.986/0.938	**0.995**±0.0003/**0.961**±0.0026
	Transistor	0.970/0.935	0.983/0.967	0.886/0.781	0.964/0.967	**0.985**/0.960	0.923/0.788	0.940/0.830	0.983±0.0005/**0.972**±0.0015
	Zipper	0.984/0.966	0.978/0.948	0.981/0.935	0.986/0.963	0.988/0.944	0.986/0.957	0.987/0.957	**0.993**±0.0003/**0.977**±0.0002
	Mean	0.969/0.947	0.976/0.932	0.970/0.915	0.981/0.940	0.982/0.931	0.979/0.946	0.982/0.955	**0.992**±0.0007/**0.976**±0.0006
	Image-level Mean	0.978	0.975	0.964	0.988	0.989	0.968	0.974	**0.993**±0.0012

Table 2. AUROC and PRO results under the multi-class setting on the MVTecAD dataset. $*$ Please see explanation in Table 1.

Category	Supervised AD Methods (Ten Abnormal Samples)			
	FCDD* [24]	DevNet* [27]	DRA* [12]	BGAD (Ours)
Carpet	0.981/0.952	-/-	-/-	**0.996**±0.0002/**0.989**±0.0004
Grid	0.949/0.897	-/-	-/-	**0.995**±0.0002/**0.986**±0.0001
Leather	0.984/0.973	-/-	-/-	**0.998**±0.0001/**0.994**±0.0003
Tile	0.977/0.938	-/-	-/-	**0.994**±0.0077/**0.978**±0.0021
Wood	0.950/0.901	-/-	-/-	**0.982**±0.0053/**0.970**±0.0007
Bottle	0.966/0.939	-/-	-/-	**0.994**±0.0009/**0.971**±0.0011
Cable	0.963/0.980	-/-	-/-	0.986±0.0010/0.977±0.0030
Capsule	0.970/0.922	-/-	-/-	**0.992**±0.0021/**0.964**±0.0033
Hazelnut	0.970/0.958	-/-	-/-	**0.995**±0.0040/**0.982**±0.0028
Metal nut	0.966/0.934	-/-	-/-	**0.996**±0.0003/**0.970**±0.0012
Pill	0.975/0.960	-/-	-/-	**0.996**±0.0002/**0.988**±0.0005
Screw	0.963/0.925	-/-	-/-	**0.993**±0.0003/**0.968**±0.0010
Toothbrush	0.967/0.907	-/-	-/-	**0.995**±0.0003/**0.961**±0.0026
Transistor	0.942/0.935	-/-	-/-	0.983±0.0005/**0.972**±0.0015
Zipper	0.968/0.948	-/-	-/-	**0.993**±0.0003/**0.977**±0.0002
Mean	0.966/0.938	-/-	-/-	**0.992**±0.0007/**0.976**±0.0006
Image-level Mean	0.965	0.948	0.961	**0.993**±0.0012

Table 3. Pixel-level AUROC results on the BTAD dataset. $\cdot/\cdot$ means pixel-level AUROC and PRO. The results of our model are averaged over three independent runs.

Categories	AE-MSE	AE-SSIM	VT-ADL	NFAD	BGAD (Ours)
1	0.490	0.530	**0.990**	0.972/0.767	0.982±0.0027/**0.830**±0.0318
2	0.920	0.960	0.940	0.967/0.578	**0.979**±0.0018/**0.648**±0.0173
3	0.950	0.890	0.770	0.996/0.988	**0.998**±0.0003/**0.993**±0.0005
Mean	0.780	0.790	0.900	0.978/0.778	**0.986**±0.0015/**0.824**±0.0163

class, they even perform less effectively than the unsupervised baseline NFAD on most of the datasets. By contrast, our model can outperform the baseline NFAD across all the datasets. The comparison results validate that our model's better generalizability to unseen anomalies and the serious bias issue can be alleviated by our method.

4.5. Ablation Study

Experiments On Hard Subsets. The experimental results in Table 1 has already demonstrated the effectiveness

Table 4. AUROC results under the one-class setting, where models are trained with only one anomaly class and tested to detect other anomaly classes. $\cdot/\cdot$ means image-level and pixel-level AUROCs.

	Known Class	Baseline	Ten Training Anomaly Samples					
		NFAD	DevNet	FLOS	SAOE	MLEP	DRA	BGAD (Ours)
Carpet	Color	0.998/**0.993**	0.767/-	0.760/-	0.467/-	0.689/-	0.886/-	**1.000/0.993**
	Cut	**0.998**/0.995	0.819/-	0.688/-	0.793/-	0.653/-	0.922/-	**0.998/0.996**
	Hole	0.997/0.993	0.814/-	0.733/-	0.831/-	0.674/-	0.922/-	**0.998/0.995**
	Metal	0.998/0.993	0.863/-	0.678/-	0.883/-	0.764/-	0.933/-	**1.000/0.994**
	Thread	**1.000**/0.995	0.972/-	0.946/-	0.834/-	0.967/-	0.989/-	**1.000/0.996**
	Mean	0.998/0.994	0.847/-	0.761/-	0.762/-	0.751/-	0.935/-	**0.999/0.995**
Metal_nut	Bent	0.977/0.959	0.904/-	0.827/-	0.901/-	0.956/-	0.990/-	**1.000/0.972**
	Color	0.977/0.963	0.978/-	0.9788/-	0.879/-	0.945/-	0.967/-	**0.999/0.973**
	Flip	0.976/0.977	0.987/-	0.942/-	0.795/-	0.805/-	0.913/-	**0.995/0.982**
	Scratch	**1.000**/0.965	0.991/-	0.943/-	0.845/-	0.805/-	0.911/-	**1.000/0.972**
	Mean	0.983/0.966	0.965/-	0.922/-	0.855/-	0.878/-	0.945/-	**0.998/0.975**

Table 5. AUROC and PRO results on subsets from the MVTecAD dataset. The details of subset selection are provided in Appendix.

Dataset / Metric	MVTecAD		Hard Subsets		Unseen Subsets	
	NFAD	BGAD	NFAD	BGAD	NFAD	BGAD
Image AUROC	0.968	0.992(+2.5%)	0.948	0.984(**+3.6%**)	0.948	0.971(+**2.3**%)
Pixel AUROC	0.979	0.992(+1.3%)	0.960	0.986(**+2.6%**)	0.960	0.982(+**2.2**%)
PRO	0.946	0.976(+3.0%)	0.863	0.949(**+8.6%**)	0.863	0.930(+**6.7**%)

of our model. However, to further demonstrate the ability of our method to detect complex anomalies, we construct two more difficult subsets from the MVTecAD dataset and conduct experiments on these two subsets. The details of subset selection are provided in Appendix. The results are shown in Table 5. It can be found that the detection and localization performance gain on these hard subsets is larger than that on the original dataset with a margin of 1.1%, 1.3%, and 5.6% respectively. This ablation study demonstrates that our model is more beneficial for harder anomaly classes.

Generalization to Hard Subsets. We use the easy subsets as the training set and validate results on the hard subsets to explore the generalizability of the model. The easy subsets are formed by excluding the hard subsets mentioned

Table 6. AUROC results under the one-class setting. ·/· means image-level and pixel-level AUROCs.

Method	Category				
	Carpet	Metal_nut	Capsule	Screw	Transistor
NFAD (baseline)	0.998/0.994	0.983/0.966	0.941/0.990	0.885/0.989	0.984/0.929
BGAD†	0.998/0.994	0.997/0.927	0.868/0.963	0.823/0.980	0.933/0.847
BGAD (Ours)	**0.999/0.995**	**0.998/0.975**	**0.988/0.991**	**0.947/0.991**	**0.994/0.942**

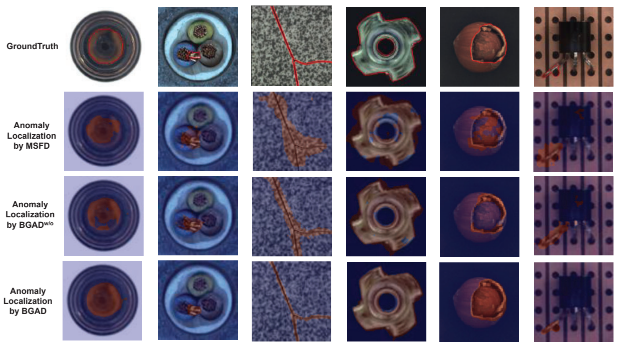
in the last paragraph from the original dataset. The experimental results are shown in Table 5. It can be found that even only trained with easy anomalies, our BGAD can generalize well to hard anomalies with performance gain by 2.3% and 2.2% AUROC, and 6.7% PRO.

Effect of Semi-Push-Pull Mechanism. We implement a variant of BGAD (termed as BGAD†), it doesn't utilize the BG-SPP loss while employing the conventional contrastive loss. The comparison results are shown in Table 6, and more results are in Appendix (Table 11). The BGAD† performs less effectively than the BGAD, and even worse than the baseline NFAD (especially for some complex categories, *e.g.*, Capsule, Screw, Transistor). The reason is that the BGAD†'s full pushing mechanism will encourage the anomalous features to deviate from the normal distribution at least a large enough bound, which may make the model more inclined to generate larger anomaly scores and thus lead to the model being easier to over-fit the known anomalies. Therefore, the BGAD† may generate larger anomaly scores for normal features, which will significantly reduce the AUROC metrics. However, the semi-push-pull mechanism in our BGAD only changes the ambiguous region, this has less impact on the full normal and abnormal distributions. Thus, the BG-SPP loss doesn't make the model have the inclination to generate larger anomaly scores, is more conducive to alleviating over-fitting of the model to the known anomalies.

4.6. Qualitative Results

We visualize some anomaly localization results in Figure 4 with the MVTecAD dataset. Our BGAD can generate more accurate anomaly localization maps (see columns of $\{1,3,4,5,6\}$ in Figure 4), or even generate anomaly maps better than ground truth (see columns of $\{2\}$ in Figure 4).

To illustrate the effectiveness of our method more intuitively, we visualize normal and abnormal feature distributions and log-likelihood distributions in Figure 5, 6. From Figure 5, it can be found that the supervised DevNet [27] is biased by the known anomalies, failing to distinguish unseen anomalies from the normal data. But our method can effectively mitigate this issue and generate more discriminative features than the unsupervised MSFD [47]. From Figure 6, it can be found that the ambiguous log-likelihood regions can be diminished by our BGAD.



GroundTruth
Anomaly Localization by MSFD
Anomaly Localization by BGAD$^{w/o}$
Anomaly Localization by BGAD

Figure 4. Qualitative results. The anomaly localization results generated by MSFD, BGAD$^{w/o}$, and BGAD are shown for comparison. In the first row, the areas enclosed by the red lines are ground-truth.

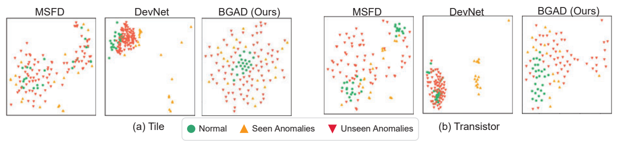


Figure 5. Feature distributions learned by unsupervised MSFD [47], supervised DevNet [27] and our BGAD.

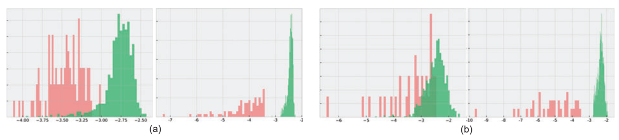


Figure 6. Log-likelihood histograms from (a) tile and (b) transistor category. Left is the log-likelihood histogram w/o anomaly samples, right is the log-likelihood histogram with anomaly samples.

5. Conclusion

We propose a novel and more discriminative AD model termed as BGAD to tackle the *insufficient discriminability* issue and the *bias* issue simultaneously. Compared with unsupervised AD models, our model can learn more discriminative features for distinguishing anomalies by exploiting a few anomalies effectively. Compared with supervised AD methods, our method can mitigate the bias issue with the explicit separating boundary and semi-push-pull mechanism. We hope our boundary guiding mechanism can inspire subsequent studies of supervised AD.

Acknowledgements

This work was supported in part by the National Natural Science Fund of China (61971281), the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and the Science and Technology Commission of Shanghai Municipality (22DZ2229005).

References

[1] Samet Akcay, Amir Atapour-Abarghouei, and Toby P. Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. *In ACCV*, page 622–637, 2018. 1, 2

[2] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv: 1907.02392*, 2019. 11, 12

[3] Liron Bergman, Niv Cohen, and Yedid Hoshen. Deep nearest neighbor anomaly detection. *arXiv preprint arXiv: 2002.10445*, 2020. 1, 2

[4] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad - a comprehensive real-world dataset for unsupervised anomaly detection. *In CVPR*, 2019. 1, 6, 11

[5] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. *In CVPR*, 2020. 1, 2, 6

[6] Paul Bergmann, Sindy Lowe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *In International Conference on Computational Vision Technologies and Applications*, 2019. 1, 2, 6

[7] Shuo Chen, Gang Niu, Chen Gong, Jun Li, Jian Yang, and Masashi Sugiyama 1 3. Large-margin contrastive learning with distance polarization regularizer. *In International Conference on Machine Learning*, 2017. 12

[8] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv: 2005.02357v3*, 2020. 1, 2

[9] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Pratical automated data augmentation with a reduced search space. *In CVPR*, 2020. 5

[10] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. *In 1st International Workshop on Industrial Machine Learning*, 2021. 1, 2, 6, 7

[11] Sergiu Deitscha, Vincent Christlein, Stephan Berger, Claudia Buerhop-Lutz, Andreas Maier, Florian Gallwitza, and Christian Riess. Automatic classification of defective photovoltaic module cells in electroluminescence images. *In Solar Energy*, pages 455–468, 2019. 6, 11

[12] Choubo Ding, Guansong Pang, and Chunhua Shen. Catching both gray and black swans: open-set supervised anomaly detection. *In CVPR*, 2022. 1, 3, 6, 7, 11, 14

[13] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *In International Conference on Learning Representations*, 2015. 3

[14] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *In International Conference on Learning Representations*, 2017. 2, 3, 11, 12

[15] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. *In IEEE Winter Conference on Application of Computer Vision*, 2022. 1, 2, 3, 11

[16] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *In International Conference on Learning Representations*, 2019. 1, 2

[17] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *In Conference and Workshop on Neural Information Processing Systems*, 2019. 1, 2, 3

[18] Jorn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. i-revnet: Deep invertible networks. *In International Conference on Learning Representations*, 2018. 11

[19] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *In Conference and Workshop on Neural Information Processing Systems*, 2019. 11, 12

[20] Sungwook Lee, Seunghyun Lee, and Byung Cheol Song. Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *arXiv preprint arXiv: 2206.04325*, 2022. 6, 7

[21] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. *In CVPR*, 2021. 11

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. *In ICCV*, 2017. 6

[23] Wen Liu, Weixin Luo, Zhengxin Li, Peilin Zhao, and Shenghua Gao1. Margin learning embedding prediction for video anomaly detection with a few anomalies. *In International Joint Conference on Artificial Intelligence*, 2019. 6

[24] Philipp Liznerski, Lukas Ruff, Robert A. Vandermeulen, Billy Joe Franks, Marius Kloft, and Klaus-Robert Muller. Explainable deep one-class classification. *In International Conference on Learning Representations*, 2021. 1, 2, 3, 6, 7

[25] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. Vt-adl: A vision transformer network for image anomaly detection and localization. *arXiv preprint arXiv:2104.10036*, 2021. 6, 11

[26] Duc Tam Nguyen, Zhongyu Lou, Michael Klar, and Thomas Brox. Anomaly detection with multiple-hypotheses predictions. *In International Conference on Machine Learning*, 2019. 2

[27] Guansong Pang, Choubo Ding, Chunhua Shen, and Anton van den Hengel. Explainable deep few-shot anomaly detection with deviation networks. *arXiv preprint arXiv:2108.00462*, 2021. 1, 3, 6, 7, 8

[28] Guansong Pang, Chunhua Shen, and Anton van den Hengel. Deep anomaly detection with deviation networks. *In Proc. ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, 2019. 3

[29] Stanislav Pidhorskyi, Ranya Almohsen, Donald A. Adjeroh, and Gianfranco Doretto. Generative probabilities novelty detection with adversarial autoencoders. *In Conference and Workshop on Neural Information Processing Systems*, 2018. 2

[30] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. Panda: Adapting pretrained features for anomaly detection and segmentation. *In CVPR*, 2021. 1

[31] Oliver Rippel, Patrick Mertens, and Dorit Merhof. Modeling the distribution of normal data in pre-trained deep features for anomaly detection. *arXiv preprint arXiv: 2005.14140*, 2020. 2

[32] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Scholkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. *In IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2, 6, 7

[33] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. *In IEEE Winter Conference on Application of Computer Vision*, 2021. 1, 2

[34] Lukas Ruff, Robert A. Vandermeulen, Billy Joe Franks, Klaus-Robert Muller, and Marius Kloft. Rethinking assumptions in deep anomaly detection. *arXiv preprint arXiv:2006.00339*, 2020. 1, 2, 3

[35] Lukas Ruff, Robert A. Vandermeulen, Nico Gornitz, Lucas Deecke, and Shoaib A. Siddiqui. Deep one-class classification. *In International Conference on Machine Learning*, 2021. 2, 3

[36] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. *In International Conference on Learning Representations*, 2021. 1, 2, 3

[37] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. *In IEEE Winter Conference on Application of Computer Vision*, 2018. 2

[38] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H. Rohban, and Hamid R. Rabiee. Multiresolution knowledge distillation for anomaly detection. *In IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 6

[39] Thomas Schlegl, Philipp Seebock, Sebastian M. Waldstein, Georg Langs, and Ursula Schmidt-Erfurthb. Fast unsupervised anomaly detection with generative adversarial networks. *In Medical Image Analysis*, 2017. 2

[40] Thomas Schlegl, Philipp Seebock, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. *In International Conference on Information Processing in Medical Imaging*, 2017. 2, 6

[41] Hannah M. Schluter, Jeremy Tan, Benjamin Hou, and Bernhard Kainz. Natural synthetic anomalies for self-supervised anomaly detection and localization. *In ECCV*, 2022. 5, 13, 14

[42] Bernhard Scholkopf, John C. Plattz, John Shawe-Taylory, Alex J. Smolax, and Robert C. Williamsonx. Estimating the support of a high-dimensional distribution. *In Neural Computation*, page 1443–1471, 2001. 2

[43] Javier Silvestre-Blanes1, Teresa Albero-Albero1, Ignacio Miralles, Rubén Pérez-Llorens, and Jorge Moreno. A public fabric database for defect detection methods and results. *In Autex Research Journal*, 2019. 6, 11

[44] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin†. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *In Conference and Workshop on Neural Information Processing Systems*, 2020. 6

[45] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *In International Conference on Machine Learning*, 2019. 11

[46] David M.J. Tax and Robert P.W. Duin. Support vector data description. *In Machine Learning*, pages 45–66, 2004. 2

[47] Guodong Wang, Shumin Han, Errui Ding, and Di Huang. Student-teacher feature pyramid matching for unsupervised anomaly detection. *In British Machine Vision Conference*, 2021. 1, 2, 6, 7, 8

[48] Jie Yang, Yong Shi, and ZhiQuan Qi. Dfr: Deep feature reconstruction for unsupervised anomaly segmentation. *arXiv preprint arXiv: 2012.0712*, 2020. 2

[49] Xincheng Yao, Chongyang Zhang, Ruoqi Li, Jun Sun, and Zhenyu Liu. One-for-all: Proposal masked cross-class anomaly detection. *In AAAI*, 2023. 2

[50] Jihun Yi and Sungroh Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. *In Asian Conference on Computer Vision*, 2021. 1, 2

[51] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv: 2111.07677*, 2021. 1, 2

[52] Vitjan Zavrtanik, Matej Kristan, and Danijel Skocaj. Draem: A discriminatively trained reconstruction embedding for surface anomaly detection. *In International Conference on Computational Vision*, 2021. 1, 5, 6, 7, 11, 13, 14

[53] Houssam Zenati, Manon Romain, Chuan Sheng Foo, Bruno Lecouat, and Vijay Ramaseshan Chandrasekhar. Adversarially learned anomaly detection. *In ICDM*, pages 727–736, 2018. 2

[54] Houssam Zenati, Manon Romain, Chuan Sheng Foo, Bruno Lecouat, and Vijay Ramaseshan Chandrasekhar. Pytorch image models. *https://github.com/rwightman/pytorch-image-models*, 2019. 11

[55] Ev Zisselman and Aviv Tamar. Deep residual flow for out of distribution detection. *In IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2