Generating and Reweighting Dense Contrastive Patterns for Unsupervised Anomaly Detection

Songmin Dai[1*], **Yifan Wu**[1*], **Xiaoqiang Li**[1†], **Xiangyang Xue**[2]

[1]School of Computer Engineering and Science, Shanghai University
[2]School of Computer Science, Fudan University
laodar@shu.edu.cn, VictorWu@shu.edu.cn, xqli@shu.edu.cn, xyxue@fudan.edu.cn

Abstract

Recent unsupervised anomaly detection methods often rely on feature extractors pretrained with auxiliary datasets or on well-crafted anomaly-simulated samples. However, this might limit their adaptability to an increasing set of anomaly detection tasks due to the priors in the selection of auxiliary datasets or the strategy of anomaly simulation. To tackle this challenge, we first introduce a prior-less anomaly generation paradigm and subsequently develop an innovative unsupervised anomaly detection framework named GRAD, grounded in this paradigm. GRAD comprises three essential components: (1) a diffusion model (PatchDiff) to generate contrastive patterns by preserving the local structures while disregarding the global structures present in normal images, (2) a self-supervised reweighting mechanism to handle the challenge of long-tailed and unlabeled contrastive patterns generated by PatchDiff, and (3) a lightweight patch-level detector to efficiently distinguish the normal patterns and reweighted contrastive patterns. The generation results of PatchDiff effectively expose various types of anomaly patterns, e.g. structural and logical anomaly patterns. In addition, extensive experiments on both MVTec AD and MVTec LOCO datasets also support the aforementioned observation and demonstrate that GRAD achieves competitive anomaly detection accuracy and superior inference speed.

Introduction

Image anomaly detection plays a crucial role in various fields, including industrial product defect detection, medical image lesion detection, security screening using X-ray images, and video surveillance (Zheng et al. 2018; Bergmann et al. 2019; Sato et al. 2018; Kiran, Thomas, and Parakkal 2018; Akcay, Atapour-Abarghouei, and Breckon 2018). However, securing real-world anomalous data for training is typically challenging and scarce due to the inability to cover a sufficiently diverse range of potential anomaly patterns. Consequently, the setting of one-class learning, which employs only normal samples for model training, has proven to be better suited for most industrial anomaly detection tasks (Bergmann et al. 2019, 2022). In recent years, many high-accuracy industrial anomaly detection methods heavily rely on ImageNet (Deng et al. 2009) pretrained feature extractor. Nevertheless, such reliance may limit their generalization capabilities in scenarios (Bergmann et al. 2022) where ImageNet pretrained features are insufficiently informative, or on other types of image-like data (Bergmann et al. 2021; Horwitz and Hoshen 2023). Additionally, some methods have achieved promising results on the MVTec AD (Bergmann et al. 2019) without using pretrained feature extractors. These methods utilize manually-selected external out-of-distribution (OOD) datasets (Liznerski et al. 2021) or carefully designed anomaly-simulated data to sample anomaly patterns (Li et al. 2021; Zavrtanik, Kristan, and Skočaj 2021; Yang et al. 2023). However, previous anomaly acquisition strategies can be considered as ad-hoc solutions that overly rely on priors or visual inspection of test images, such as in MVTec AD, where most anomalies are low-level structure anomalies (*e.g.*, scratches, dents, and contaminations). Such reliance may cause these strategies to fail in detecting other types of anomalies, such as logical anomalies recently proposed in the MVTec LOCO (Bergmann et al. 2022). These logical anomalies are represented as violations of logical constraints in

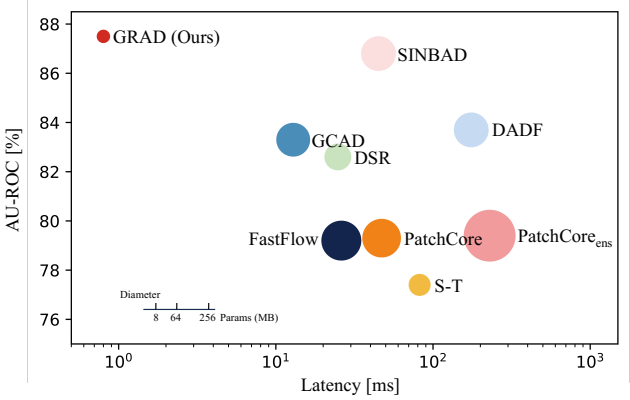

Figure 1: Anomaly detection performance vs. latency per image on an NVIDIA Tesla V100 GPU. Each bubble's area is proportional to the number of parameters in each detector, and each AU-ROC value is an average of the image-level detection AU-ROC values on MVTec LOCO (Bergmann et al. 2022).

*The first two authors contributed equally to this paper.
†Corresponding author.

images, which not only challenges the ananomaly simulation-based methods but also the pretrained representations by auxiliary datasets. Therefore, it becomes necessary to devise image anomaly detection techniques that are independent of both pretrained Imagenet feature extractors and ad-hoc anomaly acquisition strategies.

In this paper, we introduce a novel framework named **GRAD** (**G**enerating and **R**eweighting dense contrastive patterns for unsupervised **A**nomaly **D**etection), which achieves SOTA performance in both anomaly detection accuracy and inference runtime, as depicted in Fig. 1. We first put forward a novel anomaly generation paradigm: retaining the structure information within each small patch of the image while disregarding the global structure information of the whole image. Based on this paradigm, we design an anomaly generator called PatchDiff. This generator enforces a constraint on the receptive field size of the diffusion model (Ho, Jain, and Abbeel 2020) and removes the attention layers (Dong, Cordonnier, and Loukas 2021), thus ensuring that only the local structure within each patch is retained, while the global structure is discarded. As illustrated in Fig. 2, with different sizes of the receptive field, PatchDiff can generate diverse dense contrastive patterns that cover a range of anomaly types, *e.g.*, the structural and logical anomalies proposed in MVTec LOCO. Subsequently, we expect to utilize the generated local anomaly patterns to learn a patch-level anomaly detector. However, the contrastive patterns generated by PatchDiff may also be normal and we cannot provide patch-wise ground truth for them. Consequently, the generated contrastive patterns are unlabeled. Furthermore, the local patterns in both normal and generated data could often be long-tailed. Considering the previous two points, we introduce a self-supervised reweighting mechanism to mitigate the negative impacts of fake anomalous patches (patches without effective anomaly patterns) and imbalanced distribution. The mechanism utilizes density information of the features extracted by the detector during the training phase to assign different weights to the contrastive patches. It filters the fake anomalous patches and rebalances the distribution of the contrastive patches. Finally, to obtain high-throughput anomaly detection models better applied in practical industrial scenarios, we design a lightweight Fully Convolutional Network (FCN)-based patch-level detector with a pure encoder architecture. It consists of only 8 convolutional layers but performs on par with larger models in industrial anomaly detection. Furthermore, to deal with tasks that involve mixed-level anomalies, we can also integrate multiple detectors with different receptive fields. We empirically find that a single-level detector is enough to achieve competitive accuracy on MVTec AD dataset, while three detectors can be integrated to handle both structural and logical anomalies in MVTec LOCO.

The main contributions of this paper can be summarized as follows:

- We propose a novel paradigm for generating anomaly patterns without scenario-specific priors. Based on this, we develop PatchDiff which can effectively expose a range of local anomaly patterns.
- We introduce a self-supervised reweighting mechanism for the generated contrastive data to rebalance them and filter out the fake anomalous patches. This mechanism enables we can efficiently use the unlabeled and long-tailed contrastive patterns for anomaly detection.
- We design a lightweight encoder-based patch-level detector trained with only the normal data and generated contrastive data, which relies on no external dataset, heavy pretrained backbone, or memory-consuming decoder architecture.

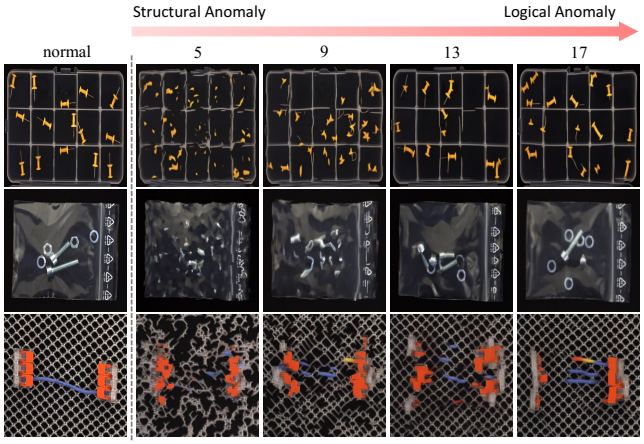

Figure 2: Anomaly contrastive images generated by our PatchDiff on MVTec LOCO. The number n above the images indicates that this column is generated based on the corresponding $n \times n$ receptive field size. We show that employing varying sizes of limited receptive fields effectively enables the PatchDiff to expose anomalies at different levels: generators with smaller sizes tend to expose structural anomalies, while generators with larger sizes tend to expose logical anomalies.

Related Works

Reconstruction-based. A well-trained autoencoder (AE) on normal data is supposed to produce lower reconstruction errors on the normal data than the anomalous data (Baur et al. 2018; Andrews, Morton, and Griffin 2016; An and Cho 2015). However, in practice, it may also reconstruct anomalies very well or even better (Pidhorskyi, Almohsen, and Doretto 2018). To alleviate this problem, recent works developed many advanced variants of AE by using generative priors or novel architectures (Perera et al. 2019; Gong et al. 2019; Hou et al. 2021; Zavrtanik, Kristan, and Skoaj 2021; Pirnay and Chai 2022; Ristea et al. 2022; You et al. 2022).

Pretrained feature-based. State-of-the-art methods for industrial anomaly detection tend to use features of a deep network pretrained on external datasets (*e.g.*, ImageNet). These methods (Defard et al. 2020; Rudolph, Wandt, and Rosenhahn 2021; Gudovskiy, Ishizaka, and Kozuka 2022; Roth et al. 2022; Hyun et al. 2023; Zhang et al. 2023) effectively utilize the general low-level visual features encoded by the pretrained network to do the anomaly detection and achieve appealing performance on MVTec AD (Bergmann

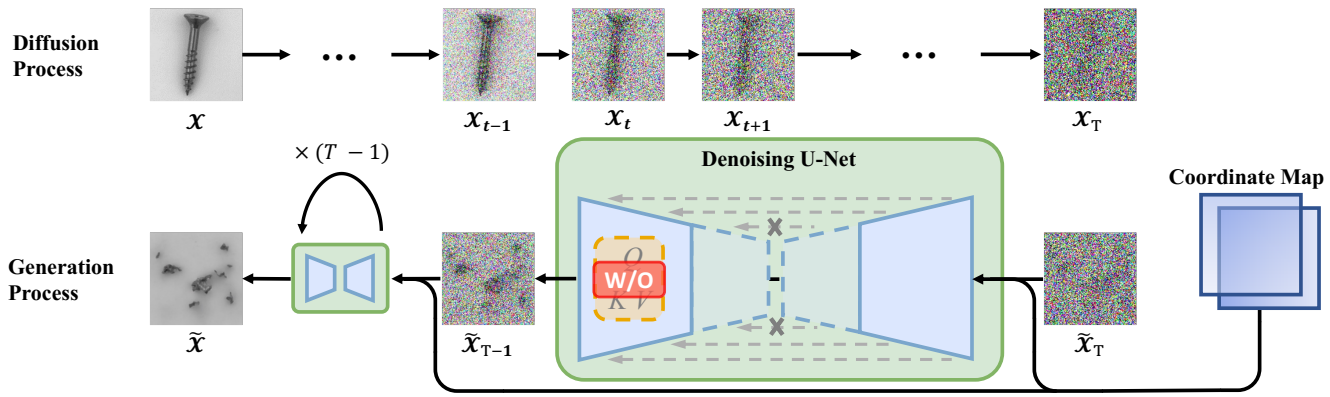

Figure 3: An illustration of the proposed anomaly generator, PatchDiff. Compared with usual Diffusion models, PatchDiff limits the receptive field of the U-Net used to denoise, which preserves only the local structures rather than the global structures. PatchDiff can effectively produce higher-level novel visual structures coming from the recombinations of specific-level local structures. We can use PatchDiffs with various receptive filed sizes to generate multilevel dense contrastive patterns, which are useful for exposing the multilevel anomalies like the structure anomaly and the logical anomaly in MVTec LOCO.

et al. 2019). However, they are hard to directly apply in other image-like domains (e.g. the depth map) (Bergmann et al. 2021; Horwitz and Hoshen 2023) or to cover the higher-level anomaly type, logical anomalies (Bergmann et al. 2022).

Anomaly simulation-based. To overcome the limitations of pre-trained features and ensure that the model produces well-defined and expected results outside the normal distribution, several anomaly simulation methods (Liznerski et al. 2021; Li et al. 2021; Zavrtanik, Kristan, and Skočaj 2021; Yang et al. 2023) are proposed. They employ various ad-hoc strategies to simulate specific types of anomaly patterns tailored to different datasets. Most of them heavily rely on human priors and can only expose specific anomaly patterns, making them also challenging to generalize to different scenarios.

Approaches

Our method can be primarily divided into two stages: (1) Generating diverse contrastive images based on our novel proposed anomaly generation paradigm to cover the anomaly patterns at interest levels. (2) Training lightweight patch-level detectors with our proposed reweighting mechanism to fully utilize the unlabeled and long-tailed generated contrastive patterns. In the following, we will describe the key parts of GRAD in detail.

Generating Anomaly Contrastive Images

In contrast to previous ad-hoc anomaly acquisition strategies (Li et al. 2021; Zavrtanik, Kristan, and Skočaj 2021; Yang et al. 2023), we introduce a novel and prior-less anomaly generation paradigm: preserving the structure information within each small image patch while disregarding the global structure information of the entire image. To implement this, we propose a diffusion model (Ho, Jain, and Abbeel 2020) based generator called PatchDiff. As shown in Fig. 3, the diffusion and denoise process is very similar to DDPM, the differences mainly come from the U-Net architecture in the following aspects:

(1) To prevent the U-Net from utilizing long-range information for recovering global structures during denoising, we deliberately remove self-attention used in DDPM (Ho, Jain, and Abbeel 2020). Self-attention is a powerful tool for capturing long-range contextual information, but for our specific task, it is unnecessary (Dong, Cordonnier, and Loukas 2021), since local consistency is all we need.

(2) To further ensure that the U-Net focuses on recovering the local patterns within the corresponding patches during denoising, we directly reduce the depth of both the encoder and decoder of the U-Net. In this way, each latent neuron of the bottleneck has a limited receptive field, and thus it denoises using only the local content and retaining only local structures.

(3) To enable the U-Net to effectively model position-dependent cues, we incorporate a 2-channel coordinate map as additional information alongside the input. This coordinate map is a tensor with dimensions matching that of the input image, where each element represents the coordinate of the corresponding pixel. Noteworthy, the output of our U-Net is still a 3-channel image as same as the original U-Net in DDPM.

Then we modify the training loss of original Diffusion models by introducing a global tiny noise $\boldsymbol{\epsilon}_g$ during the noise injection process. It is motivated by the observation that there is a tendency for overall color deviation in the generated results. Consequently, to avoid the color deviation, the training loss of PatchDiff at each denoising step t becomes

$$\mathbb{E}_{\boldsymbol{\epsilon}_1,\boldsymbol{\epsilon}_g}\left\|\boldsymbol{\epsilon}_1-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_1+\boldsymbol{\epsilon}_g,t\right)\right\|^2,$$

where $\epsilon_1 \sim \mathcal{N}(\mathbf{0},\mathbf{I})$, ϵ_θ and $\bar{\alpha}_t$ are the same as in DDPM. As depicted in Fig. 2, the images generated by PatchDiff effectively avoid the presence of low-level anomalous cues that often occur in simulation strategies, easily noticeable edges when tailoring two images together. Instead, PatchDiff focuses more on the slightly higher-level anomaly patterns. By setting multiple receptive filed sizes to the U-Net, PatchDiff can efficiently expose both structural and logical anomalies

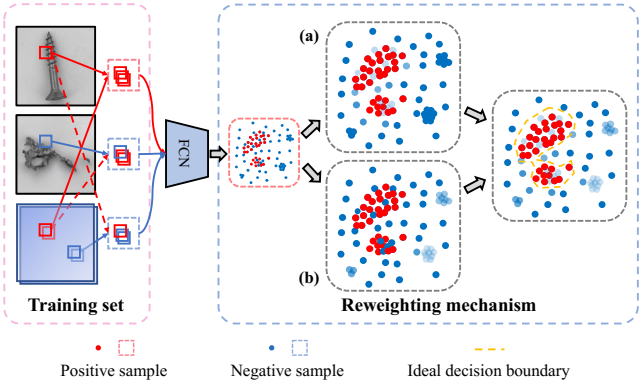

Figure 4: Schematic overview of two components during training patch-level detectors. The left portion is the training set which consists of one type of positive patch and two types of negative patches. The right portion is the reweighting mechanism which comprises mechanism (a) to filter the fake anomaly patterns and mechanism (b) to rebalance the long-tailed training data.

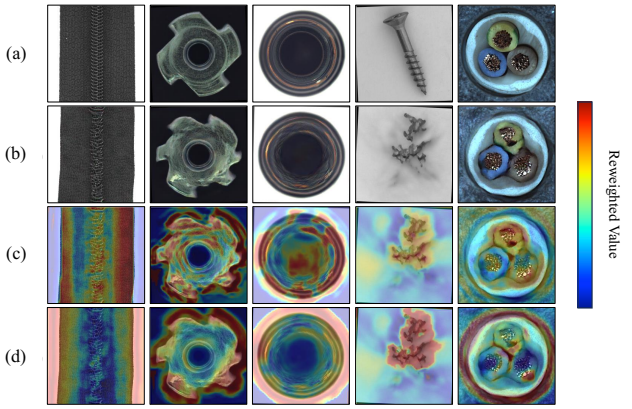

Figure 5: The reweighted map to show the effects of reweighting mechanism. (a) and (b) respectively displays the origin images and the generated contrastive images. (c) and (d) respectively depicts the effects when filtering fake anomaly patterns and rebalancing long-tailed training data. Our reweighting mechanism learns to identify patterns to be disregarded, indicated by the blue regions, and patterns to be emphasized, represented by the red regions, through a self-supervised approach

in MVTec LOCO. This enables PatchDiff to produce more comprehensive local abnormalities without using any prior knowledge of test anomalies. Additionally, the reduction in the depth of architecture and the removal of attention layers contribute to a decrease in the model's complexity and calculation cost, leading to improved training and sampling speed. Furthermore, it is worth noting that the training process of PatchDiff uses only fitting loss like DDPM(Ho, Jain, and Abbeel 2020), which is very stable and easy to implement. In summary, PatchDiff is a prior-less, easy-to-implemented, relatively-fast multilevel local anomaly pattern generation method.

Training Patch-level Detector

A naive idea to utilize the contrastive images generated by PatchDiff is directly labeling them as the anomalous class and training image-level detectors. But it does not fully exploit the dense and local anomaly patterns nor provide useful anomaly scores for localization. Instead, we opt to train patch-level anomaly detectors that detect level-specific local anomalies by patch-wisely classifying the normal images and contrastive images. Our patch-level detector is implemented with an 8-layer Fully Convolutional Network, FCN (Long, Shelhamer, and Darrell 2015), in a pure encoder way. At the training stage, the detector takes input patches of a fixed size, precisely 34×34 pixels, and produces an output anomaly score corresponding to each individual patch. To address local anomalies of multiple concerned levels (*e.g.*, both structural and logical anomalies in MVTec LOCO), we choose to maintain the detector architecture but resize the original images into lower resolutions, which indirectly achieves the adjustment of the receptive field sizes. This approach enables us to train additional detectors capable of capturing higher-level anomaly patterns without redesigning the detector's architecture and further reduces the computational cost. In the following, we will introduce how to train the patch-level detector.

Preparing the Training set Similar to the input during the generation phase, we use a 2-channel coordinate map F as an additional input. As illustrated in the left portion of Fig. 4, we prepare three types of 5-channel patches as training inputs, including one type of positive patch and two types of negative patches. Let I denote an image data, and $\mathcal{I}^+$ and $\mathcal{I}^-$ represent sets of normal samples and generated samples from PatchDiff, respectively. Subsequently, the positive patches set $\mathcal{C}^+$ and negative patches set $\mathcal{C}^-$ are defined as

$$\mathcal{C}^+ = \left\{c \mid c = \text{RandCrop}(I \oplus F), I \in \mathcal{I}^+\right\},$$
$$\mathcal{C}^- = \left\{c \mid c = \text{RandCrop}(I) \oplus \text{RandCrop}(F), I \in \mathcal{I}^+\right\}$$
$$\cup \left\{c \mid c = \text{RandCrop}(I \oplus F), I \in \mathcal{I}^-\right\},$$

where $\oplus$ denotes concatenation along the channel axis. The negative patches are constructed in two ways: (1) the patches from generated samples along with their corresponding coordinate maps, and (2) the patches from normal samples with incorrect coordinate maps. Specifically, the patches from the latter way are believed to provide examples that break the dependence between patch content and position. This explicitly enhances the detector's utilization of the auxiliary information from the coordinate maps and improves its ability to capture position-aware cues.

Reweighting the Contrastive Patches There are two potential challenges during training the patch-level detector D: (1) The images generated by PatchDiff are pixel-unlabeled, leading to the presence of fake anomaly patterns (e.g. the background region in the generated images) among the negative patches, which will mislead the detector. (2) Some important anomaly patterns may appear more rarely and lie in the

low-density regions of the data manifold, causing the detector to overlook such patterns during the training process. To mitigate these challenges, we propose a feature density-based reweighting mechanism that incorporates two reweighting strategies, as shown in the right part of Fig. 4. This mechanism relies on the feature distributions extracted from the last latent layer of our patch-level detector on both positive and negative samples. Let us denote $\mathcal{M}^+$ and $\mathcal{M}^-$ as the feature sets of positive and negative samples, respectively. Then the two reweighting strategies can be performed as follows:

(1) Filtering the fake anomaly patterns. As depicted in Fig. 4(a), we introduce a reweighting factor $w_i^{\text{noisy-}}$ for each given negative patch $\boldsymbol{c}_i^-$, to assign smaller weights to the patches whose features are too close to or even within normal features set $\mathcal{M}^+$. The reweighting factor can be formulated as

$$w_i^{\text{noisy-}} = \frac{1}{\sum_{\boldsymbol{z}\in\mathcal{M}^+} \exp(\beta_{\text{density}}\text{sim}(\boldsymbol{z}, \boldsymbol{z}_i^-))}, \tag{1}$$

where $\boldsymbol{z}_i^-$ is the feature vector of the negative patch $\boldsymbol{c}_i^-$, $\text{sim}(\boldsymbol{z}, \boldsymbol{z}') := \boldsymbol{z}\cdot\boldsymbol{z}'/\|\boldsymbol{z}\|\|\boldsymbol{z}'\|$ is the density kernel based on the cosine similarity and $\beta_{\text{density}} > 0$ is a hyper-parameter for controlling kernel bandwidth.

(2) Rebalancing the long-tailed training patches. As depicted in Fig. 4(b), we introduce a reweighting factor $w_i^{\text{tail-}}$ for each given negative patch $\boldsymbol{c}_i^-$ to downweight the patches whose features are in the high-density regions. Empirically, we find introducing a reweighting factor $w_j^{\text{tail+}}$ for each positive patch $\boldsymbol{c}_j^+$ is also helpful. Therefore we have the following two additional reweighting factors for the training patches

$$\begin{aligned} w_i^{\text{tail-}} &= \frac{1}{\sum_{\boldsymbol{z}\in\mathcal{M}^-} \exp(\beta_{\text{density}}\text{sim}(\boldsymbol{z}, \boldsymbol{z}_i^-))}, \\ w_j^{\text{tail+}} &= \frac{1}{\sum_{\boldsymbol{z}\in\mathcal{M}^+} \exp(\beta_{\text{density}}\text{sim}(\boldsymbol{z}, \boldsymbol{z}_j^+))}. \end{aligned} \tag{2}$$

The effects of our reweighting mechanism are shown in Fig 5. By incorporating these two kinds of reweighting factors, our reweighted binary classification loss $\mathcal{L}_{\text{RBCE}}$ can be formulated as

$$\begin{aligned} \mathcal{L}_{\text{RBCE}} = &-\frac{1}{\lambda^+}\sum_{j=1}^{|\mathcal{C}^+|} w_j^{\text{tail+}} \log(1 - f(\boldsymbol{c}_j^+)) \\ &-\frac{1}{\lambda^-}\sum_{i=1}^{|\mathcal{C}^-|} w_i^{\text{tail-}} w_i^{\text{noisy-}} \log(f(\boldsymbol{c}_i^-)), \end{aligned} \tag{3}$$

where λ^+ and λ^- are the normalization constants to keep the total weights of each class equal to 1:

$$\lambda^+ = \sum_{j=1}^{|\mathcal{C}^+|} w_j^{\text{tail+}}, \quad \lambda^- = \sum_{i=1}^{|\mathcal{C}^-|} w_i^{\text{tail-}} w_i^{\text{noisy-}}. \tag{4}$$

In practice, the $\mathcal{M}^+$ and $\mathcal{M}^-$ are both implemented with a memory bank that store the features of previous training steps in a queue.

Regularization on Features and Gradients We further utilize a classical unsupervised representation learning method named denoising autoencoder (Vincent et al. 2008) to regularize the learned feature by detector D. To achieve that, we introduce a simple MLP-based network R that recovers the original input patches from the feature vectors extracted from the last latent layer of D. Let f_Z denote the function extracting features from input patches, f_R denote the function recovering input patches from features, and $\mathcal{C}$ denote the collection of all training patches $\mathcal{C}^+ \cup \mathcal{C}^-$. The feature regularization loss can be formulated as

$$\mathcal{L}_{\text{feat}} = \frac{1}{|\mathcal{C}|}\sum_{\boldsymbol{c}\in\mathcal{C}} \|f_R(f_Z(\boldsymbol{c}+\boldsymbol{\epsilon_c}) + \boldsymbol{\epsilon_z}) - \boldsymbol{c}\|^2, \tag{5}$$

where $\boldsymbol{\epsilon_c}$ and $\boldsymbol{\epsilon_z}$ are respectively the noise perturbations added to the feature layer and the input layer. The auxiliary denoising task regularizes the last hidden layer of the detector to extract informative and robust representations. We highlight the auxiliary network R will be dropped in the inference stage so that will not increase the inference runtime.

Additionally, we propose a gradient regularization loss to smooth the learned decision function f, which further discourages the detector from learning imperceptible distinctions between normal patterns and fake anomaly patterns. The gradient regularization loss can be formulated as

$$\mathcal{L}_{\text{grad}} = \frac{1}{|\mathcal{C}^+|}\sum_{\boldsymbol{c}\in\mathcal{C}^+} \|\nabla_{\boldsymbol{c}} f(\boldsymbol{c})\|^2. \tag{6}$$

It penalizes the gradient norms of the decision scores with respect to the input data, which is often used to improve the Lipschitz smoothness and robustness, and thus the generalization performance of decision functions (Dai et al. 2023; Arjovsky and Bottou 2017; Ross and Doshi-Velez 2018).

The Overall Training Loss We calculate the overall training loss for the patch-level anomaly detector by aggregating the aforementioned three types of losses as

$$\mathcal{L} = \mathcal{L}_{\text{RBCE}} + \alpha_1\mathcal{L}_{\text{feat}} + \alpha_2\mathcal{L}_{\text{grad}}, \tag{7}$$

where α_1 and α_2 are hyper-parameters to adjust the impact of $\mathcal{L}_{\text{feat}}$ and $\mathcal{L}_{\text{grad}}$.

Experiments

In this section, we first briefly introduce the experimental details (See Appendix for more details). Then we report the anomaly detection accuracies and the ablation study on each component.

Dataset

To validate the effectiveness and generalizability of our approach, we perform experiments on both MVTec AD (Bergmann et al. 2019) and MVTec LOCO (Bergmann et al. 2022). There are 15 sub-datasets in MVTec AD and 5 sub-datasets in MVTec LOCO and each sub-dataset presents a diverse set of anomalies. Particularly, the training sets among them contain only normal images, while the test sets contain both normal and various types of industrial defects. Pixel-level annotations are provided in the test set.

Category	SPADE (2020)	PaDiM (2020)	S-T (2020)	PatchCore (2022)	GCAD (2022)	DADF (2023)	SINBAD (2023)	GRAD (Ours)
breakfast box	78.2	65.7	68.6	81.3	83.9	75.3	**92.0**	81.2
juice bottle	88.3	88.9	91.0	95.6	**99.4**	98.6	94.9	97.6
pushpins	59.3	61.2	74.9	72.3	86.2	81.0	78.8	**99.7**
screw bag	53.2	60.9	71.2	64.9	63.2	77.3	**85.4**	76.6
splicing connectors	65.4	67.8	81.1	82.4	83.9	86.4	**92.0**	85.4
average	68.8	68.9	77.3	79.3	83.3	83.7	86.8	**87.5**

Table 1: Image-level AU-ROC performance for anomaly detection of different methods on MVTec LOCO (Bergmann et al. 2022). The best results are in bold.

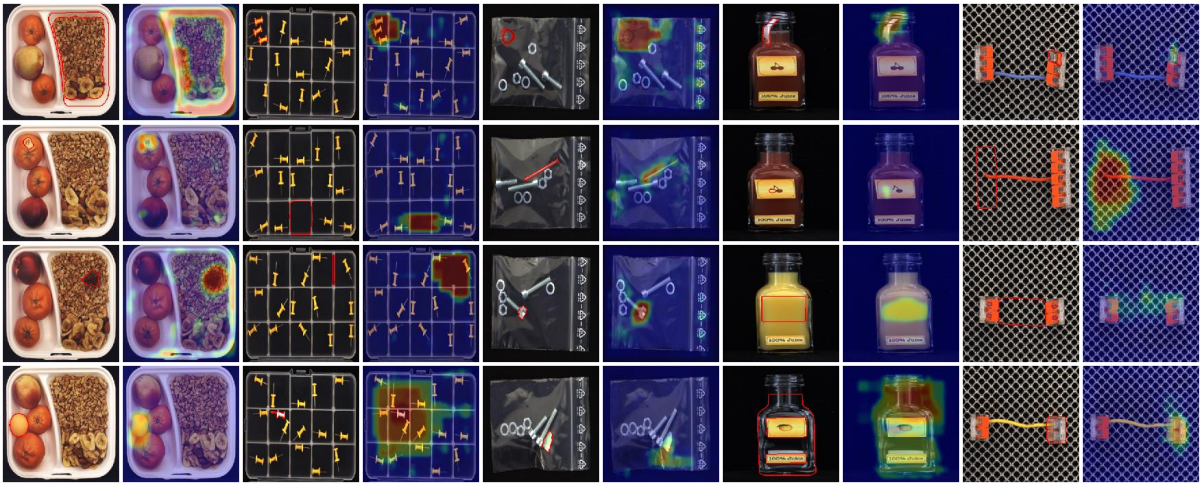

Figure 6: Defect localization results of GRAD on MVTec LOCO (Bergmann et al. 2022).

Training Settings

We simply define level-n PatchDiff as the PatchDiff with a receptive field of $n \times n$ pixels, and the images generated by it belong to level-n. Similarly, we define level-n detector as the patch-level detector with an indirect receptive field of $n \times n$ pixels.

PatchDiff. For each sub-dataset in MVTec AD, we train 3 levels of PatchDiffs (level-5, 9, 13). For each sub-dataset in MVTec LOCO, we need to train 3 different levels of detectors, and consequently, we train 4 levels of PatchDiffs (level-5, 9, 13, 17). In particular, 2 of them use level-5, 9, and 13 PatchDiffs and another one uses level-9, 13, and 17 PatchDiffs. For all PatchDiffs, we generally train them for a total of 10,000 training steps. For each sub-dataset, we sample 1,000 images for each level-n.

Patch-level Detector. Each sub-dataset in MVTec AD and MVTec LOCO contains limited training images. To train competitive detectors from scratch for each small sub-dataset, we adopt general data augmentations on both normal and generated images like previous works(Bergmann et al. 2019, 2022). For level-$34, 68$, and 136 detectors, the images are respectively resized into 256×256, 128×128, and 64×64. We train the detector on batches of size $128 \times (k + 2)$ for 2,000 epochs and report the accuracy of the final epoch. Each batch contains 128 randomly cropped positive patches from 4 normal images and $128 \times (k + 1)$ negative patches from 4 normal images and $4k$ contrastive images, where k equals the number of levels of used generated contrastive images. Specifically, we use $k = 3$ for all experiments as mentioned before.

Evaluation Settings

The image-level anomaly score directly takes the max value of a score map from the patch-based anomaly detector, and the pixel-level detection result is obtained by up-sampling the score map and then applying a Gaussian blur with a kernel size of 16. Consistent with existing methods (Bergmann et al. 2019, 2022), we use AU-ROC as the evaluation metric for the evaluation of image-level anomaly detection and pixel-level anomaly localization.

Main Results

The anomaly detection results. We compare GRAD with different methods on MVTec LOCO and MVTec AD, as shown in Table 2. For both datasets, GRAD has the best average image-level AU-ROC score, demonstrating the effectiveness of GRAD in anomaly detection. In table 1, it is important to note that the fairness of the comparison might be compromised to some extent, as all the compared methods utilize ImageNet pretrained feature extractors. However, GRAD still achieves superior performance by 0.7% even without such advantages, which shows that ImageNet pre-

Method	Pixel-level AU-ROC	Image-level AU-ROC
IGD (2022)	93.1	93.4
PSVDD (2020)	92.5	93.2
FCDD (2021)	92.1	95.7
CutPaste (2021)	95.2	96.0
NSA (2022)	96.3	97.2
DRAEM (2021)	**97.3**	98.0
DSR (2022)	-	98.2
GRAD (Ours)	96.8	**98.7**

Table 2: Anomaly detection performance on MVTec AD dataset (Bergmann et al. 2019). The best results are in bold.

Method	latency (ms↓)	FPS↑
S-T (2020)	82.2	12.2
FastFlow (2021)	26.1	38.3
DSR (2022)	24.8	40.3
GCAD (2022)	12.9	77.5
PatchCore (2022)	47.1	21.2
GRAD (Ours)	**0.799**	**1251.6**

Table 3: Inference speed on NVIDIA Tesla V100. The data of our method is obtained on MVTec LOCO dataset with three patch-level detectors (patch size: 34, input size: 256, 128, and 64).

trained features inadequately address the intricacies of logical anomaly detection within MVTec LOCO, and further demonstrates that our contrastive images generated by PatchDiff do expose both structural and logical anomalies effectively. In particular, GRAD achieves excellent results (+13.5%) on the sub-dataset of pushpins, which exactly fits our observation that the generated images for pushpins perfectly expose several abnormal logical situations in the testing set, *e.g.*, the additional pushpin in the top left compartment and no pushpins in the top right compartment, as shown in level-17 generated pushpin image of Fig. 2. In addition, we show the defect localization results in Fig. 6. In table 2, all the methods we compared do not rely on pretrained features and external data. Although GRAD does not achieve the best result for anomaly localization (pixel-level AUROC), it is still competitive among them.

Inference runtimes. We compare with different methods and report the inference latency and FPS in Table 3. Obviously, GRAD achieves a remarkable throughput performance due to its extremely lightweight architecture, and thereby, GRAD's inference speed is more than 16 times faster than GCAD's.

Ablation Study

We first perform an extensive ablation study to validate the effectiveness of two reweighting factors and the regularization technique on MVTec LOCO. The results are shown in Table 4. More details and comprehensive ablation results can be found in Appendix. We utilize the baseline as the beginning and then add regularization, noisy reweighting and long-tail reweighting one by one.

Effects of regularization techniques. One of the novel contributions presented in this paper is the regularization on features and gradients, which helps our encoder-based detector extract an informative and robust representation and build a smooth decision boundary for the data manifold. As demonstrated in Table 4, the integration of these techniques translates into improvements of +3.4/+3.1/+0.9 on the MVTec LOCO dataset.

	AUROC Level-34	Level-68	Level-136
baseline†	78.2	77.8	64.3
+ Regularization	81.6	80.9	65.2
+ Noisy Reweighting	82.5	82.5	72.1
+ Long-tail Reweighting	85.2	85.4	75.1

Table 4: Ablation study on components. Detection AUROC results on MVTec LOCO dataset of three patch-level detectors are presented. †The baseline setting uses no regularization techniques and reweighting strategies.

The effect of reweighting mechanism. Our reweighting mechanism comprises two essential components: (1) noisy reweighting, which aims to filter fake anomaly patches, and (2) long-tail reweighting, designed to rectify the imbalanced distribution of input data. When integrating the noisy reweighting, our detectors display enhancements of +0.9/+1.6/+6.9 on the MVTec LOCO dataset, as presented in Table 4. Furthermore, with the incorporation of long-tail reweighting, our detectors achieve improvements of +2.7/+2.9/+3.0, as shown in the same table. These outcomes underscore the disruptive influence of fake anomaly patches and the presence of long-tail distributions on detector performance. It is evident that our reweighting mechanism adeptly mitigates these challenges from both fronts, offering substantial advantages to our detectors.

Particularly, in Table 4, Level-136 detectors exhibit relatively poorer performance in anomaly detection. This result can be attributed to their input size, which is merely 64×64, resulting in insufficient resolution to offer informative structural anomaly details. However, this is in line with our intentions, as the purpose of Level-136 detectors is not to emphasize minute details, but rather to capture the logical relationships among components within the receptive fields of size 136×136.

Conclusion

In this paper, we propose a novel unsupervised anomaly detection framework, GRAD, by generating and reweighting dense contrastive patterns. The proposed generation method PatchDiff is able to generate multilevel contrastive patterns which exposes a range of local anomaly patterns. The proposed reweighting strategies fully utilize the unlabeled and long-tailed contrastive patterns and help the patch-level anomaly detector better learn the exposed local anomaly patterns. GRAD requires no scenario-specific prior, external datasets, or heavy pretrained feature extractor. It achieves competitive anomaly detection and localization accuracy with a superior inference speed.

Acknowledgements

This work is supported in part by Shanghai science and technology committee under grant No. 21511100600. We appreciate the High Performance Computing Center of Shanghai University, and Shanghai Engineering Research Center of Intelligent Computing System for providing the computing resources and technical support.

References

Akcay, S.; Atapour-Abarghouei, A.; and Breckon, T. P. 2018. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *ACCV*.

An, J.; and Cho, S. 2015. Variational Autoencoder based Anomaly Detection using Reconstruction Probability. In *SNU Data Mining Center, Tech. Rep.*

Andrews, J. T.; Morton, E. J.; and Griffin, L. D. 2016. Detecting anomalous data using auto-encoders. *International Journal of Machine Learning and Computing*, 6(1): 21.

Arjovsky, M.; and Bottou, L. 2017. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations (ICLR)*.

Baur, C.; Wiestler, B.; Albarqouni, S.; and Navab, N. 2018. Deep Autoencoding Models for Unsupervised Anomaly Segmentation in Brain MR Images. In *BrainLes@MICCAI*.

Bergmann, P.; Batzner, K.; Fauser, M.; Sattlegger, D.; and Steger, C. 2022. Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. *International Journal of Computer Vision*, 130(4): 947–969.

Bergmann, P.; Fauser, M.; Sattlegger, D.; and Steger, C. 2019. MVTec AD–A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *CVPR*.

Bergmann, P.; Fauser, M.; Sattlegger, D.; and Steger, C. 2020. Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings. In *CVPR*.

Bergmann, P.; Jin, X.; Sattlegger, D.; and Steger, C. 2021. The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization. *arXiv preprint arXiv:2112.09045*.

Chen, Y.; Tian, Y.; Pang, G.; and Carneiro, G. 2022. Deep One-Class Classification via Interpolated Gaussian Descriptor. In *AAAI*.

Cohen, N.; and Hoshen, Y. 2020. Sub-Image Anomaly Detection with Deep Pyramid Correspondences. *ArXiv*, abs/2005.02357.

Dai, S.; Li, X.; Zhou, Y.; Ye, X.; and Liu, T. 2023. GradPU: positive-unlabeled learning via gradient penalty and positive upweighting. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Defard, T.; Setkov, A.; Loesch, A.; and Audigier, R. 2020. PaDiM: a Patch Distribution Modeling Framework for Anomaly Detection and Localization. In *ICPR Workshops*.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*.

Dong, Y.; Cordonnier, J.-B.; and Loukas, A. 2021. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*, 2793–2803. PMLR.

Gong, D.; Liu, L.; Le, V.; Saha, B.; Mansour, M. R.; Venkatesh, S.; and van den Hengel, A. 2019. Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 1705–1714.

Gudovskiy, D.; Ishizaka, S.; and Kozuka, K. 2022. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 98–107.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Horwitz, E.; and Hoshen, Y. 2023. Back to the feature: classical 3d features are (almost) all you need for 3d anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2967–2976.

Hou, J.; Zhang, Y.; Zhong, Q.; Xie, D.; Pu, S.; and Zhou, H. 2021. Divide-and-Assemble: Learning Block-wise Memory for Unsupervised Anomaly Detection. In *ICCV*.

Hyun, J.; Kim, S.; Jeon, G.; Kim, S. H.; Bae, K.; and Kang, B. J. 2023. ReConPatch: Contrastive Patch Representation Learning for Industrial Anomaly Detection. *arXiv preprint arXiv:2305.16713*.

Kiran, B.; Thomas, D.; and Parakkal, R. 2018. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2): 36.

Li, C.-L.; Sohn, K.; Yoon, J.; and Pfister, T. 2021. CutPaste: Self-Supervised Learning for Anomaly Detection and Localization. In *CVPR*.

Liznerski, P.; Ruff, L.; Vandermeulen, R. A.; Franks, B. J.; Kloft, M.; and Muller, K.-R. 2021. Explainable Deep One-Class Classification. In *ICLR*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*, 3431–3440.

Perera, P.; Nallapati, R.; Xiang, B.; and NONE. 2019. OCGAN: One-class Novelty Detection Using GANs with Constrained Latent Representations. In *CVPR*.

Pidhorskyi, S.; Almohsen, R.; and Doretto, G. 2018. Generative Probabilistic Novelty Detection with Adversarial Autoencoders. In *NeurIPS*.

Pirnay, J.; and Chai, K. Y. 2022. Inpainting Transformer for Anomaly Detection. In *ICIAP*.

Ristea, N.-C.; Madan, N.; Ionescu, R. T.; Nasrollahi, K.; Khan, F. S.; Moeslund, T. B.; and Shah, M. 2022. Self-Supervised Predictive Convolutional Attentive Block for Anomaly Detection. In *CVPR*.

Ross, A.; and Doshi-Velez, F. 2018. Improving the adversarial robustness and interpretability of deep neural networks

by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Roth, K.; Pemula, L.; Zepeda, J.; Schölkopf, B.; Brox, T.; and Gehler, P. 2022. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14318–14328.

Rudolph, M.; Wandt, B.; and Rosenhahn, B. 2021. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 1907–1916.

Sato, D.; Hanaoka, S.; Nomura, Y.; Takenaga, T.; Miki, S.; Yoshikawa, T.; Hayashi, N.; and Abe, O. 2018. A primitive study on unsupervised anomaly detection with an autoencoder in emergency head CT volumes. In *Medical Imaging 2018: Computer-Aided Diagnosis*.

Schlüter, H. M.; Tan, J.; Hou, B.; and Kainz, B. 2022. Natural Synthetic Anomalies for Self-supervised Anomaly Detection and Localization. In *ECCV*.

Tzachor, N. C.; and Hoshen, Y. 2023. Set features for fine-grained anomaly detection. *arXiv preprint arXiv:2302.12245*.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, 1096–1103.

Yang, M.; Liu, J.; Yang, Z.; and Wu, Z. 2023. SLSG: Industrial Image Anomaly Detection by Learning Better Feature Embeddings and One-Class Classification. *arXiv preprint arXiv:2305.00398*.

Yao, H.; Luo, W.; and Yu, W. 2023. Visual Anomaly Detection via Dual-Attention Transformer and Discriminative Flow. *arXiv preprint arXiv:2303.17882*.

Yi, J.; and Yoon, S. 2020. Patch SVDD: Patch-level SVDD for Anomaly Detection and Segmentation. In *ACCV*.

You, Z.; Cui, L.; Shen, Y.; Yang, K.; Lu, X.; Zheng, Y.; and Le, X. 2022. A Unified Model for Multi-class Anomaly Detection. In *NeurIPS*.

Yu, J.; Zheng, Y.; Wang, X.; Li, W.; Wu, Y.; Zhao, R.; and Wu, L. 2021. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv:2111.07677*.

Zavrtanik, V.; Kristan, M.; and Skoaj, D. 2021. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognit.*, 112: 107706.

Zavrtanik, V.; Kristan, M.; and Skočaj, D. 2022. Dsr–a dual subspace re-projection network for surface anomaly detection. In *European conference on computer vision*, 539–554. Springer.

Zavrtanik, V.; Kristan, M.; and Skočaj, D. 2021. DRAEM - A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 8330–8339.

Zhang, H.; Wu, Z.; Wang, Z.; Chen, Z.; and Jiang, Y.-G. 2023. Prototypical residual networks for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16281–16291.

Zheng, P.; Yuan, S.; Wu, X.; Li, J. Y.; and Lu, A. 2018. One-Class Adversarial Nets for Fraud Detection. In *AAAI*.