Low-Shot Unsupervised Visual Anomaly Detection via Sparse Feature Representation

Fanghui Zhang, Haiyue Zhu, Yigang Cen, Shichao Kan, Linna Zhang,
Prahlad Vadakkepat, and Tong Heng Lee

Abstract*—Visual anomaly detection is an essential component in modern industrial manufacturing. Existing studies using notions of pairwise similarity distance between a test feature and nominal features have achieved great breakthroughs. However, the absolute similarity distance lacks certain generalizations, making it challenging to extend the comparison beyond the available samples. This limitation could potentially hamper anomaly detection performance in scenarios with limited samples. This paper presents a novel Sparse Feature Representation Anomaly Detection (SFRAD) framework which formulates the anomaly detection as a sparse feature representation problem; and notably proposes an Anomaly Score by Orthogonal Matching Pursuit (ASOMP) as a novel detection metric. Specifically, SFRAD calculates the Gaussian kernel distance between the test feature and its sparse representation in the nominal feature space for anomaly detection. Here, the Orthogonal Matching Pursuit (OMP) algorithm is adopted to achieve the sparse feature representation. Moreover, to construct a low-redundancy memory bank storing the basis features for sparse representation, a novel Basis Feature Sampling (BFS) algorithm is proposed by considering both the maximum coverage and the optimum feature representation simultaneously. As a result, SFRAD incorporates both the advantages of absolute similarity and linear representation; and this enhances the generalization in low-shot scenarios. Extensive experiments on the MVTec AD, KolektorSDD, KolektorSDD2, MVTec LOCO AD, VISA, MNIST, and CIFAR-10 datasets demonstrate that our proposed SFRAD outperforms the previous methods and achieves state-of-the-art unsupervised anomaly detection performance. Notably, significantly improved outcomes and results have also been achieved on low-shot anomaly detection. Code is available at https://github.com/fanghuisky/SFRAD.

***Index Terms*—Anomaly Detection, Unsupervised Learning, Low-Shot Learning, Feature Representation, Orthogonal Matching Pursuit**

I. INTRODUCTION

VISUAL anomaly detection and localization refer to judging whether a given image conforms to the prescribed pattern or distribution of normal samples as well as locating abnormal areas if exist, which have wide areas of applications in industrial manufacturing [1]–[5], medical diagnosis [6]–[8], video surveillance [9]–[12], etc. In industrial manufacturing systems, abnormal products resulting from raw materials or manufacturing technologies often occur on various production and assembly lines. Therefore, the visual anomaly detection system is critically demanded to ensure intelligent production and quality inspection, which can significantly save the workforce, improve efficiency, and save cost. In recent years, some deep learning technologies, such as convolutional neural networks [13], transformers [14], and deep metric learning [15], etc., have provided new solutions for visual anomaly detection and achieved extraordinary performance. However, visual anomaly detection is still considered a very challenging problem due to the difficulty of obtaining a significant number of annotated samples and the inherent uncertainty surrounding anomaly types. As a result, the traditional supervised approaches based deep learning technologies in general are difficult to apply for visual anomaly detection.

Consequently, unsupervised approaches have become the mainstream technology for identifying abnormal samples in visual anomaly detection [16], which are mainly categorized into background reconstruction based methods and embedding feature based methods. In literature, background reconstruction based methods have shown great promise in anomaly detection, which mainly employ autoencoders [13], [17]–[20], GANs [7], [21]–[24], or self-supervised solutions [25]–[27] to locate anomalies in images. Autoencoder-based solutions use only defect-free images for reconstruction network training and thus are supposed to fail to reconstruct abnormal areas correctly during inference. However, due to the strong generalization ability of autoencoders, there is a high chance that abnormal areas will be reconstructed, resulting in a failure in anomaly detection. To alleviate such "over-generalization" issue of the autoencoder, researchers also explore limiting the expression ability in autoencoder latent space, such as using memory storage [28], [29], clustering features [30], etc. However, under-expression leads to the loss of high-frequency detailed information in the reconstructed image [31]. Moreover, some research works integrate GAN [6], [7], [32] into

---

Fanghui Zhang is with the School of Artificial Intelligence, Henan University, Zhengzhou, Henan 475004, China, with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China, with the Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China, and also with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: fhzhang@henu.edu.cn; 18112013@bjtu.edu.cn).

Haiyue Zhu is with the Adaptive Robotics and Mechatronics Group, Singapore Institute of Manufacturing Technology (SIMTech), Agency for Science, Technology and Research (A*STAR), Singapore 138634 (e-mail: zhu_haiyue@simtech.a-star.edu.sg).

Yigang Cen is with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China, and also with Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China (e-mail: ygcen@bjtu.edu.cn).

Shichao Kan is with the School of Computer Science and Engineering, Central South University, Hunan, China (e-mail: kanshichao@csu.edu.cn).

Linna Zhang is with the School of Mechanical Engineering, Guizhou University, 550025, Guiyang, China (e-mail: zln770808@163.com).

Prahlad Vadakkepat and Tong Heng Lee are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: prahlad@nus.edu.sg; eleleeth@nus.edu.sg).

Corresponding authors: Haiyue Zhu, Yigang Cen

Manuscript received April 19, 2021; revised August 16, 2021.

the framework to increase the fidelity of image reconstruction. Self-supervised solutions [25]–[27] aim to synthesize anomaly samples or generate random masking samples and use them as anomalous samples during training. However, due to the strong generalization ability of convolutional neural networks, some abnormal areas also have a high chance of being reconstructed, which fails in anomaly detection.

Recently, embedding feature based methods [33]–[37] have significantly improved the unsupervised anomaly detection performance. The key principle of these works is to match multiscale features between test samples and nominal samples extracted by the deep models. Typical embedding feature methods include memory banks [35], [36], student-teacher networks [33], [34], [38], normalizing flows [39]–[42], etc. Memory bank based methods [35], [36] obtain sufficient features extracted from a pretrained convolutional neural network (CNN) and store them in a memory bank. Subsequently, the similarity distances between the test features and memory bank features are calculated to classify normal or abnormal. For student-teacher networks [33], [34], [38], the knowledge from a powerful teacher network is distilled into student networks during the training of the normal samples, and the anomalies can be detected by capturing the feature discrepancy between the teacher and student networks when inference with abnormal samples. Normalizing flows [39]–[42] embed normal image features into standard normal distribution and use the probability to detect anomalies.

In literature, visual anomaly detection is well developed as surveyed above. However, most existing anomaly detection algorithms only focus on data-sufficient scenarios where plenty number of normal samples are available for training to densely describe the normal space in a comprehensive manner, and thus can reject the anomaly effectively. Nevertheless, for many real-world anomaly detection applications, meeting such data-sufficient conditions is difficult or costly due to the data-scarcity nature of the industrial processes, and it cannot wait until sufficient samples have been collected to develop the autonomous inspection. As a result, anomaly detection from low-shot samples is practically desired to improve the flexibility and adaptiveness of Artificial Intelligence (AI), so that can make AI anomaly detection more affordable for real-world applications. However, compared to the data-sufficient anomaly detection, low-shot anomaly detection [39], [43]–[48] specially faces enormous challenges due to the sample sparsity. On the one hand, the sparse samples available for training demand the algorithm to have additional generalization capability to generalize and reconstruct the entire normal space. On the other hand, as mentioned earlier, the “strong-generalization” capability is a well-known evil for anomaly detection that potentially misrepresents the anomaly into normal space, thus degrades the discrimination effectiveness. As a result, successful low-shot anomaly detection requires addressing such contradictions properly within one framework.

In this work, to address the above issues, we propose a novel Sparse Feature Representation Anomaly Detection (SFRAD) approach for the low-shot unsupervised visual anomaly detection, which treats the anomaly detection as a sparse feature representation problem using the nominal feature bases extracted from the normal sample patches. Moreover, a novel anomaly metric, Anomaly Score by Orthogonal Matching Pursuit (ASOMP), is introduced to effectively reflect the anomaly level, which calculates the Gaussian kernel distance between the test feature and its sparse representation in the nominal feature space. Compared with previous methods that directly compute pairwise absolute feature similarity distances, ASOMP expands the scope of anomaly comparison beyond finitely available features to encompass the nominal feature space. This significantly enhances the generalization ability in low-shot anomaly detection scenarios. In addition, a low-redundancy Basis Feature Sampling (BFS) algorithm is proposed in SFRAD to select the nominal basis features, where its memory bank is constructed by considering both the maximum coverage criterion and the optimum feature representation criterion simultaneously. Together with the sparse feature representation, SFRAD enhances the generalization in low-shot scenarios by incorporating the advantages of both absolute similarity and linear representation.

Overall, the main contributions of this paper are summarized as follows:

1) A sparse feature representation based low-shot unsupervised visual anomaly detection, SFRAD, is proposed. This is the first attempt to combine deep features and sparse feature representations for anomaly detection.
2) A novel OMP-based anomaly score calculation, ASOMP, is introduced to effectively reflect the anomaly using Gaussian kernel distance for feature representation anomaly detection.
3) A low-redundancy BFS algorithm is proposed for memory bank construction by considering both the maximum coverage and optimum feature representation simultaneously.

The rest of this paper is organized as follows. The related work about anomaly detection is reviewed in Section II, and Section III presents the proposed SFRAD. Extensive experiments are performed and analyzed in Section IV, and finally, Section V concludes the paper.

II. RELATED WORK

A. Anomaly Detection

Unsupervised visual anomaly detection and segmentation can be mainly classified into two categories [26], [49], [50], i.e., background reconstruction based methods and embedding feature based methods.

1) Background Reconstruction Based Methods: Typical background reconstruction based methods include autoencoders [13], [17]–[19], GANs [7], [21]–[23], and self-supervised techniques [25]–[27]. Their main idea is to adapt the distribution of normal images by building a reconstruction network based on all nominal images. These methods assume that abnormal images cannot be reconstructed by the trained reconstruction network. When inference, the difference between the test image and the reconstruction image is calculated for anomaly detection. Anomaly detection methods based on background reconstruction are intuitive, easily understandable, and effective for regular texture classes. However, these methods not only require tedious training from scratch but also generalize poorly across the complex texture and object categories [51], [52].

Earlier work by Paul et al. [13] proposed to combine an autoencoder and a structural-similarity-based perceptual loss function for visual anomaly detection. However, the autoencoder can sometimes generalize well and is likely to reconstruct the anomalies well. To mitigate the drawback of [13], Gong et al. [29] proposed to augment the autoencoder with a memory module and introduced a new memory-augmented autoencoder, i.e., MemAE. Liu et al. [21] proposed a deep semantic segmentation network based on a generative adversarial network for fabric defect detection, which can detect different anomaly types. Shashanka et al. [53] proposed a novel convolutional adversarial variational autoencoder with guided attention, i.e., CAVGA, which can localize the anomaly with a latent variable to preserve the spatial information. Vitjan et al. [25] formulated anomaly detection as a reconstruction-by-inpainting problem (RIAD). Partial image regions are randomly removed. Then RIAD reconstructs the image from partial inpaintings to address the drawbacks of autoencoder methods. Jonathan et al. [26] posed anomaly detection as a patch-inpainting problem and proposed the inpainting transformer (InTra) to inpaint covered patches in a large sequence of image patches. Huang et al. [27] proposed a self-supervised masking approach through random masking and then restoring to detect anomalies. Song et al. [19] presented an anomaly composition and decomposition network to inspect texture defects.

2) Embedding Feature Based Methods: Embedding feature-based methods benefit from the well-discriminative feature space, which mainly include memory bank [35], [36] and knowledge distillation [33], [34], [38]. The main idea of these methods is to use features extracted from the neural network for feature matching between test features and nominal features. Niv et al. [35] proposed the semantic pyramid anomaly detection method (SPADE), which used correspondences based on a multiscale feature pyramid and achieved good performance on unsupervised anomaly detection. Thomas et al. [54] proposed an approach for patch distribution modeling, i.e., PaDiM, which uses a pretrained CNN to extract features for patch embedding and multivariate Gaussian distributions. Then, a probabilistic representation of the normal class can be obtained. Correlations between the different semantic levels of CNN are exploited to detect anomalies. Karsten et al. [36] proposed PatchCore, which used a maximally representative memory bank of normal features. Paul et al. [34] proposed a robust student-teacher framework. Anomalies are detected when there are differences between the outputs of student networks and those of the teacher network. In [33], the student-teacher framework is extended in terms of accuracy and efficiency. When given a teacher network, the knowledge is distilled into a single student network with identical architecture to learn the distribution of normal images. Zhou et al. [55] proposed a teacher-student model, which aims to push representations on anomaly areas of the teacher and student network as far as possible and pull representations on normal areas as close as possible. The proposed SFRAD in this work is also classified under this category. Differently, SFRAD also focuses on low-shot anomaly detection, i.e., only a few normal images are available for training.

B. Low Shot Anomaly Detection

Low-shot anomaly detection [39], [43]–[47] is to identify anomalies or defects in the test image when the amount of normal images available is limited. Marco et al. [39] employed a multi-scale feature extractor that enabled the normalizing flow to assign likelihoods to the images. Besides, they designed a scoring function based on these likelihoods to indicate anomalies. Shelly et al. [46] proposed a hierarchical generative model to capture the multi-scale patch distribution of each training image. Further, they used image transformations and optimized scale-specific patch-discriminators to distinguish between real and fake patches of the image. Huang et al. [47] leveraged registration, an image alignment task as the proxy task, to train a category-agnostic anomaly detection model. During testing, they compared the registered features of the test image and its corresponding support (normal) images to identify the anomalies. Different from the existing works that mainly rely on similarity distances for anomaly detection, we introduce a novel anomaly measurement metric via sparse feature representation, which improves the generalization ability of embedding feature based anomaly detection in the low-data regime.

III. PROPOSED METHOD

The visual anomaly detection problem is formulated as follows. Let $\mathcal{I} = \{\boldsymbol{I}_1, \boldsymbol{I}_2, \cdots, \boldsymbol{I}_M\}$ represent the normal (anomaly-free) image set with M samples available for training, where $\boldsymbol{I}_i$ denotes an anomaly-free image. Consider a test set $\mathcal{T} = \{(\boldsymbol{I}'_i, y_i)\}$ includes both normal and abnormal images, where y_i is the ground-truth label, i.e., $y_i \in \{0, 1\}$ indicates whether the image $\boldsymbol{I}'_i$ is normal (0) or abnormal (1). The objective of anomaly detection aims to learn a model $\Psi(\cdot)$ from $\mathcal{I}$ only, so that $\Psi(\boldsymbol{I}'_i) \rightarrow y_i$ distinguishes if the test image is normal or abnormal. Moreover, when $\Psi(\boldsymbol{I}') = 1$, an accompanying anomaly localization map Ω is simultaneously predicted from $\Psi(\cdot)$ as output. Motivated by practical industrial needs, the low-shot anomaly detection problem in this work aims to address the challenging situation where the image number M for $\mathcal{I}$ is a relatively small number, e.g., $M = 1, 2, 5, 10$, etc. Such extreme condition requires the learned model with strong generalization capability, which is beneficial for both data-scarce and data-redundant situations.

Previous algorithms utilize deep features extracted from a pretrained network for feature matching and achieve breakthroughs in anomaly detection. One of the most representative methods, PatchCore, extracts locally-aware features from the ImageNet pretrained network and uses core subsampling to construct a memory bank with nominal patch-level features. When inference, PatchCore uses the absolute minimum similarity distance between a test feature and all nominal features in the memory bank for anomaly detection. The success of this approach is under the assumption that the nominal features in

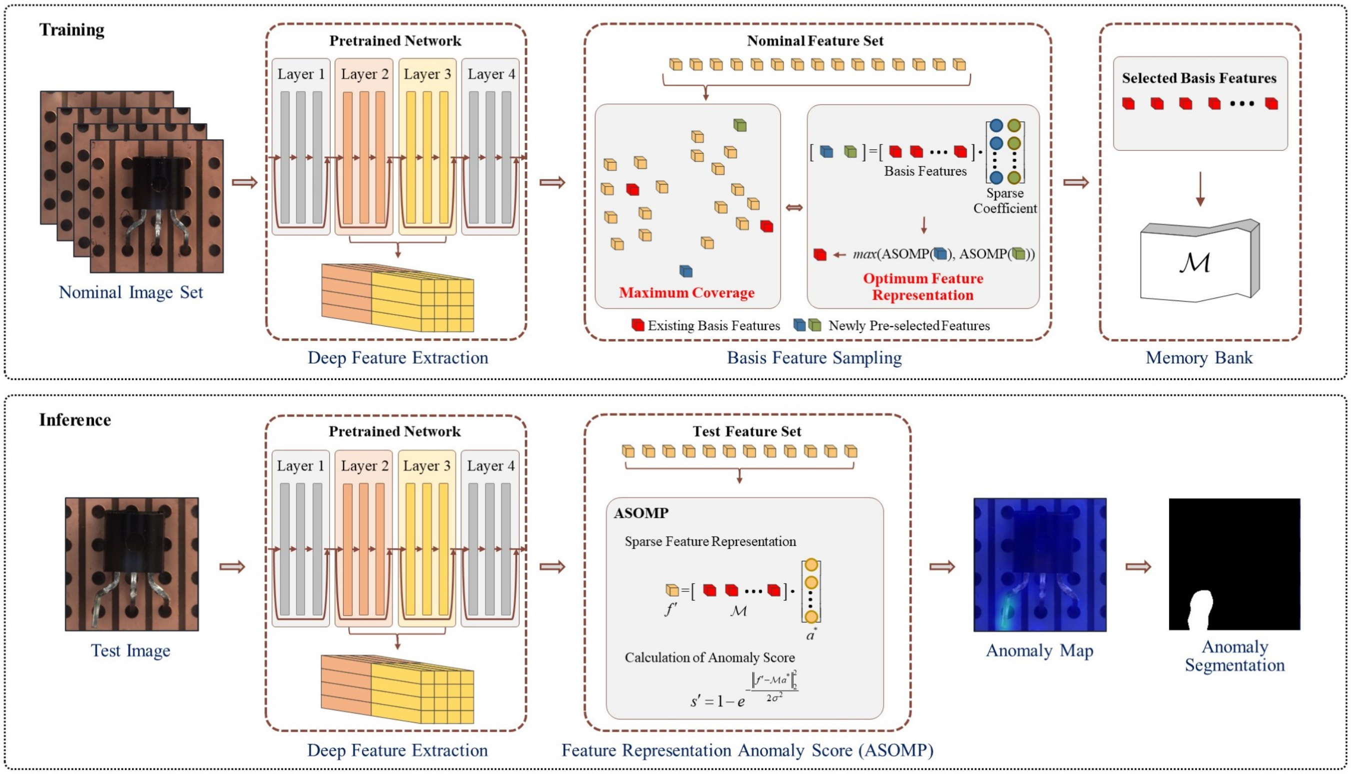

Fig. 1. The overall framework of our proposed SFRAD, which employs sparse feature representation as a novel metric for anomaly detection. Firstly, the nominal feature set is obtained by extracting deep features of normal images using a pretrained network. Next, a low-redundant memory bank $\mathcal{M}$ is constructed using the proposed Basis Feature Sampling (BFS) approach, which considers both criteria of the maximum coverage as well as the optimum feature representation. When Inference, the sparse representation is conducted for all patch features of the test samples to calculate the anomaly scores using ASOMP, so that the anomaly map and anomaly segmentation are obtained using these anomaly scores.

the memory bank are extracted from plenty of normal samples, which are able to almost fully cover the entire nominal space. In this case, the absolute similarity distance is naturally a good indicator of anomaly. However, this prerequisite is hard to meet in the low-shot scenarios, which likely leads to the situation that the nominal features in the memory bank only partially or sparsely cover the entire nominal space. In this case, the absolute similarity distance may fail to reflect the real anomaly level as it lacks the elasticity to generalize from the partial nominal space into the entire space.

To improve the generalization capability of anomaly detection in low-shot scenarios, this paper proposes to use the sparse feature representation distance as a metric to reflect the anomaly level, where the feature representation is native with the generalization capability to represent unseen features from the bases. To effectively represent the feature with the reflection of the anomaly, a memory bank $\mathcal{M}$ is constructed to store the basis features (nominal features) extracted from the normal samples, which is considered as a dictionary in Hilbert space, i.e., all normal features can be represented linearly by the non-redundant nominal features in memory bank $\mathcal{M}$. The idea to distinguish the anomaly in this approach is to use the nominal feature in $\mathcal{M}$ to linearly represent a test feature $\boldsymbol{x}'$, i.e., $\boldsymbol{x}^* = \mathcal{M}\boldsymbol{a}^*$ where $\boldsymbol{a}^* = \min_{\boldsymbol{a}} ||\boldsymbol{x}' - \mathcal{M}\boldsymbol{a}||_2$ is the sparse coefficient. The feature representation distance between the represented feature $\boldsymbol{x}^*$ and the original feature $\boldsymbol{x}'$ is used as an anomaly indicator. When $\boldsymbol{x}'$ is normal, it is supposed to be able to be linearly represented by the basis features in $\mathcal{M}$, which means the distance $||\boldsymbol{x}' - \boldsymbol{x}^*||_2$ should be small. On the contrary, if $\boldsymbol{x}'$ is abnormal, it is supposed to not be in this captured linear space. In other words, $\boldsymbol{x}'$ cannot be linearly represented by the basis features in $\mathcal{M}$ with small error; and accordingly, the distance $||\boldsymbol{x}' - \boldsymbol{x}^*||_2$ will be large to reflect the anomaly. The proposed Sparse Feature Representation based Anomaly Detection approach is denoted as SFRAD, and its framework is illustrated in Fig. 1. Compared to the absolute similarity distance between the test feature and its nearest normal feature (one to one), our method greatly expands the horizon of the distance metric across the whole nominal feature space (n to one), which can effectively promote generalization capability for the low-shot anomaly detection.

A. Feature Extraction for Nominal Set

Similar to previous research [35], [36], [54], deep features extracted by pretrained models can be employed for anomaly detection. In this work, a Wide-ResNet50×2 network pretrained on the ImageNet dataset is used in our SFRAD to extract the features at each layer, denoted as,

$$\boldsymbol{F}_{l,i} = \mathcal{F}_l(\boldsymbol{I}_i)|_{l\in\{1,2,3,4\}}, \tag{1}$$

Algorithm 1: Anomaly Score calculation by Orthogonal Matching Pursuit (ASOMP)

```
Input: Memory bank $\mathcal{M} = [\boldsymbol{f}_1^+, \cdots, \boldsymbol{f}_{N_M}^+] \in \mathbb{R}^{c \times N_M}$;
       test feature $\boldsymbol{f} \in \mathbb{R}^c$; $n_{max}$, $\epsilon$
Output: Anomaly score $s$,
        $\boldsymbol{a}^* = \arg\min_{\boldsymbol{a}:\mathrm{Supp}(\boldsymbol{a}) \subseteq T_n} \|\boldsymbol{f} - \mathcal{M}\boldsymbol{a}\|_2$
Initialize $n = 0$, residual $\mathbf{R}_0 f = \boldsymbol{f}$, support set $T_0 = \emptyset$.
while $n < n_{max}$ and $\|\mathbf{R}_n f\|_2 > \epsilon$ do
    $T_{n+1} = T_n \bigcup \{i^*\}$,
      where $i^* = \underset{i=1,\dots,N_M}{\arg\max} \left|\left\langle \mathbf{R}_n f, \boldsymbol{f}_i^+ \right\rangle\right|$.
    $\mathbf{R}_{n+1} f = (\boldsymbol{I} - P_{T_{n+1}})\boldsymbol{f}$,
      where $P_{T_{n+1}}$ is the projection onto the span of
      the vectors $\{\boldsymbol{f}_j^+, j \in T_{n+1}\}$.
    $n \leftarrow n + 1$.
end
$s = 1 - e^{-\frac{\|\mathbf{R}_n f\|_2^2}{2\sigma^2}}$
```

Algorithm 2: The framework of our proposed SFRAD.

```
Input: Nominal image set $\mathcal{I} = \{\boldsymbol{I}_1, \cdots, \boldsymbol{I}_M\}$; test
       image $\boldsymbol{I}' \in \mathbb{R}^{W \times H}$; sampling rate $\alpha$.
Output: Anomaly map $\Omega$, Image-level anomaly score
        $\Phi$
// BFS for Memory Bank
Compute feature maps $\boldsymbol{F}$ of nominal images according
  to Eq. (2).
Compute nominal feature set $\mathcal{X}$ according to Eq. (3).
$\mathcal{M} \leftarrow \{\}$
for $i \in [0, \cdots, \alpha \times len(\mathcal{X})]$ do
    $\mathcal{N} \leftarrow \underset{\boldsymbol{f}^- \in \mathcal{X}-\mathcal{M}}{\arg\max\text{-}k} \ \underset{\boldsymbol{f}^+ \in \mathcal{M}}{\min} \|\psi(\boldsymbol{f}^-) - \psi(\boldsymbol{f}^+)\|_2$,
    $\boldsymbol{f}^* = \underset{\boldsymbol{f}_j \in \mathcal{N}}{\arg\max} \ \mathrm{ASOMP}(\boldsymbol{f}_j, \mathcal{M})$
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{\boldsymbol{f}^*\}$
end
// Inference stage
Compute feature map $\boldsymbol{F}^{'} \in \mathbb{R}^{w \times h \times c}$ of test image $\boldsymbol{I}'$
  according to Eq. (2).
Compute test feature set $\mathcal{X}' = \{\boldsymbol{f}_1', \cdots, \boldsymbol{f}_{N'}'\}$
  according to Eq. (3).
Compute $s_i'$ by $\mathrm{ASOMP}(\boldsymbol{f}_i', \mathcal{M})$.
Obtain $\Omega = [s_1', \cdots, s_{N'}']$ and $\Phi = max(\Omega)$.
Reshape $\Omega$ into the shape of $(w, h)$, subsequently
  upsample $\Omega$ as $(W, H)$.
```

and $\mathcal{F}_l$ denotes the l-th layer of the feature extractor, and $\boldsymbol{F}_{l,i} \in \mathbb{R}^{w_l \times h_l \times c_l}$, where w_l, h_l, and c_l are the width, height, and channel of $\boldsymbol{F}_{l,i}$, respectively. In addition, the locally aware patch features (LAPF) [36] method is adopted to increase the receptive field size and robustness, which has two steps, i.e., (1) gathering feature vectors from the neighborhood and (2) performing adaptive average pooling.

To effectively utilize the features extracted from different layers, upsampling is used to obtain the same size by bilinear interpolation. Same to [36], we only take the two intermediate feature layers, i.e., $l = 2$ and 3, and the Adaptive Pooling (AP) is used to remove the redundant information. Finally, the features extracted from different layers are fused together as $\boldsymbol{F}_i$, denoted as

$$\boldsymbol{F}_i = \underset{l=2,\,3}{\phi_c}\left(\phi_p\left(\phi_u(\boldsymbol{F}_{l,i})\right)\right) \in \mathbb{R}^{w \times h \times c}, \tag{2}$$

where ϕ_c, ϕ_p, and ϕ_u denote the concatenation, adaptive pooling, and upsampling operations, respectively. Let $\boldsymbol{f}_i(p) \in \mathbb{R}^c$ denote the channel feature vector at position p among $w \times h$ of feature map $\boldsymbol{F}_i$, i.e., $\boldsymbol{F}_i = [\boldsymbol{f}_i(p)]_{p \in \{w \times h\}}$, where $\{w \times h\}$ stands for $\{1, 2, \cdots, w \times h\}$. The nominal feature set $\mathcal{X}$ can be formed as

$$\mathcal{X} = \bigcup_{p \in \{w \times h\},\ I_i \in \mathcal{I}} \boldsymbol{f}_i(p). \tag{3}$$

B. Basis Feature Sampling for Memory Bank

In this paper, we introduce the concept of Hilbert space for our novel basis feature sampling to construct the low-dimension memory bank $\mathcal{M}$ from the nominal feature set $\mathcal{X}$. Different from [36], in order to enhance the generalization capability, the selection criteria of memory bank $\mathcal{M}$ in this work not only considers (A) the maximum coverage of the $\mathcal{X}$ in its entirety, but also focuses on (B) the optimum non-redundant feature representation, i.e., the features in nominal feature set $\mathcal{X}$ can be represented linearly by the basis features in $\mathcal{M}$, while the basis feature in $\mathcal{M}$ cannot be represented linearly by others.

For Criterion A on the maximum coverage, we follow [36] to employ the minimax facility location coreset selection to ensure approximate coverage of the basis feature subset. Differently, in order to incorporate Criterion A and B integrately, we introduce an intermediate feature set $\mathcal{N}$ in each iteration to temporarily store the top-k optimal feature basis from Criterion A, i.e.,

$$\mathcal{N} \leftarrow \underset{\boldsymbol{f}^- \in \mathcal{X}-\mathcal{M}}{\arg\max\text{-}k} \ \underset{\boldsymbol{f}^+ \in \mathcal{M}}{\min} \|\psi(\boldsymbol{f}^-) - \psi(\boldsymbol{f}^+)\|_2, \tag{4}$$

where $\arg\max\text{-}k$ represents the sorting operation to select the top-k maximum, and $\psi(\cdot)$ denotes the random linear projections. From $\mathcal{N}$, we further apply the selection of optimum non-redundant feature representation to fulfill Criterion B.

Specifically, for every feature $\boldsymbol{f}_j$ selected in $\mathcal{N}$, an Orthogonal Matching Pursuit (OMP) algorithm is employed to linearly represent $\boldsymbol{f}_j \in \mathcal{N}$ using the current basis features in $\mathcal{M}$,

$$\boldsymbol{a}_j^* = \underset{\boldsymbol{a}_j}{\arg\min} \|\boldsymbol{f}_j - \mathcal{M}\boldsymbol{a}_j\|_2^2 \ \text{ s.t. } \ \|\boldsymbol{a}_j\|_0 \leq n. \tag{5}$$

Here, n is the sparsity, and $\boldsymbol{a}_j^*$ are the computed optimal coefficients using OMP algorithm, which greedily selects one basis feature with the greatest similarity and computes the coefficient for the selected basis iteratively until all n basis features are selected. Subsequently, the residual $\mathbf{R}_n f_j$ between the pre-selected feature $\boldsymbol{f}_j$ and its optimal representation is calculated as follows,

$$\mathbf{R}_n f_j(\boldsymbol{f}_j, \mathcal{M}) = \boldsymbol{f}_j - \mathcal{M}\boldsymbol{a}_j^*. \tag{6}$$

The representation distance between feature $\boldsymbol{f}_j$ and the current basis set $\mathcal{M}$ is calculated by the Gaussian kernel function,

$$s_j = 1 - e^{-\frac{\|\mathbf{R}_n f_j\|_2^2}{2\sigma^2}}, \tag{7}$$

where σ is a width parameter. A larger distance s_j indicates $\boldsymbol{f}_j$ cannot be closely represented by basis features in $\mathcal{M}$, which will also be used as the anomaly score. The above process is named Anomaly Score calculation by Orthogonal Matching Pursuit (ASOMP), denoted as,

$$\mathrm{ASOMP}(\boldsymbol{f}_j, \mathcal{M}) = 1 - e^{-\frac{\|\boldsymbol{f}_j - \mathcal{M}\boldsymbol{\alpha}_j^*\|_2^2}{2\sigma^2}}, \tag{8}$$

which is detailed in Algorithm 1. Therefore, we use the largest ASOMP to select the best representation basis feature to ensure Criterion B, denoted as

$$\begin{aligned} &\boldsymbol{f}^* = \underset{\boldsymbol{f}_j \in \mathcal{N}}{\arg\max}\ \mathrm{ASOMP}(\boldsymbol{f}_j, \mathcal{M}) \\ &\mathcal{M} \leftarrow \mathcal{M} \cup \{\boldsymbol{f}^*\} \end{aligned}, \tag{9}$$

The entire basis feature sampling (BFS) for the memory bank is summarized in Algorithm 2.

C. Anomaly Detection with SFRAD

In the inference stage, the image-level anomaly detection score Φ and the pixel-level anomaly segmentation score Ω for a test image $\boldsymbol{I}'$ can be calculated using the obtained memory bank $\mathcal{M}$. Specifically, to calculate Φ and Ω, the feature map $\boldsymbol{F}'$ of $\boldsymbol{I}'$ is firstly obtained using Eq. (2) and then to form test feature set $\mathcal{X}'$ as described in Eq. (3), where $\boldsymbol{F}' \in \mathbb{R}^{w \times h \times c}$ and the size of $\mathcal{X}'$ is $N' = |\mathcal{X}'| = w \times h$. For each $\boldsymbol{f}_i' \in \mathcal{X}'$, its anomaly score is calculated as,

$$s_i' = \mathrm{ASOMP}(\boldsymbol{f}_i', \mathcal{M}). \tag{10}$$

Finally, anomaly map Ω and anomaly score Φ of test image $\boldsymbol{I}'$ can be obtained as $\Omega = [s_1', \cdots, s_{N'}']$ and $\Phi = max(\Omega)$. Subsequently, Ω is reshaped and upsampled as (W, H). By setting a segmentation threshold τ_1, the pixel of test image $\boldsymbol{I}'$ is considered as an abnormal pixel when the value of the corresponding position at anomaly map Ω is greater than τ_1. Similarly, a classification threshold τ_2 is set. When anomaly score $\Phi > \tau_2$, test image $\boldsymbol{I}'$ is classified as an anomaly image. The τ_1 and τ_2 are estimated using the acceptable False-Positive Rate (FPR) on the training dataset [49]. The overall inference procedure is summarized in Algorithm 2 as well.

IV. EXPERIMENTAL RESULTS

In this section, we conduct extensive experiments on the tasks of full-shot and low-shot anomaly detection to demonstrate the effectiveness of our proposed SFRAD. We evaluate the performance of our proposed SFRAD on seven commonly-used anomaly detection datasets, i.e., MVTec Anomaly Detection (MVTec AD) [2], Kolektor Surface-Defect Dataset (KolektorSDD) [56], Kolektor Surface-Defect Dataset 2 (KolektorSDD2) [57], MVTec Logical Constraints Anomaly Detection Dataset (MVTec LOCO AD) [58], Visual Anomaly Dataset (VISA) [59], CIFAR-10 [60], and MNIST dataset. We also conduct ablation studies for sparsity n in OMP, BFS, ASOMP, LAPF, and backbone network structures to verify their influence on the anomaly detection performance of SFRAD.

A. Experiment Settings

1) Dataset Description: The MVTec AD [2] dataset is the benchmark used to verify the effectiveness of unsupervised anomaly detection and segmentation methods in industrial inspection. It contains more than 5,000 images, which is made up of 15 different categories (5 for textures and 10 for objects). Each category comprises a set of defect-free training images and a test set of images with various kinds of defects, as well as images without defects. The KolektorSDD [56] and the KolektorSDD2 [57] are two classic surface defect datasets. The KolektorSDD consists of 399 images captured in a controlled industrial environment in a real-world case, including 347 defect-free images and 52 images with visible defects. In our experiments, the KolektorSDD is divided into three folds to perform 3-fold cross-validation, and only defect-free images are used for unsupervised training. The KolektorSDD2 contains more than 3000 images, and only defect-free images are used for unsupervised training. The MVTec LOCO AD [58] dataset is intended for the evaluation of unsupervised anomaly localization algorithms. The dataset includes both structural and logical anomalies. It contains 3644 images from five different categories inspired by real-world industrial inspection scenarios. The VISA [59] dataset contains 12 subsets corresponding to 12 different objects. There are 10,821 images with 9,621 normal and 1,200 anomalous samples. The MNIST dataset is made up of 10 different handwritten digit categories, which contains 60,000 training images and 10,000 testing images. The CIFAR-10 [60] dataset consists of 60,000 32×32 images in 10 classes, with 6000 images per class. There are 50,000 training images and 10,000 test images.

For MVTec AD, KolektorSDD, KolektorSDD2, MVTec LOCO AD, and VISA datasets, only normal images (without any image-level or pixel-level labels) are used for training. For MNIST and CIFAR-10 datasets, the experiments are conducted under the one-class classification detection setting, i.e., one category images in the training dataset are regarded as normal for training, the same category images in the test dataset are regarded as normal images, and the rest images are regarded as abnormal images for inference.

2) Evaluation Metrics: In this paper, the Area Under the Receiver Operating Characteristic Curve (ROC-AUC) and the Per-Region-Overlap (PRO) score are adopted as the evaluation metrics. The ROC-AUC is calculated at the image level and the pixel level. As for the region-level PRO score, the normalized PRO-AUC up to an average per-pixel FPR of 30% is reported following [61].

3) Implementation Details: In all experiments, we use a Wide-ResNet50×2 network pretrained on ImageNet to extract features. The images in the MVTec AD, MVTec LOCO AD, and VISA datasets are resized and cropped to 256×256 and 224×224, respectively. The images in KolektorSDD are resized to 384×160. Note that when inference for RegAD, the images in KolektorSSD are resized to 256×256

TABLE I
THE IMAGE-LEVEL ANOMALY DETECTION ON THE MVTEC AD DATASET. RESULTS ARE LISTED AS AUC-ROC SCORES (%).

Category	AE-L2 [13]	AnoGAN [6]	Patch SVDD [62]	DFR [49]	RIAD [25]	SPADE [35]	Cut Paste [63]	STPM [33]	InTra [26]	PaDiM [54]	Patch Core [36]	ADPS [65]	DAF [66]	GLCF [67]	Ours
Bottle	80.0	80.0	98.6	99.8	99.9	97.2	98.3	**100.**	**100.**	99.8	**100.**	**100.**	99.2	**100.**	**100.**
Cable	56.0	47.6	90.3	79.5	88.4	84.8	80.6	92.3	84.2	92.2	99.5	93.7	94.3	**100.**	99.0
Capsule	62.0	44.2	76.7	96.3	81.9	89.7	96.2	88.0	86.5	91.5	**98.1**	96.0	89.8	95.5	**98.1**
Carpet	50.0	33.7	92.9	97.8	84.2	92.8	93.1	98.9	98.8	**99.9**	98.7	97.4	99.2	99.8	98.3
Grid	78.0	87.1	94.6	94.6	99.6	47.3	99.9	**100.**	**100.**	95.7	98.2	**100.**	**100.**	99.7	98.3
Hazelnut	88.0	25.9	92.0	**100.**	83.3	88.1	97.3	**100.**	95.7	93.3	**100.**	99.6	**100.**	**100.**	**100.**
Leather	44.0	45.1	90.9	99.4	**100.**	95.4	**100.**	99.9	**100.**	**100.**	**100.**	**100.**	**100.**	**100.**	**100.**
Metalnut	73.0	28.4	94.0	93.0	88.5	71.0	99.3	**100.**	96.8	99.2	**100.**	99.7	**100.**	**100.**	99.8
Pill	62.0	71.1	86.1	91.7	83.8	80.1	92.4	93.8	90.2	94.3	96.6	95.3	**98.0**	96.3	97.8
Screw	69.0	10.0	81.3	94.8	84.5	66.7	86.3	88.2	95.7	84.3	**98.1**	89.5	91.1	95.3	96.6
Tile	77.0	40.1	97.8	93.3	98.7	96.5	93.4	95.5	98.2	97.3	98.7	99.8	**100.**	99.8	99.3
Toothbrush	98.0	43.9	**100.**	**100.**	**100.**	88.9	98.3	87.8	99.7	97.2	**100.**	95.3	**100.**	94.5	99.7
Transistor	71.0	69.1	91.5	80.5	90.9	90.3	95.5	93.7	95.8	97.8	**100.**	97.6	95.5	**100.**	99.8
Wood	74.0	56.6	96.5	98.7	93.0	95.8	98.6	99.2	98.0	98.8	99.2	97.7	**99.9**	99.3	99.5
Zipper	80.0	71.5	97.8	90.2	98.1	96.6	99.4	93.6	99.4	90.9	99.4	**100.**	97.7	98.2	99.6
Mean	70.8	50.3	92.1	94.0	91.7	85.3	95.1	95.3	95.8	95.5	**99.1**	97.4	97.6	98.6	**99.1**

TABLE II
THE PIXEL-LEVEL ANOMALY SEGMENTATION ON THE MVTEC AD DATASET. RESULTS ARE LISTED AS AUC-ROC SCORES (%).

Category	AE-SSIM [13]	AE-L2 [13]	AnoGAN [6]	Exp VAE [69]	DFR [49]	RIAD [25]	SPADE [35]	STPM [33]	PaDiM [54]	Patch Core [36]	ADPS [65]	DAF [66]	GLCF [52]	Ours
Bottle	93.0	86.0	86.0	87.0	97.0	98.4	98.4	98.8	98.3	98.6	**99.5**	99.0	98.8	98.6
Cable	82.0	86.0	78.0	90.0	92.0	84.2	97.2	95.5	96.7	**98.5**	94.6	97.3	98.2	98.3
Capsule	94.0	88.0	84.0	74.0	99.0	92.8	99.0	98.3	98.5	98.9	94.1	97.1	98.9	**99.1**
Carpet	87.0	59.0	54.0	78.0	97.0	96.3	97.5	98.8	99.1	99.1	**99.5**	99.2	98.2	99.0
Grid	94.0	90.0	57.9	73.0	98.0	98.8	93.7	99.0	97.3	98.7	99.2	**99.3**	98.9	98.8
Hazelnut	97.0	95.0	87.0	98.0	99.0	96.1	99.1	98.5	98.2	98.7	**99.6**	98.9	98.9	98.6
Leather	78.0	75.0	64.0	95.0	98.0	99.4	97.6	99.3	99.2	99.3	**99.9**	99.7	99.0	99.4
Metalnut	89.0	86.0	76.0	94.0	93.0	92.5	98.1	97.6	97.2	98.4	97.5	**99.2**	97.8	98.7
Pill	91.0	85.0	87.0	83.0	97.0	95.7	96.5	97.8	95.7	97.6	**99.3**	98.5	98.1	98.5
Screw	96.0	96.0	80.0	97.0	99.0	98.8	98.9	98.3	98.5	99.4	98.7	98.7	99.4	**99.5**
Tile	59.0	51.0	50.0	80.0	87.0	89.1	87.4	97.3	94.1	95.9	**99.6**	99.5	95.1	95.8
Toothbrush	92.0	93.0	90.0	94.0	99.0	98.9	97.9	98.9	98.8	98.7	99.1	**99.2**	98.9	98.8
Transistor	90.0	86.0	80.0	93.0	80.0	87.7	94.1	82.5	**98.5**	96.4	92.2	89.4	**98.5**	96.6
Wood	73.0	73.0	62.0	77.0	93.0	85.8	88.5	97.2	94.9	95.1	**99.3**	97.7	94.8	95.5
Zipper	88.0	77.0	78.0	78.0	96.0	97.8	96.5	98.5	98.5	98.9	**99.6**	98.7	98.8	98.8
Mean	86.9	81.6	74.3	86.1	95.0	94.1	96.0	97.1	97.6	98.1	98.1	98.1	98.2	**98.3**

because the pretrained model on all images in the MVTec AD dataset except the bottle category is used. The bilinear interpolate method is used for upsampling the anomaly map to its input resolution. The sampling rate is set to 10%, and k is set to 2 in this paper. For full-shot anomaly detection, we calculate σ^2 in Gaussian kernel distance by setting the maximum Gaussian kernel distance of nominal features to 0.7. In few-shot experiments, due to the small number of normal features (only 78 features in the memory bank for the one-shot setting), it may not be able to estimate the appropriate sigma for low-shot anomaly detection. Therefore, we use the Euclidean distance instead of the Gaussian kernel function (last step) in ASOMP to calculate the anomaly score.

4) Baseline Selection: We compare the proposed SFRAD with state-of-the-art methods on the unsupervised anomaly detection task including: AE-SSIM [13], AE-L2 [13], AnoGAN [6], Patch-SVDD [62], DFR [49], RIAD [25], SPADE [35], CutPaste [63], STPM [33], InTra [26], PaDiM [54], PatchCore [36], Semi-Orthogonal [64], ADPS [65], DAF [66], GLCF [67], DifferNet [39], FastFlow [42], RegAD [47], DeepSVDD [68] and TDG [46], etc. For the MVTec AD dataset, the results of AE-SSIM, AE-L2, AnoGAN, and US are obtained from [33], [49], [54]. The results of other methods are reported from the original papers. For the KolektorSDD and KolektorSDD2, the results of US, PaDiM, and Semi-Orthogonal are obtained from [64]. Results not specified separately are obtained by running the experiments with the codes provided in the original articles.

B. Comparison of Full-Shot Anomaly Detection

1) The MVTec AD Dataset: We quantitatively analyze the performance of our proposed SFRAD and all compared methods. The results of image-level ROC-AUC, pixel-level ROC-AUC, and region-level PRO-AUC on the MVTec AD dataset are reported in Tables I, II, and III, respectively. The best and second-best results are marked in bold and underlined, respectively.

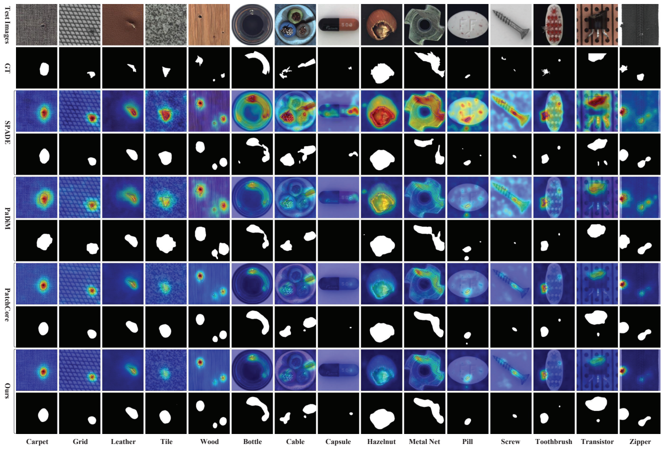

Fig. 2. Qualitative results on the MVTec AD dataset for our method across different categories. Each column shows an example of different categories.

TABLE III
THE REGION-LEVEL ANOMALY SEGMENTATION ON THE MVTEC AD DATASET. RESULTS ARE LISTED AS PRO SCORES (%).

Category	AE-SSIM [13]	AE-L2 [13]	AnoGAN [6]	VAE [70]	US [61]	DFR [49]	SPADE [35]	STPM [33]	PaDiM [54]	Patch Core [36]	ADPS [65]	DAF [66]	Ours
Bottle	83.3	91.0	62.0	70.5	91.8	93.0	95.5	95.1	94.8	96.1	**97.8**	95.5	94.8
Cable	47.8	82.5	38.3	77.9	86.5	81.0	90.9	87.7	88.8	92.6	82.9	89.7	**96.7**
Capsule	86.0	86.2	30.5	77.9	91.6	**97.0**	93.7	92.2	93.5	95.5	93.4	84.1	95.1
Carpet	64.7	45.6	20.4	61.9	69.5	93.0	94.6	95.8	96.2	96.6	**97.5**	97.3	94.8
Grid	84.8	58.1	22.6	40.8	81.8	93.0	86.7	96.6	94.6	95.9	**97.2**	96.7	95.1
Hazelnut	91.6	91.7	69.8	77.0	93.7	**97.0**	95.3	94.3	92.6	93.9	95.2	96.1	95.7
Leather	56.1	81.8	37.8	64.9	81.8	97.0	97.2	98.0	97.8	98.9	**99.3**	98.9	97.3
Metalnut	60.3	83.0	32.0	57.5	89.5	90.0	94.3	94.5	85.6	91.3	91.5	**95.6**	94.5
Pill	83.0	89.3	77.6	79.3	93.5	96.0	94.6	**96.5**	92.7	94.1	96.0	91.1	95.8
Screw	88.7	75.4	46.6	66.4	92.8	96.0	96.0	93.0	94.4	**97.9**	94.3	93.4	97.3
Tile	17.5	89.7	17.7	24.2	91.2	79.0	75.9	92.1	86.0	87.4	**97.9**	96.8	80.2
Toothbrush	78.4	82.1	74.9	85.3	86.3	93.0	**93.5**	92.2	93.1	91.4	93.4	92.7	90.0
Transistor	72.5	72.8	54.9	61.0	70.1	79.0	87.4	69.5	84.5	83.5	85.4	76.7	**91.8**
Wood	60.5	72.7	38.6	57.8	72.5	91.0	87.4	93.6	91.1	89.6	**96.6**	94.5	88.8
Zipper	66.5	83.8	46.7	60.8	93.3	90.0	92.6	95.1	95.9	97.1	**98.3**	95.1	95.6
Mean	69.3	79.1	44.7	64.2	85.7	91.0	91.7	92.4	92.1	93.4	**94.4**	93.0	93.6

As shown in Table I, both SFRAD and PatchCore achieve a mean result of 99.1%, with the latter method showing state-of-the-art performance so far. The above results demonstrate that the proposed SFRAD also achieves state-of-the-art performance in image-level anomaly detection. Furthermore, our proposed SFRAD exhibits good stability and achieves the best performance in four categories and the second-best performance in eight categories. For the pixel-level segmentation comparison on the MVTec AD dataset, the SFRAD achieves the best mean result. This shows that the proposed SFRAD can detect anomalies excellently at the pixel level. At the region level, our proposed SFRAD still outperforms the PatchCore in mean PRO score on the MVTec AD dataset.

Qualitative visualization comparisons of our proposed SFRAD with three state-of-the-art methods, i.e., SPADE, PaDiM, and PatchCore, are shown in Fig. 2. Five texture images and ten object images are displayed in 15 columns in Fig. 2. The test images are shown in the first row, and their corresponding ground truths are shown in the second row. Anomaly maps and segmentation results of SPADE, PaDiM, and PatchCore are shown in the third to eighth rows, respectively. Anomaly maps and segmentation results of our method

TABLE IV
THE PIXEL-LEVEL ANOMALY SEGMENTATION ON THE KOLEKTORSDD AND KOLEKTORSDD2. RESULTS ARE LISTED AS AUC-ROC SCORES (%). WE REPORT THE SCORE FOR EACH FOLD OF THE KOLEKTORSDD, THEIR MEAN AND STANDARD DEVIATION (STD.), AND THE MEAN SCORE FOR THE KOLEKTORSDD2 WITH THE STANDARD DEVIATION WITH THREE RANDOM SEEDS.

Category	US [61]	PaDiM [54]	Semi-Orthogonal [64]	PatchCore [36]	ADPS [65]	Ours
Fold 1	90.4	93.9	95.3	96.9	-	**97.2**
Fold 2	88.3	93.5	95.1	98.1	-	**98.4**
Fold 3	90.2	96.2	97.6	98.2	-	**98.8**
Mean	89.6±1.2	94.5±1.5	96.0±1.4	97.8±0.7	96.3±0.0	**98.1±0.8**
KolektorSDD2	95.0±0.5	95.6±0.0	98.1±0.0	99.5±0.0	99.2±0.0	**99.6±0.0**

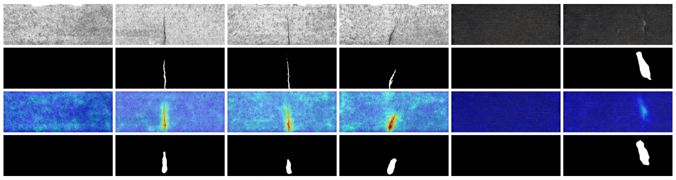

Fig. 3. Qualitative results on the KolektorSDD and KolektorSDD2 for our method. Rows one to four are test images, ground truths, anomaly maps, and segmentation results. The first four columns show the results for a normal image and three anomaly images in the KolektorSDD. The last two columns show the results for a normal image and an anomaly image in the KolektorSDD2.

TABLE V
THE IMAGE-LEVEL ANOMALY DETECTION ON THE MVTEC LOCO AD DATASET.

Category	SPADE		PatchCore		Ours	
	Logical	Structural	Logical	Structural	Logical	Structural
Breakfast_box	63.1	69.0	80.9	72.3	82.5	74.4
Juice_bottle	79.5	81.4	91.5	97.9	91.6	98.2
Pushpins	53.2	68.2	69.0	82.9	68.6	87.6
Screw_bag	51.4	76.9	57.6	90.4	57.8	91.0
Splicing_connectors	64.0	64.9	78.0	95.3	78.2	96.4
Mean	62.2	72.1	75.4	87.8	**75.8**	**89.5**

TABLE VI
THE PERFORMANCE COMPARISON OF ANOMALY DETECTION AND LOCALIZATION ON THE VISA DATASET.

Category	Image-level ROC-AUC			Pixel-level ROC-AUC			Region-level PRO-AUC		
	SPADE	PatchCore	Ours	SPADE	PatchCore	Ours	SPADE	PatchCore	Ours
Candle	81.6	99.0	98.9	99.0	99.2	99.3	96.5	95.2	95.3
Capsules	61.6	75.2	80.0	99.2	98.6	99.2	89.5	81.5	88.8
Cashew	92.1	97.5	97.7	93.5	98.7	98.9	89.1	94.2	93.6
Chewinggum	93.6	99.5	99.6	98.3	98.7	98.9	68.6	82.5	85.3
Fryum	83.0	94.1	96.4	91.9	95.0	95.4	84.6	87.7	89.5
Macaroni1	62.6	94.5	97.6	99.8	99.2	99.6	97.2	94.1	97.2
Macaroni2	49.6	72.0	82.1	98.6	97.4	99.0	86.4	92.2	97.1
Pcb1	92.7	97.8	98.8	99.2	99.7	99.8	93.1	93.5	94.6
Pcb2	86.0	95.5	96.9	96.7	98.7	98.8	88.3	89.1	90.4
Pcb3	78.8	96.9	97.9	99.1	98.9	99.0	89.0	89.6	90.4
Pcb4	96.9	99.5	99.5	95.2	98.1	98.2	78.7	86.5	87.1
Pipe_fryum	77.5	99.9	99.9	98.4	99.0	99.2	94.4	94.5	94.7
Mean	79.7	93.4	**95.4**	97.4	98.4	**98.8**	88.0	90.0	**92.0**

are shown in the last two rows, respectively. We can see that the proposed SFRAD pays more attention to the abnormal regions of images. Although there are significant differences in the location, shape, and size of anomalies, our method still significantly detects anomalous regions. Compared with the other three methods, there are fewer false detections (column 6, bottle, SPADE; column 11, pill, all three compared methods) and missed detections (column 8, capsule, PaDiM) in our proposed SFRAD. In addition, our method is simpler to operate, without alignment operations and a large memory to store features in the memory bank.

2) The KolektorSDD and KolektorSDD2: We compare the proposed SFRAD method with five unsupervised anomaly detection methods, i.e., US, PaDiM, Semi-Orthogonal, PatchCore, and ADPS, on two industrial datasets, KolektorSDD and KolektorSDD2, to verify its effectiveness. The results of all compared methods on these two datasets are listed in Table IV. The detection results on KolektorSDD and KolektorSDD2 show that SFRAD achieves the highest pixel-level ROC-AUC, outperforming the other five comparison methods. On the KolektorSDD2, SFRAD achieves the highest pixel-level ROC-AUC. Overall, our method achieves at least 97.2% pixel-level results on two industrial datasets, which demonstrates that SFRAD is applicable to different anomaly detection datasets. We also present the qualitative results of SFRAD, as shown in Fig. 3. It can be observed that SFRAD can detect complete anomalous regions. This proves that SFRAD can be adapted to anomaly detection in real industrial scenarios with different materials and different products.

3) The MVTec LOCO AD Dataset: For the image-level anomaly detection experiments on the MVTec LOCO AD dataset, as shown in Table V, the proposed SFRAD has demonstrated exceptional performance. When compared with the SPADE and PatchCore, SFRAD achieves the highest average scores in both the logical and structural dimensions, with 75.8% and 89.5% respectively. Particularly in the structural dimension, SFRAD significantly outperforms the other methods, indicating a pronounced advantage in identifying structural anomalies within images. Moreover, SFRAD has shown high accuracy across most categories, such as in the "Juice_bottle" category, where SFRAD achieves a high score of 98.2% in the structural dimension.

4) The VISA Dataset: The quantitative results on the VISA dataset are shown in Table VI. SFRAD shows excellent performance on image-level, pixel-level and region-level anomaly detection, especially on Candle, Cashew, Chewinggum, and Pcb1, etc.

5) MNIST and CIFAR-10: The proposed SFRAD is evaluated on the MNIST and CIFAR-10 datasets specifically for one-class classification tasks. The evaluation focuses solely on image-level anomaly detection metrics for these datasets. Table VII and Table VIII show the results on the MNIST and CIFAR-10 datasets, respectively. It can be seen that SFRAD exhibits impressive performance overall. Remarkably, SFRAD outperforms other methods across all ten categories of the MNIST dataset, and its average result is 4.5% higher than DeepSVDD. Simultaneously, SFRAD also achieved superior performance on the CIFAR-10 dataset. In conclusion, SFRAD has demonstrated strong and competitive performance in image-level anomaly detection on both the MNIST and CIFAR-10 datasets.

C. Comparison of Low-Shot Anomaly Detection

1) The MVTec AD Dataset: The performances of low-shot anomaly detection (using only a small number of normal images as the training set) are investigated. Referring to PatchCore, we validate the influence of our proposed SFRAD on low-shot anomaly detection using the number of normal images from 1 to 50 (i.e., corresponding to 0.4% to 21% of the total normal images). The performances of low-shot anomaly

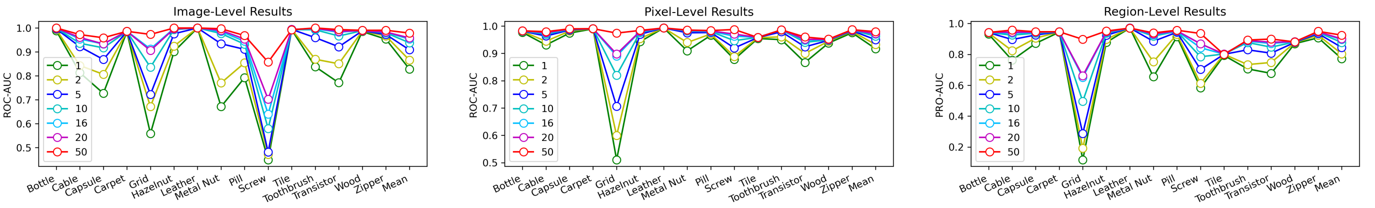

Fig. 4. The influence of low-shot anomaly detection on the MVTec AD dataset.

TABLE VII
THE IMAGE-LEVEL ANOMALY DETECTION ON THE MNIST DATASET.

Category	AnoGAN	OC-SVM	DeepSVDD	SFRAD
0	96.6	98.6	98.0	**99.9**
1	99.2	99.5	99.7	**99.9**
2	85.0	82.5	91.7	**99.0**
3	88.7	88.1	91.9	**99.1**
4	89.4	94.9	94.9	**99.1**
5	88.3	77.1	88.5	**99.1**
6	94.7	96.5	98.3	**99.6**
7	93.5	93.7	94.6	**99.2**
8	84.9	88.9	93.9	**99.2**
9	92.4	93.1	96.5	**99.3**
Mean	91.27	91.29	94.8	**99.3**

TABLE VIII
THE IMAGE-LEVEL ANOMALY DETECTION ON THE CIFAR-10 DATASET.

Category	AnoGAN	OC-SVM	DeepSVDD	SFRAD
Airplane	67.1	61.6	61.7	**85.0**
Automobile	54.7	63.8	65.9	**86.3**
Bird	52.9	50.0	50.8	**74.4**
Cat	54.5	55.9	59.1	**67.1**
Deer	65.1	66.0	60.9	**87.0**
Dog	60.3	62.4	65.7	**72.2**
Frog	58.5	74.7	67.7	**89.2**
Horse	62.5	62.6	67.3	**84.7**
Ship	75.8	74.9	75.9	**85.1**
Truck	66.5	75.9	73.1	**85.2**
Mean	61.8	64.8	64.8	**81.6**

detection are shown in Fig. 4. For most texture classes (carpet, leather, tile, wood) and some target classes (bottle, zipper), robust performances on the three evaluation metrics can be achieved with a small number of images, even one normal image. For the hazelnut and pill categories, the performances of low-shot anomaly detection are slightly affected. Even using only one normal image as the nominal dataset, our method is still competitive. For cable, capsule, toothbrush, and transistor categories, when the number of images in the nominal dataset is less than five, the image-level results are partly affected. For example, in the capsule category, comparing using one image with using five images as the nominal dataset, the image-level ROC-AUC dropped by about 16%. The above phenomenon is due to the fact that randomly selecting one image as the nominal dataset cannot represent all normal images for some categories. For the grid category, when the number of images is less than 20, all three metric results are affected to a certain extent. The reason is that the difference between normal patch features and abnormal patch features is small, and it is easy to misclassify when the number of nominal images is insufficient.

TABLE IX
LOW-SHOT ANOMALY DETECTION PERFORMANCE ON THE MVTEC AD DATASET.

Method\Shots Retained	1 0.4	2 0.8	5 2.1	10 4.1	16 6.6	20 8.3	50 21
Image-level ROC-AUC							
SPADE [35]	74.9±1.1	76.2±1.2	77.5±0.7	77.7±0.7	78.3±0.6	77.7±0.8	78.0±0.3
PaDiM [54]	76.0±1.3	78.8±1.2	81.7±0.9	84.2±0.9	86.5±0.6	87.4±0.5	91.4±0.5
DifferNet [39]	73.1±1.7	75.5±1.4	79.8±1.5	83.1±1.2	85.4±0.6	86.0±1.0	88.6±0.7
FastFlow [42]	63.6±2.9	72.2±2.9	86.4±0.8	87.1±0.9	88.2±0.9	88.8±0.8	90.0±0.6
RegAD [47]	76.7±1.5	81.5±1.2	85.5±0.8	88.0±0.5	89.9±0.6	90.7±0.4	89.5±0.2
PatchCore [36]	**83.4±1.0**	**86.9±1.0**	**90.9±1.0**	93.7±0.6	95.2±0.5	95.4±0.2	97.4±0.4
Ours	82.8±0.9	86.6±1.0	**90.9±1.0**	**93.8±0.6**	**95.3±0.5**	**95.8±0.3**	**97.9±0.3**
Pixel-level ROC-AUC							
SPADE [35]	89.5±0.2	92.1±0.3	93.8±0.3	94.9±0.3	95.9±0.1	96.1±0.1	96.6±0.0
PaDiM [54]	87.9±0.4	90.2±0.4	92.4±0.2	93.8±0.2	94.8±0.9	95.1±0.1	96.4±0.0
FastFlow [42]	70.6±2.7	82.6±2.0	92.0±0.7	94.3±0.7	95.9±0.3	96.2±0.4	97.2±0.3
RegAD [47]	**93.1±0.3**	**94.7±0.3**	**96.2±0.2**	**96.9±0.1**	96.2±0.1	**97.1±0.1**	96.4±0.1
PatchCore [36]	91.5±0.4	93.1±0.3	94.7±0.3	96.0±0.4	96.7±0.3	96.9±0.2	97.8±0.1
Ours	91.6±0.3	93.2±0.3	94.9±0.4	96.2±0.4	**97.0±0.3**	**97.1±0.2**	**97.9±0.1**
Region-level PRO-AUC							
SPADE [35]	76.7±0.5	82.0±0.5	85.7±0.5	88.1±0.6	**89.7±0.3**	90.1±0.2	91.3±0.1
PaDiM [54]	70.9±0.8	76.1±1.0	81.4±0.6	84.4±0.6	86.6±0.4	87.5±0.4	90.2±0.1
FastFlow [42]	51.3±2.6	53.2±2.4	77.8±1.4	83.9±1.3	87.7±1.0	88.9±1.0	91.3±0.8
RegAD [47]	**79.8±0.8**	**83.6±0.8**	**87.7±0.4**	**89.6±0.3**	88.8±0.2	**90.3±0.1**	87.6±0.2
PatchCore [36]	77.3±0.6	80.3±0.7	84.0±0.8	87.2±0.8	88.8±0.8	89.3±0.7	91.7±0.2
Ours	77.2±0.4	80.3±0.7	84.5±0.9	87.7±1.1	89.6±1.0	90.1±0.7	**92.4±0.2**

TABLE X
LOW-SHOT ANOMALY DETECTION PERFORMANCE ON THE KOLEKTORSDD.

Method\Shots Retained	1 0.4	2 0.8	5 2.1	10 4.1	16 6.6	20 8.3	50 21
Image-level ROC-AUC							
SPADE [35]	57.2±3.3	56.3±4.6	58.1±3.8	59.9±1.8	60.0±1.4	60.1±1.0	60.7±0.5
PaDiM [54]	64.3±4.8	71.5±2.8	76.2±1.8	78.7±1.6	81.7±1.3	82.8±1.2	86.9±0.8
DifferNet [39]	73.5±3.5	77.7±3.0	83.1±2.0	86.2±1.1	87.7±1.0	88.0±0.4	89.8±1.0
FastFlow [42]	55.2±4.3	64.5±9.1	79.9±3.6	83.3±2.9	86.3±3.1	89.1±2.5	89.1±1.8
RegAD [47]	70.9±2.6	59.6±1.8	62.4±1.9	71.0±1.1	76.4±1.1	70.6±1.5	76.9±0.8
PatchCore [36]	78.3±3.7	82.7±4.2	87.5±2.0	89.6±1.0	89.9±0.6	90.7±0.7	90.9±0.8
Ours	**79.1±4.3**	**83.6±3.6**	**88.3±2.3**	**90.2±0.8**	**91.1±0.8**	**91.3±0.5**	**92.3±0.6**
Pixel-level ROC-AUC							
SPADE [35]	94.4±0.3	94.9±0.4	95.6±0.2	96.0±0.2	96.3±0.1	96.3±0.1	96.7±0.1
PaDiM [54]	90.9±0.9	93.5±0.5	95.1±0.3	95.7±0.2	96.3±0.1	96.4±0.2	97.1±0.1
FastFlow [42]	63.8±8.5	81.3±7.6	90.0±2.0	92.1±1.7	94.6±1.8	95.7±1.1	95.5±0.6
RegAD [47]	82.8±2.9	70.5±1.4	82.9±0.7	87.7±0.3	87.5±0.2	85.7±0.3	84.7±0.2
PatchCore [36]	95.6±0.5	96.2±0.4	96.9±0.3	97.2±0.2	97.2±0.3	97.2±0.2	97.5±0.1
Ours	**96.1±0.5**	**96.7±0.5**	**97.3±0.3**	**97.5±0.2**	**97.7±0.1**	**97.7±0.1**	**97.9±0.1**
Region-level PRO-AUC							
SPADE [35]	64.5±1.1	66.6±1.2	69.2±1.0	71.0±1.0	72.3±0.5	72.5±0.7	74.8±0.5
PaDiM [54]	56.4±3.6	63.9±1.9	69.2±1.0	71.6±1.0	73.5±0.8	74.5±1.1	77.0±0.7
FastFlow [42]	50.8±4.6	51.7±3.9	58.4±4.3	64.8±5.0	70.0±6.1	74.2±4.0	70.9±3.0
RegAD [47]	31.9±2.8	18.4±1.0	33.2±1.2	43.6±1.3	42.2±0.7	37.7±0.7	40.2±0.4
PatchCore [36]	72.1±2.4	75.1±2.0	77.8±1.8	79.6±1.1	80.1±1.5	80.5±1.0	81.2±1.0
Ours	**74.4±2.7**	**77.9±2.2**	**79.8±1.4**	**81.5±1.2**	**82.7±0.6**	**82.9±0.6**	**83.5±0.7**

Furthermore, the results of our method and six other compared methods (i.e., SPADE, PaDiM, DifferNet, FastFlow, RegAD, and PatchCore) on low-shot anomaly detection are shown in Table IX. The mean score and standard deviation with twenty random seeds when $K \leq 10$ and ten random seeds when $K > 10$ for each category are reported. The same setting is used in the comparison methods. For image-level anomaly detection, our proposed SFRAD has consistently

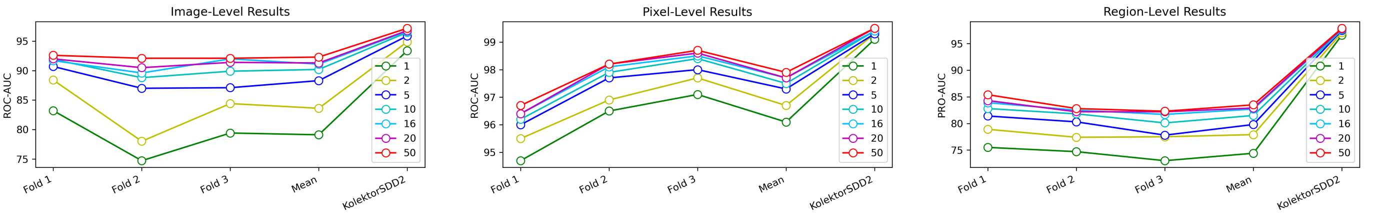

Fig. 5. The influence of low-shot anomaly detection on the KolektorSDD and KolektorSDD2.

TABLE XI
LOW-SHOT ANOMALY DETECTION PERFORMANCE ON THE KOLEKTORSDD2.

Method\Shots Retained	1 0.4	2 0.8	5 2.1	10 4.1	16 6.6	20 8.3	50 21
Image-level ROC-AUC							
SPADE [35]	76.6±3.1	77.9±2.6	79.5±1.6	79.7±1.2	80.4±0.8	80.4±1.1	80.6±0.5
PaDiM [54]	84.7±2.3	86.8±0.8	87.7±0.5	88.5±0.6	89.3±0.4	89.5±0.4	91.1±0.2
DifferNet [39]	89.7±1.3	89.9±0.9	90.4±0.5	90.6±0.9	90.6±0.8	90.9±0.7	92.1±0.6
FastFlow [42]	54.7±12.6	66.0±17.5	85.6±5.4	83.5±3.9	85.6±4.5	87.7±4.5	92.5±3.3
RegAD [47]	86.6±0.7	90.1±0.5	89.0±0.4	92.4±0.4	90.4±0.3	89.7±0.3	90.1±0.2
PatchCore [36]	**93.6±1.2**	94.5±1.2	95.3±0.8	96.1±0.4	96.3±0.4	96.4±0.3	96.8±0.4
Ours	93.4±1.7	**94.9±1.3**	**95.9±0.8**	**96.5±0.5**	**96.7±0.3**	**96.8±0.2**	**97.2±0.3**
Pixel-level ROC-AUC							
SPADE [35]	97.7±0.2	98.0±0.2	98.5±0.1	98.6±0.1	98.7±0.0	98.8±0.0	98.9±0.0
PaDiM [54]	95.8±0.4	97.4±0.3	98.1±0.1	98.4±0.1	98.6±0.1	98.7±0.0	98.9±0.0
FastFlow [42]	62.2±18.0	83.8±11.9	93.8±2.7	93.3±2.8	95.8±2.1	96.3±1.7	97.2±1.4
RegAD [47]	97.4±0.2	96.5±0.5	97.8±0.2	98.2±0.1	98.0±0.1	97.8±0.1	98.3±0.1
PatchCore [36]	99.0±0.2	99.1±0.2	**99.3±0.1**	99.3±0.1	**99.4±0.0**	99.4±0.0	99.4±0.0
Ours	**99.1±0.2**	**99.3±0.1**	**99.3±0.1**	**99.4±0.0**	**99.4±0.0**	**99.5±0.0**	**99.5±0.0**
Region-level PRO-AUC							
SPADE [35]	95.3±0.3	96.0±0.6	96.7±0.3	97.0±0.3	97.1±0.2	97.3±0.2	97.5±0.2
PaDiM [54]	93.0±0.8	95.8±0.4	96.8±0.2	97.2±0.2	97.4±0.1	97.5±0.2	97.8±0.1
FastFlow [42]	49.8±0.6	52.8±9.2	89.4±4.0	89.5±3.0	92.4±3.0	93.9±2.4	95.3±1.5
RegAD [47]	91.8±0.5	91.3±0.5	91.9±0.3	93.4±0.3	92.1±0.4	91.9±0.5	94.0±0.7
PatchCore [36]	96.3±0.6	96.8±0.6	97.0±0.4	97.3±0.3	97.4±0.3	97.4±0.3	97.6±0.2
Ours	**96.6±0.6**	**97.1±0.6**	**97.5±0.4**	**97.6±0.2**	**97.7±0.2**	**97.8±0.2**	**97.9±0.2**

demonstrated superior results compared to the other methods listed, especially when $K > 2$. For pixel-level anomaly detection, our method maintains competitiveness with existing methods, except for $K \leq 10$. When $K \leq 10$, our method is only inferior to RegAD. It's worth noting that RegAD incorporates a substantial number of additional images for training its model. In addition, as K increases, the rate of performance improvement in RegAD slows down, and in some instances, a decline in performance is observed. In contrast, our method leverages the ImageNet pretrained model to extract image features, which yields favorable results. As K continues to rise, our method's performance exhibits a consistent upward trend. In addition, our method exhibits sensitivity towards categories with excessively large or small abnormal regions (such as grid, screw, and transistor), as depicted in Fig. 4. When not accounting for the impact of these categories, the performance of our method consistently outperforms that of the comparison methods listed. When $K > 10$, our method overtakes the listed comparative methods by a significant margin in terms of performance. For the region-level anomaly detection, when $K \leq 10$, the performance of our method lags behind that of SPADE. This difference in performance can be attributed to SPADE's utilization of both earlier layers (Layer1) and deeper layers (Layer2, Layer3) to calculate anomaly maps for features of each layer. These anomaly maps are then aligned for pyramid matching to detect anomalies. In contrast, our approach only employs deep layers (Layer2 and Layer3). It's worth noting that in situations where available images are limited, the incorporation of supplementary information can enhance the method's performance. Therefore, when $K \leq 10$, our method does not get the optimal results. When $K > 10$, our method achieves competitive results, further demonstrating the superiority of our proposed SFRAD on low-shot anomaly detection.

2) The KolektorSDD and KolektorSDD2: Fig. 5 shows the results of our method for low-shot anomaly detection on the KolektorSDD and KolektorSDD2. As the number of nominal images increases, anomaly detection performance continues to improve. When the number of nominal images exceeds 10, the performance tends to be stable. Additionally, compared to others, our proposed SFRAD is more robust on the KolektorSDD2.

Table X and Table XI record the comparison results of different methods on KolektorSDD and KolektorSDD2, respectively. For RegAD, the model is trained using aggregated data from all categories in MVTec AD, except for the bottle category, and directly applicable to KolektorSDD and KolektorSDD2. For FastFlow, the model undergoes 500 iterations without any parameter adjustments, and no validation set is designated for optimal model selection. It can be seen that our proposed SFRAD outperforms SPADE, PaDiM, DifferNet, FastFlow, and RegAD on low-shot anomaly detection by a large margin. Moreover, the results of our SFRAD are better than those of PatchCore in most cases. These comparative results further demonstrate the superiority of our method on low-shot anomaly detection.

3) The MVTec LOCO AD dataset: In this subsection, we conduct an in-depth evaluation of SFRAD's performance for low-shot anomaly detection on the MVTec LOCO AD dataset. As illustrated in Fig. 6, SFRAD achieves satisfactory results in detecting logical and structural anomalies. The accuracy of SFRAD steadily increases with the number of normal samples. Compared to existing methods such as SPADE, RegAD, and PatchCore, SFRAD demonstrates strong competitiveness under one-shot, five-shot, and ten-shot settings. Notably, when $k \geq 5$, SFRAD achieves higher accuracy in both logical and structural anomaly detection. Overall, our method exhibits high performance and stability in terms of average accuracy.

4) The VISA Dataset: Fig. 7 compares the performance of our method with SPADE, RegAD, and PatchCore on the VISA dataset. Through a comprehensive evaluation for few-shot anomaly detection, SFRAD consistently demonstrates high accuracy across one-shot, five-shot, and ten-shot settings, showcasing exceptional robustness. In particular, SFRAD's

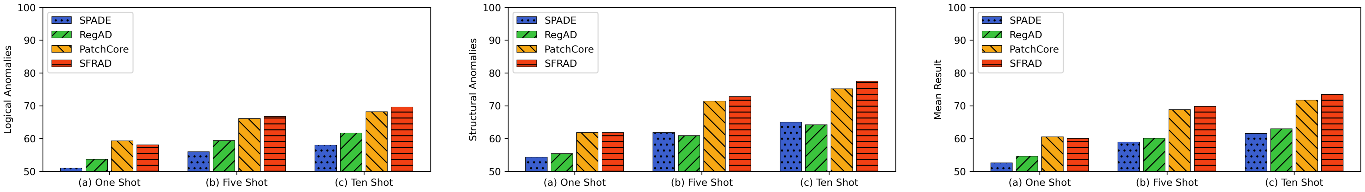

Fig. 6. Low-shot anomaly detection Performance on the MVTec LOCO AD dataset (Image-level ROC-AUC).

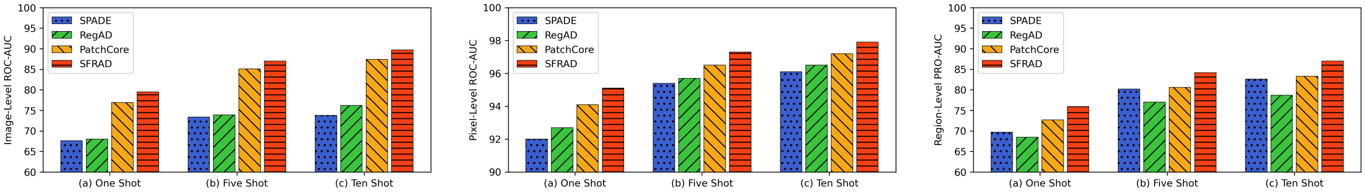

Fig. 7. Low-shot anomaly detection Performance on the VISA dataset.

anomaly detection capability under low-shot conditions is notably superior to other methods. This highlights its potential for practical applications, especially in industrial inspection scenarios where sample acquisition is limited.

5) MNIST and CIFAR-10: Fig. 8 shows the results of our method in comparison to PatchSVDD, DeepSVDD, and TDG on the MNIST dataset. Our method outperforms all comparison methods under the low shot setting. This demonstrates that our method can still achieve excellent results in one-class classification with a limited number of images. Furthermore, we evaluate SFRAD on the CIFAR-10 dataset, as shown in Fig. 9. The results obtained by SFRAD significantly surpass those of the PatchCore DeepSVDD method. On the one-shot setting, our method is not as good as TDG. This is because TDG uses conditional generators to extend a normal image to multiple images, while our method only extracts features of an image. In some natural images, an image (or the subsets) cannot represent the feature variation of normal samples [39]. On the five-shot and ten-shot settings, our method's results are on par with those of TDG. These underscore the superiority of our approach.

D. Analysis of Computational Complexity

In this subsection, we conduct an analysis of the computational complexity of SFRAD and compare its inference time with other benchmark methods. Firstly, we examine the computational complexity of SFRAD during the training phase, denoted by $O(\cdot)$ notation, which hides constant factors and smaller terms [71]. As depicted in Algorithm 2, the computational complexity is solely dependent on the predetermined size of the memory bank $\mathcal{M}$ and the size of the anomaly-free image set $\mathcal{I}$. Therefore, the computational complexity of the training phase is $O(|\mathcal{M}|\,|\mathcal{I}|)$. During the testing phase, the computational complexity is only related to the size of memory bank $\mathcal{M}$, so the time complexity is $O(|\mathcal{M}|)$ in the testing phase.

We evaluate the inference speed of our method and other comparison methods under the $K = 20$ settings. The statistics for inference speed are presented in Table XII. In general, when $n = 1$, our method consistently achieves a frame rate of over five Frames Per Second (FPS), which surpasses the inference speed of SPADE, PaDiM, DifferNet, and RegAD, but falls short of FastFlow and PatchCore. Furthermore, as the value of n is increased, our method demonstrates enhanced anomaly detection capabilities, albeit at the cost of increased inference time. Nevertheless, it's worth mentioning that the inference speed of our method remains faster than that of some comparison methods. In terms of performance, our method outperforms the listed comparative methods by at least 0.4% at the image level and 0.2% at the pixel level. Our method provides the flexibility so that users can adjust the configuration to balance the performance and inference speed for different application requirements.

E. Ablation Study

We conduct extensive ablation experiments on the MVTec AD dataset and analyze the influence of each component of our method on anomaly detection to demonstrate the superiority of the proposed SFRAD. The main contents of ablation research include the following six aspects: 1) the influence of sparsity n in the OMP algorithm, 2) the influence of BFS, 3) the influence of ASOMP, 4) the influence of LAPF, 5) the influence of backbone network structures, and 6) the influence of distribution shift.

1) Influence of Sparsity n in OMP: This subsection verifies the influence of sparsity n in OMP on anomaly detection performance. As shown in Fig. 10, the value of n is more robust to most categories, such as bottle, carpet, hazelnut, leather, metal nut, and zipper. For these categories, stable performance and good results are obtained in three metrics regardless of the value of n. For capsule, grid, and pill categories, when $n \leq 3$, the image-level results are decreased. For the pixel-level results, different n values have little influence on the performance. When $n = 1$, the performance of the proposed SFRAD only drops slightly for pill, tile, transistor, and wood categories. For the region-level results, when $n = 1$, the performance of our SFRAD for the grid, metal nut, pill, and tile categories drops slightly. To sum up, when the n

TABLE XII
THE INFERENCE SPEED PER IMAGE ON MVTEC AD.

Method	SPADE	PaDiM	DifferNet	FastFlow	RegAD	PatchCore	Ours ($n = 1$)	Ours ($n = 5$)
Speed (FPS)	0.41	0.59	3.11	20.96	2.59	6.31	5.03	1.11

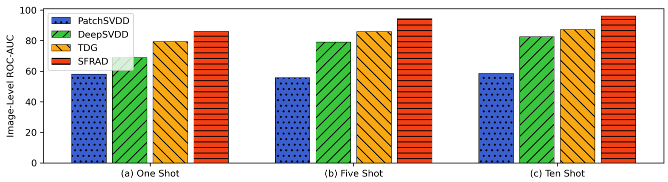

Fig. 8. Low-shot anomaly detection Performance on the MNIST dataset.

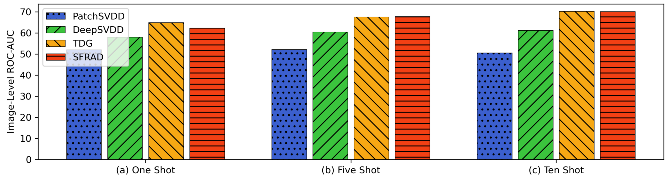

Fig. 9. Low-shot anomaly detection Performance on the CIFAR-10 dataset.

value is small, it has a certain influence on some categories, and when $n > 5$, all categories can achieve stable results. The above analysis concludes that, when n is large, the feature representation can more accurately fit a normal feature. Thus, the distance between a normal feature and its feature representation is smaller than that of an abnormal feature. Therefore, it is easier to distinguish normal features from abnormal features by setting a larger n. Considering the time cost, we set $n = 5$ in this paper.

2) Influence of BFS: This subsection discusses the influence of BFS in our proposed SFRAD on anomaly detection performance. First, we set $\alpha = \{0.1\%, 0.5\%, 1\%, 5\%, 10\%, 20\%\}$ to verify the influence of sampling rate α in BFS on the performance of SFRAD, as shown in Fig. 11. when sampling rate α is extremely small, the performance of the method is affected to a certain extent on the minority categories. When sampling rate $\alpha > 1\%$, robust results can be achieved in almost all categories. Furthermore, we compare the performance of Greedy Coreset Selection (GCS) [36] and BFS for the memory bank on three metrics when sampling rate $\alpha = \{1\%, 10\%\}$, and the results are shown in Fig. 12. Our proposed SFRAD achieves the highest performance in ten categories for the image-level results. In terms of pixel-level results, the mean result of our method is not much different from that of GCS. For the region-level metric, our method achieves the best performance in 9 categories, which is significantly higher than GCS. In general, our proposed BFS demonstrates certain competitiveness.

3) Influence of ASOMP: In this subsection, we construct some baseline methods — Similarity Distance, L2 Distance, Cosine Distance, and Mahalanobis Distance, where the ASOMP-based anomaly score in SFRAD is replaced with those different distances, respectively. SFRAD is compared with those baseline methods to verify the influence of ASOMP on anomaly detection performance. The comparison results are shown in Fig. 13. For the image-level results, ASOMP achieves higher results on nine categories than other distances, where the mean result of ASOMP is also the highest. For the pixel-level results, the mean performance of ASOMP is slightly higher than Similarity Distance, L2 Distance, and Mahalanobis Distance. Moreover, the highest results are achieved in seven categories, of which the result of the pill category is significantly higher than other distances. In terms of region-level results, the results of ASOMP on ten categories are all higher than the latter, and the mean result is also at least 0.5% higher than other distances. Therefore, our proposed ASOMP is more robust than the other four distances.

4) Influence of LAPF: In this subsection, we explore the influence of using LAPF on the anomaly detection performance of SFRAD. To assess the influence of LAPF on various layers of the feature extractor, we conduct separate experiments for different values of l, specifically $l = \{2, 3\}$, $l = 2$, and $l = 3$. The results of SFRAD with (w) and without (wo) LAPF are presented in Fig. 14. Across all categories, LAPF demonstrates the ability to enhance anomaly detection performance to some extent. More specifically, LAPF effectively improves anomaly detection performance for categories with larger targets or defects (e.g., transistor) while imposing limitations on categories with minor defects (e.g., grid). Additionally, for shallow feature extractors (e.g., $l = 2$), SFRAD's performance with LAPF is notably better due to LAPF's capacity to increase the receptive field size. Conversely, for deep feature extractors (e.g., $l = 3$), the performance of SFRAD with LAPF remains largely unaffected.

5) Influence of Backbone Network Structures: We investigate the impact of utilizing the Vision Transformer (ViT), the Masked AutoEncoder (MAE), and the image encoder (ResNet50) of the Contrastive Language-Image Pretraining model (CLIP-ResNet50) as backbone network structures for feature extraction on the anomaly detection performance of SFRAD, as depicted in Fig. 15. When employing ViT, CLIP-ResNet50, or Wide-ResNet50×2 (Ours) as the feature extractor, we observe that both methods yielded nearly comparable results, with both surpassing the results obtained using MAE. We attribute this divergence in performance to the fact that ResNet and ViT have undergone classification-oriented pretraining, which aligns more effectively with the requirements of our task. In contrast, MAE's emphasis on reconstructing input images through its encoder-decoder architecture may explain its comparatively lower performance. Nevertheless, it's worth noting that even when using MAE as an encoder, the results still outperform most of the comparison methods. In summary, achieving satisfactory results is possible using transformers and autoencoders.

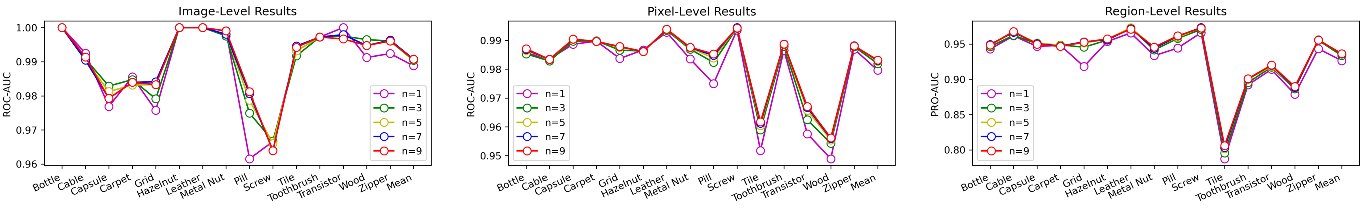
Image-Level Results Pixel-Level Results Region-Level Results

Fig. 10. The influence of sparsity n in OMP on the MVTec AD dataset.

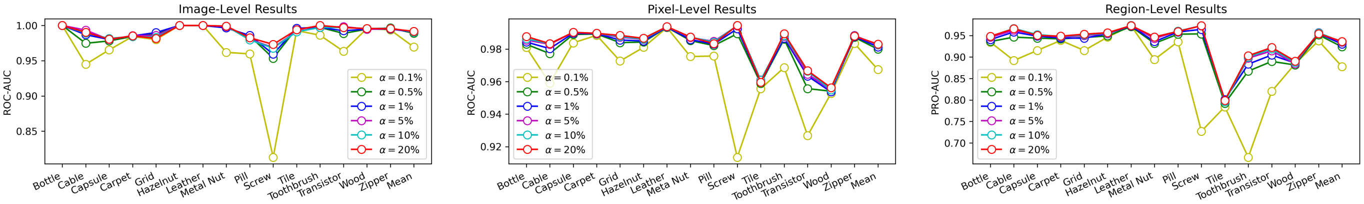
Image-Level Results Pixel-Level Results Region-Level Results

Fig. 11. The influence of α in BFS on the MVTec AD dataset.

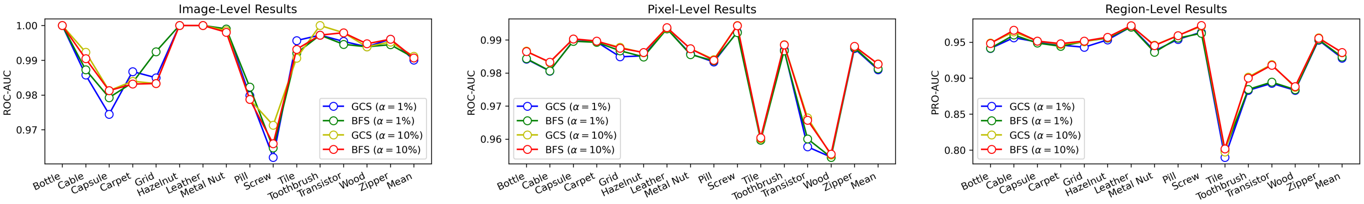
Image-Level Results Pixel-Level Results Region-Level Results

Fig. 12. The influence of BFS on the MVTec AD dataset.

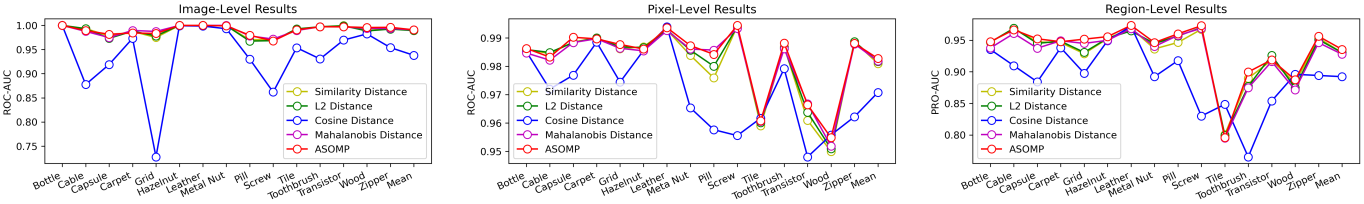
Image-Level Results Pixel-Level Results Region-Level Results

Fig. 13. The influence of ASOMP on the MVTec AD dataset.

6) Influence of Distribution Shift: We conduct a thorough investigation into the influence of distribution shifts on the performance of our proposed method. In our experiments, we intentionally introduce noise to the test set images, altering their distribution from that of normal images. To systematically evaluate this influence, inspired by [72], we select four distinct types of corruption for testing, i.e., Brightness, Contrast, Defocus Blur, and Gaussian noise. Additionally, each corruption type is tested at three different severity levels. Fig. 16 and 17 present the average results across different categories and the specific results for each category in the MVTec AD dataset, respectively. In Fig. 16, we observe a gradual decline in SFRAD's performance as the severity of corruption to the test images increases. Notably, SFRAD demonstrates resilience to Defocus Blur, with its impact being relatively minor. This indicates SFRAD exhibits a degree of robustness in handling varying levels of image corruption, particularly in cases of Defocus Blur. In Fig. 17, variations in the sensitivity of different categories to corruption become apparent. For instance, SFRAD performs well regardless of the type of noise added or the extent of image degradation for the carpet and leather categories. However, the screw category shows a higher susceptibility to corruption, possibly attributed to the smaller and less conspicuous abnormal areas. This makes normal images more prone to being misclassified as anomalies after degradation. In summary, our algorithm demonstrates commendable anomaly detection performance on the out-of-distribution (OOD) test dataset.

V. CONCLUSION

In this paper, a novel framework named SFRAD is proposed for unsupervised visual anomaly detection. The proposed SFRAD methodology uses the pretrained Wide-ResNet50×2 network to extract the multiscale features of nominal images.

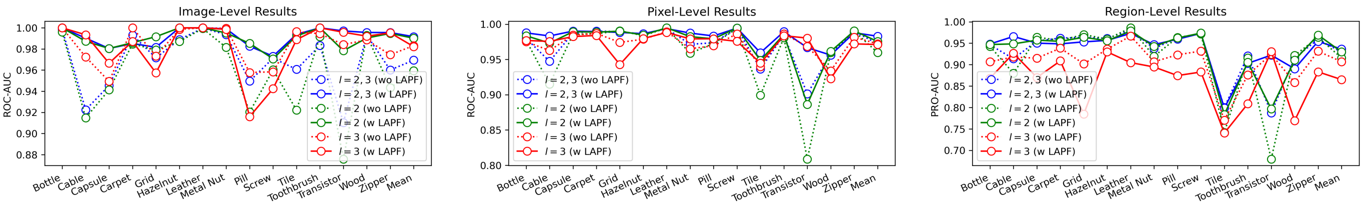

Fig. 14. The influence of LAPF on the MVTec AD dataset.

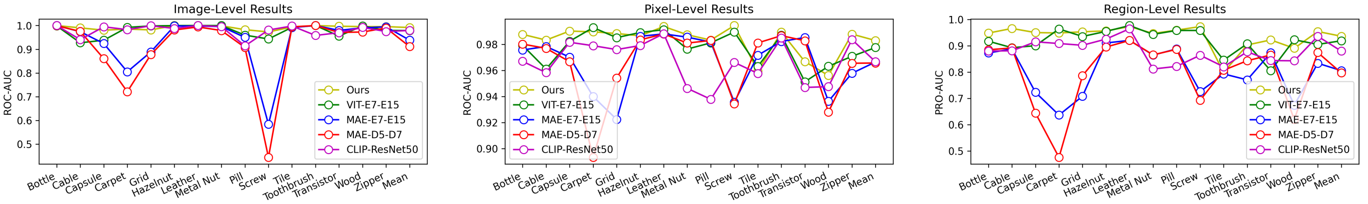

Fig. 15. The influence of backbone network structures on the MVTec AD dataset. "ViT-E7-E15" means two feature layers from ViT, i.e., $l = blocks.7$ and $blocks.15$, are used to extract features. "MAE-D5-D7" means two feature layers from the decoder of MAE, i.e., $l = blocks.5$ and $blocks.7$, are used to extract features.

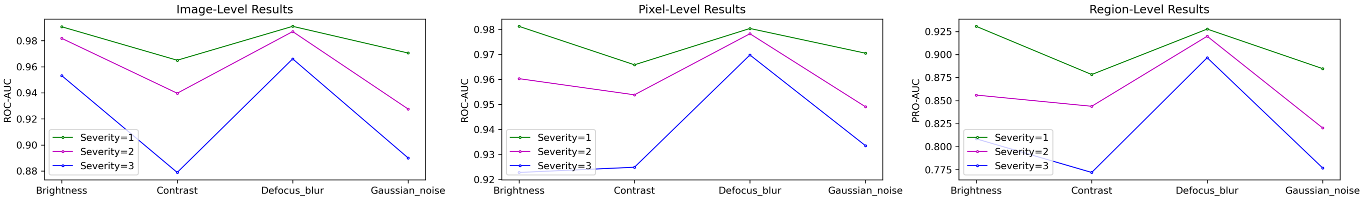

Fig. 16. The influence of four corruptions on the MVTec AD dataset (the average results).

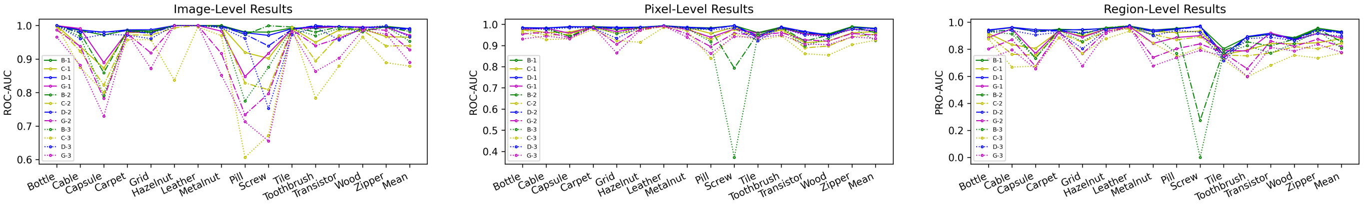

Fig. 17. The influence of four corruptions on the MVTec AD dataset (the results for each category).

The memory bank can be obtained by combining minimax facility location coreset selection and ASOMP to sample basis features. During inference, the anomaly score of the test feature can be calculated by using the memory bank and ASOMP. Finally, anomaly pixels can be located according to anomaly scores. Experimental results on the MVTec AD, KolektorSDD, KolektorSDD2, MVTec LOCO AD, VISA, MNIST, and CIFAR-10 datasets show the effectiveness of our proposed SFRAD methodology. The limitations of our proposed SFRAD methodology, in its current development, are along the lines of two aspects. Firstly, the selection of the σ value in the Gaussian kernel function for different categories can be further explored to improve low-shot anomaly detection. Secondly, the aspect on how best to realize robust anomaly detection in complex scenarios with the proposed SFRAD method remains a challenging work. These would be pertinent issues where we would devote attention in our future work.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China 2021YFE0110500, in part by the National Natural Science Foundation of China under Grant 62202499, 62062021, in part by Beijing Natural Science Foundation under Grant L231012, in part by the Hunan Provincial Natural Science Foundation of China under Grant 2022JJ40632.

REFERENCES

[1] D. Carrera, F. Manganini, G. Boracchi, and E. Lanzarone, "Defect detection in SEM images of nanofibrous materials," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 551–561, 2017.

[2] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, “Mvtec AD - A comprehensive real-world dataset for unsupervised anomaly detection,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*. Computer Vision Foundation / IEEE, 2019, pp. 9592–9600.

[3] Y. Yan, D. Wang, G. Zhou, and Q. Chen, “Unsupervised anomaly segmentation via multilevel image reconstruction and adaptive attention-level transition,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[4] T. Bao, J. Chen, W. Li, X. Wang, J. Fei, L. Wu, R. Zhao, and Y. Zheng, “Miad: A maintenance inspection dataset for unsupervised anomaly detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 993–1002.

[5] P. Bergmann, X. Jin, D. Sattlegger, and C. Steger, “The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization,” in *VISIGRAPP 2022*. SCITEPRESS, 2022, pp. 202–213.

[6] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” in *Information Processing in Medical Imaging - 25th International Conference, IPMI 2017*, vol. 10265. Springer, 2017, pp. 146–157.

[7] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, “f-anogan: Fast unsupervised anomaly detection with generative adversarial networks,” *Medical image analysis*, vol. 54, pp. 30–44, 2019.

[8] K. Zhou, J. Li, Y. Xiao, J. Yang, J. Cheng, W. Liu, W. Luo, J. Liu, and S. Gao, “Memorizing structure-texture correspondence for image anomaly detection,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 6, pp. 2335–2349, 2022.

[9] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, “Latent space autoregression for novelty detection,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*. Computer Vision Foundation / IEEE, 2019, pp. 481–490.

[10] A. Li, Z. Miao, Y. Cen, X. Zhang, L. Zhang, and S. Chen, “Abnormal event detection in surveillance videos based on low-rank and compact coefficient dictionary learning,” *Pattern Recognition*, vol. 108, p. 107355, 2020.

[11] K. Xu, T. Sun, and X. Jiang, “Video anomaly detection and localization based on an adaptive intra-frame classification network,” *IEEE Trans. Multim.*, vol. 22, no. 2, pp. 394–406, 2020.

[12] Z. Fang, J. Liang, J. T. Zhou, Y. Xiao, and F. Yang, “Anomaly detection with bidirectional consistency in videos,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 3, pp. 1079–1092, 2022.

[13] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, “Improving unsupervised defect segmentation by applying structural similarity to autoencoders,” in *VISIGRAPP 2019*. SciTePress, 2019, pp. 372–380.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5998–6008.

[15] S. Kan, Z. He, Y. Cen, Y. Li, V. Mladenovic, and Z. He, “Contrastive bayesian analysis for deep metric learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2022.

[16] X. Tao, X. Gong, X. Zhang, S. Yan, and C. Adak, “Deep learning for unsupervised anomaly localization in industrial images: A survey,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–21, 2022.

[17] S. Mei, H. Yang, and Z. Yin, “An unsupervised-learning-based approach for automated defect inspection on textured surfaces,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 6, pp. 1266–1277, 2018.

[18] H. Yang, Y. Chen, K. Song, and Z. Yin, “Multiscale feature-clustering-based fully convolutional autoencoder for fast accurate visual inspection of texture surface defects,” *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 3, pp. 1450–1467, 2019.

[19] K. Song, H. Yang, and Z. Yin, “Anomaly composition and decomposition network for accurate visual inspection of texture defects,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, 2022.

[20] D. Li, Q. Tao, J. Liu, and H. Wang, “Center-aware adversarial autoencoder for anomaly detection,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 6, pp. 2480–2493, 2022.

[21] J. Liu, C. Wang, H. Su, B. Du, and D. Tao, “Multistage gan for fabric defect detection,” *IEEE Transactions on Image Processing*, vol. 29, pp. 3388–3400, 2020.

[22] C. Lv, F. Shen, Z. Zhang, D. Xu, and Y. He, “A novel pixel-wise defect inspection method based on stable background reconstruction,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

[23] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” in *Information Processing in Medical Imaging - 25th International Conference, IPMI 2017*, vol. 10265. Springer, 2017, pp. 146–157.

[24] Z. Yang, T. Zhang, I. S. Bozchalooi, and E. Darve, “Memory-augmented generative adversarial networks for anomaly detection,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 6, pp. 2324–2334, 2022.

[25] V. Zavrtanik, M. Kristan, and D. Skocaj, “Reconstruction by inpainting for visual anomaly detection,” *Pattern Recognition*, vol. 112, p. 107706, 2021.

[26] J. Pirnay and K. Chai, “Inpainting transformer for anomaly detection,” in *Image Analysis and Processing - ICIAP 2022*, vol. 13232. Springer, 2022, pp. 394–406.

[27] C. Huang, Q. Xu, Y. Wang, Y. Wang, and Y. Zhang, “Self-supervised masking for unsupervised anomaly detection and localization,” *IEEE Transactions on Multimedia*, pp. 1–1, 2022.

[28] D. S. Tan, Y. Chen, T. P. Chen, and W. Chen, “Trustmae: A noise-resilient defect classification framework using memory-augmented autoencoders with trust regions,” in *IEEE Winter Conference on Applications of Computer Vision, WACV 2021*. IEEE, 2021, pp. 276–285.

[29] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. van den Hengel, “Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection,” in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019*. IEEE, 2019, pp. 1705–1714.

[30] Y. Liao, A. Bartler, and B. Yang, “Anomaly detection based on selection and weighting in latent space,” in *17th IEEE International Conference on Automation Science and Engineering, CASE 2021*. IEEE, 2021, pp. 409–415.

[31] X. Tao, D.-P. Zhang, W. Ma, Z. Hou, Z. Lu, and C. Adak, “Unsupervised anomaly detection for surface defects with dual-siamese network,” *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2022.

[32] J. W. Barker and T. P. Breckon, “PANDA: perceptually aware neural detection of anomalies,” in *International Joint Conference on Neural Networks, IJCNN 2021*. IEEE, 2021, pp. 1–8.

[33] G. Wang, S. Han, E. Ding, and D. Huang, “Student-teacher feature pyramid matching for unsupervised anomaly detection,” *CoRR*, vol. abs/2103.04257, 2021.

[34] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, “Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*. IEEE, 2020, pp. 4182–4191.

[35] N. Cohen and Y. Hoshen, “Sub-image anomaly detection with deep pyramid correspondences,” *CoRR*, vol. abs/2005.02357, 2020.

[36] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. V. Gehler, “Towards total recall in industrial anomaly detection,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*. IEEE, 2022, pp. 14 298–14 308.

[37] J. Yang, Y. Shi, and Z. Qi, “Learning deep feature correspondence for unsupervised anomaly detection and segmentation,” *Pattern Recognit.*, vol. 132, p. 108874, 2022.

[38] H. Deng and X. Li, “Anomaly detection via reverse distillation from one-class embedding,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2022, pp. 9727–9736.

[39] M. Rudolph, B. Wandt, and B. Rosenhahn, “Same same but differnet: Semi-supervised defect detection with normalizing flows,” in *IEEE Winter Conference on Applications of Computer Vision, WACV 2021*. IEEE, 2021, pp. 1906–1915.

[40] D. A. Gudovskiy, S. Ishizaka, and K. Kozuka, “CFLOW-AD: real-time unsupervised anomaly detection with localization via conditional normalizing flows,” in *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022*. IEEE, 2022, pp. 1819–1828.

[41] M. Rudolph, T. Wehrbein, B. Rosenhahn, and B. Wandt, “Fully convolutional cross-scale-flows for image-based defect detection,” in *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022*. IEEE, 2022, pp. 1829–1838.

[42] J. Yu, Y. Zheng, X. Wang, W. Li, Y. Wu, R. Zhao, and L. Wu, “Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows,” *CoRR*, vol. abs/2111.07677, 2021.

[43] H. Tang, C. Yuan, Z. Li, and J. Tang, “Learning attention-guided pyramidal features for few-shot fine-grained recognition,” *Pattern Recognit.*, vol. 130, p. 108792, 2022.

[44] Z. Li, H. Tang, Z. Peng, G.-J. Qi, and J. Tang, “Knowledge-guided semantic transfer network for few-shot image recognition,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023.

[45] P. Xing and Z. Li, "Visual anomaly detection via partition memory bank module and error estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[46] S. Sheynin, S. Benaim, and L. Wolf, "A hierarchical transformation-discriminating generative model for few shot anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8495–8504.

[47] C. Huang, H. Guan, A. Jiang, Y. Zhang, M. Spratling, and Y.-F. Wang, "Registration based few-shot anomaly detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 303–319.

[48] P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, K. Müller, and M. Kloft, "Exposing outlier exposure: What can be learned from few, one, and zero outlier images," *Trans. Mach. Learn. Res.*, vol. 2022, 2022.

[49] Y. Shi, J. Yang, and Z. Qi, "Unsupervised anomaly segmentation via deep feature reconstruction," *Neurocomputing*, vol. 424, pp. 9–22, 2021.

[50] Y. Zheng, X. Wang, R. Deng, T. Bao, R. Zhao, and L. Wu, "Focus your distribution: Coarse-to-fine non-contrastive learning for anomaly detection and localization," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022, pp. 1–6.

[51] Q. Wan, L. Gao, X. Li, and L. Wen, "Unsupervised image anomaly detection and segmentation based on pre-trained feature mapping," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2022.

[52] Q. Wan, L. Gao, and L. Li, Xinyuand Wen, "Industrial image anomaly localization based on gaussian clustering of pretrained feature," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 6, pp. 6182–6192, 2022.

[53] S. Venkataramanan, K. Peng, R. V. Singh, and A. Mahalanobis, "Attention guided anomaly localization in images," in *Computer Vision - ECCV 2020*, vol. 12362. Springer, 2020, pp. 485–503.

[54] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "Padim: A patch distribution modeling framework for anomaly detection and localization," in *Pattern Recognition. ICPR International Workshops and Challenges, 2021*, vol. 12664. Springer, 2020, pp. 475–489.

[55] Q. Zhou, S. He, H. Liu, T. Chen, and J. Chen, "Pull & push: Leveraging differential knowledge distillation for efficient unsupervised anomaly detection and localization," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2022.

[56] D. Tabernik, S. Sela, J. Skvarc, and D. Skocaj, "Segmentation-based deep-learning approach for surface-defect detection," *J. Intell. Manuf.*, vol. 31, no. 3, pp. 759–776, 2020.

[57] J. Bozic, D. Tabernik, and D. Skocaj, "Mixed supervision for surface-defect detection: From weakly to fully supervised learning," *Computers in Industry*, vol. 129, p. 103459, 2021.

[58] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization," *Int. J. Comput. Vis.*, vol. 130, no. 4, pp. 947–969, 2022.

[59] Y. Zou, J. Jeong, L. Pemula, D. Zhang, and O. Dabeer, "Spot-the-difference self-supervised pre-training for anomaly detection and segmentation," in *Computer Vision - ECCV 2022 - 17th European Conference,*, vol. 13690. Springer, 2022, pp. 392–408.

[60] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[61] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*. IEEE, 2020, pp. 4182–4191.

[62] J. Yi and S. Yoon, "Patch SVDD: patch-level SVDD for anomaly detection and segmentation," in *Computer Vision - ACCV 2020*, vol. 12627. Springer, 2020, pp. 375–390.

[63] C. Li, K. Sohn, J. Yoon, and T. Pfister, "Cutpaste: Self-supervised learning for anomaly detection and localization," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*. Computer Vision Foundation / IEEE, 2021, pp. 9664–9674.

[64] J. Kim, D. Kim, S. Yi, and T. Lee, "Semi-orthogonal embedding for efficient unsupervised anomaly segmentation," *CoRR*, vol. abs/2105.14737, 2021.

[65] P. Xing, H. Tang, J. Tang, and Z. Li, "Adps: Asymmetric distillation postsegmentation for image anomaly detection," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2024.

[66] Y. Cai, D. Liang, D. Luo, X. He, X. Yang, and X. Bai, "A discrepancy aware framework for robust anomaly detection," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 3, pp. 3986–3995, 2024.

[67] H. Yao, W. Yu, W. Luo, Z. Qiang, D. Luo, and X. Zhang, "Learning global-local correspondence with semantic bottleneck for logical anomaly detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 5, pp. 3589–3605, 2024.

[68] L. Ruff, N. Görnitz, L. Deecke, S. A. Siddiqui, R. A. Vandermeulen, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, vol. 80. PMLR, 2018, pp. 4390–4399.

[69] W. Liu, R. Li, M. Zheng, S. Karanam, Z. Wu, B. Bhanu, R. J. Radke, and O. I. Camps, "Towards visually explaining variational autoencoders," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*. IEEE, 2020, pp. 8639–8648.

[70] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014,*.

[71] H. Yu, Q. Li, Y. Tan, J. Gan, J. Wang, Y.-a. Geng, and L. Jia, "A coarse-to-fine model for rail surface defect detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 3, pp. 656–666, 2019.

[72] T. Cao, J. Zhu, and G. Pang, "Anomaly detection under distribution shift," in *IEEE/CVF International Conference on Computer Vision*. IEEE, 2023, pp. 6488–6500.