

Learning Unified Reference Representation for Unsupervised Multi-class Anomaly Detection

Liren He^{1*}, Zhengkai Jiang^{2*}, Jinlong Peng^{2*}, Wenbing Zhu^{1,3*}, Liang Liu², Qiangang Du¹, Xiaobin Hu², Mingmin Chi^{1†}, Yabiao Wang^{4,2†}, and Chengjie Wang^{2†}

¹ Fudan University, Shanghai, China

`mmchi@fudan.edu.cn`

² Tencent Youtu Lab, Shanghai, China

`{caseywang, jasoncjwang}@tencent.com`

³ Rongcheer, Suzhou, China

⁴ Zhejiang University, Hangzhou, China

Abstract. In the field of multi-class anomaly detection, reconstruction-based methods derived from single-class anomaly detection face the well-known challenge of “learning shortcuts”, wherein the model fails to learn the patterns of normal samples as it should, opting instead for shortcuts such as identity mapping or artificial noise elimination. Consequently, the model becomes unable to reconstruct genuine anomalies as normal instances, resulting in a failure of anomaly detection. To counter this issue, we present a novel unified feature reconstruction-based anomaly detection framework termed RLR (**R**econstruct features from a **L**earnable **R**eference representation). Unlike previous methods, RLR utilizes learnable reference representations to compel the model to learn normal feature patterns explicitly, thereby prevents the model from succumbing to the “learning shortcuts” issue. Additionally, RLR incorporates locality constraints into the learnable reference to facilitate more effective normal pattern capture and utilizes a masked learnable key attention mechanism to enhance robustness. Evaluation of RLR on the 15-category MVTEC-AD dataset and the 12-category VisA dataset shows superior performance compared to state-of-the-art methods under the unified setting. Code is available at RLR.

Keywords: Multi-class Anomaly Detection · Feature Reconstruction

1 Introduction

Unsupervised anomaly detection and localization strive to learn the patterns of normal samples from the training set then treat outliers as anomalies during inference, which is widely applied in industrial manufacturing [1, 25, 33, 37],

* Equal contribution.

† Corresponding authors. This work was supported by Natural Science Foundation of China under contract 62171139.

medical image analysis [6], among other fields. However, in practical industrial anomaly detection scenarios, multi-class anomaly detection is not only more prevalent but also more valuable, as it only requires training one model for N classes, whereas single-class methods require training N models for N classes. Thus, this paper focuses on the promising and challenging multi-class anomaly detection.

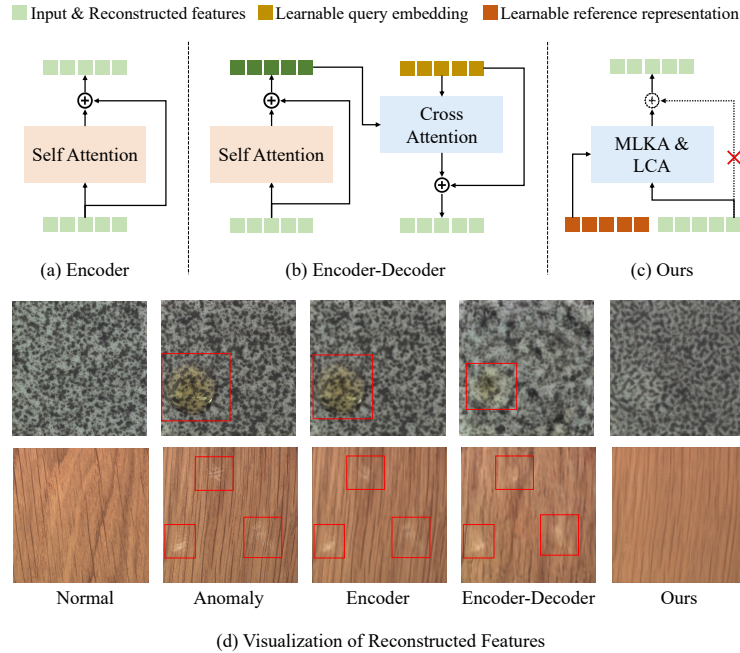


Fig. 1: Motivation and effectiveness of our method. Existing frameworks are shown in (a) and (b), fall into learning shortcuts issue. Our framework is depicted in (c), which utilizes learnable reference representation for feature reconstruction to address this issue. (d) shows the visualizations of the reconstructed features, which includes the Normal Sample, the input Anomaly Sample, as well as the features recovered from the Encoder, Encoder-Decoder (specifically UniAD [29]) and our proposed methods.

Most existing mainstream anomaly detection methods are single-class methods, they can be categorized into embedding-based, knowledge distillation-based, and reconstruction-based approaches. Directly applying single-class method to multi-class anomaly detection leads to significant performance degradation, since the normal patterns for multiple classes is much more complicated than single class. Embedding-based methods [3, 8, 13, 17, 18] employ statistical methods, such as Gaussian distribution [3], normalizing flow [8], and memory bank [18] to obtain the normal patterns from pre-trained model. However, these methods require high computational or storage resources in multi-class anomaly detection.

Knowledge distillation-based methods [2, 4, 21, 23] distill the normal features extracted by a pre-trained teacher model into a student model and assume the student model only learns normal patterns. However, this assumption may fail in multi-class anomaly detection. Reconstruction-based methods [26, 27, 30–32] train models solely using normal samples, with expectation that the model will learn to recognize normal patterns. Therefore they can reconstruct anomalies as normal instances, enabling the detection of anomalies during inference. However, the fundamental objective of unsupervised reconstruction training is essentially identity mapping or artificial noise eliminating. This carries the risk that the model may not effectively learn normal patterns, but instead encounter the problem of *learning shortcuts* to achieve training objectives, such as identical mapping, resulting in anomalies still being recovered as anomalies.

Since existing single-class anomaly detection methods cannot be directly applied to multi-class anomaly detection, recent efforts have proposed unified frameworks for multi-class anomaly detection, such as UniAD [29] and OmniAL [36]. These methods are based on reconstruction, and the tendency of reconstruction models to learn shortcuts during the training phase becomes more pronounced in multi-class anomaly detection. Therefore, they aim to weaken this tendency by increasing the difficulty of learning shortcuts. However, they cannot completely eliminate this possibility because fundamentally, their output is primarily determined by the input, so the model still can learn shortcuts during the training phase to perfectly reconstruct the input.

To address this issue, we propose a simple yet effective feature reconstruction method based on learnable reference representation for multi-class anomaly detection. Existing Transformer-based feature reconstruction frameworks, both vanilla Encoder (Figure 1(a), like AnoViT [12]) and Encoder-Decoder (Figure 1(b), like UniAD [29]), fall into “learning shortcuts” since their output is a direct mapping of the input. Therefore they can learn shortcuts such as identity mapping or simple noise elimination, allowing it to perfectly reconstruct features during training without learning the normal patterns. In contrast, our method reconstructs features from learnable reference to ensure our model’s output is primarily influenced by the learnable reference. As shown in Figure 1(c), the learnable reference representation serve as the Key and Value to recover the features, meanwhile we remove residual connections. Figure 1(d) demonstrates the visualization of the reconstructed features, our method presents a higher accuracy in reconstructing anomaly features into normal features.

Specifically, we propose the framework called RLR, which Reconstruct features from Learnable Reference representation. Our RLR is based on Transformer Encoder without residual connections. We utilize two novel components, namely Local Cross Attention (LCA) and Masked Learnable Key Attention (MLKA), to reconstruct features. In the LCA module, we compute the Cross Attention between the input features and the learnable reference. Additionally, considering that the feature maps extracted by the pre-trained CNN model exhibit locality [18], we introduce a locality constraint for the learnable reference to obtain more effective and accurate reference normal patterns. In the

MLKA module, we utilize the learnable reference as the Key to compute neighbor masked Attention without residual connection, allowing us to capture more detailed information to assist in feature reconstruction. This helps prevent the reconstructed features from being overly smoothed when relying solely on LCA.

We conducted experiments on two widely used industrial anomaly detection datasets, namely MVTec-AD [1] and VisA [37]. Our approach outperformed the previous state-of-the-art unified framework and separate anomaly detection models adapted for multi-class task on both datasets. Specifically, we obtain 98.6% Image-AUROC and 98.5% Pixel-AUROC on the MVTec-AD, which is a significant improvement compared to previous methods.

Overall, our contributions are summarized as follows:

- We propose a unified anomaly detection framework that reconstructs features from learnable reference representation, thereby forcing the model to learn normal patterns instead of learning shortcuts.
- We introduce Local Cross Attention to enable the model to learn more accurate and effective reference representation, and Masked Learnable Key Attention to assist the model in reconstructing more detailed features.
- Our unified anomaly detection framework achieves SOTA performance on popular industrial anomaly detection datasets, MVTec-AD and VisA.

2 Related Work

Unsupervised Anomaly Detection. We divide the mainstream unsupervised anomaly detection approaches into embedding based, knowledge distillation based and reconstruction based methods. The first involves conducting statistical analysis on embedding of normal samples that extracted by pre-trained models. For instance, MDND [17] and PaDiM [3] rely on multivariate Gaussian distribution, Patchcore [18] utilizes memory bank, and [8, 13, 19, 20] count on normalizing flow. However, these methods require substantial resources and intricate design tricks. The second distillates the normal features extracted by pre-trained teacher model into student model [2]. However, despite these approaches require tricks to ensure the student model only learns normal patterns, such as multiresolution [21] and reverse distillation [4, 23], it still cannot guarantee that. The third applies models like AutoEncoder [31], GAN [7, 26], Diffusion [11, 32] and Transformer [30] to reconstruct anomaly pixels or features into normal instances. However, existing methods fail to safeguard that the model learns normal patterns instead of shortcuts, while our RLR utilizes learnable reference to guarantee that.

Transformer based Anomaly Detection. Recently, Transformer [24, 34] has gained widespread attention for its ability to model long-range dependencies, leading to its utilization in reconstruction-based anomaly detection methods. InTra [16] employs a Transformer Encoder to reconstruct masked images. AnoViT [12] and VT-ADL [15] use Transformer Encoder to reconstruct features, which are further reconstructed into images using CNNs. ADTR [30] highlights Transformer Encoder-based methods suffer from “identity mapping” problem

and proposes an Encoder-Decoder architecture for feature reconstruction. However, the outputs of these architectures are still primarily derived from the input mapping, leaving room for potential shortcuts in the learning process. This tendency becomes more pronounced, particularly in the case of multi-class anomaly detection, where the challenge of learning multi-class normal patterns increases, further reinforcing the inclination towards learning shortcuts, while our RLR addresses that.

Unified Anomaly Detection. UniAD [29] first introduces a unified framework for multi-class anomaly detection and highlights the increased likelihood of encountering the “learning shortcut” problem in the unified setting, since learning multi-class normal patterns is more challenging than learning shortcuts. UniAD proposes a solution based on the Transformer Encoder-Decoder structure with Neighbor Mask Attention and Feature Noise to increase the difficulty of the model learning shortcuts. However, it does not fundamentally eliminate the possibility to learn shortcuts as described above. OmniAL [36] is an image reconstruction based method. It proposes complex and realistic anomaly synthesis, which successfully work in the multi-class anomaly detection. The primary objective of OmniAL is to enhance the learning of multi-class normal patterns by introducing challenging anomalies that make it difficult for the model to learn shortcuts. However this approach heavily relies on the quality of generated anomalous samples and requires significant prior knowledge and resources. Our RLR reconstructs features from learnable reference, forcing the model learn normal patterns rather than shortcuts.

3 Method

In this paper, we propose a unified anomaly detection framework based on feature reconstruction. The key insight is to reconstruct features from learnable reference representation, thereby enforcing this reference to learn normal patterns. This approach avoids the problem of previous unsupervised reconstruction methods, which employ the self-reconstruction task as the training objective and lead to “learning shortcuts” issue. To enhance the model’s ability to learn normal reference patterns and reconstruct normal features, we further introduce two simple yet effective attention mechanisms. Figure 2 shows the overview of the proposed RLR with four steps: multi-scale feature extraction, feature reconstruction with our parallel Masked Learnable Key Attention and Local Cross Attention, loss calculation between the recovered features and the original features, and score map forecast at the inference phase.

3.1 Feature Extraction

Multi-Scale Feature Extraction. Following the previous patch feature based approaches [3, 18, 29], we apply the fixed CNNs that pre-trained on ImageNet [5] like ResNet [10] or EfficientNet [22] to extract the multi-scale feature maps of the input images. The feature maps are defined as $\mathcal{F}_{i,j} \in \mathbb{R}^{C_j \times H_j \times W_j}$,

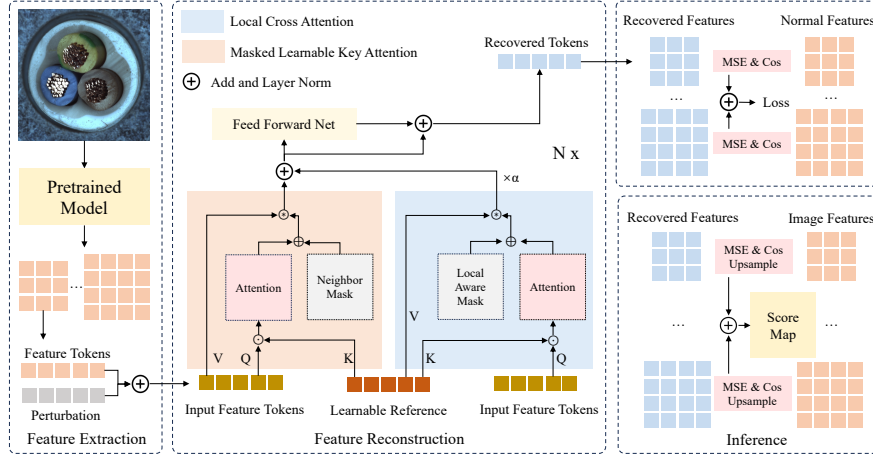


Fig. 2: Framework of our approach. RLR consists of Multi-Scale Feature Extraction through pre-trained model, Feature Reconstruction with combination of Masked Learnable Key Attention and Local Cross Attention, Loss and Score Map calculation between recovered features and original features.

where i is the index of the input image x_i , j is the index of the multi-scale feature maps, and C_j, H_j, W_j are the channel dimension, height and width of the j th feature map. Same as [18], we apply the local neighbourhood aggregation to feature maps to improve robustness. Formally, for a patch at location (h, w) , we denote its neighborhood as

$$\mathcal{N}_p^{(h,w)} = \{(a, b) | a \in [h - \lfloor p/2 \rfloor, \dots, h + \lfloor p/2 \rfloor], b \in [w - \lfloor p/2 \rfloor, \dots, w + \lfloor p/2 \rfloor]\}, \quad (1)$$

where p is the neighbor window size. We use Adaptive Average Pooling as the aggregation function f_{agg} to obtain locally aware feature for patch located at (h, w) with its neighborhood $\mathcal{N}_p^{(h,w)}$, which is formulated as

$$\mathcal{F}_{org,h,w}^{i,j} = f_{agg} \left(\{\mathcal{F}_{i,j}^{(a,b)} | (a, b) \in \mathcal{N}_p^{(h,w)}\} \right), \quad (2)$$

where $\mathcal{F}_{org}^{i,j} \in \mathbb{R}^{C_j \times H_j \times W_j}$ are the feature maps that are expected to be reconstructed.

Feature Perturbation. Using the feature $\mathcal{F}_{org}^{i,j}$ directly as input can lead to the problem of identical mapping due to the presence of residual shortcuts in the vanilla Transformer. Although we have addressed this issue with our shortcut-free attention, feature perturbation is still beneficial for enhancing generalization and robustness. Inspired by [14, 29], we simply sample the noise tokens \mathcal{P} from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, where σ is the adjustable variance. By adding random noise to $\mathcal{F}_{org}^{i,j}$, we define the input feature tokens $\mathcal{F}_{in}^{i,j} \in \mathbb{R}^{H_j W_j \times C_j}$ as

$$\mathcal{F}_{in}^{i,j} = Reshape(\mathcal{F}_{org}^{i,j}) + \mathcal{P}. \quad (3)$$

3.2 Feature Reconstruction

To avoid ‘‘learning shortcut’’ issue described above, we replace self-attention in vanilla Transformer with our proposed parallel Masked Learnable Key Attention (MLKA) and Local Cross Attention (LCA). We combine the output of MLKA and LCA, where output of LCA multiplied by a hyperparameter α greater than 1, to ensure model focuses more on LCA. Note that there is not residual shortcut between the input and the output of the attention module. Hence the new block is stacked by our proposed attention and Feed Forward Network (FFN) with norm & add similar to vanilla Transformer. For the input feature tokens \mathcal{F}_{in}^j at each feature level j , after K consecutive blocks, we obtain the recovered tokens $\mathcal{F}_{out}^j \in \mathbb{R}^{N_j \times C_j}$, where N_j is the number of patches of j_{th} feature map, i.e., $N_j = H_j \times W_j$. We formally define the calculation process of k_{th} block as

$$\mathcal{Z}_k^j = LN \left(MLKA(\mathcal{Y}_{k-1}^j, \mathcal{R}_h^j) + \alpha LCA(\mathcal{Y}_{k-1}^j, \mathcal{R}_h^j) \right) \quad (4)$$

and

$$\mathcal{Y}_k^j = LN \left(FFN(\mathcal{Z}_k^j) + \mathcal{Z}_k^j \right), \quad (5)$$

where $\mathcal{Z}_k^j \in \mathbb{R}^{N_j \times C_h}$ is the output of the attention module, C_h means the hidden layer dimension, LN represents LayerNorm, $\mathcal{R}_h^j \in \mathbb{R}^{N_j \times C_h}$ indicates the learnable reference tokens after projected to hidden feature space, and $\mathcal{Y}_k^j \in \mathbb{R}^{N_j \times C_h}$ is the output of the k_{th} block and the input of the $(k+1)_{th}$ block. We use a single \mathcal{R}_h^j for all K blocks, so it is independent of k . Furthermore, we acquire \mathcal{Y}_0^j from \mathcal{F}_{in}^j and \mathcal{F}_{out}^j from \mathcal{Y}_K^j , which is defined as

$$\mathcal{Y}_0^j = FFN(\mathcal{F}_{in}^j), \mathcal{F}_{out}^j = FFN(\mathcal{Y}_K^j). \quad (6)$$

Then we reshape the \mathcal{F}_{out}^j to the recovered feature maps $\mathcal{F}_{rec}^j \in \mathbb{R}^{C_j \times H_j \times W_j}$ for training and inference.

Masked Learnable Key Attention. The residual connection in Transformer allows the output directly contains input, making it easier to learn shortcuts, thus a natural idea to figure it out is to remove this connection. Unfortunately, the skip connection is quite important for training deep self-attention network, as it helps prevent deep Transformer from converging to rank collapse [9]. To overcome this challenge, we propose a modified self-attention called Masked Learnable Key Attention (MLKA).

Specifically, we randomly initialize a learnable reference feature representation for each feature level j and define it as $\mathcal{R}^j \in \mathbb{R}^{N_j \times C_j}$, which has the same size as \mathcal{F}_{in}^j because their patch positions need to correspond. Then we use a fully connected layer to project \mathcal{R}^j to the hidden feature space into $\mathcal{R}_h^j \in \mathbb{R}^{N_j \times C_h}$, and a single \mathcal{R}_h^j is used for all K layers. Through training on the normal samples, the learnable reference eventually represents the normal feature pattern. We enhance the vanilla self-attention structure by applying the learnable normal reference feature tokens as the Key vectors. This allows us to eliminate the

residual connection and prevent it from falling into rank collapse at the same time. Additionally, inspired by [29] we add a neighbor masked attention map to the original attention map between Query and Key vectors in order to make a token invisible to itself and its neighbors when calculating attention. The process of MLKA in k_{th} block is formulated as

$$\mathcal{Q}_k, \mathcal{K}_k, \mathcal{V}_k = W_k^Q \mathcal{Y}_{k-1}^j, W_k^K \mathcal{R}_h^j, W_k^V \mathcal{Y}_{k-1}^j \quad (7)$$

$$\mathcal{A}_k = \text{Softmax} \left(\mathcal{Q}_k \mathcal{K}_k^T / \sqrt{d_k} + \mathcal{M}_{nei} \right) \quad (8)$$

$$\mathcal{O}_k = \mathcal{A}_k \mathcal{V}_k \quad (9)$$

where parameters W_k^Q, W_k^K and W_k^V are responsible for embedding \mathcal{Y}_{k-1}^j and \mathcal{R}_h^j into Query, Key and Value vectors, and $\mathcal{M}_{nei} \in \mathbb{R}^{N_j \times N_j}$ is the neighbor mask map that contains zero and negative infinity. In particular, the neighbor patches of the i_{th} patch in $\mathcal{M}_{nei}[i] \in \mathbb{R}^{N_j}$ are marked as negative infinity so that they will be zero after Softmax function, meaning the attention weights of neighbor patches to the i_{th} patch in $\mathcal{A}_k[i] \in \mathbb{R}^{N_j}$ are zero, aiming to ignore patch $_i$'s neighbors. The size of the neighbor window for each feature level j is a hyperparameter, and we provide specific settings in the experimental section. Since MLKA does not have residual connection and with neighbor mask, a token will not directly contain itself after passing through MLKA. Furthermore, due to the learnable key only consists of normal features, the abnormal tokens will receive a low similarity scores when calculating attention. As a result, the output of MLKA will not retain abnormal features. These advantages enable MLKA to recover abnormal tokens to their normal state and prevent the occurrence of "learning shortcuts".

Local Cross Attention. Although MLKA partially addresses the issue of learning shortcuts, there may still be possibility of residual abnormal features remaining in the output tokens, as the Value vectors correspond to input features, similar to UniAD [29]. To overcome this limitation, we propose the Local Cross Attention (LCA) module to reconstruct the feature tokens from learnable reference representation tokens. The basic idea is to treat input feature tokens as Query vectors and learnable reference tokens as Key and Value vectors, and then apply a cross attention between them. However, since the feature maps obtained from CNN-based backbone represent local patch features, incorporating locality constraints can enhance the learning effectiveness of reference. Specifically, the LCA introduces a local aware mask to the attention between input feature and learnable reference, so that a token can only see the reference tokens within a local window to reconstruct its normal representation. The process of LCA can be formulated as

$$\mathcal{Q}_k, \mathcal{K}_k, \mathcal{V}_k = W_k^Q \mathcal{Y}_{k-1}^j, W_k^K \mathcal{R}_h^j, W_k^V \mathcal{R}_h^j \quad (10)$$

$$\mathcal{A}_k = \text{Softmax} \left(\mathcal{Q}_k \mathcal{K}_k^T / \sqrt{d_k} + \mathcal{M}_{loc} \right) \quad (11)$$

$$\mathcal{O}_k = \mathcal{A}_k \mathcal{V}_k \quad (12)$$

where $\mathcal{M}_{loc} \in \mathbb{R}^{N_j \times N_j}$ is the local aware mask similar to \mathcal{M}_{nei} but the neighbor patches are marked as zero while others are negative infinity, meaning the attention weights of non-neighbor patches to the i_{th} patch in $\mathcal{A}_k[i] \in \mathbb{R}^{N_j}$ are zero, thereby focusing attention only on the neighbor patches. Since LCA forces the network to reconstruct the feature tokens from learnable reference representation, the reference eventually contains the normal features, thereby the recovered feature will get rid of abnormal features remarkably.

3.3 Training and Loss Function

Following the Feature Reconstruction module, we acquire the recovered multi-scale feature maps $\mathcal{F}_{rec}^j \in \mathbb{R}^{C_j \times H_j \times W_j}$. We utilize Mean Square Error and Cosine Similarity to measure the loss between reconstructed features \mathcal{F}_{rec}^j and the original extracted features \mathcal{F}_{org}^j . The loss function is defined as

$$\mathcal{L}_{cos}^j = 1 - \frac{1}{H_j W_j} \times \frac{\mathcal{F}_{rec}^j \cdot \mathcal{F}_{org}^j}{\|\mathcal{F}_{rec}^j\|_2 \|\mathcal{F}_{org}^j\|_2} \quad (13)$$

$$\mathcal{L} = \sum_{j=1}^L \left(\frac{\|\mathcal{F}_{rec}^j - \mathcal{F}_{org}^j\|_2}{H_j W_j} + \mathcal{L}_{cos}^j \right), \quad (14)$$

where \mathcal{L}_{cos} is the cosine loss and \mathcal{L} is the total loss. After training on the dataset that only contains normal samples, we obtain the network parameters and the reference representation of normal patterns.

3.4 Inference and Scoring Function

During inference, we extract the multi-scale feature maps F_{org}^j from a test image without introducing any noise. We employ the reconstruction network to recover the corresponding normal feature maps F_{rec}^j . We also utilize Mean Square Error and Cosine Similarity to achieve the abnormal score map since the recovered feature map F_{rec}^j supposed to only contains normal characteristics. The difference between the original and reconstructed features indicates that the original abnormal features have been recovered as normal. The magnitude of this difference serves as indicator of the anomaly score, with a higher difference corresponding to a higher score. We calculate the score map by the following formula

$$\mathcal{S}_{cos}^j = 1 - \frac{\mathcal{F}_{rec}^j \cdot \mathcal{F}_{org}^j}{\|\mathcal{F}_{rec}^j\|_2 \|\mathcal{F}_{org}^j\|_2} \quad (15)$$

$$\mathcal{S} = \frac{\sum_{j=1}^L \text{Upsample}(\|\mathcal{F}_{rec}^j - \mathcal{F}_{org}^j\|_2 + \mathcal{S}_{cos}^j)}{L}, \quad (16)$$

where $\mathcal{S} \in \mathbb{R}^{H \times W}$ is the final score map. For each feature level j , we upsample the j_{th} score map to the original image size, and then calculate the average of all score maps as the final score map.

4 Experiment

4.1 Datasets and Metrics

To validate the effectiveness of our proposed RLR, we perform experiments on two major industrial anomaly detection datasets MVTec-AD [1] and VisA [37].

MVTec-AD [1] is a comprehensive industrial anomaly detection dataset that contains 5 texture and 10 object real world industrial products. The training set contains 3,629 high-resolution normal images, while the testing set consists of 467 normal images and 1,258 anomaly images. Each product exhibits multiple types of anomalies with various size, shape, and other morphological attributes.

VisA [37] contains 12 objects and a total of 10,821 images with 9,621 normal samples and 1,200 anomalous samples. It contains 3 subsets consists of complex structure, multiple instance and aligned object. The anomalous images contain various flaws, including surface defects such as scratches, dents, color spots or crack, and structural defects like misplacement or missing parts. We follow the VisA 1-class protocol to do the unsupervised anomaly detection experiments.

Metrics. Follow the previous anomaly detection works [3, 14, 18, 29], we use the common metric, Area Under the Receiver Operating Curve (AUROC), to evaluate the performance of the approaches. It contains Image-AUROC for measuring detection performance and Pixel-AUROC for measuring localization performance.

4.2 Implementation Details

We train a unified anomaly detection model for all categories. The input images are resized to 256×256 and features are extracted by EfficientNet-B6 [22] that pretrained on ImageNet [5]. We reconstruct the feature maps from stage 1 to 3. Respectively the hidden layer dimensions for feature reconstruction are set to 128, 256 and 512, with corresponding neighbor window sizes of 5, 7 and 11. The hyperparameter α is set to 2, and the number of encoder layers in the feature reconstruction module is 4. We train the model using the Adam optimizer for 200 epochs, with an initial learning rate of $1e-4$.

4.3 Main Results

We compared our proposed method RLR with several previous state-of-the-art (SOTA) anomaly detection approaches, including unified frameworks such as UniAD [29] and OmniAL [36], as well as the SOTA separate methods under multi-class anomaly detection setting, such as MDK [21], DRAEM [31], Patch-Core [28], SimpleNet [14] and so on. We conducted these comparisons on the MVTec-AD and VisA datasets, and our method outperforms these methods in both anomaly detection and localization metrics, achieving new SOTA results. The specific results of MVTec-AD are presented in Table 1 and VisA in Table 2.

Anomaly Detection. Our method has achieved SOTA results on both the MVTec-AD and VisA datasets in terms of the anomaly detection metric

Table 1: Quantitative results with SOTA methods on benchmark MVTec-AD. Anomaly detection and localization results are displayed as Image-AUROC% / Pixel-AUROC%. The best results are highlighted in **bold**.

Category	PaDiM [3]	MKD [21]	DRAEM [31]	Patchcore [18]	SimpleNet [14]	UniAD [29]	OmmiAL [36]	Ours	
Object	Bottle	97.9 / 96.1	98.7 / 91.8	97.5 / 87.6	100 / 98.4	86.5 / 88.1	99.7 / 98.1	100 / 99.2	100 / 99.0
	Cable	70.9 / 81.0	78.2 / 89.3	57.8 / 71.3	99.2 / 97.3	71.5 / 79.3	95.2 / 97.3	98.2 / 97.3	99.3 / 99.0
	Capsule	73.4 / 96.9	68.3 / 88.3	65.3 / 50.5	85.6 / 95.2	77.8 / 89.4	86.9 / 98.5	95.2 / 96.9	97.0 / 99.2
	Hazelnut	85.5 / 96.3	97.1 / 91.2	93.7 / 96.9	100 / 98.9	94.3 / 95.9	99.8 / 98.1	95.6 / 98.4	100 / 99.0
	Metal Nut	88.0 / 84.8	64.9 / 64.2	72.8 / 62.2	99.9 / 98.4	87.8 / 87.0	99.2 / 94.8	99.2 / 99.1	99.7 / 98.3
	Pill	68.8 / 87.7	79.7 / 69.7	82.2 / 94.4	93.3 / 95.7	80.2 / 90.7	93.7 / 95.0	97.2 / 98.9	98.9 / 98.3
	Screw	56.9 / 94.1	75.6 / 92.1	92.0 / 95.5	82.9 / 95.9	72.8 / 85.7	87.5 / 98.3	88.0 / 98.0	94.8 / 99.5
	Toothbrush	95.3 / 95.6	75.3 / 88.9	90.6 / 97.7	88.9 / 98.2	87.8 / 96.4	94.2 / 98.4	100 / 99.4	93.1 / 98.9
	Transistor	86.6 / 92.3	73.4 / 71.7	74.8 / 64.5	96.7 / 89.3	79.7 / 83.3	99.8 / 97.9	93.8 / 93.3	99.7 / 98.6
	Zipper	79.7 / 94.8	87.4 / 86.1	98.8 / 98.3	91.9 / 95.5	88.5 / 84.3	95.8 / 96.8	100 / 99.5	98.5 / 98.2
Texture	Carpet	93.8 / 97.6	69.8 / 95.5	98.0 / 98.6	96.1 / 98.7	87.6 / 89.5	99.8 / 98.5	98.7 / 99.4	99.7 / 99.0
	Grid	73.9 / 71.0	83.8 / 82.3	99.3 / 98.7	97.1 / 96.6	79.1 / 69.9	98.2 / 96.5	99.9 / 99.4	99.8 / 98.7
	Leather	99.9 / 84.8	93.6 / 96.7	98.7 / 97.3	100 / 99.4	95.2 / 96.6	100 / 98.8	99.0 / 99.3	100 / 99.4
	Tile	93.3 / 80.5	89.5 / 85.3	99.8 / 98.0	99.9 / 95.7	97.9 / 91.6	99.3 / 91.8	99.6 / 99.0	100 / 96.7
	Wood	98.4 / 89.1	93.4 / 80.5	99.8 / 96.0	98.4 / 93.5	97.5 / 87.0	98.6 / 93.2	93.2 / 97.4	98.9 / 95.5
Mean	84.2 / 89.5	81.9 / 84.9	88.1 / 87.2	95.3 / 96.4	85.6 / 87.6	96.5 / 96.8	97.2 / 98.3	98.6 / 98.5	

Table 2: Quantitative results with SOTA methods on benchmark VisA. Anomaly detection and localization results are displayed as Image-AUROC%/Pixel-AUROC%. The best results are highlighted in **bold**.

Category	DRAEM [31]	JNLD [35]	OmmiAL [36]	UniAD [29]	Ours	
Complex structure	PCB1	83.9 / 94.0	82.9 / 98.0	77.7 / 97.6	94.2 / 99.4	97.0 / 99.7
	PCB2	81.7 / 94.1	79.1 / 95.0	81.0 / 93.9	93.3 / 97.8	97.4 / 99.1
	PCB3	87.7 / 94.1	90.1 / 98.5	88.1 / 94.7	87.2 / 98.2	96.4 / 99.1
	PCB4	87.1 / 72.3	96.2 / 97.5	95.3 / 97.1	99.2 / 97.8	99.7 / 98.4
Multiple instance	Macaroni1	68.6 / 89.8	90.5 / 93.3	92.6 / 98.6	91.6 / 99.2	97.7 / 99.8
	Macaroni2	60.3 / 83.2	71.3 / 92.1	75.2 / 97.9	83.9 / 97.9	86.3 / 99.3
	Capsules	89.6 / 96.6	91.4 / 99.6	90.6 / 99.4	73.1 / 98.1	85.9 / 99.2
	Candles	70.2 / 82.6	85.4 / 94.5	86.8 / 95.8	96.9 / 99.1	98.2 / 99.5
Aligned object	Cashew	67.3 / 68.5	82.5 / 94.1	88.6 / 95.0	93.2 / 98.6	96.1 / 99.4
	Chewing gum	90.0 / 92.7	96.0 / 98.9	96.4 / 99.0	99.0 / 99.1	99.8 / 99.2
	Fryum	86.2 / 83.2	91.9 / 90.0	94.6 / 92.1	89.3 / 97.5	96.5 / 97.4
	Pipe fryum	87.1 / 72.3	87.5 / 92.5	86.1 / 98.2	97.3 / 99.1	99.4 / 99.5
Mean	80.5 / 87.0	87.1 / 95.2	87.8 / 96.6	91.5 / 98.5	95.9 / 99.2	

Image-AUROC (I-A). Furthermore, our approach has shown significant improvements compared to the previous unified framework. In contrast to the feature reconstruction-based method UniAD [29], our method has increased the I-A score from 96.5% to 98.6% on MVTec-AD and 91.5% to 95.9% on VisA. On the other hand, separate models such as PaDiM [3] and DRAEM [31] have exhib-

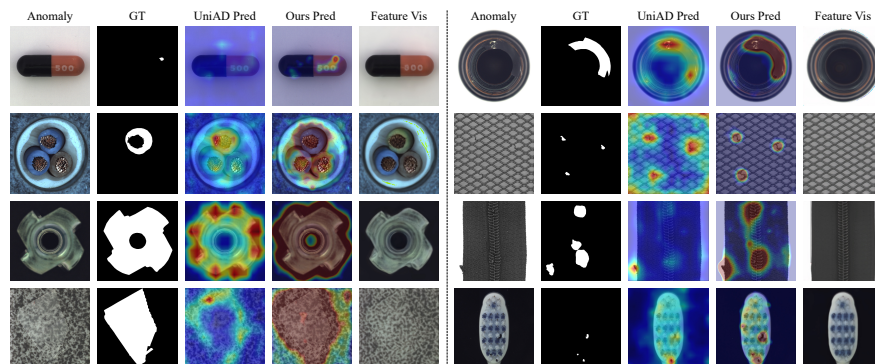


Fig. 3: Qualitative results on MVTec-AD. We visualize several anomalies (Anomaly) along with their corresponding Ground Truth (GT), the detection results of UniAD (UniAD Pred), the detection results of our method (Ours Pred), and the visualization of ours reconstructed features (Feature Vis).

ited a notable decrease in performance, indicating that these models struggle to effectively learn normal patterns across multiple classes.

Anomaly Localization. We have also achieved SOTA results on the pixel-level localization metric Pixel-AUROC (P-A), on both datasets. Since OmniAL [36] is a pixel-level reconstruction and detection model, it has an advantage in anomaly localization. However, our method, with its more accurate detection performance, has ultimately improved the P-A score of OmniAL on the MVTec-AD from 98.3% to 98.5%, and on the VisA from 96.6% to 99.2%. Moreover, compared to the feature reconstruction-based UniAD, our improvements are even more significant on MVTec-AD.

Qualitative Results on MVTec-AD. To further demonstrate the superiority of our method, we visualized the reconstruction and detection results of the MVTec-AD [1] dataset, as shown in Figure 3. It can be observed that UniAD fails to detect some anomalies as these anomaly features are not recovered to normal features. In contrast, our method successfully reconstructs normal features. This indicates that UniAD has not fully learned the normal patterns for these classes but instead learned shortcuts that prevent the reconstruction of anomalies as normal features. In contrast, our method recovers these samples to normal instances, demonstrates the effectiveness of our RLR in successfully addressing the issue of models learning shortcuts. We also conducted qualitative analysis on the VisA [37] dataset. Due to space limitations, please refer to the supplementary materials for details.

4.4 Ablation Study

Ablations of Each Component. We conducted further ablation experiments in the unified setting on the MVTec-AD dataset to validate the effectiveness of each proposed module. The results are shown in Table 3, where each mod-

Table 3: Ablation study of each component on MVTec-AD. Self Att. means vanilla Self Attention and Cross Att. means Cross Attention with learnable reference representation.

Modules				Metrics	
Residual	Self Att.	Cross Att.	MLKA LCA	I-A	P-A
✓	✓			83.5	85.3
✓		✓		85.6	86.9
		✓		95.7	97.1
			✓	96.4	97.5
				✓	97.9 98.0
		✓	✓	96.8	97.9
			✓	✓	98.6 98.5

ule only operates on the attention mechanism. “Residual” indicates the presence of residual connections, “Self Att.” computes the self-attention of input features, while in “Cross Att.”, the Query vector is the input feature, and the Key and Value vectors are learnable reference features.

Firstly, residual connections lead Transformer Encoder-based reconstruction models into the trap of learning shortcuts, whether utilizing Self Attention or Cross Attention. Secondly, the Cross Attention without residual connections in Table 3 can represent the foundational idea (baseline) of the our proposed method, which forces the model to reconstruct from learnable reference features explicitly, and we can see that it achieved decent detection performance. Compared with Self Attention with residual connections, its I-A and P-A metrics improved by 12.2% and 11.8%, demonstrating the effectiveness of our framework. Additionally, our proposed MLKA and LCA have further improved the metrics I-A and P-A by 2.9% and 1.4% compared with Cross Attention, which also proves the effectiveness of the two modules we proposed. Finally, incorporating local constraints in LCA yields better results compared to vanilla Cross Attention with or without MLKA. This indicates that adding local constraints enables the model to learn more effective references.

Analysis of Multi-Scale Feature Extraction. We investigate the results of combining feature maps at different scales, as shown in Table 4. Since the feature map of stage 4 is too small and no longer provides meaningful reconstruction information, we combine the feature maps from stages 1 to 3. From the results, it can be observed that the feature maps from stages 2 and 3 exhibit better detection performance (I-A). The feature map from stage 1 enhances the ability for anomaly localization (P-A) but may have a slight negative impact on detection (I-A).

Analysis of Hyperparameter α . We conducted experiments with different values of α , and the results are shown in Table 5. It can be observed that alpha values greater than 1 outperform α equals 1. This indicates that MLKA

Table 4: Different combination choices of multi-scale feature maps on MVTec-AD. Metrics are presented in the form of Image-AUROC% / Pixel-AUROC%.

Combination	stage 1, 2	stage 2, 3	stage 1, 2, 3
Metrics	97.6 / 97.9	98.9 / 98.1	98.6 / 98.5

Table 5: Different parameter α on MVTec-AD. Metrics are presented in the form of Image-AUROC% / Pixel-AUROC%.

$\alpha =$	1	2	3
Metrics	98.3 / 98.4	98.6 / 98.5	98.5 / 98.4

still has the potential for learning shortcuts, therefore it is necessary to pay more attention on LCA. However, excessively high α value can render MLKA ineffective, resulting in a decline in metrics. Additionally, all α values exhibit high metrics, indicating the robustness of our approach.

Analysis of Feature Perturbation. Since our method has already eliminated the possibility of the model relying on shortcuts, adding noise to the input is not necessary. However, perturbations can still enhance the model’s reconstruction ability to some extent. After removing the perturbations, our method achieved a I-A / P-A result of 98.3%/98.2% on the MVTec-AD, which is only 0.3% lower than when noise was added in both I-A and P-A metrics. In contrast, in the context of UniAD, perturbations are used to increase the difficulty of the model learning shortcuts, making them more important in the UniAD framework.

5 Conclusion

In this paper, we propose the RLR framework to address the issue of learning shortcuts in feature-based reconstruction for multi-class anomaly detection methods. The RLR framework Reconstructs features from Learnable Reference representaion. To further improve the accuracy and effectiveness of the references, as well as the precision and detail of the reconstructed features, we design the Local Cross Attention module and the Mask Learnable Key Attention module. Our experimental results demonstrate the state-of-the-art (SOTA) performance of our method on various datasets. Both qualitative and quantitative analysis validate the effectiveness of our approach.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No.62171139).

References

1. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9592–9600 (2019)
2. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4183–4192 (2020)
3. Defard, T., Setkov, A., Loesch, A., Audigier, R.: Padim: a patch distribution modeling framework for anomaly detection and localization. In: International Conference on Pattern Recognition. pp. 475–489. Springer (2021)
4. Deng, H., Li, X.: Anomaly detection via reverse distillation from one-class embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9737–9746 (2022)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
6. Fernando, T., Gammulle, H., Denman, S., Sridharan, S., Fookes, C.: Deep learning for medical anomaly detection—a survey. *ACM Computing Surveys (CSUR)* **54**(7), 1–37 (2021)
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. *Communications of the ACM* **63**(11), 139–144 (2020)
8. Gudovskiy, D., Ishizaka, S., Kozuka, K.: Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 98–107 (2022)
9. He, B., Martens, J., Zhang, G., Botev, A., Brock, A., Smith, S.L., Teh, Y.W.: Deep transformers without shortcuts: Modifying self-attention for faithful signal propagation. In: The Eleventh International Conference on Learning Representations (2023)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33**, 6840–6851 (2020)
12. Lee, Y., Kang, P.: Anovit: Unsupervised anomaly detection and localization with vision transformer-based encoder-decoder. *IEEE Access* **10**, 46717–46724 (2022)
13. Lei, J., Hu, X., Wang, Y., Liu, D.: Pyramidflow: High-resolution defect contrastive localization using pyramid normalizing flow. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14143–14152 (2023)
14. Liu, Z., Zhou, Y., Xu, Y., Wang, Z.: Simplenet: A simple network for image anomaly detection and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20402–20411 (2023)
15. Mishra, P., Verk, R., Fornasier, D., Piciarelli, C., Foresti, G.L.: Vt-adl: A vision transformer network for image anomaly detection and localization. In: 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE). pp. 01–06. IEEE (2021)

16. Pirnay, J., Chai, K.: Inpainting transformer for anomaly detection. In: International Conference on Image Analysis and Processing. pp. 394–406. Springer (2022)
17. Rippel, O., Mertens, P., Merhof, D.: Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 6726–6733. IEEE (2021)
18. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14318–14328 (2022)
19. Rudolph, M., Wandt, B., Rosenhahn, B.: Same same but different: Semi-supervised defect detection with normalizing flows. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 1907–1916 (2021)
20. Rudolph, M., Wehrbein, T., Rosenhahn, B., Wandt, B.: Fully convolutional cross-scale-flows for image-based defect detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1088–1097 (2022)
21. Salehi, M., Sadjadi, N., Baselizadeh, S., Rohban, M.H., Rabiee, H.R.: Multiresolution knowledge distillation for anomaly detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14902–14912 (2021)
22. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. pp. 6105–6114. PMLR (2019)
23. Tien, T.D., Nguyen, A.T., Tran, N.H., Huy, T.D., Duong, S., Nguyen, C.D.T., Truong, S.Q.: Revisiting reverse distillation for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24511–24520 (2023)
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
25. Wang, C., Zhu, W., Gao, B.B., Gan, Z., Zhang, J., Gu, Z., Qian, S., Chen, M., Ma, L.: Real-riad: A real-world multi-view dataset for benchmarking versatile industrial anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22883–22892 (2024)
26. Yan, X., Zhang, H., Xu, X., Hu, X., Heng, P.A.: Learning semantic context from normal samples for unsupervised anomaly detection. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 3110–3118 (2021)
27. Yao, X., Li, R., Qian, Z., Luo, Y., Zhang, C.: Focus the discrepancy: Intra- and inter-correlation learning for image anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6803–6813 (2023)
28. Yi, J., Yoon, S.: Patch svdd: Patch-level svdd for anomaly detection and segmentation. In: Proceedings of the Asian conference on computer vision (2020)
29. You, Z., Cui, L., Shen, Y., Yang, K., Lu, X., Zheng, Y., Le, X.: A unified model for multi-class anomaly detection. *Advances in Neural Information Processing Systems* **35**, 4571–4584 (2022)
30. You, Z., Yang, K., Luo, W., Cui, L., Zheng, Y., Le, X.: Adtr: Anomaly detection transformer with feature reconstruction. In: International Conference on Neural Information Processing. pp. 298–310. Springer (2022)
31. Zavrtnik, V., Kristan, M., Skočaj, D.: Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8330–8339 (2021)
32. Zhang, H., Wang, Z., Wu, Z., Jiang, Y.G.: Diffusionad: Denoising diffusion for anomaly detection. *arXiv preprint arXiv:2303.08730* (2023)

33. Zhang, J., He, H., Gan, Z., He, Q., Cai, Y., Xue, Z., Wang, Y., Wang, C., Xie, L., Liu, Y.: Ader: A comprehensive benchmark for multi-class visual anomaly detection. arXiv preprint arXiv:2406.03262 (2024)
34. Zhang, J., Wang, C., Li, X., Tian, G., Xue, Z., Liu, Y., Pang, G., Tao, D.: Learning feature inversion for multi-class anomaly detection under general-purpose coco-ad benchmark. arXiv preprint arXiv:2404.10760 (2024)
35. Zhao, Y.: Just noticeable learning for unsupervised anomaly localization and detection. In: 2022 IEEE International Conference on Multimedia and Expo (ICME). pp. 01–06. IEEE (2022)
36. Zhao, Y.: Omnia: A unified cnn framework for unsupervised anomaly localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3924–3933 (2023)
37. Zou, Y., Jeong, J., Pemula, L., Zhang, D., Dabeer, O.: Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In: European Conference on Computer Vision. pp. 392–408. Springer (2022)