Dual-branch Normalizing Flow for Anomaly Detection and Localization from Images

Yao Li[a], Wei Ma[a], Shuai He[b], Shiyong Lan[a,*], Wenwu Wang[c], Yixin Qiao[a] and Guonan Deng[a]

[a]College of Computer Science, Sichuan University, Chengdu, 610065, China
[b]School of Emergency Management, Chengdu University, Chengdu, 610106, China
[c]Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, UK

ARTICLE INFO

Keywords:
Anomaly detection
Anomaly localization
Normalizing flow
Unsupervised learning
Spatial attention

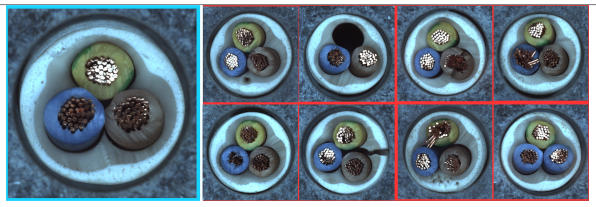
ABSTRACT

In visual anomaly detection, unexpected or abnormal patterns are identified from given image samples. Existing methods for visual anomaly detection are mainly divided into two types: semantic inter-class (i.e., image-level) anomaly detection and texture intra-class anomaly detection. The normalizing flow-based method can effectively map normal training data to a Gaussian distribution due to its excellent distribution expression ability, thus interpretably identifying lower likelihood abnormal data from normal data. However, with normalizing flow based methods, it is challenging to achieve high precision detection and localization of anomalies due to the unpredictable scale of the anomalies within texture anomalies. The large scale span of potential anomalous textures makes it difficult to balance the learning of distributions from both global and local features in existing normalizing flow methods. To address this limitation, we propose a dual-branch architecture to model the density mapping of global and local features, respectively. Our proposed model can achieve coarse-grained and fine-grained image anomaly detection and localization, via modeling both the global features and local texture attributes of the input images with a dual branch normalizing flow. Furthermore, we design a Dynamic Spatial Attention Module (DSAM) in each branch of the flow module to enhance the model's ability in capturing anomaly features. Extensive experiments on two public datasets have demonstrated that our model is effective in detecting various real-world sample defects, especially in unsupervised visual anomaly detection tasks, achieving substantially promising results. The codes are available at https://github.com/SYLan2019/DNFAD.

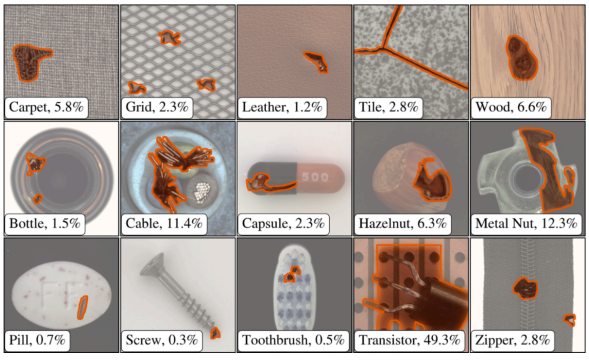
1. Introduction

In industrial manufacturing applications, automated detection of product surface defects (such as scratches, cracks, and uneven coating) is a key step in ensuring product quality. For example, on electronic component production lines, even tiny solder joint defects can lead to equipment malfunctions, while in the textile industry, anomalies in fabric texture can directly impact the quality of the final products [1; 2]. In the medical field, potential abnormal patterns on medical images are of great importance for the effective diagnosis of diseases [3]. Visual anomaly detection, as a classic problem in computer vision, is an important component of visual content understanding. It is primarily utilized to distinguish samples that do not meet specified norms, facilitating rapid identification of anomalous instances and aiding in dataset cleaning [1], industrial anomaly detection [2], and medical image anomaly detection [3]. Due to the scarcity of abnormal samples in practical applications, existing mainstream anomaly detection methods adopt unsupervised learning [4; 5], which only uses normal samples for training.

Recently, unsupervised anomaly detection methods have achieved great success in industrial and medical applications. For example, the Semantic-Aware Normalizing Flow (SANF) [4], as a recent flow-based method, improves the performance of image level anomaly detection by combining the semantic and spatial features of images. The Generative Adversarial Network (GAN) based Omni-frequency Channel-selection Reconstruction (OCR-GAN) [5] utilizes frequency decoupling and interaction between multiple frequencies to achieve state-of-the-art performance in reconstruction-based methods. However, existing methods face two main



(a) The ambiguity in the types of anomalies



(b) The variation in the size and shape of anomalies

Figure 1: Two main challenges in anomaly detection

*Corresponding author.
liyao518@stu.scu.edu.cn (Y. Li); mawei12138@stu.scu.edu.cn (W. Ma); heshuai@cdu.edu.cn (S. He); lanshiyong@scu.edu.cn (S. Lan); w.wang@surrey.ac.uk (W. Wang); Qiaoyixin@stu.scu.edu.cn (Y. Qiao); 2023223045102@stu.scu.edu.cn (G. Deng)

challenges: 1) the ambiguity in the types of image anomalies, 2) the wide range of size and shape of the anomalies within images. The first challenge is due to the fact that the anomaly types may not be fixed, and the boundary between the normality and anomaly could be unclear. For example, in Figure 1(a), a normal cable (left) has eight known anomalies (right), while its unknown potential anomalies are countless since the causes of anomalies are unpredictable. As for the second challenge, the variation in anomaly sizes and shapes is illustrated in Figure 1(b), which is caused by the inherent size of the detected object and the degree of its unpredictable behavior [6].

To address the first challenge, the key lies in the representational ability of visual features to facilitate the discrimination of various anomalies in images. Some existing methods [7; 8] rely on the backbones pre-trained on a large number of natural images such as ImageNet [9], which have already learned visual information from various scenarios, in order to obtain robust feature representations and alleviate the impact from the ambiguity of anomaly types. However, there is inevitably a domain gap between the data used for pre-training the model and the data to which the model is applied in practical scenarios. Several recent flow-based methods [10; 4] fuse the semantic and spatial features to alleviate the ambiguity of anomaly types. For instance, SANF [4] designs an attention module in each flow block to enhance semantic representation by fusing spatial features from pre-trained backbones, thereby achieving better performance in image-level anomaly identification than flow-based baselines. However, SANF mainly focuses on enhancing the semantic features of images to improve the performance of image level anomaly identification, and is not concerned with the spatial features of images used for anomaly localization, so SANF cannot achieve anomaly localization.

To address the second challenge, existing methods have introduced multi-scale approaches to extract information from images at various scales [11; 12] and utilize hierarchical features from pre-trained models [13]. However, these multi-scale based methods not only suffer from the loss of inherent high-frequency information during the upsampling and downsampling processes [5], but also lead to increasingly complex models and significant computational overhead [11; 13]. In addition, Zoom Text Detector (ZTD) [14] and CM-Net [15] propose to concatenate features from different convolutional layers, and design objective functions to focus on information from different perspectives. For example, CM-Net employs a multi-perspective feature (MPF) module to enable the model to recognize the central mask (CM) from contour features, patch-level local features, and gap features of text instances, thus achieving excellent performance of text detection. Inspired by the idea of multi-view learning [15; 16], we introduce a novel dual-branch flow-based approach to address the issue of wide range variations in anomaly sizes, by focusing on the features of input data with two granularity levels. Specifically, these two branches are named Global Branch and Local Branch, respectively. The Global Branch primarily models the global features of the images to cope with large-scale anomalies, utilizing normalizing flow to assign high-density values to the global features of the images. The Local Branch focuses on modeling the features of the images at the pixel-level to be sensitive to weak anomalies, and then estimating the distribution density of each pixel in the features via normalizing flow. Finally, the features obtained from these two branches are fused for final anomaly identification and localization. Therefore, this dual-branch flow-based method can effectively address the problem caused by large-scale variations in abnormal sizes.

This work is the first one that utilizes normalizing flow with dual-branch optimization objectives to simultaneously capture global and local anomaly information, which can effectively represent different granularity-level features of input data to address the uncertainty of anomalous scales. Different branches adopt different optimization strategies, among which the first branch is designed to use map flow loss, which considers the entire feature as a whole. The second branch uses the pixel flow loss to estimate the density of each pixel in the feature map. The detailed implementation will be discussed in Section 3.4. The proposed method has been evaluated on public datasets, i.e., MVTec AD [2] and Btad [17]. Experimental results demonstrate that our proposed method achieves superior performance in image anomaly detection, including anomaly identification and localization.

Our contributions can be summarized as follows:

- We propose a dual-branch normalizing flow model for visual anomaly detection (DNFAD), which is capable of modeling image features from both local and global perspectives, effectively addressing the shortcomings of the flow-based anomaly detection methods in identifying anomalous samples and accurately locating anomalies simultaneously.
- We design a dynamic spatial attention module (DSAM) that combines dynamic convolution and spatial attention mechanisms, enhancing the model's ability to capture spatial information and thus improving anomaly identification and localization performance.
- Extensive experiments demonstrate that our model outperforms existing flow-based baselines, and generalizes better than non-normalizing flow-based methods in real-world scenarios, where potential anomaly types are often unpredictable.

2. RELATED WORK

Due to the imbalance between normal and anomalous samples, unsupervised or semi-supervised methods, which only utilize normal samples or few-shot abnormal samples to design and construct models, have gradually become the mainstream approach for visual anomaly detection. Based on their principles and training tasks, the current unsupervised methods can be mainly divided into reconstruction-based methods, knowledge distillation-based methods, memory-based methods, and the normalizing flow-based methods.

2.1. Reconstruction-based Methods

The existing reconstruction-based methods usually apply autoencoders [18], variational autoencoder (VAE) [19; 20] and generative adversarial network (GAN) [21; 1; 22] as a generator. During training, these models take normal images as input and reconstruct them. During testing, these models compute the anomaly score for each image sample based on the reconstruction error between the generated image and the original input image. Since these models only encounter normal samples during training, it learns how to reconstruct normal image samples but lacks the ability to reconstruct anomalous image samples. Therefore, the reconstruction error for anomalous samples will be much higher than for normal samples, resulting in higher anomaly scores for anomalous samples. However, due to the use of Mean Squared Error (MSE) [23] loss, these models perform well for pixel-level reconstruction, even for anomalous samples, for which the reconstruction errors could be close to or even smaller than those for normal samples [7].

2.2. Synthesis-based Methods

The synthesis-based methods, such as CutPaste [24] and FAIR (i.e. frequency aware image restoration) [25], introduce simulated anomalies during training, artificially creating anomalies on images to train the model. The optimization goal of the model is to reconstruct samples without anomalies, and the model uses the reconstruction error between the reconstructed samples and the original samples as the criterion for anomaly detection. However, due to the diversity of anomaly types, it is impossible to simulate all potential anomalies, as a result, such methods are prone to overfitting, especially for some of the simulated anomalies. For example, FAIR [25] can achieve high performance on the MVTec AD dataset [2], but it does not generalize well to other datasets. Recently, ADShift [26] introduces a similarity loss in the training phase to constrain the differences in latent features between the original sample and the augmented sample, thereby preventing the model from being affected by domain distribution shifts. ReContrast [27] proposes a new Contrastive Pairs strategy to optimize the network based on cross reconstruction errors, avoiding the use of commonly employed data augmentation that may introduce potential anomalies. However, ADShift and ReContrast augment samples by simulating various perturbations from different domains, which inevitably poses a potential risk of corrupting the original normal texture, and mislead the model in learning from the real domain due to the introduction of unrealistic simulated domain conditions.

2.3. Knowledge Distillation-based Methods

These methods utilize network models pre-trained on ImageNet [9] such as AlexNet [28], ResNet [29], and VGG [30] as teacher model. Then, they construct a student model with a similar or identical architecture to the teacher model. During training, these methods utilize normal samples as input to distill the representation knowledge from the teacher model to the student model. Specifically, the objective is to minimize the discrepancy between the features extracted by the teacher and student networks at different stages of the model [31]. During testing, the student model, which has not been exposed to anomalous samples, inherently possesses a significant gap in representation compared to the teacher model for anomalous samples. Some methods employ the reverse distillation paradigm [7], where the structure remains unchanged but the flow of data between the teacher and student models is altered to amplify the difference in representation of anomalous samples, aiming to enhance the anomaly detection and localization capabilities. However, because these methods typically require constructing two network models, their costs are relatively high, and the inference time is also relatively long.

2.4. Memory-based Methods

Memory-based methods aim to use a memory bank to reduce the size of representative features learned from normal samples during training. In inference, these methods first select the item stored in the memory bank that is most similar to the given query sample, and then use the difference between the query sample and its most similar item to derive the anomaly score. The Deep SVDD [32] (i.e. deep support vector data description) method compares the differences in global features of samples while Patch SVDD [33] compares differences at the pixel level. PatchCore [8] selects local block features from normal training data and utilizes a greedy core subset sampling method to choose a representative subset belonging to normal samples. However, such methods require additional computational cost, when comparing the test features with the closest features in the memory bank.

2.5. Normalizing Flow-based Methods

Normalizing flow models have been utilized to estimate the density of normal features, assigning higher probabilities to normal features and lower probabilities to anomalous features. However, Kirichenko et al. [34] pointed out that normalizing flow models trained on raw RGB images often assign higher probability values to anomalous images. To address this issue, Rudolph et al [35; 11] proposed to apply normalizing flow to high-dimensional features rather than raw images. They introduced DifferNet [35], which implements a normalizing flow network for image-level anomaly detection based on pre-trained features. Rudolph et al. further proposed CSFlow [11], which rescales images to three different sizes to address continuously changing sizes of anomalies. Gudovskiy et al. proposed CFlow [13] for anomaly detection, which utilizes multi-level feature maps with different receptive fields given by a pre-trained feature extractor. SANF [4] and MF4 [36] (i.e. multi-frequency feature fusion) adopt Vision Transformer (ViT) [37] as the backbone to obtain the pre-trained features, and utilize the normalizing flow to learn the density distribution of the semantic features enhanced by a attention mechanism, thereby obtaining the final anomaly detection score. However, SANF mainly focuses on One Class Classification (OCC) issues and cannot achieve anomaly localization.

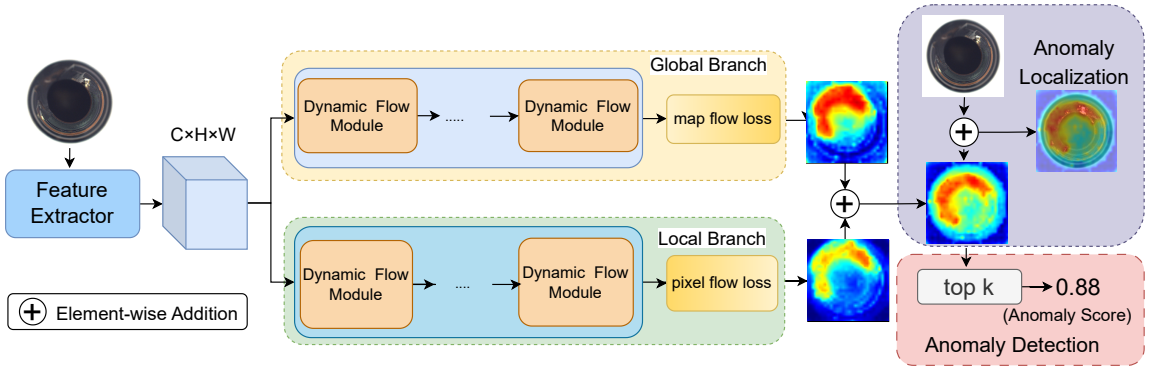


Figure 2: Overview of our proposed DNFAD. The model primarily consists of two parts, namely, the feature extractor and the dual-branch normalizing flow. The former is responsible for extracting spatial features from images. The latter akin to the decoders, estimates the density of the features separately from the local and global perspectives. Finally, the outputs from the two flow-based branches are fused for anomaly detection and localization.

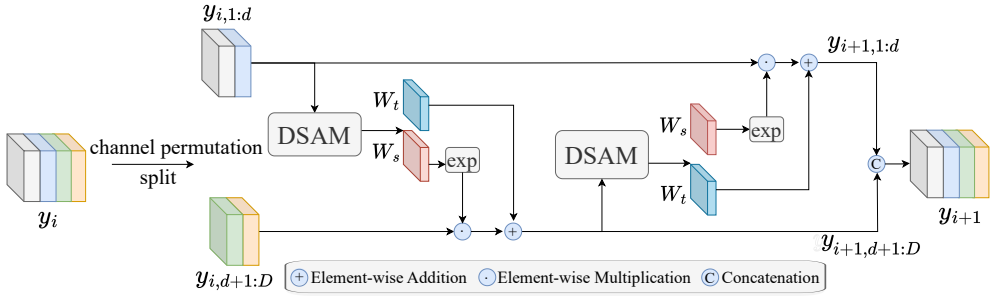


Figure 3: The dynamic flow module. The DSAM component is designed to dynamically adjust the scaling parameters W_s and translation parameters W_t in the affine coupling layer of the normalizing flow, enabling it to adaptively capture the changes in the input features.

3. Proposed Method

3.1. Overall Architecture of the Proposed Model

Figure 2 illustrates the proposed dual-branch anomaly localization model based on normalizing flow and spatial attention mechanism. This model mainly consists of two parts: the first is the feature extractor and the second is the dual-branch normalizing flow. The feature extractor is responsible for extracting spatial features from images, while the dual-branch flow functions akin to decoders models the local and global features separately.

Specifically, we use the DINO (i.e., self-distillation with no labels) model [38], pre-trained on large-scale natural images, as the feature extractor, taking into account the complexities and variations in visual scenes. Subsequently, these features are fed into the dual-branch flow for analysis, where each branch consists of dynamic spatial attention flow modules, as detailed in Sections 3.2 and 3.3. However, these two branches adopt different optimization strategies. Specifically, the first branch (i.e., Global Branch) is designed to utilize the map flow loss, which treats the entire feature map as a whole and learns the distribution density of the entire feature map of normal training samples, while the second branch (i.e., Local Branch) is designed to employ the pixel flow loss, drawing inspiration from idea of patch encoding in [39], to estimate the density of each pixel in the feature map separately. The detailed definition of the loss function can be found in Section 3.4.

3.2. Dynamic Flow Module

The flow module in each branch of our model refers to the classic normalizing flow network, i.e., real-valued Non-Volume Preserving (real-NVP) [40]. Unlike real-NVP, we introduce an attention mechanism to dynamically adjust the scaling parameter W_s and translation parameter W_t in the affine coupling layer of real-NVP. Therefore, we call our flow module as a dynamic flow module. Figure 3 shows the architecture of this module, where we design a Dynamic Spatial Attention Module (DSAM), which will be discussed in detail in Section 3.3. Assume the input to DSAM is denoted as y_i and the output as y_{i+1}, for the i-th dynamic flow module. In this module, the input features are shuffled along the channels, then they are evenly divided into $y_{i,1:d}$ and $y_{i,d+1:D}$ along the channel dimension, where D is the number of channels, using $split(\cdot)$ function as follows,

$$[y_{i,1:d}, y_{i,d+1:D}] = split(y_i) \tag{1}$$

After this, $y_{i,1:d}$ is input into the dynamic spatial attention module to estimate the parameters W_s and W_t needed for affine transformations, which are then applied to $y_{i,d+1:D}$ to obtain $y_{i+1,d+1:D}$. Subsequently, the newly obtained $y_{i+1,d+1:D}$ is input into another dynamic spatial attention module to estimate new W_s and W_t, which are applied to $y_{i,1:d}$ to obtain $y_{i+1,1:d}$. Finally, the module concatenates the $y_{i+1,1:d}$ and $y_{i+1,d+1:D}$ along the channel dimension to obtain the feature y_{i+1}. Thus, following [40],

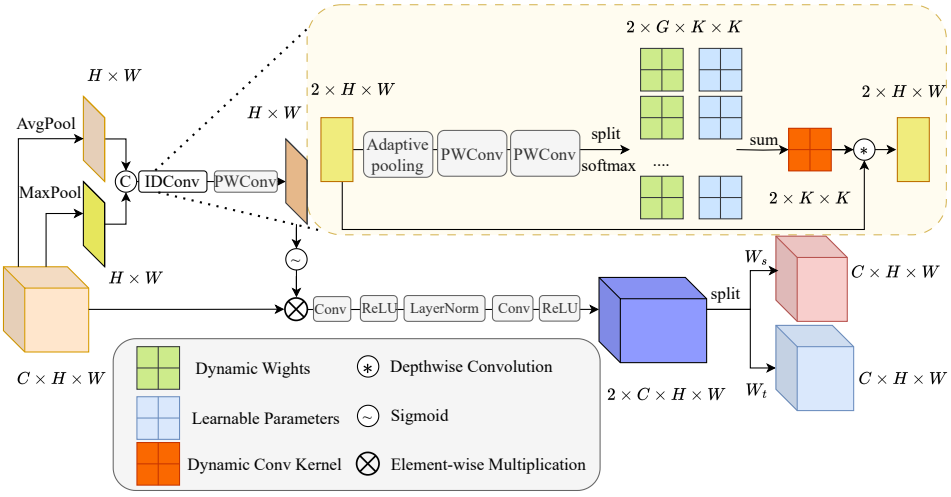


Figure 4: The dynamic spatial attention module (DSAM). The IDConv is introduced into our spatial attention module to dynamically aggregate spatial contextual information from the input features.

the output of each dynamic flow module can be obtained:

$$(W_{s1}, W_{t1}) = DSAM(y_{i,1:d}) \tag{2}$$

$$y_{i+1,d+1:D} = y_{i,d+1:D} \odot e^{W_{s1}} + W_{t1} \tag{3}$$

$$(W_{s2}, W_{t2}) = DSAM(y_{i+1,d+1:D}) \tag{4}$$

$$y_{i+1,1:d} = y_{i,1:d} \odot e^{W_{s2}} + W_{t2} \tag{5}$$

$$y_{i+1} = Concat(y_{i+1,1:d}, y_{i+1,d+1:D}) \tag{6}$$

where $\odot$ denotes element-wise multiplication, and Concat$(\cdot)$ indicates the concatenation of two features along the channel dimension. Lastly, the proposed method employs soft-clamping operation to maintain model stability [40].

$$\sigma_\alpha(W_s) = \frac{2\alpha}{\pi} arctan \frac{W_s}{\alpha} \tag{7}$$

where $\sigma_\alpha(\cdot)$ denotes soft-clamping operation, α represents a hyper-parameter, and W_s denotes the parameter generated in the dynamic flow module.

3.3. Dynamic Spatial Attention Module

Inspired by SANF [4], where the attention mechanism is introduced into the normalizing flow model, we design the DSAM component in our model, whose structure is illustrated in Figure 4. However, unlike the static spatial attention used in SANF [4], we introduce the Input-dependent Depthwise Convolution (IDConv) [41] into the DSAM module to generate spatial attention based on input data, thus reflecting its dynamic fluctuations with changes in input. As shown in the processing pipeline of Figure 4, the parameters of W_s and W_t are adjusted by the introduced dynamic spatial attention module. Additionally, compared to static convolution, the dynamic convolution generates weights for convolution by analyzing global spatial information in input images, resulting in better performance [42].

Specifically, assuming the input $X \in \mathbb{R}^{C\times H\times W}$. The model first adopts a strategy similar to the classical attention module, i.e. Convolutional Block Attention Module (CBAM) [43], to average and max pool the features, resulting in two $H \times W$ features. These two features are then concatenated, and a dynamic IDConv [41], which is a variant of dynamic convolution [42; 44], is applied to adjust the spatial dimensions of the features. Subsequently, a 1×1 Point-Wise Convolution (PWConv) is employed to compress the features along the channel dimension, generating a spatial feature score map. This score map is then multiplied with the original features through a $Sigmoid(\cdot)$ activation function to adjust the original features. Additionally, we further process the features with LayerNorm, aiming to enhance model's stability [45]. Subsequently, the model processes the normalized features through two convolutional layers followed by two $ReLU(\cdot)$ activation functions, resulting in a $2 \times C \times H \times W$ feature map. This feature map is then segmented along the channel dimension to obtain the parameters W_s and W_t required for the normalizing flow, where $W_s \in \mathbb{R}^{C\times H\times W}$ and $W_t \in \mathbb{R}^{C\times H\times W}$.

As depicted in Figure 4, IDConv first aggregates the spatial contexts of the input feature through adaptive average pooling, compressing the spatial dimensions to K^2. Subsequently, the compressed features are fed into two 1×1 convolutions, resulting in an attention map $A' \in \mathbb{R}^{(G\times 2)\times K^2}$, where G represents the number of attention map groups. Then, the attention map is transformed into $\mathbb{R}^{G\times 2\times K^2}$ and passed through a softmax function along the G dimension, generating attention weights $A \in \mathbb{R}^{G\times 2\times K^2}$. Finally, A is element-wise multiplied with a set of learnable parameters $P \in \mathbb{R}^{G\times 2\times K^2}$ and summed along the G dimension, producing the convolution kernel $W \in \mathbb{R}^{C\times K^2}$.

3.4. Loss Function

The input feature as $y \in \mathbb{R}^{H\times C\times W}$, with probability density $P_Y(y)$. The output from the Global Branch is $z_M \in \mathbb{R}^{H\times C\times W}$, with probability density $P_{Z_M}(z_M)$. The relationship between the density of z_M and y is as shown in

Equation (8).

$$P_Y(y) = P_{Z_M}(z_M)\left|\ \det(\frac{\partial z_M}{\partial y})\ \right| \tag{8}$$

where $z_M = F_m(y_{sem})$ and $F_M : Y \to Z_M$ is the first branch proposed in this paper. To simplify the calculation, taking the logarithm on both sides of the equation and assuming Z follows a Gaussian distribution, i.e., $z \sim \mathcal{N}(0, I)$. Then, we construct the loss function $\mathcal{L}_M$ for the first branch model by minimizing the negative log-likelihood $-\log P_Y(y)$: $\mathcal{L}_M$:

$$\mathcal{L}_M = -\log p_Y(y) = \frac{\left\|z_M\right\|_2^2}{2} - \log\left|\det(\frac{\partial z_M}{\partial y})\right| \tag{9}$$

where $\|\cdot\|_2^2$ represents the L_2 norm, and $\left|\det(\frac{\partial z_M}{\partial y})\right|$ denotes the absolute determinant of the Jacobian matrix.

For the output feature of the second branch, denoted as z_P, this model considers the input and output of this branch as features composed of individual vectors:

$$y = \{y_1, ..., y_i, ..., y_N\}^{N=H\times W} \tag{10}$$

$$z = \{z_1, ..., z_i, ..., z_N\}^{N=H\times W} \tag{11}$$

where $y_i \in \mathbb{R}^C$ and $z_i \in \mathbb{R}^C$. Unlike the optimization approach of the first branch, which operates at the feature level, this branch optimizes the model parameters at the pixel level, aiming to maximize the likelihood of each individual z_i. Therefore, for each pair of features y_i and z_i, the loss function $\mathcal{L}_i$ is designed as follows:

$$\mathcal{L}_i = -\log p_{Y_i}(y_i) = \frac{\left\|z_i\right\|_2^2}{2} - \log\left|\det(\frac{\partial z_i}{\partial y_i})\right| \tag{12}$$

Then, the loss function $\mathcal{L}_P$ for the second branch is obtained by summing and averaging over the $H \times W$ features:

$$\mathcal{L}_P = \frac{\sum_{i=1}^{N=H\times W} \mathcal{L}_i}{H \times W} \tag{13}$$

Finally, the loss function $\mathcal{L}$ is obtained by summing the loss functions of the two branches:

$$\mathcal{L} = \mathcal{L}_P + \mathcal{L}_M \tag{14}$$

In summary, with the loss function of the global branch, the entire feature map of each sample are treated as a whole, and then constraints are enforced on the learning of the density distribution of the feature maps. Consequently, the designed global branch is better at modeling the global characteristics of the input image. However, the local branch treats each pixel independently, and models its probability density distribution accordingly, in order to capture local subtle anomalies. With the innovative design of these two objective functions, we are able to utilize the complementarity between the two branches to improve the performance of anomaly detection.

3.5. Anomaly Localization and Anomaly Score

We adopt the method proposed in [6] for estimating the log-likelihood of transformed features z, where $\log P_Z(z) = -\frac{\|z\|_2^2}{2}$. Similarly, for the output features from the Global and Local Branch, we can obtain their corresponding log-likelihood $\log P_M$ and $\log P_P$, respectively. Specifically, the model first restores the log-likelihood maps of the two branches to the same size as the input image using bi-linear interpolation. For pixel-level localization, the log-likelihood maps ($\log P_M$, $\log P_P$) of the down-scaled dual-branch outputs will be transformed into probability maps ($e^{\log P_M}$, $e^{\log P_P}$) after applying the exponential function. Subsequently, the model will perform element-wise addition of the two probability maps:

$$P_{add} = e^{\log P_M} + e^{\log P_P} \tag{15}$$

Subsequently, for the fused probability map P_{add}, the model will scale it to obtain anomaly localization score map. For the anomaly score localization images, the model will employ the $topK$ method for computation:

$$S_{loc} = max(P_{add}) - P_{add} \tag{16}$$

$$S_{det} = \frac{1}{K}\sum_{1}^{K} topK(S_{loc}) \tag{17}$$

where $max(\cdot)$ represents taking the maximum value, and $topK(\cdot)$ represents selecting the values of the top-K pixels in the score map, with K set to 1 indicating the maximum value, and K set to the size of the input image indicating the global average value.

4. Experiment

4.1. Datasets and Metrics

We use the popular datasets employed in industrial anomaly detection: MVTec AD [2], MPDD [51] and Btad [17]. The MVTec AD dataset, provided by Bergmann et al. [2] in 2019, consists of real-world industrial production scenes for anomaly detection. It has been widely employed to evaluate the effectiveness of various methods for detecting anomalies in industrial products. This dataset primarily comprises two types of anomalies: object anomalies and texture anomalies. On the other hand, the Btad dataset contains three common types (i.e., Btad01, Btad02, Btad03) of industrial products for anomaly detection.

In the training sets of the MVTec AD and Btad datasets, only normal samples are included. However, the test set contains a combination of normal samples, anomalous samples, and pixel-level annotations for the anomalous samples. The evaluation metric is the AUROC [52] metric. Due to the completion of the anomaly localization task in this paper, in addition to using the image-level AUROC metric (I-AUROC), we also utilize the pixel-level AUROC metric (P-AUROC) for evaluating the anomaly localization performance.

4.2. Implementation Details

We use DINO [38] and ResNet [29] as our feature extractor, with a fusion of features from the 7th and 11th

Table 1
The results of I-AUROC [%] and P-AUROC[%] for various methods on the MVTec AD dataset.

Subclass	Category	Non-normalizing flow-based methods									Normalizing flow-based methods				
		PatchCore [8] CVPR2022	RDAD [7] CVPR2022	FAIR [25] arxiv2023	SSM [46] TMM2023	OCR-GAN [5] TIP2023	DeSTSeg [47] CVPR2023	NSA [48] ECCV2022	MSFR [49] KBS2024	ReContrast [27] NIPS2023	ADShift [26] ICCV2023	CSFlow [11] WACV2022	CFlow [13] WACV2022	AST [50] WACV2023	SANF [4] NeuroCom2024	Ours -
Textures	carpet	(98.7,99.0)	(98.9,98.9)	(99.7,**99.6**)	(76.3,94.4)	(99.4,-)	(98.9,96.1)	(95.6,95.5)	(99.8,98.6)	(99.8,99.3)	(-98.4)	(**100**,98.0)	(97.6,99.2)	(97.5,97.4)	(96.8,-)	(**100**,98.6)
	grid	(98.2,98.7)	(**100**,99.3)	(99.7,**99.4**)	(**100**,99.0)	(99.6,-)	(99.7,99.1)	(99.9,99.2)	(**100**,98.8)	(**100**,99.2)	(-98.4)	(99.1,96.1)	(98.1,96.1)	(99.1,95.7)	(98.1,-)	(99.8,98.9)
	leather	(**100**,99.3)	(**100**,99.4)	(**100**,99.6)	(99.9,99.6)	(97.1,-)	(**100**,**99.7**)	(99.9,99.5)	(**100**,99.2)	(**100**,99.5)	(-99.4)	(**100**,98.4)	(99.9,**99.7**)	(**100**,98.7)	(**100**,-)	(**100**,**99.7**)
	tile	(98.7,95.6)	(99.3,95.6)	(**100**,98.4)	(94.4,90.2)	(95.5,-)	(**100**,98.0)	(**100**,**99.3**)	(99.2,95.4)	(99.8,96.3)	(-94.4)	(**100**,93.9)	(97.1,96.2)	(**100**,97.1)	(**100**,-)	(99.5,98.5)
	wood	(99.2,95.0)	(99.2,95.3)	(**100**,97.3)	(95.9,86.9)	(95.7,-)	(97.1,97.7)	(97.5,90.7)	(99.3,94.6)	(99.0,95.9)	(-94.3)	(**100**,88.6)	(98.7,86.6)	(**100**,97.0)	(97.8,-)	(99.4,**98.0**)
	Average.Tex	(98.9,97.5)	(99.5,97.7)	(**99.9**,**98.8**)	(93.3,94.0)	(97.5,-)	(99.1,98.1)	(98.6,96.8)	(99.7,97.3)	(99.7,98.0)	(-,97.0)	(99.8,95.0)	(98.3,95.6)	(99.3,97.2)	(98.5,-)	(99.7,**98.8**)
Objects	bottle	(**100**,98.6)	(**100**,**98.7**)	(**100**,98.3)	(99.9,95.9)	(99.6,-)	(**100**,99.2)	(97.7,98.3)	(**100**,98.2)	(**100**,99.0)	(-,99.2)	(99.8,90.9)	(99.9,97.2)	(**100**,97.8)	(**100**,-)	(**100**,**98.7**)
	cable	(99.5,98.4)	(95.1,97.4)	(98.1,98.5)	(77.3,82.1)	(99.1,-)	(97.8,97.3)	(94.5,96.0)	(97.5,96.0)	(**99.8**,**98.9**)	(-,97.9)	(99.1,95.3)	(97.6,97.8)	(98.5,91.8)	(96.0,-)	(98.9,98.0)
	capsule	(98.1,98.8)	(96.3,98.5)	(97.0,93.9)	(91.4,98.4)	(96.2,-)	(97.0,99.1)	(95.2,97.6)	(96.9,98.3)	(97.7,98.4)	(-,98.5)	(97.1,97.9)	(97.0,**99.1**)	(**99.7**,98.6)	(97.3,-)	(99.1,98.2)
	hazelnut	(**100**,98.8)	(99.9,98.9)	(99.2,99.4)	(91.5,97.4)	(98.5,-)	(99.9,**99.6**)	(94.7,97.6)	(**100**,98.4)	(**100**,99.1)	(-,98.8)	(99.6,96.3)	(**100**,98.8)	(**100**,98.1)	(99.8,-)	(**100**,99.4)
	metal_nut	(**100**,98.4)	(**100**,97.3)	(98,98.1)	(88.7,89.6)	(99.5,-)	(99.5,98.6)	(98.7,98.4)	(99.5,96.5)	(**100**,**98.7**)	(-,96.8)	(99.1,98.6)	(98.5,98.6)	(98.5,94.9)	(98.6,-)	(99.8,96.2)
	pill	(96.6,97.4)	(96.6,98.2)	(99.0,98.4)	(89.1,97.8)	(98.3,-)	(97.2,98.7)	(99.2,98.5)	(97.8,98.9)	(98.6,**99.1**)	(-,96.3)	(98.6,95.7)	(96.2,**98.9**)	(**99.1**,96.1)	(93.2,-)	(98.2,98.1)
	screw	(98.1,99.4)	(97.0,**99.6**)	(91.6,98.8)	(85.0,98.9)	(**100**,-)	(93.6,98.5)	(90.2,96.5)	(94.8,99.5)	(98.0,**99.6**)	(-,99.3)	(97.6,94.7)	(93.1,98.9)	(99.7,93.4)	(90.0,-)	(93.4,95.1)
	toothbrush	(**100**,98.7)	(99.5,99.1)	(**100**,**99.2**)	(**100**,98.9)	(98.7,-)	(99.9,99.3)	(**100**,94.9)	(98.9,98.2)	(**100**,**99.2**)	(-,98.3)	(91.9,96.3)	(98.8,99.0)	(96.6,98.5)	(95.8,-)	(**100**,98.5)
	transistor	(**100**,96.3)	(96.7,92.5)	(98.6,95.4)	(91.0,80.1)	(98.3,-)	(98.5,89.1)	(95.1,88.0)	(95.0,90.8)	(99.7,95.4)	(-,87.5)	(99.3,**98.2**)	(92.93.3)	(99.3,93.6)	(94.3,-)	(**100**,95.1)
	zipper	(99.4,98.8)	(98.5,98.2)	(98.5,**99.4**)	(99.9,99.0)	(99.0,-)	(**100**,99.1)	(99.8,94.2)	(97.6,98.8)	(99.5,98.1)	(-,97.5)	(99.7,96.4)	(97.1,99.1)	(99.1,96.6)	(95.8,-)	(99.5,98.3)
	Average.Obj	(99.2,98.4)	(97.9,97.8)	(98.0,97.9)	(91.4,93.8)	(98.7,-)	(98.3,97.9)	(96.5,96.0)	(97.8,97.4)	(**99.3**,**98.6**)	(-,97.0)	(98.2,96.0)	(97.0,98.1)	(98.9,95.9)	(96.1,-)	(98.9,97.6)
Total Average		(99.1,98.1)	(98.5,97.8)	(98.6,98.2)	(92.0,93.9)	(98.3,-)	(98.6,97.9)	(97.2,96.3)	(98.4,97.4)	(**99.5**,**98.4**)	(98.0,97.0)	(98.7,95.5)	(97.5,97.7)	(99.2,96.4)	(96.9,-)	(99.2,98.0)

- denotes no result reported. PS: OCR-GAN and SANF do not have the capability to obtain results for the P-AUROC metric.

layers. The convolution neural network are all of kernel size 3x3. The dual-branch network constructed in this paper consists of 6 layers. The training of this network employs the AdamW optimizer [53] with hyperparameters set to $\beta_1 = 0.8, \beta_2 = 0.8, \epsilon = 10^{-8}$, and a learning rate of 0.00005. The learning epochs are set to 200.

4.3. Experimental Results and Analysis

In this section, quantitative and qualitative experiments are performed to compare the proposed model with the latest anomaly detection methods on two commonly used datasets. In addition, ablation experiments are conducted to evaluate the various designs for the dual-branch anomaly detection and localization model. These included assessing the performance of single-branch anomaly detection implementation, analyzing the impact of the dynamic spatial attention module on the network, evaluating the influence of $topK$ anomaly scoring on the model, and examining the experimental effects of the pre-trained feature extractors on the network.

4.3.1. Comparative Experiment on MVTec AD

To show the performance of the proposed model, we conducted experiments to compare it with recent state-of-the-art anomaly detection methods. These include normalizing flow based methods, such as CSFlow [11], CFlow [13], AST [50], and SANF [4], and other methods that are not based on normalizing flow, such as PatchCore [8], RDAD [7], FAIR [25], SSM [46], OCR-GAN [5], DeSTSeg [47], MSFR [49], NSA [48], ReContrast [27] and ADShift [26]. Table 1 shows the results of these compared methods obtained on the MVTec AD dataset. Figure 5 provides visual examples by our proposed method from this dataset.

The results in Table 1 show that the anomaly detection method outperforms existing mainstream methods based on normalizing flow. This is primarily attributed to the incorporation of a new branch in the dual-branch anomaly detection model, which enhances anomaly detection for smaller anomalous regions through pixel-level modeling, leading to improved anomaly detection performance. Specifically, compared to methods such as CSFlow, CFlow, and SANF ,our proposed model offers an increased image-level AUROC by approximately 0.5%, 1.7%, and 2.3%, respectively.

Our model captures global and local information through a dual-branch approach and effectively optimizes the intrinsic process of the normalizing flow, improving its ability to identify multi-scale anomalies in input samples. Therefore, our model achieves the second best image-level anomaly identification performance (in terms of the total average of I-AUROC metric) compared to all baselines on the MVTec AD dataset with rich coexistence of multi-scale anomalies. In addition, in practical applications such as product quality management, the I-AUROC metric for image-level anomaly identification usually has a higher priority than the P-AUROC metric, because picking out anomaly samples from all products is the primary step in quality management of the products.

Furthermore, compared with all flow-based baselines in Table 1 on the object-subset of the MVTec AD dataset, our proposed model achieves the best performance in terms of the I-AUROC metric, and offers the second best performance in terms of the P-AUROC metric. The P-AUROC of our model is only slightly lower than that of CFlow. This may be due to the relatively complex multi-scale feature fusion in CFlow, which is more effective in representing the texture of small objects, as supported by the results of the *pill* and *capsule* category in Table 1. However, on average, the proposed method outperforms all flow-based baselines across the entire MVTec AD dataset in both I-AUROC and P-AUROC metrics.

Compared with non-normalizing flow-based methods, FAIR [25] and ReContrast [27] introduces simulated anomaly data and achieves slightly better P-AUROC metrics than our method on the MVTec AD dataset. However, the simulated data used by FAIR and ReContrast [27] makes these models limited in their generalization to other datasets, such as the Btad dataset. ReContrast [27] intentionally restricts the model from learning to adapt to transformations between augmentations, which consequently limits its ability to generalize across domains [54].

As shown in Table 2, the performance of FAIR and ReContrast [27] in both P-AUROC and I-AUROC metrics is worse than our method on the Btad dataset. Moreover, it takes a larger number of FLOPs (see Section 4.3.3). Furthermore, the proposed method outperforms RDA [7] and

Table 2
The results of I-AUROC [%] and P-AUROC [%] for various methods on the Btad dataset.

	Non-normalizing flow-based methods										Normalizing flow-based methods				
Dataset	PatchCore [8]	RDAD [7]	FAIR [25]	SSM [46]	°OCR-GAN [5]	°DeSTSeg [47]	NSA [48]	MSFR [49]	ReContrast [27]	ADShift [26]	CSFlow [11]	CFlow [13]	AST [50]	°SANF [4]	Ours
Btad01	(97.5, 95.6)	(96.3, **96.6**)	(98.3, 95.6)	-	(95.4, -)	(97.4, 96.0)	(87.8, 80.1)	(94.1,**97.2**)	(97.8,94.0)	(95.1,96.4)	(**99.6**, 91.8)	(96.6, 94.2)	(97.8, 95.3)	(95.2, -)	(97.5, 95.8)
Btad02	(82.6, 95.1)	(86.6, **96.7**)	(76.1, 94.9)	-	(**94.6**, -)	(80.7, 98.1)	(78.8, 74.1)	(84.4,96.5)	(82.1,95.9)	(86.7,96.1)	(85.2, 88.0)	(87.4, 96.4)	(87.9, 96.6)	(78.4, -)	(90.2, 96.6)
Btad03	(**100**, 92.9)	(**100**,99.7)	(99.4, 98.7)	-	(99.6, -)	(99.1, 97.5)	(92.3, 90.2)	(**100**, 97.8)	(99.6,**99.8**)	(99.6,98.7)	(**100**, 97.8)	(99.0, 99.5)	(**100**, 96.3)	(96.7, -)	(**100**, 99.6)
Average	(93.4, 94.5)	(94.3, 97.6)	(91.3, 96.4)	-	(**96.5**, -)	(92.4, 97.5)	(86.3, 81.5)	(93.5, **97.8**)	(93.2,96.6)	(93.8,97.1)	(94.9, 92.5)	(94.3, 96.7)	(95.2, 97.3)	(90.1, -)	(95.9, 97.3)

- denotes no result reported. SSM has no source code released. OCR-GAN and SANF do not have the ability to locate anomalies (i.e. cannot obtain P-AUROC indicators).
° denotes re-running the official source codes with default parameter settings, and take the average of results from three random seed. Bold indicates first, underlined indicates second.

DeSTSeg [47] on the MVTec AD dataset which use the distillation paradigm for anomaly detection. This may be because that this distillation process cannot guarantee the correctness of information transferred from the teacher network to the student network, and thus may degrade the performance of the model. In addition, the proposed method outperforms the reconstruction-based methods SSM [46] and OCR-GAN [5] (as shown in Table 1). The SSM and OCR-GAN methods, as reconstruction-based models, cannot completely prevent abnormal information from being mistakenly reconstructed as normal. This is because the reconstruction-based method has a certain degree of tolerance for sample variation in the testing stage, due to its own generalization ability, even though they are only learned to reconstruct normal sample features during the training stage.

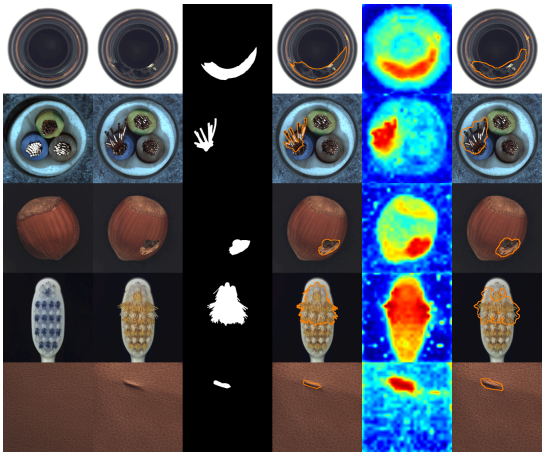


Figure 5: The anomaly localization results for some categories on the MVTec AD dataset. From left to right: a normal sample, an abnormal sample, the ground truth, the actual detection results, the anomaly score map, and the anomaly detection results.

Furthermore, in terms of image anomaly localization, the additional pixel-level branch effectively reduces redundant error information, thereby enhancing the precision of anomaly localization. Compared to existing normalizing flow networks such as CSFlow, and CFlow, the pixel-level AUROC metrics have improved by approximately 2.5%, and 0.3%, respectively. Furthermore, from the anomaly detection visualizations in Figure 5, it is evident that regions with deeper anomaly levels exhibit darker colors. The anomaly score maps effectively differentiate between normal and anomalous patterns in the images, providing a clear delineation between normal and anomalous regions.

4.3.2. Comparative Experiment on Btad

To test the model's generalization ability, this paper also compared the proposed model with other models by performing evaluations on the Btad dataset using image-level AUROC and pixel-level AUROC metrics. The Btad dataset is mainly a dataset related to texture anomalies. Table 2 shows the experimental results. In addition, Figure 7 illustrates the anomaly localization results of our method on the Btad dataset. Compared with other recent methods, our method achieves the second best results, indicating that our proposed model is also competitive for detecting these texture anomalies. In addition, our method outperforms RDAD and DeSTSeg which use distillation paradigm for anomaly detection, while the anomaly span in Btad dataset is relatively large, meaning that very subtle anomalies randomly coexist with large-scale anomalies in the dataset. As a result, the detailed information corresponding to image-level anomaly detection is easily lost in the reverse distillation process.

Moreover, compared to all normalizing flow baselines, our proposed method has achieved better performance on the Btad dataset. Table 3 shows the performance on three datasets by averaging the results. It can be seen that our method achieves the best average performance. From the experimental results, we can see that our method outperforms the flow-based baseline methods and all the other compared methods. Compared with baselines, our dual-branch normalizing flow method not only models features from a global semantic perspective, but also models features at the pixel level, thus providing improved anomaly detection and positioning performance.

4.3.3. Visualization of Anomaly Score Distribution

We have visualized the anomaly scores, achieved by the compared methods, for each sample in the MVTec AD dataset, as shown in Figure 6. It can be seen that, compared to the baselines, our model has provided a larger gap (blue to red bars) between the anomaly scores obtained for all abnormal samples and the scores of normal samples and a smaller fluctuation range (red bars) of the scores for all abnormal samples. This, to some extent, indicates that our proposed model offers better performance for anomaly detection in different application scenarios compared to the baselines.

4.3.4. Computational Complexity

We also compare the complexity of the methods, in the number of FLOPs and Params. The results are shown in Table 4. Compared with CFlow and AST, our method has a similar computational complexity, but it achieves a higher accuracy. CSFlow models the features at three different scales

Table 3
The average results in I-AUROC [%] and P-AUROC [%] for various methods across different datasets.

	Non-normalizing flow-based approach										Normalizing flow-based Mrthods				
Dataset	PatchCore [8]	RDAD [7]	FAIR [25]	SSM [46]	OCR-GAN [5]	DeSTSeg [47]	NSA [48]	MSFR [49]	ReContrast [27]	ADShift [26]	CSFlow [11]	CFlow [13]	AST [50]	SANF [4]	Ours
MVTec AD	(99.1, 98.1)	(98.5, 97.8)	(98.6, 98.2)	(92.0, 93.9)	(98.3, -)	(98.6, 97.9)	(97.2, 96.3)	(98.4, 97.4)	(**99.5**,**98.4**)	(98.0,97.0)	(98.7, 95.5)	(97.5, 97.7)	(99.2, 96.4)	(96.9, -)	(99.2, 98.0)
Btad	(93.4, 94.5)	(94.3, **97.6**)	(91.3, 96.4)	-	(**96.5**, -)	(92.4, 97.5)	(86.3, 81.5)	(93.5, 97.8)	(93.2,96.6)	(93.8,97.1)	(94.9, 92.5)	(94.3, 96.7)	(95.2, 97.3)	(90.1, -)	(95.9, 97.3)
MPDD	(94.1, 97.8)	(92.1, **98.5**)	(73.0, 94.9)	-	(87.1, -)	(93.5, 95.4)	(89.1, 88.3)	(94.7, 97.6)	(95.8,97.9)	(82.3,97.9)	(97.4, 97.9)	(82.6, 96.8)	(87.4, 95.5)	(92.0, -)	(**97.5**, 98.2)
Average	(95.5, 96.8)	(95.0, **98.0**)	(87.6, 96.5)	-	(94.0, -)	(94.9, 97.1)	(90.9, 88.7)	(95.5,97.6)	(96.2,97.6)	(91.3,97.0)	(97.0, 95.3)	(91.5, 97.1)	(93.9, 96.4)	(93.0, -)	(**97.5**, 97.8)

- denotes no result reported. Bold indicates first, underlined indicates second.

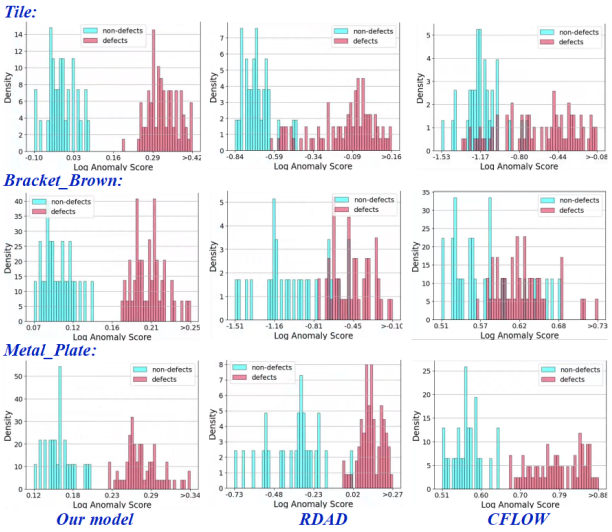


Figure 6: Comparison of anomaly score distribution on MVTec AD. From the leftmost column to the rightmost column are the distributions of abnormal scores for our model, RDAD, and CFLOW, respectively. From these results, it can be seen that our model can more clearly distinguish abnormal samples from normal ones.

simultaneously, which has a complexity almost three times of our model in FLOPs and Params, however, it offers lower detection accuracy as shown in Table 1. RDAD uses reverse distillation for anomaly detection, which only includes simple convolutional layers. However, RDAD's parameters are about twice that of our proposed model, but with almost the same number of FLOPs, due to its use of multi-layer features with pre-trained extractors such as WideResNet [55]. FAIR uses a reconstruction network that only includes some simple convolutional networks, but it has a larger number of FLOPs than our method due to its reconstruction of the original image. ReContrast [27] requires constructing two feature extractors with identical architectures to enable domain adaptation, resulting in significantly higher parameter counts and longer inference times compared to our method, making it less suitable for deployment in industrial settings. Similarly, ADShift [26] performs multiple similar augmentation operations along with feature distribution matching (FDM) to align the training and test data distributions. This process of applying multiple augmentations and complex distribution matching for each test sample introduces considerable inference time overhead.

OCR-GAN, as a synthesis-based model, has larger Params because it needs to construct Encoder-Decoders for multiple frequency channels simultaneously. Another synthesis-based model, DeSTSeg, is designed as a simple convolutional network in its teacher and student networks,

Table 4
The comparison of computional complexity between our method and the state of the art methods.

Algorithm	Methods	FLOPs(G)	Params(M)	FPS
Non-normalizing flow-based methods	RDAD [7]	39.1	150.6	16.3
	FAIR [25]	160.2	69.0	45.8
	SSM [46]	-	-	-
	OCR-GAN [5]	41.8	96.0	25.9
	DeSTSeg [47]	40.4	35.2	65.8
	NSA [48]	-	-	-
	ReContrast [27]	75.3	144.7	33.3
	ADShift [26]	24.0	83.8	25.2
Normalizing flow-based Methods	CSFlow [11]	62.3	275.2	10.6
	CFlow [13]	49.0	81.6	34.2
	AST [50]	49.8	84.6	17.5
	Ours	45.8	82.3	40.5

- denotes no result reported.

Table 5
The AUROC [%] results of pre-trained feature extractors on the MVTec AD dataset.

	I-AUROC	P-AUROC
ResNet18 [29]	97.3	96.4
WideResNet50 [55]	98.5	97.8
DINO [38]	**99.2**	**98.0**
ViT [37]	97.4	96.2

resulting in a relatively small number of parameters compared to other models. However, the calculation process of DeSTSeg requires a large feature scale, which leads to its FLOPs not being significantly different from our model. In addition, for synthesis-based models such as DeSTSeg and OCR-GAN, synthesizing sufficiently representative and reasonable abnormal data corresponding to different practical scenarios not only requires much time to design and consider, but also affects the generalization and practicality of these models. In addition, we compared the detection frame rates of various methods with the same batch size during the testing phase, and our proposed model achieved a frame rate of 40.5 FPS, which can potentially meet the requirements of real-time detection. To sum up, our model has a small computational complexity and offers competitive performance.

4.4. Ablation Studies

To validate the effectiveness of the various modules designed in this paper, we conducted ablation experiments in this section, focusing on the visual feature extractor, dual-branch structure, dynamic spatial attention and the *topK* anomaly scoring method at the end of the module. All ablation experiments were conducted on the MVTec AD dataset.

Table 6
The AUROC [%] results of the ablation experiments on the MVTec AD dataset.

	map_flow	pixel_flow	DSAM	I-AUROC	P-AUROC
Variant1	✓	×	×	98.4	97.1
Variant2	×	✓	×	98.6	97.5
Variant3	✓	✓	×	98.9	97.8
Ours	✓	✓	✓	**99.2**	**98.0**

4.4.1. Visual Feature Extractor

We investigate the relationship of the structure of the pre-trained models for feature extraction to the effectiveness of the extracted features on anomaly detection, by applying these pre-trained models for image classification and self-supervised learning. Specifically, we employed the pre-trained CNN-based image classification models including ResNet18 [29] and WideResNet50 [55], the Transformer-based image classification model ViT [37], and the pre-trained Transformer-based self-supervised model DINO [38] as our visual feature extractors. The experimental results are shown in Table 5. From Table 5, it can be seen that DINO, as a feature extractor, performs the best in the proposed model. The feature extractors such as WideResNet50 and ResNet18, which are pretrained for image classification tasks, perform well for extracting semantic information but not detailed spatial information that is required in our tasks. In comparison, the DINO extractor, pretrained with a self-supervised task to distinguish images at different spatial scales, is more suitable for our proposed model. Among the ResNet and WideResNet backbones, the features utilized are mainly the outputs from the second and third modules. The ViT model primarily utilized the features from the last module, while the DINO models mainly utilized the features from the seventh and eleventh modules. Using the DINO models as the visual feature extractor in our proposed algorithm achieved better anomaly detection results as compared to using other feature extractors, as shown in Table 5.

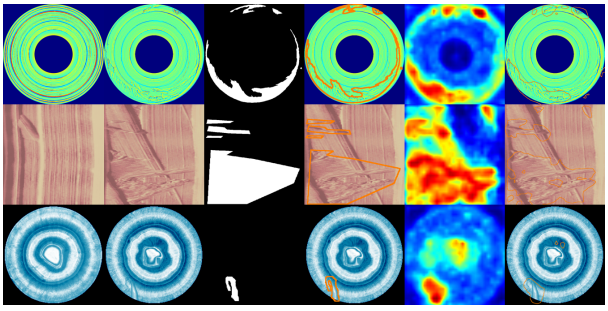


Figure 7: The anomaly localization results on the Btad dataset. From left to right: a normal sample, an abnormal sample, the ground truth, the actual detection results, the anomaly score map, and the anomaly detection results.

4.4.2. Dual-branch Normalizing Flow Structure

To assess the effectiveness of each component of the model, this section conducted ablation experiments on the dual-branch normalizing flow structure, focusing on the dual-branch structure, dynamic spatial attention, and layer normalization. The experimental results are presented in Table 6. As for *Variant1*, only map_flow was used, here map_flow corresponds to the Global Branch. *Variant2* employed only pixel_flow which corresponds to the Local Branch. *Variant3* introduced pixel_flow based on *Variant1*. *Variant4* corresponds our proposed whole model. Additionally, Figure 8 illustrates the anomaly detection results for different branches.

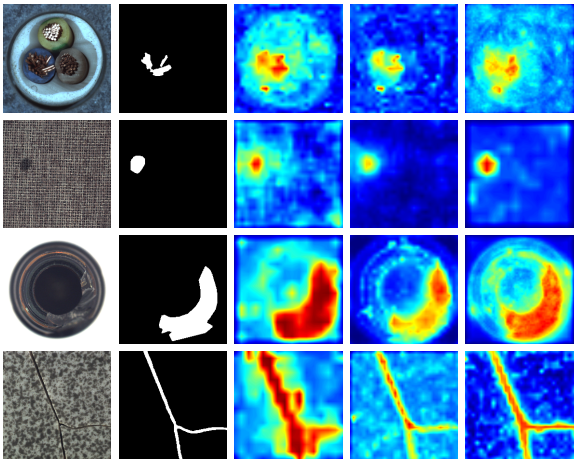


Figure 8: The results of anomaly detection in different branches. From left to right: an abnormal sample, the ground truth, the result of the first branch, the result of the second branch, and the fused result.

From Table 6, it is evident that pixel-level modeling is superior to feature map-level modeling. Assigning high probabilities individually to each pixel yields better results compared to assigning high probabilities to the entire feature map. The integration of these two approaches achieves the best anomaly detection performance. Additionally, the proposed dynamic convolutional spatial attention model further enhances its anomaly detection performance, achieving the best performance as compared to the other variants mentioned above. From Figure 8, it can be observed that, with only the first branch, the anomalous regions located by the anomaly score map tends to include larger areas compared to the true regions. Furthermore, we can see that the global branch focuses on the overall characteristics, but not local details. For example, the third image in the last row of Figure 8 lacks details of anomalies, while the local branch can distinguish the details more clearly. As a result, the feature map shown in the rightmost column, which is obtained by combining the two branches, is more accurate for anomaly detection.

4.4.3. Anomaly Score Calculation

This section conducted ablation studies on the impact of hyperparameter K on the *topK* anomaly scoring calculation, which affects the final detection results. Instead of directly setting a specific value for K, the section implicitly determined

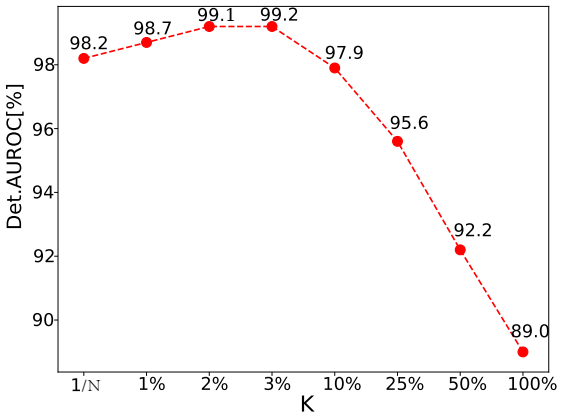


Figure 9: The I-AUCROC results for different values of K in $topK$. N is the total number of pixels in the score map.

K based on the occupancy rate over the entire anomaly map. Specifically, K was varied as $1/N$, 1%, 2%, 3%, 10%, 25%, 50%, and 100%. Here, setting $K = 1/N$ represents taking the maximum value from the score map, where N is the total number of pixels in the score map, while setting $K = 100\%$ represents taking the average value over the entire score map. Figure 9 illustrates the trend of the model's anomaly detection performance with different values of K. It is observed that the model achieves better performance when K is set to 3%. Consequently, the model uses the mean of the top 3% of the scores from the score map as the anomaly score for each sample.

4.4.4. Hyperparameter

We further investigate the impact of the depth of the network. From the results in Table 7, we can see that when the number of flow layers is low, the model achieves suboptimal results. This is likely because the modeling ability of the model is limited by the number of layers, making it difficult to map a complex distribution to a normal distribution. In addition, it can be seen that when the number of model layers is 6, the model achieves the best performance of I-AUROC (99.2%) and P-AUROC (98.0%) both in detection and localization. As the number of layers increases further, the performance of the model starts to decrease, potentially due to the overfitting problem associated with an increased model size.

To study the impact of different optimization parameters, we have added a set of ablation experiments to compare the experimental results of the proposed model at different learning rates. From the experimental results in the MVTec AD dataset shown in Table 8, we can see that when the learning rate is 5e-5, the model achieves the best performance in I-AUROC (99.2%) and P-AUROC (98.0%), both in the detection of anomalies and in the localization of anomalies. When the learning rate drops, the model needs more epochs to train and update. The model may fall to the local optimum due to the small learning rate, making it difficult to continuously improve the model's capabilities. When the learning rate increases, the model update in each step may exceed the effective range of the gradient update, which can make it difficult to converge. However, we observed that our model is less sensitive to the choice of learning rate, compared to the baseline methods. Even with suboptimal parameter configurations, our method can still surpass the performance of some models, such as SANF (I-AUROC 96.9%), CFLOW (I-AUROC 97.5%) and SSM (I-AUROC 92.0%).

Table 7
The AUROC[%] results of different coupling layers on MVTec AD dataset

Layers	I-AUROC	P-AUROC
1	93.4	96.1
2	97.6	97.8
4	98.0	97.7
6	**99.2**	**98.0**
8	98.1	97.9
10	97.5	97.7

Table 8
The AUROC[%] results of different learning rate on MVTec AD dataset

Learning rate	I-AUROC	P-AUROC
1e-5	98.1	97.3
2e-5	98.4	97.5
5e-5	**99.2**	**98.0**
1e-4	98.6	97.6
2e-4	98.2	97.6

5. Conclusion

This paper has presented a dual-branch anomaly detection model based on normalizing flow from multiple perspectives. In the dual-branch structure, the two branches are designed to model image features from the global and local perspectives, for scenarios where the anomalies vary greatly in size. In addition, the dynamic spatial attention nodule (DSAM) is embedded into each branch, by combining dynamic convolution and a newly designed attention mechanism, which can capture spatial features of various complexity in the input image. Experimental results on the well-known public datasets, such as MVTec AD and Btad, demonstrate the advantages of the proposed dual-branch normalizing flow network over the existing flow-based baselines in detection performance, and over non-flow based methods in terms of the number of parameters and FLOPs. In addition, the proposed method has better generalization ability than existing synthesis-based methods. In the future, we will further study the robustness of anomaly detection in various complex environments in engineering applications. When given a certain edge situation in a practical application scenario, we can augment the number of corresponding high distinguishable normal samples with needles to increase adaptive training.

Acknowledgements:

This work was supported by National Natural Science Foundation of China project 62371324. This work was also supported by the 2035 Innovation Pilot Program of Sichuan University, China.

References

[1] Guoliang Liu, Shiyong Lan, Ting Zhang, Weikang Huang, and Wenwu Wang. Sagan: skip-attention gan for anomaly detection. In 2021 IEEE International Conference on Image Processing, pages 2468–2472. IEEE, 2021.

[2] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9592–9600, 2019.

[3] Tiange Xiang, Yixiao Zhang, Yongyi Lu, Alan L Yuille, Chaoyi Zhang, Weidong Cai, and Zongwei Zhou. Squid: Deep feature inpainting for unsupervised anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 23890–23901, 2023.

[4] Wei Ma, Yao Li, Shiyong Lan, Wenwu Wang, Weikang Huang, and Wujiang Zhu. Semantic-aware normalizing flow with feature fusion for image anomaly detection. Neurocomputing, 590:127728, 2024.

[5] Yufei Liang, Jiangning Zhang, Shiwei Zhao, Runze Wu, Yong Liu, and Shuwen Pan. Omni-frequency channel-selection representations for unsupervised anomaly detection. IEEE Transactions on Image Processing, 32:4327–4340, 2023.

[6] Yixuan Zhou, Xing Xu, Jingkuan Song, Fumin Shen, and Heng Tao Shen. Msflow: Multiscale flow-based framework for unsupervised anomaly detection. IEEE Transactions on Neural Networks and Learning Systems, 2024.

[7] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9737–9746, 2022.

[8] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14298–14308, 2022.

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. International journal of Computer Vision, 115(3):211–252, 2015.

[10] Wei Ma, Shiyong Lan, Weikang Huang, Wenwu Wang, Hongyu Yang, Yitong Ma, and Yongjie Ma. A semantics-aware normalizing flow model for anomaly detection. In 2023 IEEE International Conference on Multimedia and Expo (ICME), pages 2207–2212, 2023.

[11] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Fully convolutional cross-scale-flows for image-based defect detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 1088–1097, 2022.

[12] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: A patch distribution modeling framework for anomaly detection and localization. In Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, Proceedings, Part IV, page 475–489, 2021.

[13] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. 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, pages 98–107, 2022.

[14] Chuang Yang, Mulin Chen, Yuan Yuan, and Qi Wang. Zoom text detector. IEEE Transactions on Neural Networks and Learning Systems, 35(11):15745–15757, 2024.

[15] Chuang Yang, Mulin Chen, Zhitong Xiong, Yuan Yuan, and Qi Wang. CM-Net: Concentric mask based arbitrary-shaped text detection. IEEE Transactions on Image Processing, 31:2864–2877, 2022.

[16] Yingming Li, Ming Yang, and Zhongfei Zhang. A survey of multi-view representation learning. IEEE Transactions on Knowledge and Data Engineering, 31(10):1863–1883, 2019.

[17] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. Vt-adl: A vision transformer network for image anomaly detection and localization. In 2021 IEEE 30th International Symposium on Industrial Electronics, pages 01–06, 2021.

[18] Andrew Ng et al. Sparse autoencoder. CS294A Lecture notes, 72(2011):1–19, 2011.

[19] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. arXiv preprint arXiv:1807.02011, 2018.

[20] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. Special lecture on IE, 2(1):1–18, 2015.

[21] Bowei Pu, Shiyong Lan, Wenwu Wang, Caiying Yang, Wei Pan, Hongyu Yang, and Wei Ma. Gannext: A new convolutional gan for anomaly detection. In International Conference on Artificial Neural Networks, pages 39–49, 2023.

[22] Caiyin Yang, Shiyong Lan, Weikang Huang, Wenwu Wang, Guoliang Liu, Hongyu Yang, Wei Ma, and Piaoyang Li. A transformer-based gan for anomaly detection. In Artificial Neural Networks and Machine Learning–ICANN 2022: 31st International Conference on Artificial Neural Networks, Bristol, UK, September 6–9, 2022, Proceedings, Part II, pages 345–357, 2022.

[23] Yufei Liang, Jiangning Zhang, Shiwei Zhao, Runze Wu, Yong Liu, and Shuwen Pan. Omni-frequency channel-selection representations for unsupervised anomaly detection. arXiv preprint arXiv:2203.00259, 2022.

[24] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9664–9674, 2021.

[25] Tongkun Liu, Bing Li, Xiao Du, Bingke Jiang, Leqi Geng, Feiyang Wang, and Zhuo Zhao. Fair: Frequency-aware image restoration for industrial visual anomaly detection. arXiv preprint arXiv:2309.07068, 2023.

[26] Tri Cao, Jiawen Zhu, and Guansong Pang. Anomaly detection under distribution shift. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6511–6523, 2023.

[27] Jia Guo, Shuai Lu, Lize Jia, Weihang Zhang, and Huiqi Li. Recontrast: Domain-specific anomaly detection via contrastive reconstruction. Advances in Neural Information Processing Systems, 36:10721–10740, 2023.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 2012.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[31] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H Rohban, and Hamid R Rabiee. Multiresolution knowledge distillation for anomaly detection. In Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, pages 14902–14912, 2021.

[32] Yu Zhou, Xiaomin Liang, Wei Zhang, Linrang Zhang, and Xing Song. Vae-based deep svdd for anomaly detection. Neurocomputing, 453:131–140, 2021.

[33] Jihun Yi and Sungroh Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In Proceedings of the Asian Conference on Computer Vision, 2020.

[34] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. Advances in

Neural Information Processing Systems, 33:20578–20589, 2020.
[35] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 1907–1916, 2021.
[36] Wei Ma, Shiyong Lan, Weikang Huang, Yitong Ma, Hongyu Yang, Wei Pan, and Yilin Zheng. Flow-based one-class anomaly detection with multi-frequency feature fusion. In 2023 IEEE International Conference on Image Processing, pages 3474–3478, 2023.
[37] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.
[38] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9650–9660, 2021.
[39] Yajie Cui, Zhaoxiang Liu, and Shiguo Lian. Patch-wise auto-encoder for visual anomaly detection. arXiv preprint arXiv:2308.00429, 2023.
[40] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. arXiv preprint arXiv:1605.08803, 2016.
[41] Meng Lou, Hong-Yu Zhou, Sibei Yang, and Yizhou Yu. Transxnet: Learning both global and local dynamics with a dual dynamic token mixer for visual recognition. arXiv preprint arXiv:2310.19380, 2023.
[42] Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. On the connection between local attention and dynamic depth-wise convolution. arXiv preprint arXiv:2106.04263, 2021.
[43] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision, pages 3–19, 2018.
[44] Chao Li, Aojun Zhou, and Anbang Yao. Omni-dimensional dynamic convolution. In Proceedings of the International Conference on Learning Representations, 2022.
[45] Jimmy Lei Ba. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.
[46] Chaoqin Huang, Qinwei Xu, Yanfeng Wang, Yu Wang, and Ya Zhang. Self-supervised masking for unsupervised anomaly detection and localization. IEEE Transactions on Multimedia, 25:4426–4438, 2023.
[47] Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. Destseg: Segmentation guided denoising student-teacher for anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3914–3923, 2023.
[48] Hannah M Schlüter, Jeremy Tan, Benjamin Hou, and Bernhard Kainz. Natural synthetic anomalies for self-supervised anomaly detection and localization. In Proceedings of the European Conference on Computer Vision, pages 474–489, 2022.
[49] Ehtesham Iqbal, Samee Ullah Khan, Sajid Javed, Brain Moyo, Yahya Zweiri, and Yusra Abdulrahman. Multi-scale feature reconstruction network for industrial anomaly detection. Knowledge-Based Systems, 305:112650, 2024.
[50] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Asymmetric student-teacher networks for industrial anomaly detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 2592–2602, 2023.
[51] Stepan Jezek, Martin Jonak, Radim Burget, Pavel Dvorak, and Milos Skotak. Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions. In 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, pages 66–71, 2021.
[52] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061, 2020.
[53] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. In Proceedings of the International Conference on Learning Representations, 2018.
[54] Jia Guo, Shuai Lu, Weihang Zhang, and Huiqi Li. Dinomaly: The less is more philosophy in multi-class unsupervised anomaly detection. arXiv preprint arXiv:2405.14325, 2024.
[55] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. arXiv preprint arXiv:1605.07146, 2016.