Semantic-Aware Normalizing Flow with Feature Fusion for Image Anomaly Detection

Wei Ma[#,a,1], Yao Li[#,a,1], Shiyong Lan [a,*,1], Wenwu Wang[b,2], Weikang Huang[a,1] and Wujiang Zhu[a,1]

[a] *College of Computer Science, Sichuan University, Chengdu, 610065, China*

[b] *Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, U.K.*

ARTICLE INFO

Keywords:
Semantic anomaly detection
Texture anomaly detection
Normalizing flow
Self-attention
Feature fusion

ABSTRACT

In the field of computer vision, anomaly detection is a binary classification task used to identify exceptional instances within image datasets. Typically, it can be divided into two aspects: texture defect detection and semantic anomaly detection. Existing methods often use pre-trained feature extractors to singly capture semantic or spatial features of images, and then employ different classifiers to handle these two types of anomaly detection tasks. However, these methods fail to fully utilize the synergistic relationship between these two types of features, resulting in algorithms that excel in one type of anomaly detection task but perform poorly in the other type. Therefore, we propose a novel approach that successfully combines these two types of features into a normalizing flow learning module to address both types of anomaly detection tasks. Specifically, we first adopt a pre-trained Vision Transformer (ViT) model to capture both texture and semantic features of input images. Subsequently, using the semantic features as input, we design a novel normalizing flow model to fit the semantic distribution of normal data. In addition, we introduce a feature fusion module based on attention mechanisms to integrate the most relevant texture and semantic information between these two types of features, significantly enhancing the model's ability to simultaneously represent the spatial texture and semantic features of the input image. Finally, We conduct comprehensive experiments on well-known semantic and texture anomaly detection datasets, namely Cifar10 and MVTec, to evaluate the performance of our proposed method. The results demonstrate that our model achieves outstanding performance in both semantic and texture anomaly detection tasks, particularly achieving state-of-the-art results in semantic anomaly detection.

1. Introduction

Detecting anomalous patterns in data holds significant importance in both science and industry, representing a crucial task in visual image understanding. This technique has found widespread applications in various domains, including but not limited to quality monitoring of industrial components [1, 2], novelty detection [3–5], human health monitoring [6, 7], and video surveillance [8, 9]. In general, detecting anomalous patterns in data from an open-world scenario can be approximately divided into five related subtopics: anomaly detection (AD), novelty detection (ND), open set recognition (OSR), out-of-distribution (OOD) detection, and outlier detection (OD), as discussed in survey papers [10, 11]. Furthermore, visual anomaly detection, as one of the widely studied anomaly detection (AD) , can be primarily categorized into two types: local texture defect detection, such as in the case of MVTec [1], and semantic anomaly detection (usually also dubbed image-level anomaly detection or one-class classification), as exemplified by Cifar10 [3]. The former focuses on identifying local texture anomalies within images, while the latter places greater emphasis on discerning the overall semantic differences among different images.

As a particular application example in industry, detecting anomalies from image samples becomes increasingly important for controlling the quality of industrial products. However, the performance of existing visual anomaly detection models [12–17] does not meet the requirement for accurate semantic anomaly detection. To this end, we focus on image-level anomaly detection, which can enable substandard products to be identified from a large number of test samples in industrial manufacturing.

In real-world applications, however, data from an anomaly detection task often exhibit uncertainty and imbalance in the distribution between the labelled abnormal and normal data [12]. Specifically, it is often difficult for us to precisely define anomalies within a dataset for any given anomaly detection task, and the number of normal samples typically far exceeds that of anomalous samples. Consequently, when traditional supervised binary classifiers are employed for anomaly detection tasks, the uncertainty and imbalance of the anomaly data can limit the ability of a model in correctly identifying anomalies. To address this issue, current visual anomaly detection methods commonly utilize unsupervised or self-supervised methods [3, 14, 17]. These methods solely utilize normal data to learn the distribution of normal data. Any data that deviate from this learned distribution are then classified as anomalies.

[*] Corresponding author: lanshiyong@scu.edu.cn. [#] Equal contribution.

[1] W. Ma, Y. Li, S. Lan, W. Huang and W. Zhu are with the College of Computer Science, Sichuan University, China. E-mail: {mawei12138, liyao518}@stu.scu.edu.cn, lanshiyong@scu.edu.cn, and {wkhuang, zhuwujiang}@stu.scu.edu.cn, respectively.

[2] W. Wang is with the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, U.K. E-mail: w.wang@surrey.ac.uk.

In addition, image-level anomaly detection is one of the most challenging tasks [1, 12], as detection failures can range from subtle defects (such as thin scratches), to partial structural changes (such as missing components), and even to image-level semantic differences (such as novel class objects). Existing state-of-the-art methods have attempted to learn the feature distribution of normal samples via generative adversarial networks (GANs) [12, 13, 18], normalizing-flows [16, 19] or other unsupervised adaptation methods [14, 15]. Nevertheless, these existing methods have not fully explored and utilized the fusion of semantic and spatial features from visual image samples to improve the performance of visual anomaly detection.

To address the aforementioned issues, we propose a new model based on the normalizing flow that can be used for semantic anomaly detection while preserving the detection performance for texture anomaly detection tasks as much as possible. Our model consists of two main components: a pre-trained feature extractor for extracting image features and a normalizing flow model for anomaly detection. For the pre-trained feature extractor, we choose a model based on the Vision Transformer (ViT) [20]. We select this model because it not only provides rich texture features (patch tokens) similar to traditional convolutional neural networks (CNNs) but also captures the semantic features (class token) of the images by leveraging its self-attention mechanism to aggregate these texture features. Unlike existing normalizing flow-based anomaly detection methods that heavily rely on spatial texture features, our method focuses on leveraging semantic features (class token) to learn the distribution of normal data for semantic anomaly detection. In other words, our method mainly targets one-class classification (OCC), treating all normal samples as one class and abnormal samples as another class. It is different from out-of-distribution (OOD) scenarios, in which multiple classes of normal data have their own distributions, and abnormal data does not follow the distributions of any of these normal classes.

Furthermore, we recognize the importance of incorporating spatial texture features when dealing with texture anomaly detection tasks. We design various feature fusion strategies to enhance the capability of our model to perceive spatial information. Firstly, we employ global average pooling to aggregate spatial texture features, which are then concatenated with the semantic features extracted by the pre-trained feature extractor. Additionally, we utilize class attention techniques and learnable queries [21] to incorporate fresh semantic information from texture features and fuse it with the original semantic information (class token). Then, we replace the learnable query with pre-trained semantic features as the query, transforming the attention aggregation process into one guided by the original semantic features.

We conduct extensive evaluations on the popular semantic anomaly detection dataset Cifar10 and texture anomaly detection dataset MVTec. We achieve an Area Under the Receiver Operating Characteristic Curve (AUROC) at 99.3% for the Cifar10 dataset which represents approximately the state-of-the-art in this field. Furthermore, our model achieves competitive results in texture anomaly detection on the MVTec dataset. Finally, we compute the average anomaly detection performance of our model for both datasets. In this regard, our model outperforms other existing methods.

Our primary contributions can be summarized as follows:

- We propose a novel anomaly detection architecture based on the normalizing flow model. This architecture effectively maps normal data to a Gaussian distribution, leveraging the explicit utilization of semantic attributes as learning objectives during the process of fitting the distribution of normal data.

- In order to enhance the performance of anomaly detection, we propose a novel feature fusion module incorporated into each layer of the normalizing flow model. This module facilitates the integration of texture features and semantic features, leading to improved fitting of the distribution function to the normal data.

- Through extensive experiments on well-known datasets for semantic anomaly detection, we demonstrate that our method surpasses state-of-the-art baselines. The results validate the superior performance and effectiveness of our approach.

This paper serves as a comprehensive extension of its conference version [22][1]. Specifically, we investigate the detection performance of our proposed model by incorporating more pre-trained extractors derived from various pre-training tasks such as image classification and some proxy tasks. Furthermore, we introduce various feature fusion methods, such as global average pooling, attention mechanisms, and learnable query vectors. Through ablation experiments, we thoroughly compare the experimental results of our model under different pre-trained extractors and feature fusion methods. This rigorous evaluation enables us to determine the most suitable pre-trained extractors and fusion methods for our model. Additionally, we conduct comparisons with more state-of-the-art (SOTA) anomaly detection methods. The results show that our method achieves the SOTA results on semantic anomaly detection tasks.

2. Related Work

The current visual anomaly detection methods can be primarily categorized into two types. The first type focuses on studying the original image data, including techniques like image reconstruction [12, 18] and some self-supervised methods utilizing the original images for various proxy tasks [14, 15] to learn the distribution of images. The second type involves combining pre-trained models with other anomaly detection methods. This includes combining a pre-trained feature extractor with traditional unsupervised clustering methods [3, 4] for anomaly detection, as well as methods that

[1] Code is available at https://github.com/SYLan2019/SANF-AD

leverage normalizing flows along with pre-trained feature extractors for anomaly detection [16, 19].

Reconstruction-based methods are the most commonly used algorithms for visual anomaly detection tasks [12, 13, 23]. These methods typically employ various GANs [24] to reconstruct instances of normal data and learn their feature distribution during the training phase. Subsequently, during the testing phase, the trained model is used to simultaneously reconstruct both normal and anomalous data, with the reconstruction error serving as a criterion for detecting anomalies. However, the training process of GAN involves an adversarial interplay between the generator and discriminator, rendering this method susceptible to some problems such as mode collapse, vanishing gradients, and exploding gradients [24]. Therefore, it is usually difficult to build a well-performed anomaly detection model by only using GAN to learn the distribution of the raw image data. In recent years, the self-attention mechanism has been used to improve the GAN models for anomaly detection, such as AnoTrans [18]. However, the self-attention-based GAN models are usually limited in capturing fine-grained details. In addition, simulated prior anomaly patches are used in [25] to learn a joint representation of an anomaly image and its corresponding anomaly-free reconstruction. Nevertheless, there are differences between the simulated anomalies and the actual anomalies, which may limit the performance of this algorithm in practical applications.

Self-supervised learning methods [14, 15] have also emerged as a promising approach for anomaly detection. These methods often utilize diverse proxy tasks to learn the distribution of normal data. These tasks include training classifiers to recognize artificially rotated images [26] and employing contrastive learning to increase the representation distance between normal data and artificially generated anomalous data via image augmentation operations [14]. However, a common challenge faced by these methods is the tendency towards overfitting due to the limited diversity of artificially constructed data. To address this issue, Reiss et al. [3] introduced feature extractors pre-trained on large-scale datasets such as ImageNet [27], leveraging the rich prior information embedded in these pre-trained models, and clustered these features with techniques such as K-Nearest Neighbors (KNN) and Gaussian Mixture Model (GMM), leading to improved performance as compared with previous self-supervised methods. Building upon this foundation, Cohen and Avidan [28] introduced a novel knowledge distillation-based approach for anomaly detection, where only the representation of normal data from the pre-trained model was distilled to the student model. This approach enhanced the student model's focus on the normal data and enlarged the representation distance of anomalous data between the student and teacher model, effectively addressing the anomaly detection task. In addition, self-supervised learning is used in [29] to represent the intra-class variation at the patch level, which improves the performance of anomaly detection, but can be difficult to represent patches at different scales. To sum up, the above techniques primarily consider semantic features derived from pre-trained feature extractors and overlook the remaining spatial texture information, limiting their effectiveness in texture anomaly detection.

Recently, Rudolph et al. [16] introduced the normalizing flow in the field of anomaly detection. They combined spatial texture features extracted by a pre-trained feature extractor with a normalizing flow model to learn the feature distribution of normal data, achieving state-of-the-art detection results in texture anomaly detection tasks. However, as pointed out in [30], the models based on normalizing flow tend to learn feature information of images at the spatial texture level, which often lead to poor performance when applied to semantic anomaly detection. Therefore, existing approaches based on the normalizing flow have inherent limitations in semantic anomaly detection and are prone to model instability when dealing with complex texture features in semantic datasets [30].

3. Proposed Method

We introduce an unsupervised method based on normalizing flow to learn the distribution of normal data during the training phase. In the testing phase, any data that deviates from the learned distribution in the training phase is considered as an anomaly, and the anomaly score is determined based on the model's loss. As highlighted in [3], an effective anomaly detection algorithm requires both accurate feature representation and a high-performance classifier. Thus, our model consists of two components: a pre-trained feature extractor for obtaining image features and a normalizing flow model for anomaly detection. In Section 3.1, we elaborate on how we select and utilize the pre-trained feature extractor. In Section 3.2, we introduce our proposed semantic-aware normalizing flow model. In Section 3.3, we introduce our proposed feature fusion method about how we integrate and combine texture and semantic features in each layer of the model. Finally, in Section 3.4, we provide insights into the optimization method utilized to train our model.

3.1. Feature Extractor

In the field of anomaly detection, current methods [3, 17] often rely on pre-trained feature extractors based on Convolutional Neural Networks (CNNs). These approaches utilize pre-trained CNN networks to extract feature maps, followed by global average pooling [3, 17, 27] to derive semantic features from the images. However, unlike conventional CNN-based techniques [27], the attention-based Vision Transformer (ViT) [20] not only captures spatial features (i.e., feature maps) but also leverage the class token to get the semantic features by aggregating all patch tokens within an image, eliminating the need for additional pooling operations to obtain semantic features [20]. Hence, we opt to employ a feature extractor based on ViT architecture to effectively capture the semantic as well as spatial texture features within the images. This allows us to detect anomalies in images from both the overall semantic and texture

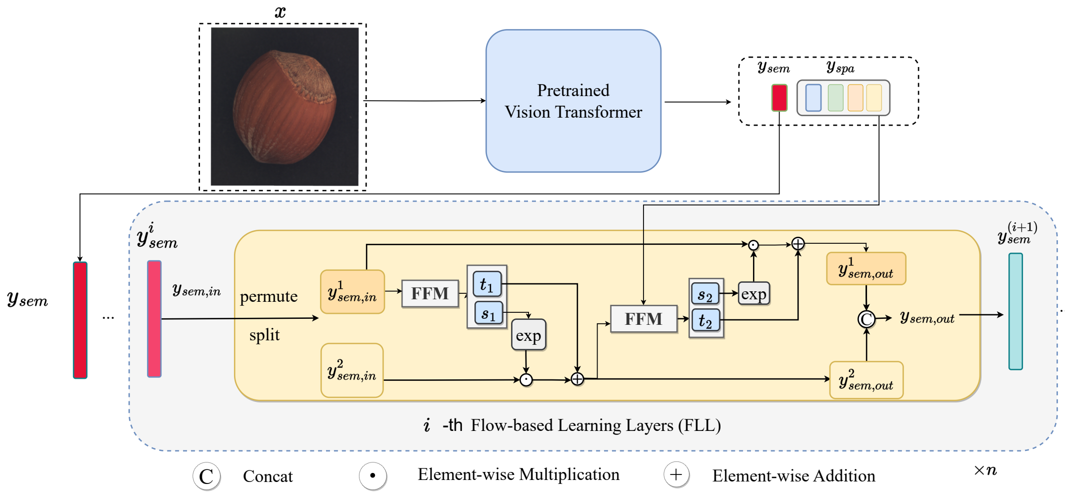

Figure 1: Overview of the proposed method: we use the pre-trained ViT to extract semantic features (class token) and spatial features (patch tokens) of image X . Then we use n stacked flow-based learning layers (FLL) to transform y_{sem} into the latent variable z . The details for the feature fusion module (FFM) are shown in Figure 2, which is used to fuse y_{sem} and y_{spa}.

perspectives. In the specific implementation, we will utilize the entire output of the last module of ViT.

Assuming the input image is denoted as X, the semantic features and spatial texture features of X can be extracted from the pre-trained ViT as

$$(y_{sem}, y_{spa}) = ViT(X) \tag{1}$$

where $y_{sem} \in R^d$ represents the class token which is used to signify semantic features, $y_{spa} \in R^{N \times d}$ denotes the patch tokens which represent spatial texture features, d represents the feature dimension, and N is the number of patch tokens within the input image for ViT.

3.2. Semantic-aware Normalizing Flow

Our proposed approach is built upon the normalizing flow method to estimate the distribution of semantic features of images. In contrast to existing normalizing flow methods that primarily emphasize spatial texture details for texture anomaly detection, our approach prioritizes the overall semantic information in the images. In other words, rather than estimating the distribution of the entire spatial texture details (patch tokens), we directly estimate the distribution of the semantic features, specifically the class token, obtained from the output of the pre-trained feature extractor. Furthermore, to enhance our model's learning of the distribution of normal data, we draw inspiration from [21], which proposes that semantic attributes can be further consolidated through spatial features. Consequently, as shown in Figure 1, we introduce a feature fusion module (FFM) within each flow-based learning layer (FLL) to seamlessly integrate additional spatial feature information. For detailed integration, please refer to Section 3.3.

As depicted in Figure 1, our normalizing flow model $F_{flow} = f_1 \circ f_2 \circ ... \circ f_K$ consists of K individual FLLs, represented as f_i, where the input is denoted as $y_{sem,in}$, representing the semantic feature, and the output is denoted as $y_{sem,out}$. The process begins by randomly permuting $y_{sem,in} \in R^d$ along the channel dimension, followed by an equal division into two vectors, namely $y^1_{sem,in} \in R^{\frac{d}{2}}$ and $y^2_{sem,in} \in R^{\frac{d}{2}}$. This can be summarized as follows:

$$[y^1_{sem,in}, y^2_{sem,in}] = split(shuffle(y_{sem,in})) \tag{2}$$

where $shuffle(\cdot)$ randomly permutes a vector along the channel dimension and $split(\cdot)$ evenly divides a vector into two along the channel dimension. Subsequently, we feed $y^1_{sem,in}$ and $y^2_{sem,in}$ into the FFM, which generates the scaling and translation parameters $[s_1, t_1]$ and $[s_2.t_2]$, respectively. These parameters are then applied to their corresponding inputs, $y^1_{sem,in}$ and $y^2_{sem,in}$, to compute the outputs $[y^1_{sem,out}, y^2_{sem,out}]$. According to [31], for simplicity in loss computation and to satisfy the affinity property, we apply the exponential function to the scaling parameters s_1 and s_2 as follows:

$$[s_1, t_1] = FFM(y^1_{sem,in}, y_{spa}) \tag{3}$$

$$y^2_{sem,out} = y^2_{sem,in} \odot e^{s_1} + t_1 \tag{4}$$

$$[s_2, t_2] = FFM(y^2_{sem,out}, y_{spa}) \tag{5}$$

$$y^1_{sem,out} = y^1_{sem,in} \odot e^{s_2} + t_2 \tag{6}$$

$$y_{sem,out} = Concat(y^1_{sem,out}, y^2_{sem,out}) \tag{7}$$

where $\odot$ denotes element-wise multiplication. Finally, we concatenate $y^1_{sem,out}$ and $y^2_{sem,out}$ along the channel dimension to obtain the output $y_{sem,out}$:

$$y_{sem,out} = Concat(y^1_{sem,out}, y^2_{sem,out}) \tag{8}$$

Similar to the approach proposed in [31], we also employ soft-clamping [32] to maintain model stability:

$$\sigma_\alpha(s) = \frac{2\alpha}{\pi} arctan \frac{s}{\alpha} \tag{9}$$

where α is the hyperparameter of the soft-clamping, and s represents the scaling parameter produced by the FFM.

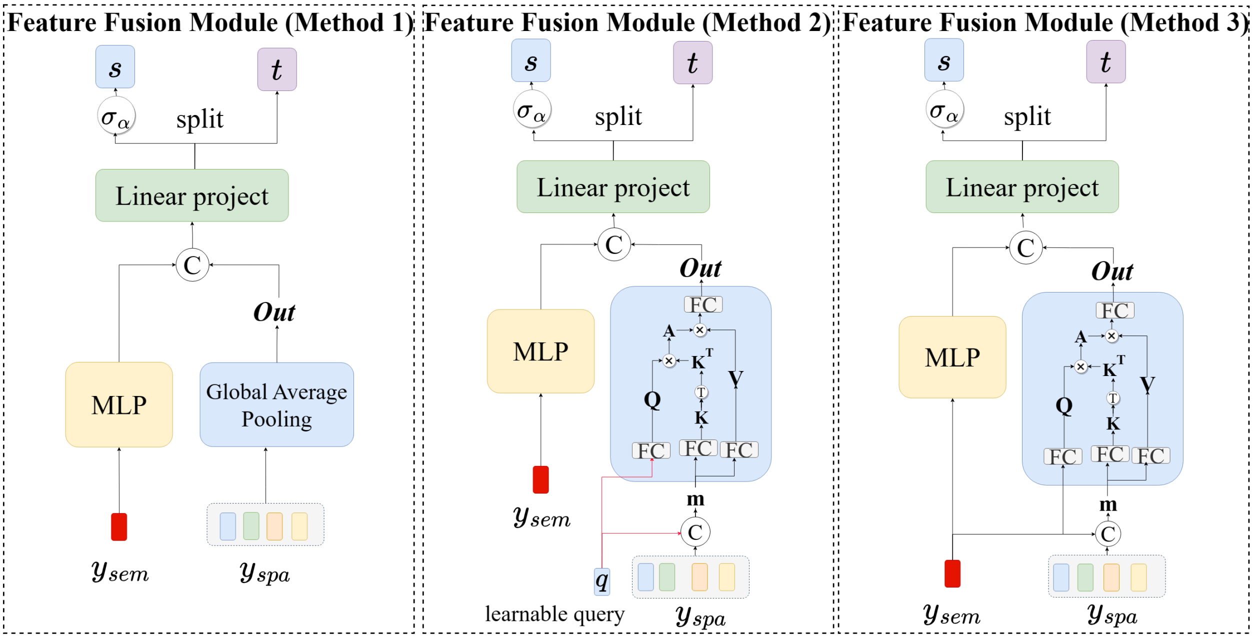

Figure 2: We designed fusion methods that combine pooling and attention techniques to integrate semtantic feature y_{sem} and spatial texture features y_{spa}.

3.3. Feature Fusion Methods

To enhance the fitting of our model to normal data, we have incorporated spatial texture features of images into the perception process. Figure 2 illustrates our approach, which involves designing multiple feature fusion methods. We assume that $y_{sem} \in R^d$ and $y_{spa} \in R^{N \times d}$, and $Out \in R^d$ that represents the output feature after fusion, where d represents the feature dimension, and N is the number of patch tokens in the input image.

Fusion Method 1: Similar to the approach described in reference [3], we employ a traditional global average pooling layer to extract the semantic information of y_{spa}, as shown below:

$$Out = GAP(y_{spa}) \tag{10}$$

where GAP represents the global average pooling layer.

Fusion Method 2: Inspired by the class attention and self-attention in [21], we aggregate the spatial features y_{spa} through learnable query and self-attention. Initially, we randomly initialize a feature vector $q \in R^d$ as the query for self-attention. We use a fully connected layer to generate Q and concatenate q with y_{spa} to obtain $m \in R^{(N+1) \times d}$. Subsequently, we use two fully connected layers to get $K \in R^{(N+1) \times d}$ and $V \in R^{(N+1) \times d}$ from m as follows:

$$m = Concat(q, y_{spa}) \tag{11}$$

$$Q = W_q q + b_q \tag{12}$$

$$K = W_k m + b_k \tag{13}$$

$$V = W_v m + b_v \tag{14}$$

where $W_q, W_k, W_v \in R^{d \times d}$ represent the parameters of the fully connected layers responsible for generating the embedding vectors in each attention module, while $b_q, b_k, b_v \in R^d$ are the corresponding biases. The variable d represents the size of the embedding vectors. Next, we calculate the attention score matrix as follows:

$$A = Softmax(QK^T)/\sqrt{d} \tag{15}$$

Afterward, the vector V, weighted by the matrix A, is input into a fully connected (FC) layer to generate $Out \in R^d$:

$$Out = W_o AV + b_o \tag{16}$$

where $W_o \in R^{d \times d}$ and $b_o \in R^d$ denote the parameters of the FC layer.

Fusion Method 3: In this fusion method, we substitute the learnable query with the pre-trained semantic feature (class token) to serve as the guiding query. The only difference from Method 2 is that when generating Q, we replace the initialized q with y_{sem}, and m is obtained by concatenating y_{sem} with y_{spa}, as shown below:

$$m = Concat(y_{sem}, y_{spa}) \tag{17}$$

$$Q = W_q y_{sem} + b_q \tag{18}$$

The calculations for K, V, A, and Out remain the same as in Method 2, which is the method presented in the conference version [22]. Finally, for y_{sem}, we encode it further using a Multilayer Perceptron (MLP) module. The output of the MLP, along with the Out obtained from the fusion methods, is concatenated and fed into the final linear projection layer (FC layer) to produce the fusion results, represented by $s \in R^d$ and $t \in R^d$. Specifically, this can be expressed as follows:

$$[s, t] = split(FC(Concat(MLP(y_{sem}), out_{CA}))) \tag{19}$$

3.4. Loss Function

We have designed a model based on normalizing flow to map the semantic feature y_{sem} of an image to a latent space z, where both z and y_{sem} have the same dimensions. Our primary objective is to optimize the model's parameters to maximize the likelihood $P_Y(y_{sem})$ of normal data. To achieve this, we define the likelihood of feature y_{sem} based on reference [33] as follows:

$$p_Y(y_{sem}) = p_Z(z)\left|\ det\frac{\partial z}{\partial y_{sem}}\ \right| \tag{20}$$

where $z = F_{flow}(y_{sem})$, and $F_{flow}\ :\ Y \rightarrow Z$ represents our proposed model based on normalizing flow. We assume $y_{sem} \sim p_Y(y_{sem})$ and $z \sim p_Z(z)$. As in [34], the normalizing flow method exhibits a bijective property. This property allows us to learn the distribution of y_{sem} by leveraging the distribution of the latent variable z. Consequently, maximizing likelihood $p_Y(y_{sem})$ is equivalent to maximizing the likelihood $p_Z(z)$. Finally, assuming $z \sim \mathcal{N}(0, I)$ and minimizing the negative log-likelihood $-logp_Y(y_{sem})$, we optimize our model by defining a loss function as follows:

$$\begin{aligned} logp_Y(y_{sem}) &= logp_Z(z) + log\left|\ det\frac{\partial z}{\partial y_{sem}}\ \right| \\ &= -\frac{\|z\|_2^2}{2} + log\frac{1}{\sqrt{2\pi}} + log\left|\ det\frac{\partial z}{\partial y_{sem}}\ \right| \end{aligned} \tag{21}$$

$$\mathcal{L}oss = \frac{\|z\|_2^2}{2} - log\left|det\frac{\partial z}{\partial y_{sem}}\right|) \tag{22}$$

where $\|\cdot\|_2^2$ denotes the L_2 norm, and $\left|det\frac{\partial z}{\partial y_{sem}}\right|$ represents the absolute determinant of the Jacobian matrix, which signifies the volume change from y_{sem} to z. For further detailed information, please refer to reference [33].

During training, we aim to maximize the likelihood of z, which is transformed from y_{sem}. Consequently, during the inference phase, the likelihood of features originating from normal data exceeds that of features originating from abnormal data. Therefore, we adopt the negative log likelihood of the features as the anomaly score.

4. Experiments and Results

In order to evaluate the effectiveness of our proposed model for semantic and texture anomaly detection, we conducted a comprehensive set of experiments as follows:

- Comparative Experiments with State-of-the-Art Visual Anomaly Detection Algorithms: In this experiment, we compared our method against current mainstream visual anomaly detection algorithms to demonstrate our model's competitiveness and effectiveness.

- Feature Fusion Ablation Experiment: This experiment aims to demonstrate the importance of fusing spatial texture features and compare the performance of different feature fusion methods.

- Performance of Different Pre-trained Visual Feature Extractors (ViTs): In this experiment, we assessed the adaptability of our model to various pre-trained ViTs obtained from different tasks, including image classification [20], self-supervised learning (DINO [35] and DINOV2 [36]), and multimodal tasks (CLIP [37]).

- Ablation Experiments for Modules in Our Model: We combine different modules to form several model variants, and then evaluate the effectiveness of each module by comparing the performance between the variants.

- Ablation Experiments with Hyperparameters: We explore how the model's performance changes under different hyperparameter settings and show its ability to generalize across diverse anomaly detection tasks without excessive tuning.

The experimental setup consisted of the following hardware configuration: Intel(R) Xeon(R) Silver 4208 CPU and NVIDIA GeForce RTX 3090 GPU. The software configuration included CUDA 11.3 and Cudnn for parallel computing, Python 3.8 programming language, and PyTorch 1.10.0 deep learning framework.

4.1. Dataset and Implementation Details

To validate the effectiveness of our model in semantic anomaly detection, we conducted extensive experiments on the widely recognized Cifar10 dataset [38]. This dataset comprises 50,000 training images and 10,000 test images, with a resolution of 32×32 pixels and a total of 10 classes. In line with the established practice in semantic anomaly detection, we employed the novelty detection setting [3], where one class is considered normal, and the remaining classes are treated as anomalies.

For assessing the applicability of our proposed fusion method in texture anomaly detection tasks, we performed comprehensive experiments on the MVTec Anomaly Detection (AD) dataset [1]. This dataset encompasses 5 texture and 10 object categories, totaling 5,354 images from the manufacturing domain. To evaluate the effectiveness of our model in this task, we followed the single-class classification protocol, also known as cold-start anomaly detection [39]. Specifically, we trained separate single-class classifiers on normal training samples of each category.

Furthermore, we selected three additional datasets commonly employed in semantic anomaly detection: CIFAR100 [40], STL10 [41], and CatsVsDogs [3]. Additionally, we utilized the Lbot dataset [12], which is specifically designed for texture anomaly detection. The example images from these six datasets are presented in Figure 3.

Hyperparameters: For fair comparisons, we use similar hyperparameter settings to those of the existing normalizing flow based anomaly detection model [16]. Table 1 provides an overview of the crucial hyperparameter settings. These include batchsize, β coefficients, soft-clamping hyperparameter α in the normalizing flow, optimizer, and network depth. We firmly believe that the careful selection of these

Figure 3: Sample images from semantic datasets (the four figures in the top row are from the Cifar10, Cifar100, STL10, and CatsVsDogs datasets, respectively) and industrial texture anomaly datasets (the two figures in the bottom row are sequentially from the MVTec and Lbot datasets).

Table 1
Model hyperparameter settings.

hyperparameters	Value
batchsize	8
optim	Adam
(β_1, β_2)	(0.9,0.999)
FLLs	4
lr	0.0005
α	2

hyperparameters enables a fair and accurate assessment of the model's performance.

Model Evaluation Metrics: To evaluate the performance of our model for anomaly detection, we utilize the commonly used metric in unsupervised anomaly detection, i.e, Area Under the Receiver Operating Characteristic Curve (AUROC) [42]. AUROC is a performance metric for classification models, which measures the accuracy of the model's classification at various thresholds. The ROC curve is a curve plotted with False Positive Rate (FPR) on the x-axis and True Positive Rate (TPR) on the y-axis. FPR represents the proportion of negative samples incorrectly predicted as positive among all negative samples, while TPR represents the proportion of positive samples correctly predicted as positive among all positive samples. By setting different thresholds on the anomaly scores output by the model, we calculate corresponding FPR and TPR values, generating a set of data points that form the ROC curve. The area under this curve represents the AUROC [42] metric.

In our specific implementation, we utilize the Sklearn deep learning framework. Additionally, we adopt the Mean-AUROC (M-AUROC) as the evaluation metric to assess the performance of our model on both semantic anomaly detection and texture anomaly detection datasets.

4.2. Experimental Results and Analysis

In order to show the effectiveness of the proposed model in details, we have conducted four aspects of analysis: 1) Comparison of performance between our algorithm and the existing state-of-the-art (SOTA) baselines. 2) Ablation study. We conduct in-depth experiments on the performance of different feature fusion modules in our model. We show the effectiveness of each module, and analyze the performance of different pre trained feature extractors in our proposed model. 3) Analysis of the computational complexity of our model. 4) Experimental comparison of the model using different hyperparameters.

4.2.1. Comparative Experiments with SOTA Baselines

We performed comprehensive comparisons with recent algorithms in both semantic anomaly detection and texture anomaly detection tasks. Here, we selected ViT-Large as the feature extractor and the third fusion method for our model. For semantic anomaly detection, we considered recent algorithms such as Transformly [28], Panda [3], MSAD [17], and CLIP-OE [43]. Notably, Transformly [28] and CLIP-OE [43] employ pre-trained ViT feature extractors. In the realm of texture anomaly detection, our comparisons encompassed algorithms such as Differnet [16], CSFlow [31], MKD [44], SIMPLENET [39], RD4AD [45], and OCRGAN [46]. Among these, Differnet [16] and CSFlow [31] are based on the normalizing flow for anomaly detection. The performance evaluation of these methods was conducted only using the unlabeled normal data across various anomaly detection datasets. The experimental results, including the M-AUROC (Mean-AUROC) metric representing the average performance of each model in both anomaly detection scenarios, are summarized in Table 2. The MSAD-V1 and MSAD-V2 methods in this table correspond to using pre-trained CNN and ViT within MSAD, respectively.

In the task of semantic anomaly detection on the Cifar10 dataset, our proposed semantic-aware flow model exhibits a significant improvements over existing flow-based anomaly detection methods such as Differnet and CSFlow. Specifically, our method achieves an improvement of 29.8% and 3.8% in terms of AUROC compared to these two methods, demonstrating the clear superiority of our proposed flow model in semantic anomaly detection tasks. Furthermore, our method surpasses other existing methods for semantic anomaly detection and achieves state-of-the-art results on the Cifar10 dataset. For the MVTec dataset in the texture anomaly detection task, our method also achieves competitive performance. Compared to existing semantic anomaly detection methods like Panda, MSAD-V1, MSAD-V2, Transformly, and Clip-OE, our method shows significant improvements of 10.4%, 9.7%, 13.8% 9.0%, and 10.1%, respectively. This indicates that our method is not limited to semantic anomaly detection tasks alone and can be successfully applied to texture anomaly detection tasks as well. However, our method falls slightly short in reaching the state-of-the-art performance in the field of texture anomaly detection.

The compared methods in texture anomaly detection tasks emphasize the analysis of spatial information in images and examine various details across the entire spatial feature map to identify anomalies. Although our method also incorporates spatial texture features, we focus on perceiving the semantic information of the images rather than directly analyzing the complete spatial texture features. Therefore,

Table 2
The AUROC comparison results [%] between our method and recent excellent baselines on various datasets.

Model	Cifar10	MVTec	⋄Cifar100	⋄STL10	⋄CatsVsDogs	⋄Lbot	M-AUROC
Transformly (CVPR2022 Workshop) [28]	98.3	87.9	97.3	99.2	99.5	89.2	95.2
Panda (CVPR2021) [3]	96.2	86.5	94.1	97.6	97.3	97.2	94.8
MSAD-V1 (AAAI2023) [17]	98.6	87.2	96.4	98.9	99.3	97.1	96.0
MSAD-V2 (AAAI2023) [17]	98.6	83.1	97.6	99.2	99.4	86.4	94.1
CLIP-OE (TMLR2022) [43]	98.6	86.8	-	-	-	-	$^{\Delta}$92.6
Differnet (WACV2021) [16]	69.5	94.7	68.3	81.4	85.3	93.4	82.1
CSFlow (WACV2022) [31]	95.5	98.7	93.2	98.9	98.2	99.1	97.2
MKD (CVPR2021) [44]	84.5	87.8	-	-	-	-	$^{\Delta}$86.1
SIMPLENET (CVPR2023) [39]	86.5	99.6	70.2	84.9	63.7	82.6	81.2
RD4AD (CVPR2022) [45]	86.5	98.5	80.6	80.4	42.2	88.2	79.4
OCRGAN [46]	89.4	98.3	-	-	-	-	$^{\Delta}$93.9
Ours	99.3	96.9	98.7	99.7	99.6	97.0	98.5

– denotes no official source code released. $^{\Delta}$ denotes only the average of the first two columns.
⋄ denotes re-running the official source codes with default parameter settings on this dataset.

Table 3
Performance comparison of different feature fusion methods.

	Cifar10	MVTec	Lbot
Without feature fusion	99.1	89.2	89.2
Feature Fusion Method 1	99.3	95.2	92.1
Feature Fusion Method 2	99.3	95.0	93.8
Feature Fusion Method 3	99.3	96.9	97.0

our method falls slightly short in matching the performance of the state-of-the-art methods in texture anomaly detection tasks. However, the average detection performance (M-AUROC) of our method on both the Cifar10 and MVTec datasets demonstrates its superiority over all existing methods.

4.2.2. Ablation Study

1) Feature Fusion Comparative Experiments:

We devised three fusion methods to integrate spatial features into our proposed flow-based model. For more detailed information, please refer to Section 3.3. We first compared the results of these three fusion methods. Based on the experimental results in Table 3, the simple global average pooling in Method 1 exhibited slightly inferior performance in the texture anomaly detection task on the MVTec dataset. This could be attributed to the fact that average pooling tends to lose some spatial texture information. While in Method 2, the utilization of learnable query vectors and self-attention mechanism from [21] also yielded less satisfactory results, potentially due to the direct use of randomly initialized learnable query vector, which hindered the model's ability to swiftly perceive spatial information. In contrast, our third fusion method, where we directly employed the semantic feature y_{sem} from the perceptual process instead of the randomly initialized learnable vector as the query to guide the extraction of relevant information from spatial texture features through self-attention mechanism, achieved superior performance. Therefore, we decided to select the third fusion method as our default fusion method.

As shown in Table 3 , by only incorporating the semantic feature from the feature extractor and the normalizing flow method (i.e., no feature fusion module used), our proposed model achieved an impressive AUROC score of 99.1% on the widely used semantic anomaly detection dataset Cifar10. Nevertheless, the model exhibited relatively lower performance on the texture anomaly detection datasets, MVTec and Lbot. However, we incorporated spatial texture features and significantly enhanced the model's performance on the MVTec and Lbot datasets, with an approximately 7.7% and 7.8% increase in the AUROC metric, respectively. In summary, integrating spatial features into our semantic-based normalizing flow model can not only improve the performance of semantic anomaly detection, but also greatly enhance its ability to detect texture anomalies. Therefore, this experiment validates that our designed feature fusion module can effectively integrate some useful semantic information from the spatial features to enhance semantic distribution learning of our model.

Furthermore, in order to visualize the response of our feature fusion module to anomaly discrimination, we plot histogram in Figure 4 to show the distribution of normal and abnormal scores of the test data on three datasets: Lbot, Cifar10, and MVTec. Specifically, the top row of Figure 4 shows the distribution of the scores for the normal and abnormal test samples from the Lbot dataset. The middle row of Figure 4 shows the histogram results on the Cifar10 dataset when bird is used as the abnormal class. The bottom row of Figure 4 shows the histogram results on the grid subset of the MVTec dataset. The left column of Figure 4 shows the results of not using feature fusion module (FFM) while the right one displays the results of using feature fusion. As shown in Figure 4, the left histogram shows that the overlapping zone between these two distributions is larger than that in the right one. That's to say, if no FFM is used in the model, it is more difficult to distinguish the anomaly from normal data.

Finally, we compare the impact of using semantic features y_{sem} and spatial features y_{spa} as the main input of the flow-based learning layer. Specifically, we swapped the semantic and texture features by using the spatial feature y_{spa} as the input of FLL and incorporating y_{sem} through FFM. We

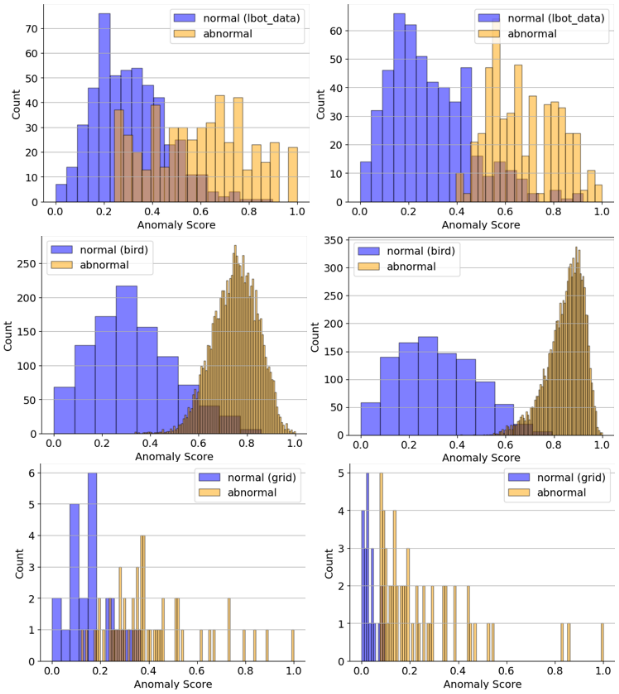

Figure 4: The distribution of normal and abnormal scores in the Lbot, Cifar10, and MVTec datasets is presented from top to bottom. The left shows the results of not using feature fusion while the right one displays the results of using feature fusion. Clearly, it is earier to distinguish the anomaly from normal data when the feature fusion module is used in the model.

Table 4
The AUROC results [%] are obtained when we use y_{sem} and y_{spa} as the main inputs for each flow-based learning layer of our model, respectively.

	Cifar10	MVTec
Using y_{sem} as the input	99.3	96.9
Using y_{spa} as the input	96.3	91.2

conducted experiments on the Cifar10 and MVTec datasets, and the results are presented in Table 4. When utilizing spatial texture information as input, the model achieved competitive results in the texture anomaly detection task on the MVTec dataset. However, it performed poorly in the semantic anomaly detection task on the Cifar10 dataset. This suggests that this method (i.e., using the spatial features as the input of FLL) is not suitable for semantic anomaly detection tasks.

2) Performance of Different Pre-trained Feature Extractors:

We further evaluated the performance of the proposed model using different pre-trained feature extractors, all of which are based on Vision Transformers. Firstly, we utilized ViT models pre-trained on the Imagenet21K dataset [20] for image classification, including ViT-Base and ViT-Large. Secondly, we employed ViT models pre-trained on self-supervised tasks, such as DINO [35] and DINOV2 [36]. Finally, we selected a ViT model pre-trained on the multimodal task Clip [37]. Comparative experiments were conducted for each feature extractor to assess the model's performance in semantic and texture anomaly detection tasks, employing the third fusion method. Please refer to Table 5 for specific experimental results. The M-AUROC metric represents the average performance across both anomaly detection scenarios.

Table 5
The AUROC results [%] obtained by using different pretraining feature extractors on the MVTec and Cifar10 datasets.

Pre-trained feature extractor	Cifar10	MVTec	M-AUROC
DINO	95.0	96.6	95.8
DINOV2	99.1	97.6	98.3
Clip	96.3	95.9	96.1
ViT-Base	98.5	94.9	96.7
ViT-Large	99.3	96.9	98.1

As shown in Table 5, the ViT model pre-trained on image classification tasks exhibited the highest performance on the semantic anomaly detection dataset Cifar10, achieving an impressive experimental result of 99.3%. For the texture anomaly detection dataset MVTec, our model demonstrated favorable results when utilizing feature extractors pre-trained on various tasks. Among them, the DINOV2 [36] pre-trained visual feature extractor yielded the best experimental result of 97.6%. These findings highlight the adaptability of our model to different pre-trained feature extractors for texture anomaly detection tasks.

The ViT model pre-trained on image classification tasks is highly effective in extracting semantic information from images. As a result, our experiments achieved outstanding performance in semantic anomaly detection when employing the model. Conversely, the DINO series used the ViT models pre-trained on various self-supervised tasks, emphasizing the capture of spatial information in images. Among these models, DINOV2 stands out as the latest and most powerful model, combining multiple pretraining tasks. By utilizing the pre-trained DINOV2 model, we achieved remarkable results in texture anomaly detection tasks. On the other hand, the Clip model, trained for multi-modal tasks, requires additional modal information, such as textual context associated with the image. It has been confirmed in [43] that incorporating semantic information yields excellent performance in semantic anomaly detection task.

To further compare the impact of DINO and ViT-Large feature extractors, we conducted comparative experiments on multiple datasets (i.e. Cifar10, Cifar100, MVTec, Lbot). In addition, our method based on DINO is compared to a method that directly combines DINO and KNN (i.e., DINO+KNN), which can be used to demonstrate the effectiveness of our flow-based learning module combined with DINO. This result is shown in Table 6. From the average performance on multiple datasets, we can observe that the proposed method based on DINO is better than the DINO+KNN method. Moreover, with ViT-Large, the proposed method further improves the results. Therefore, to maintain our model's semantic awareness, we decide to

Table 6
The AUROC results [%] using DINO and ViT-Large as the feature extractors, performed on four datasets.

	Cifar10	Cifar100	MVTec	Lbot	Average
DINO+KNN	96.2	96.4	92.7	99.3	96.4
DINO+Ours	95.0	97.6	96.6	99.4	97.2
ViT-Large+Ours	99.3	98.7	96.9	97.0	98.0

Table 7
The AUROC results [%] on Cifar10 for variants of our model.

	CNN	ViT (sem)	ViT (spa)	KNN	$Flow$	AUROC
Variant1	✓			✓		95.7
Variant2		✓		✓		98.7
Variant3			✓	✓		64.1
Variant4		✓			✓	99.1
Variant5			✓		✓	96.3
Variant6		✓	✓		✓	99.3

select the ViT-Large pre-trained on image classification task as the primary feature extractor.

3) Ablation Experiments for Modules in Our Model:

To evaluate the contribution of each module in our model, we designed six variants, each representing a combination of different modules. We conducted ablation experiments on the Cifar10 and MVTec datasets. The experimental results are shown in Tables 7 and 8, respectively. Here, "CNN" refers to the pre-trained ResNet152, "$ViT(sem)$" refers to the use of the semantic features extracted by the pre-trained ViT-Base as the input to the system. The "$ViT(spa)$" denotes that the spatial texture features extracted by pre-trained ViT-Base is used as the input to the system. The "$Flow$" represents that FLL is used as the classifier for anomaly recognition. With "KNN", the K-Nearest Neighbors (KNN) is used as the classifier. Each variant is composed of different modules. For example, Variant1 adopts CNN as feature extractor, and then uses KNN to classify whether the extracted features are from abnormal samples.

In terms of Tables 7 and 8, semantic features are more effective as compared with CNN based features for anomaly detection. In addition, with semantic features ViT (sem) as the main input, flow-based learning modules can further improve the anomaly detection performance. By comparing the experimental results from Variant2 to Variant5 in Table 8, we can see that the spatial features play an important role in texture anomaly detection tasks. For example, Variant5 even performs better than Variant6 on MVTec. However, using only spatial features does not yield satisfactory results in semantic anomaly detection tasks (see Table 7). The results of Variant6 in Tables 7 and 8 indicate that our proposed feature fusion module achieves better average performance on the Cifar10 and MVTec datasets. Therefore, these ablation experiments further demonstrate the advantage of our proposed model over the baselines for both semantic anomaly detection and texture anomaly detection.

Table 8
The AUROC results [%] on MVTec for variants of our model.

	CNN	ViT (sem)	ViT (spa)	KNN	$Flow$	AUROC
Variant1	✓			✓		87.1
Variant2		✓		✓		83.1
Variant3			✓	✓		93.1
Variant4		✓			✓	89.2
Variant5			✓		✓	97.4
Variant6		✓	✓		✓	96.9

4.2.3. *Computational Complexity Analysis*

To analyze computational complexity of our proposed model, we have conducted experiments and analyzed the inference time and parameter quantity of our model. We use the common FLOPs calculation tool (such as thop[2]) to calculate the FLOPs and number of parameters for all the models during training. The inference time of the model here is defined as the time from the start of anomaly detection processing on a single input image to the end of obtaining its output result during testing. The results are presented in Table 9, where the results of all baselines were calculated using their official source code on the data with same batch size under the same hardware environment.

Compared to the baseline CSFlow [31], our proposed flow-based learning model uses ViT (sem) as the main input instead of ViT (spa). As a result, the number of model parameters and inference time used are greatly reduced, as shown in Table 9. Specifically, the number of parameters can be reduced by four fifths, and the inference time is also reduced by about four fifths. This is mainly because the amount of input (i.e., mainly semantic features) we use is significantly reduced compared to the usual spatial features in the baseline CSFlow.

As the simplest flow-based learning model, the Differnet [16] only uses fully connected (FC) layers in each flow learning module, so its FLOPs and inference time are very small, but its anomaly detection performance is relatively poor. For SIMPLENET [39], due to its simple network structure, only a shallow feature adapter (one type of CNN-based module) is used to transfer the pre-trained extracted local features to the target domain, and then a simple binary classifier is used for anomaly discrimination. However, the official code provided by the model requires a significant amount of time for post-processing the output to refine spatial texture localization, thus its inference time is also relatively long. In addition, the Flops calculation is not only related to the number of network parameters, but also to the size of features processed in the model. Our model mainly processes semantic feature information, and the size of the processed features is much smaller than the size of the spatial feature in the SIMPLENET model. Although the network parameter size of our model (48.28M) is approximately 12 times that of SIMPLENET (3.94M). However, our model's FLOPS is not more than 2.5 times larger than SIMPLENET's.

[2] https://github.com/Lyken17/pytorch-OpCounter

Table 9
The comparison of computational complexity between our method and the latest baselines.

	FLOPs	Params	Infer-time
Panda (CVPR2021) [3]	11.60G	60.19M	31ms
MSAD (AAAI2023) [17]	11.86G	85.65M	30ms
Differnet (WACV2021) [16]	171.22M	172.21M	29ms
CSFlow (WACV2022) [31]	65.90G	275.22M	270ms
SIMPLENET (CVPR2023) [39]	1.01G	3.94M	68ms
Ours	2.46G	48.28M	55ms

Table 10
the AUROC results [%] for varying numbers of FLL layers.

Dataset	numbers of FLL layers (n_{layers})			
	2	4	6	8
Cifar10	99.2	99.3	99.3	99.1
MVTec	96.2	96.9	96.5	96.7

Table 11
The AUROC results [%] for different hyperparameter configurations on Cifar10

	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$
$(\beta_1 = 0.8, \beta_2 = 0.8)$	99.13	99.34	99.30
$(\beta_1 = 0.9, \beta_2 = 0.999)$	99.31	99.23	99.35

In addition, compared to the latest baseline MSAD [17], which utilizes pre-trained feature extractors, our model has a similar number of parameters and inference time, but it has a significant reduction in FLOPs. This is mainly because MSAD requires a large amount of computation to train and refine its pre-trained feature extractor network. Moreover, because MSAD requires fine-tuning without freezing the parameters of its pre-trained feature extractor, the number of MSAD model parameters is 1.8 times that of our model parameters.

4.2.4. Hyperparameters Discussion

To evaluate the robustness of our model across various scenarios, we have conducted a set of comparative experiments under different parameter settings and datasets. These experiments assess the impact of key parameters on both semantic anomaly detection (Cifar10) and texture anomaly detection (MVTec) tasks. Specifically, we investigate the effects of α in soft-clamping, β_1 and β_2 in the Adam optimizer, and the number of FLL layers (n_{layers}) in the network.

As shown in Tables 10, 11 and 12, our model exhibits minimal fluctuations in the AUROC metric across different parameter configurations, with variations within approximately 1%. This narrow range of variation demonstrates that our method does not necessitate excessive fine-tuning and can achieve efficient transfer in diverse anomaly detection tasks. Based on the model's performance and taking computational efficiency into account, we have set the hyperparameters as follows: $\alpha = 2, \beta_1 = 0.9, \beta_2 = 0.999$, and $n_{layers} = 4$.

Table 12
The AUROC results [%] for different hyperparameter configurations on MVTec

	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$
$(\beta_1 = 0.8, \beta_2 = 0.8)$	96.43	95.19	96.01
$(\beta_1 = 0.9, \beta_2 = 0.999)$	96.94	95.75	96.56

5. Conclusion

We have presented a novel normalizing flow model specifically designed to learn the distribution of semantic features of normal data, aiming to address the challenge of semantic anomaly detection in images. Our model achieves state-of-the-art detection performance in an unsupervised mode, surpassing other methods on popular semantic anomaly detection datasets. In addition, we investigate the utilization of spatial features to enhance the learning of semantic distributions, enabling our model to achieve competitive performance on the dataset of texture anomaly detection tasks. However, it is important to acknowledge that our model has not yet reached optimal performance in texture anomaly detection. Our model is mainly based on semantic learning, focusing on the overall distribution of normal data, and rarely paying attention to the specific locations of local texture anomalies. As a result, our proposed model is currently not conducive to anomaly localization. Therefore, further advancements in incorporating spatial features are essential to enhance the model's performance in texture anomaly detection. We will continue to develop new methods to enhance the performance of our model in texture anomaly detection.

Acknowledgements:

We sincerely thank the associate editor and reviewers for the helpful comments to further improve our manuscript. This research 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] 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*, 2019, pp. 9592–9600.

[2] Zilong Zhang, Zhibin Zhao, Xingwu Zhang, Chuang Sun, and Xuefeng Chen, "Industrial anomaly detection with domain shift: A real-world dataset and masked multi-scale reconstruction," *arXiv preprint arXiv:2304.02216*, 2023.

[3] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen, "Panda: Adapting pretrained features for anomaly detection and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2806–2814.

[4] Hossein Mirzaei, Mohammadreza Salehi, Sajjad Shahabi, Efstratios Gavves, Cees GM Snoek, Mohammad Sabokrou, and Mohammad Hossein Rohban, "Fake it until you make it: Towards accurate near-distribution novelty detection," in *NeurIPS ML Safety Workshop*.

[5] Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le, "A unified model for multi-class anomaly detection,"

Advances in Neural Information Processing Systems, vol. 35, pp. 4571–4584, 2022.

[6] Zhe Li, Chong Wang, Mei Han, Yuan Xue, Wei Wei, Li-Jia Li, and Li Fei-Fei, "Thoracic disease identification and localization with limited supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8290–8299.

[7] 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*, 2023, pp. 23890–23901.

[8] Yusha Liu, Chun-Liang Li, and Barnabás Póczos, "Classifier two sample test for video anomaly detections.," in *BMVC*, 2018, p. 71.

[9] Ignacio Aguilera-Martos, Marta García-Barzana, Diego García-Gil, Jacinto Carrasco, David López, Julián Luengo, and Francisco Herrera, "Multi-step histogram based outlier scores for unsupervised anomaly detection: Arcelormittal engineering dataset case of study," *Neurocomputing*, vol. 544, pp. 126228, 2023.

[10] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu, "Generalized out-of-distribution detection: A survey," *arXiv preprint arXiv:2110.11334*, 2021.

[11] Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou, "A unified survey on anomaly, novelty, open-set, and out of-distribution detection: Solutions and future challenges," *Trans. Mach. Learn. Res.*, vol. 2022, 2022.

[12] 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 (ICIP)*. IEEE, 2021, pp. 2468–2472.

[13] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*. Springer, 2019, pp. 622–637.

[14] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin, "Csi: Novelty detection via contrastive learning on distributionally shifted instances," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11839–11852, 2020.

[15] 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*, 2021, pp. 9664–9674.

[16] 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*, 2021, pp. 1907–1916.

[17] Tal Reiss and Yedid Hoshen, "Mean-shifted contrastive loss for anomaly detection," *arXiv preprint arXiv:2106.03844*, 2021.

[18] 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*. Springer, 2022, pp. 345–357.

[19] 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*, 2022, pp. 98–107.

[20] 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.

[21] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou, "Going deeper with image transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 32–42.

[22] Wei Ma, Shiyong Lan, Weikang Huang, Wenwu Wang, Hongyu Yang, Yitong Ma, and Yongji Ma, "A semantics-aware normalizing flow model for anomaly detection," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2023, pp. –accepted.

[23] Bowei Pu, Shiyong Lan, Wenwu Wang, Caiying Yang, Wei Pan, Hongyu Yang, and Wei Ma, "Gannext: A new convolutional gan for anomaly detection," in *Artificial Neural Networks and Machine Learning–ICANN 2023: 32st International Conference on Artificial Neural Networks, Crete, Greece, September 26–29, 2023, Proceedings, Part III*. Springer, 2023, pp. 39–49.

[24] Martin Arjovsky and Léon Bottou, "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.

[25] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj, "DrÆm – a discriminatively trained reconstruction embedding for surface anomaly detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 8310–8319.

[26] Izhak Golan and Ran El-Yaniv, "Deep anomaly detection using geometric transformations," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[28] Matan Jacob Cohen and Shai Avidan, "Transformaly-two (feature spaces) are better than one," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4060–4069.

[29] 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.

[30] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson, "Why normalizing flows fail to detect out-of-distribution data," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20578–20589, 2020.

[31] 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*, 2022, pp. 1088–1097.

[32] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe, "Guided image generation with conditional invertible neural networks (2019)," *arXiv preprint arXiv:1907.02392*, 2018.

[33] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[34] Durk P Kingma and Prafulla Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[35] 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*, 2021, pp. 9650–9660.

[36] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al., "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al., "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[38] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi, "Cifar10-dvs: an event-stream dataset for object classification," *Frontiers in Neuroscience*, vol. 11, pp. 309, 2017.

[39] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang, "Simplenet: A simple network for image anomaly detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20402–20411.

[40] Pierluca D'oro, Ennio Nasca, Jonathan Masci, and Matteo Matteucci, "Group anomaly detection via graph autoencoders," in *Advances in Neural Information Processing Systems Workshop*, 2019, vol. 2.

[41] Adam Coates, Andrew Ng, and Honglak Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[42] David MW Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020.

[43] Philipp Liznerski, Lukas Ruff, Robert A Vandermeulen, Billy Joe Franks, Klaus-Robert Müller, and Marius Kloft, "Exposing outlier exposure: What can be learned from few, one, and zero outlier images," *arXiv preprint arXiv:2205.11474*, 2022.

[44] 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*, 2021, pp. 14902–14912.

[45] 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*, 2022, pp. 9737–9746.

[46] 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.

[47] Van Der Maaten Laurens and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2605, pp. 2579–2605, 2008.

Wei Ma received a B.Sc. degree in network engineering from Dalian Maritime University, Dalian, China, in 2020. He is currently a Master student in computer science and technology at Sichuan University, China. His current research interests include computer vision, anomaly detection, and unsupervised learning.

Yao Li received a B.Eng. degree in computer science and technology from Chengdu University Of Technology, Chengdu, China, in 2023. He is now pursuing his master degree in Sichuan University, China. His current research interests include computer vision, anomaly detection, and unsupervised learning.

Shiyong Lan (IEEE Member) received the Ph.D. degree from Sichuan University, China, in 2012. He is now an Associate Professor of computer application within the College of Computer Science, Sichuan University, China. His research interests include signal processing, image processing and understanding, visual object tracking, modeling of spatio-temporal sequences, intelligent transportation system (ITS) and machine learning.

Wenwu Wang (Senior Member, IEEE) is currently a Professor of signal processing and machine learning, and the Co-Director of the Machine Audition Lab, Centre for Vision Speech and Signal Processing, University of Surrey, Guildford, U.K. He is also an AI Fellow with the Surrey Institute for People Centred Artificial Intelligence. His research interests include signal processing, machine learning and perception, artificial intelligence, machine audition (listening), and statistical anomaly detection. He is the elected Chair of IEEE Signal Processing Society (SPS) Machine Learning for Signal Processing Technical Committee, a Board Member of IEEE SPS Technical Directions Board, and an elected Member of the IEEE SPS Signal Processing Theory and Methods Technical Committee. He was the Senior Area Editor during 2019–2023, and an Associate Editor during 2014–2018, for IEEE Transactions on Signal Processing. He is an invited Keynote or Plenary Speaker on more than 20 international conferences and workshops, and a Member of the technical program committee for more than 100 international conferences and workshops.

Weikang Huang received his B.Sc. degree in internet of things from the Sichuan Agricultural University, China, in 2020. He is currently a M.E. candidate with the National Defense Key Laboratory for Synthetic Vision Graphics and Imaging Science, Sichuan University, China. His research interests include image processing and machine learning.

Wujiang Zhu received a B.Sc. degree in material science and technology from Sichuan University,China, in 2021. He is currently a Master student in software engineering at Sichuan university, China. His current research interests include spatial-temporal sequence prediction and imputation and computer vision.

Table 13
The category-specific AUROC comparison results [%] between our model and the latest baselines on the MVTec dataset.

Methods Categories	CSFlow [31]	SIMPLENET [39]	RD4AD [45]	MSAD [17]	Panda [3]	Differnet [16]	Transformly [28]	Ours
bottle	99.8	100	100	99.2	98.1	99.0	98.8	100
cable	99.1	99.9	95.0	80.4	86.2	95.9	90.1	96.0
capsule	97.1	97.7	96.3	82.4	88.0	86.9	85.0	97.3
carpet	100	99.7	98.9	95.6	88.5	92.9	99.2	96.8
grid	99	99.7	100	58.2	54.4	84.0	72.0	98.1
hazelnut	99.6	100	99.9	91.7	96.9	99.3	93.0	99.8
leather	100	100	100	99.7	97.9	97.1	100	100
metal_nut	99.1	100	100	81.9	81.2	96.1	95.2	98.6
pill	8.6	99.0	96.6	76.3	80.4	88.8	83.2	93.2
screw	97.6	98.2	97.0	58.7	66.8	96.3	56.1	90.0
tile	100	99.8	99.3	97.4	98.8	99.4	96.2	100
toothbrush	91.9	99.7	99.5	94.1	84.7	98.6	92.7	95.8
transistor	99.3	100	96.7	73.1	91.5	91.1	76.7	94.3
wood	100	100	99.2	72.9	94.8	99.8	95.7	97.8
zipper	99.7	99.9	98.5	85.1	93.3	95.1	85.2	95.8
Average	98.7	99.6	98.5	83.1	86.5	94.9	87.9	96.9

Table 14
The category-specific AUROC comparison results [%] between our model and the latest baselines on the Cifar10 dataset.

Methods Categories	CSFlow [31]	SIMPLENET [39]	RD4AD [45]	MSAD [17]	Panda [3]	Differnet [16]	Transformly [28]	Ours
plane	93.3	65.3	87.3	98.4	97.4	75.2	96.6	99.3
car	96.5	69.0	90.1	99.5	98.4	68.1	98.5	99.4
bird	92.8	69.2	75.4	97.8	93.9	63.4	96.6	99
cat	90.7	70.8	56.8	97.1	90.6	60.7	96.0	98.7
deer	94.7	73.0	82.8	97.9	97.5	78.3	98.5	99.4
dog	95.5	69.4	71.7	97.2	94.4	66.2	98.3	98.9
frog	98.4	83.4	86.9	99.6	97.5	73.5	98.5	99.7
horse	98.6	73.4	85.9	99.7	97.5	64.3	99.1	99.7
ship	97.5	72.4	89.6	99.5	97.6	75.3	98.4	99.6
truck	96.9	69.0	88.1	99.3	97.4	69.4	99.1	99.3
Average	95.5	71.4	81.4	98.6	96.2	69.5	98.3	99.3

A. Appendix: Supplement category-specific results on the Cifar10 and MVTec datasets

In order to further analyze the performance of our model in semantic anomaly detection, we have provided category-specific results on the Cifar10 and MVTec datasets, as shown in Tables 13 and 14, respectively. As for the MVTec dataset, it needs to be noted that unlike some existing studies which divide the MVTec data into texture class data and object class data for separate discussions, in our experiments, all the MVTec data are combined as the input data for our model. That is to say, we treat equally all the samples of anomalies (including local texture anomalies and object category anomalies) as input samples of abnormal class. For example, "leather" and "cable" respectively represent a texture class and an object class in MVTec, both of which were uniformly used as input samples in our experiments (as shown in Tables 13 and 14).

Table 13 and 14 can serve as detailed comparisons of the anomaly detection performance for specific categories in the dataset. Based on the experimental results in these two tables, which also effectively support the conclusion in Table 2, our model outperforms all current baselines in terms of average detection performance (M-AUROC) on the datasets (such as Cifar10 and MVTec). However, it should be pointed out that the results in Tables 13 and 14 are obtained via re-running the official codes for each baseline with its default parameter-setting, so some of the final average results may differ from Table 2. The reason for such deviation may be that the default parameter-setting in the baseline's official code may not necessarily be optimal for a certain dataset.

From the results of these two tables, it can be seen that our proposed semantic-aware flow model can obtain competitive performance on both texture anomaly (MVTec) and semantic anomaly (Cifar10) datasets. For the MVTec dataset in the texture anomaly detection task, compared to existing semantic anomaly detection methods like Panda, MSAD, and Transformly, our method gives significant improvements of 10.4%, and 13.8%, 9.0%, respectively. Whereas, our method falls slightly short in the field of texture defect detection, when compared to the latest texture defect detection methods like SIMPLENET and RD4AD. However, as for the task of semantic anomaly detection on the Cifar10 dataset, our method surpasses all baselines and achieves state-of-the-art results.

B. Visualize the representations of the models on the Cifar10 and MVTec datasets

Using t-SNE [47], we visualized the clustering distribution (i.e. the representation of the final layer) for our model, and the baselines CSFlow and MSAD as in Figures 5 and 6. As shown in these two figures, the t-SNE plots of our model show less overlap between abnormal examples and real normal examples. Therefore, we can

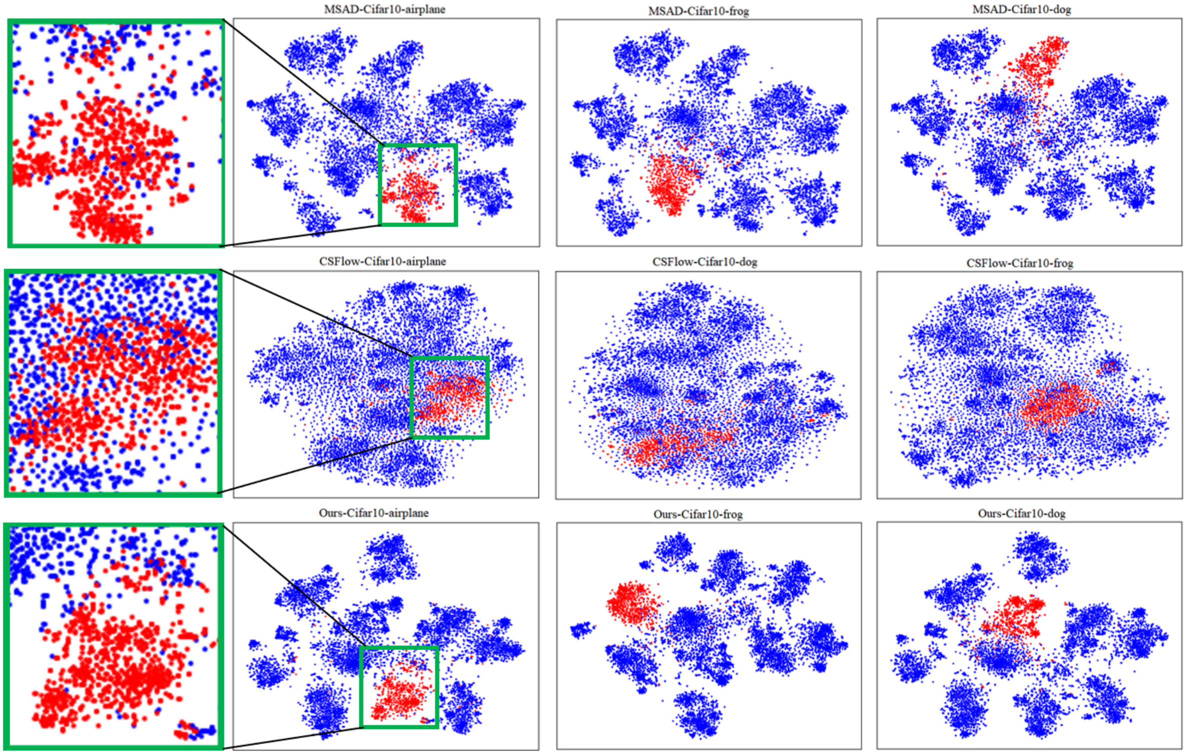

Figure 5: The t-SNE visualization of the representations obtained by three models (i.e. MSAD, CSFlow and the proposed method) on the Cifar10 dataset. We plot embeddings of normal class (blue) and abnormal class (red). A zoom-in view of the green area in the second column is shown in the leftmost column.

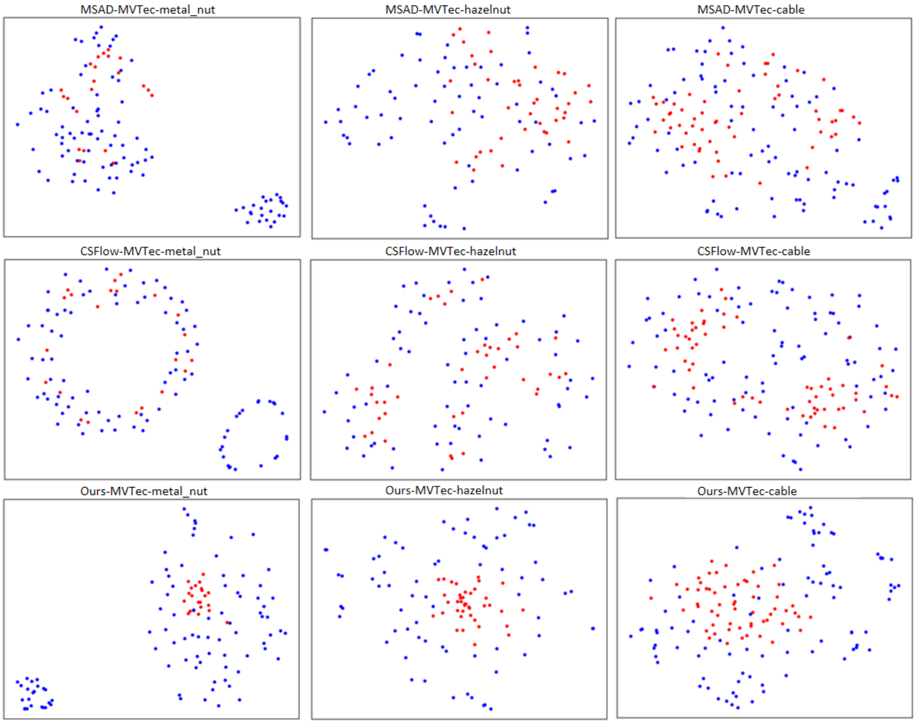

Figure 6: The t-SNE visualization of representations obtained by the models (MSAD, CSFlow and the proposed method) on the MVTec dataset. We plot embeddings of normal class (blue) and abnormal class (red).

clearly see that our model distinguishes between abnormal and normal samples better than the baseline methods.