ADTR: Anomaly Detection Transformer with Feature Reconstruction

Zhiyuan You[1], Kai Yang[2], Wenhan Luo[3], Lei Cui[2], Yu Zheng[1], and Xinyi Le*[1]

[1] Shanghai Jiao Tong University, Shanghai, China
[2] SenseTime Research, Shanghai, China
[3] Tencent, Shenzhen, China

Abstract. Anomaly detection with only prior knowledge from normal samples attracts more attention because of the lack of anomaly samples. Existing CNN-based pixel reconstruction approaches suffer from two concerns. First, the reconstruction source and target are raw pixel values that contain indistinguishable semantic information. Second, CNN tends to reconstruct both normal samples and anomalies well, making them still hard to distinguish. In this paper, we propose Anomaly Detection TRansformer (ADTR) to apply a transformer to reconstruct pre-trained features. The pre-trained features contain distinguishable semantic information. Also, the adoption of transformer limits to reconstruct anomalies well such that anomalies could be detected easily once the reconstruction fails. Moreover, we propose novel loss functions to make our approach compatible with the normal-sample-only case and the anomaly-available case with both image-level and pixel-level labeled anomalies. The performance could be further improved by adding simple synthetic or external irrelevant anomalies. Extensive experiments are conducted on anomaly detection datasets including MVTec-AD and CIFAR-10. Our method achieves superior performance compared with all baselines.

Keywords: Anomaly Detection · Transformer · Attention Mechanism.

1 Introduction

Unsupervised anomaly detection [4,8,15] aims to identify anomalies using prior knowledge from only normal samples. Due to the extreme lack of anomalies in production lines, anomaly detection is attracting more and more interests.

From the view of statistics, anomalies may be seen as distribution outliers of normal samples. In this setting, CNN-based reconstruction models like Auto-Encoder (AE), Variational Auto-Encoder (VAE), and Generative Adversarial Network (GAN) are usually adopted to model the distribution of normal samples [8,13,16,19]. These methods train a model with only normal samples based on the assumption of generalization gap, which means that the reconstruction succeeds with only normal samples but fails with anomalies. The anomaly detection is performed with a distance metric between a sample and its reconstruction.

* Corresponding Author.

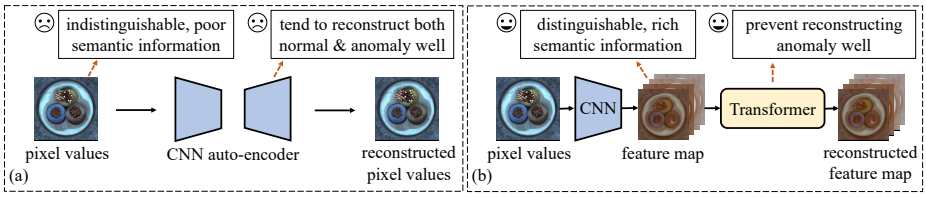


Fig. 1: **(a) CNN-based pixel reconstruction methods** tend to reconstruct both normal samples and anomalies well, making them still hard to distinguish. Also, the pixel values contain indistinguishable semantic information. **(b) Our method** reconstructs features with distinguishable semantic information. Besides, the adoption of transformer limits the reconstruction of anomalies.

As show in Fig. 1a, one concern about these approaches is the poor representation ability. The reconstruction targets are raw pixel values with poor semantic information. Therefore, these pixel reconstruction approaches usually fails when normal and anomalous regions share similar pixel values but different semantic information like different textures. In another aspect, it has been verified that the feature extractor pre-trained on large public datasets could extract distinguishable features for normal samples and anomalies [5,30]. Thus we propose to reconstruct pre-trained features instead of raw pixel values.

Taking CNN as the reconstruction model brings another issue (Fig. 1a). CNN tends to take shortcuts to learn a somewhat "identical mapping", which means the anomalous regions are also reconstructed quite well [16]. The great success of transformer in computer vision inspires us to propose a transformer-based reconstruction model. The query embedding in attention layer of transformer could limit the tendency of "identical mapping", which helps distinguish normal samples and anomalies (See Sec. 3.2).

Besides, more anomaly samples are available with the runs of production lines [5], bringing anomaly detection the demands of compatibility with both the normal-sample-only case (only normal samples are available) and the anomaly-available case (normal samples and a few anomalies are available). Therefore, a unified approach that is compatible with both cases would be a better solution.

In this paper, we propose a concise but powerful transformer-based anomaly detection approach. As shown in Fig. 1b, a frozen pre-trained CNN backbone is adopted to extract features, then a transformer is used for feature reconstruction. The proposed approach has strong representation abilities, and could limit the tendency of "identical mapping". Moreover, novel loss functions are proposed for the compatibility with the anomaly-available case. The performance could be further improved by adding simple synthetic or external irrelevant anomalies. Our approach achieves state-of-the-art anomaly detection performance in anomaly detection datasets including MVTec-AD [4] and CIFAR-10 [18].

2 Related Work

Existing anomaly detection approaches could be generally divided into two categories: reconstruction-based ones and projection-based ones.

Reconstruction-based approaches assume that the reconstruction model trained with normal samples has a generalization gap with anomalies, thus fails to reconstruct anomalies. AE [6,13,16,25] and GAN [26,29,39] are intuitive choices of reconstruction models. Zhou et al. [40] and Xia et al. [35] respectively adopt the structural information and semantic segmentation information for better reconstruction. Zaheer et al. [39] utilize a discriminator to distinguish good or bad quality of reconstruction, and the predicted possibility of bad quality serves as an anomaly score. Gong et al. [16] and Park et al. [25] introduce a memory module to select the most similar embedding in embedding storage of normal samples to restrict the generalization on anomalies. Dehaene et al. [12] refine the selection method with an iterative gradient-based approach.

Projection-based approaches project samples into an embedding space, where normal samples and anomalies are more distinguishable. SVDD [28] extracts feature representation with the one-class classification objective. Yi and Yoon [37] propose a patch-based SVDD with multiple kernels. Liu et al. [21] and Kwon et al. [19] find that the back-propagated gradients of normal samples and anomalies are more distinguishable. FCDD [22] is trained to enlarge the embedding differences between normal samples and anomalies, where the mapped samples are themselves an explanation heat map. Bergmann et al. [5] utilize a teacher-student network, assuming that the embedding differences between normal samples and anomalies would be enlarged through knowledge distillation. Salehi et al. [30] extend the knowledge distillation to multi-layer, multi-scale scheme, enlarging the distillation gap between normal samples and anomalies. PaDiM [11] models normal distribution using pre-trained features, then utilize a distance metric to measure the anomalies. Wang et al. [34] compare the embeddings of local pattern and global pattern to detect anomalies.

Transformer in anomaly detection. Transformer [33] has been successfully used in computer vision [9]. Some attempts also try to utilize transformer for anomaly detection. InTra [27] adopts transformer to recover the image by recovering all masked patches one by one. VT-ADL [24] and AnoVit [38] both apply transformer encoder to reconstruct images. However, these methods mainly focus on indistinguishable raw pixels, and do not figure out why transformer brings improvement. In contrast, we reconstruct pre-trained features instead of raw pixels. We also confirm the efficacy of the query embedding in attention layer to prevent the "identical shortcut".

3 Method

In this part, we first introduce the architecture of ADTR, followed by the analysis of why transformer could limit to reconstruct anomalies well. Finally, we propose two loss functions to extend our approach compatible with available anomalies.

3.1 Architecture

Embedding. A frozen pre-trained CNN backbone is first utilized for feature extraction (Fig. 2a). Here we use EfficientNet-B4 [32] pre-trained on ImageNet.

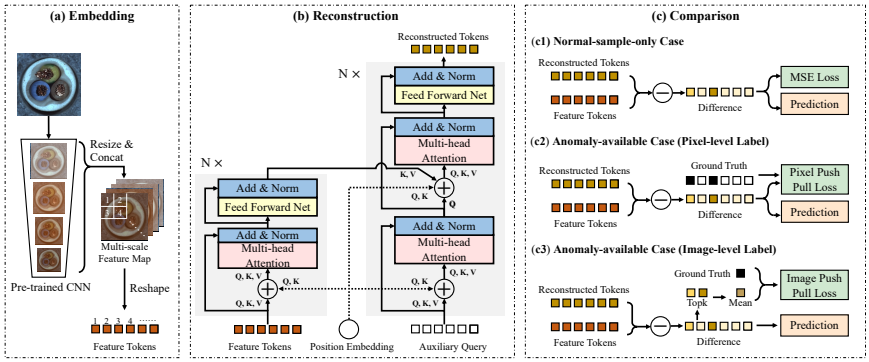


Fig. 2: **Overview of our method.** (a) Embedding: a pre-trained CNN backbone is applied to extract the multi-scale features. (b) Reconstruction: a transformer is utilized to reconstruct the feature tokens with an auxiliary learnable query embedding. (c) Comparison: our approach is compatible with both normal-sample-only case and anomaly-available case. The anomaly score maps are obtained through the differences between extracted and reconstructed features.

The features from *layer1* to *layer5* are resized to the same size, then concatenated together to form a multi-scale feature map, $\boldsymbol{f} \in \mathbb{R}^{C\times H\times W}$. Note that here we define *layer* as the combination of stages with the same size of features. We adopt multi-scale feature map because feature maps from different layers have different levels of receptive fields thus are sensitive to different anomalies.

Reconstruction. The reconstruction stage is shown in Fig. 2b. The feature map, $\boldsymbol{f} \in \mathbb{R}^{C\times H\times W}$, is first split to $H \times W$ feature tokens. To reduce the computation consumption, a 1×1 convolution is applied to reduce the dimension of these tokens before they are fed into the transformer. Also, their dimensions are recovered by another 1×1 convolution when output by transformer. The transformer encoder embeds the input feature tokens into a latent feature space. Each encoder layer follows the standard architecture [33] with multi-head attention, feed forward network (FFN), residual connection, and normalization. The transformer decoder follows the standard architecture [33] with an auxiliary query embedding. The auxiliary query is a learned embedding with the same size of the input feature tokens. The transformer decoder transforms this learned query embedding to reconstruct the feature tokens using multi-head self-attention and encoder-decoder attention mechanisms. The learned position embedding [9] is included because transformer is permutation-invariant.

Comparison. In normal-sample-only case, the model is trained with the MSE loss, $\mathcal{L}_{norm}$, between the backbone extracted features, $\boldsymbol{f}$, and the reconstructed features, $\hat{\boldsymbol{f}} \in \mathbb{R}^{C\times H\times W}$, as follows,

$$\mathcal{L}_{norm} = \frac{1}{H\times W}||\boldsymbol{f} - \hat{\boldsymbol{f}}||_2^2. \tag{1}$$

Inference. We first define the feature difference map, $\boldsymbol{d}(i,u)$, as,

$$\boldsymbol{d}(i,u) = \boldsymbol{f}(i,u) - \hat{\boldsymbol{f}}(i,u), \tag{2}$$

where i represents the index of channel, u is the index of spatial position (height together with width for simplicity). *Anomaly localization* aims to localize anomalous regions, producing an anomaly score map, $\boldsymbol{s}(u)$, which assigns an anomaly score for each pixel, u. $\boldsymbol{s}(u)$ is calculated as the $L2$ norm of the feature difference vector, $\boldsymbol{d}(:,u)$.

$$\boldsymbol{s}(u) = ||\boldsymbol{d}(:,u)||_2. \tag{3}$$

Anomaly detection aims to detect whether an image contains anomalous regions. We intuitively take the maximum value of the averagely pooled $\boldsymbol{s}(u)$ as the anomaly score of the whole image.

3.2 Preventing "Identical Mapping" with Transformer

We suspect that, compared with CNN, the query embedding in attention layer makes transformer difficult to learn an "identical mapping". We denote the features in a normal image as $\boldsymbol{x}^+ \in \mathbb{R}^{K\times C}$, where K is the feature number, C is the channel dimension. The features in an anomalous image are denoted as $\boldsymbol{x}^- \in \mathbb{R}^{K\times C}$. We take a 1-layer network as the reconstruction net, which is trained on $\boldsymbol{x}^+$ with the MSE loss and tested to detect anomalous regions in $\boldsymbol{x}^-$.

Convolutional layer in CNN. We first visit a fully-connected layer, whose weights and bias are denoted as $\boldsymbol{w} \in \mathbb{R}^{C\times C}, \boldsymbol{b} \in \mathbb{R}^C$, respectively. When using this layer as the reconstruction model of normal samples, it can be written as,

$$\hat{\boldsymbol{x}} = \boldsymbol{x}^+\boldsymbol{w} + \boldsymbol{b} \in \mathbb{R}^{K\times C}. \tag{4}$$

With the MSE loss pushing $\hat{\boldsymbol{x}}$ to $\boldsymbol{x}^+$, the model may take shortcut to regress $\boldsymbol{w} \to \boldsymbol{I}$ (identity matrix), $\boldsymbol{b} \to \boldsymbol{0}$. Ultimately, this model could also reconstruct $\boldsymbol{x}^-$ well, failing in anomaly detection. A convolutional layer with 1×1 kernel is equivalent to a fully-connected layer. Besides, An $n \times n$ ($n > 1$) kernel has more parameters and larger capacity, and can complete whatever 1×1 kernel can. Thus, the convolutional layer also has the chance to learn a shortcut.

Transformer with query embedding contains an attention layer with a learnable query embedding, $\boldsymbol{q} \in \mathbb{R}^{K\times C}$. This attention layer can be denoted as,

$$\hat{\boldsymbol{x}} = \texttt{softmax}(\boldsymbol{q}(\boldsymbol{x}^+)^T/\sqrt{C})\boldsymbol{x}^+ \in \mathbb{R}^{K\times C}. \tag{5}$$

To push $\hat{\boldsymbol{x}}$ to $\boldsymbol{x}^+$, the attention map, $\texttt{softmax}(\boldsymbol{q}(\boldsymbol{x}^+)^T/\sqrt{C})$, should approximate $\boldsymbol{I}$ (identity matrix), so $\boldsymbol{q}$ must be highly related to $\boldsymbol{x}^+$. Considering that $\boldsymbol{q}$ in the trained model is relevant to normal samples, the model could not reconstruct $\boldsymbol{x}^-$ well. The ablation study in Sec. 4.4 shows that without the attention layer or the query embedding, the performance of transformer respectively drops by 2.4% or 3%, which is almost the same as CNN. This reflects that the query embedding in attention layer helps prevent transformer from learning an "identical shortcut".

3.3 Adaptation with Anomaly-available Case

In practice, anomalies gradually increase with the runs of production lines, which brings the demands of compatibility with these increasing anomalies. Thus we adapt ADTR to ADTR+ for compatibility with the anomaly-available case.

Adaptation with pixel-level labels. Inspired by [22], we firstly calculate a pseudo-Huber loss, $\boldsymbol{\phi}(u)$, using the feature difference map, $\boldsymbol{d}(i,u)$.

$$\boldsymbol{\phi}(u) = ((\frac{1}{C}\sum_{i}^{C}|\boldsymbol{d}(i,u)|)^2 + 1)^{\frac{1}{2}} - 1. \tag{6}$$

The pseudo-Huber loss, $\boldsymbol{\phi}(u)$, is designed as a difference map, which is easy to train and extend. Then the reconstruction loss function with pixel-level labels is denoted as $\mathcal{L}_{px}$ and could be described as a "push-pull loss" as,

$$\mathcal{L}_{px} = \frac{1}{HW}\sum_{u}^{HW}(1-\boldsymbol{y}(u))\boldsymbol{\phi}(u) - \alpha\log(1-\exp(-\frac{1}{HW}\sum_{u}^{HW}\boldsymbol{y}(u)\boldsymbol{\phi}(u))), \tag{7}$$

where the first term pulls the reconstructed normal features to the extracted features, and the second term pushes the reconstructed anomalous features away from the original features, $\boldsymbol{y}(u)$ is the pixel-level label (0 for normal sample and 1 for anomaly) and α is a weight term.

Adaptation with image-level labels. Since anomaly samples could contain both anomalous and normal regions, simply treating all regions of anomaly samples as anomalous regions confuses the model. Considering that larger values in $\boldsymbol{\phi}(u)$ are more likely to be anomalous regions, we firstly collect k maximum values of $\boldsymbol{\phi}(u)$, then calculate their mean as the anomaly score of the image.

$$q = \frac{1}{k}\sum \text{top\_k}(\boldsymbol{\phi}). \tag{8}$$

Then the image-level loss, $\mathcal{L}_{img}$, could be calculated as,

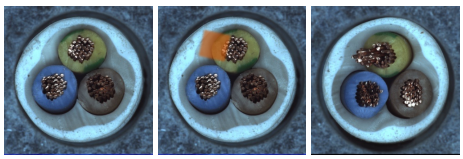
$$\mathcal{L}_{img} = (1-y)q - \alpha y\log(1-\exp(-q)), \tag{9}$$

where y is the image-level label (0 for normal sample and 1 for anomaly) and α is a weight term. In $\mathcal{L}_{img}$, the first term pulls the reconstructed features of normal samples towards the extracted features, while the second term pushes the reconstructed features of anomalies away from the extracted features.

4 Experiment

4.1 Dataset

MVTec-AD [4] is a multi-category, multi-defect, industrial anomaly detection dataset with 15 categories. The ground-truth includes both image labels and anomaly segmentation. In *normal-sample-only case*, we follow the original setting to use normal



(a) Normal sample (b) Confetti noise (c) Real anomaly

Fig. 3: **Synthetic anomalies** by adding confetti noise on normal samples.

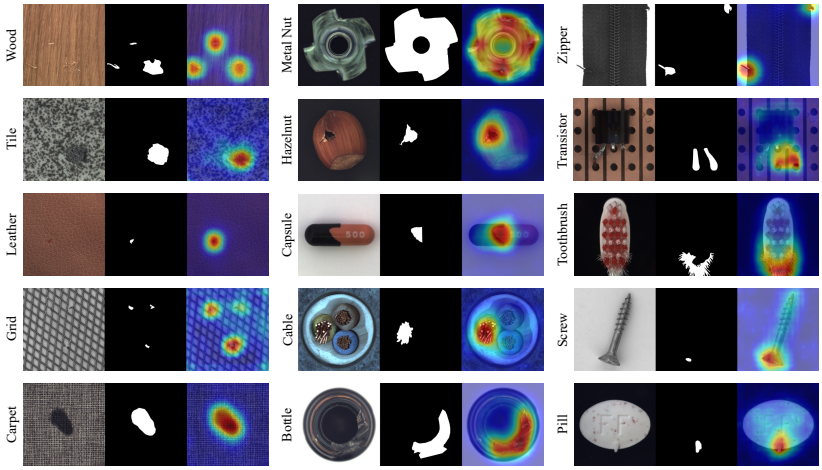


Fig. 4: **Anomaly detection results on MVTec-AD** [4]. From left to right: the anomaly sample, the ground-truth, and the anomaly score map of ADTR.

samples for training, and test on both normal and anomaly samples. In *anomaly-available case*, following [22], we synthesize anomalies by adding confetti noise on normal samples (Fig. 3).

CIFAR-10 [18] is a classical classification dataset with 10 classes. Each class has 5000 images for training and 1000 images for testing. In *normal-sample-only case*, following [19], the training set of one class is used for training, and the test set contains normal images of the same class and the same number of anomaly images randomly sampled from other classes. In *anomaly-available case*, an irrelevant dataset, CIFAR-100 [18], is used as an auxiliary dataset. We randomly select the same number of images from CIFAR-100 as anomalies.

4.2 Anomaly Detection on MVTec-AD

The performance of our method is evaluated on anomaly detection and localization tasks of MVTec-AD [4].

Setup. The sizes of the image and feature map are selected as 256×256 and 16×16, respectively. The numbers of the encoder layer and decoder layer (N in Fig. 2) in transformer are both set as 4. The features from *layer1* to *layer5* of EfficientNet-B4 [32] are resized and concatenated to form a 720-channel feature map. The reduced channel dimension is set as 256. AdamW optimizer [23] with weight decay 1×10^{-4} is used for training with batch size 16. In *normal-sample-only case*, models are trained with $\mathcal{L}_{norm}$ in Eq. (1) for 500 epochs. The learning rate is 1×10^{-4} initially, and dropped by 0.1 after 400 epochs. In *anomaly-available case*, the pixel-level loss, $\mathcal{L}_{px}$, in Eq. (7) is adopted for training, where α is chosen as 0.003. The trained model in normal-sample-only case is firstly loaded. Then the model is trained for 300 epochs with the learning rate of 1×10^{-4} for first 200 epochs and 1×10^{-5} for last 100 epochs.

Table 1: **Anomaly localization results under pixel-level AUROC metric on MVTec-AD [4].**

	Texture					Object										
	Carp.	Grid	Leat.	Tile	Wood	Bott.	Cable	Caps.	Haze.	Meta.	Pill	Screw	Toot.	Tran.	Zipp.	Mean
SSIM-AE [6]	87	94	78	59	73	93	82	94	97	89	91	96	92	90	88	86
AnoGAN [31]	54	58	64	50	62	86	78	84	87	76	87	80	90	80	78	74
VEVAE [21]	78	73	95	80	77	87	9	74	98	94	83	97	94	93	78	86
SMAI [20]	88	97	86	62	80	86	92	93	97	92	92	96	96	85	90	89
GDR [12]	74	96	93	65	84	92	91	92	98	91	93	95	99	92	87	89
P-Net [40]	57	**98**	89	**97**	**98**	**99**	70	84	97	79	91	**1.00**	99	82	90	89
FCDD [22]	96	91	98	91	88	97	90	93	95	94	81	86	94	88	92	92
SCADN [36]	64.9	79.6	76.3	67.7	67.2	69.6	81.4	68.7	88.4	75.4	74.7	87.6	90.1	68.9	67.0	75.2
PSVDD [37]	92.6	96.2	97.4	91.4	90.8	98.1	96.8	95.8	97.5	98.0	95.1	95.7	98.1	97.0	95.1	95.7
SPADE [10]	97.5	93.7	97.6	87.4	88.5	98.4	**97.2**	99.0	**99.1**	**98.1**	96.5	98.9	97.9	94.1	96.5	96.0
KDAD [30]	95.6	91.8	98.1	82.8	84.8	96.3	82.4	95.9	94.6	86.4	89.6	96.0	96.1	76.5	93.9	90.7
Loc-Glo [34]	96	78	90	80	81	93	94	90	84	91	93	96	96	**1.00**	**99**	91
ADTR(ours)	98.7	95.0	98.1	93.8	91.2	98.0	96.8	**99.1**	98.6	97.0	98.3	99.3	98.5	97.9	97.2	97.2
ADTR+(ours)	**98.8**	94.2	**98.6**	95.9	93.0	98.0	97.0	**99.1**	98.8	96.8	**98.7**	99.3	**99.2**	97.8	97.6	**97.5**

Qualitative results on MVTec-AD are shown in Fig. 4. Our approach successfully detects different kinds of anomalies with high localization accuracy. Especially, for the shown "Metal Nut" example, where the anomaly is a flipped normal sample, our approach detects the "flip" anomaly successfully though there are no obvious vision anomalies like texture disorder nor color change.

Quantitative results of anomaly localization are given in Tab. 1. Our approach is compared with SSIM-AE [6], AnoGAN [31], VEVAE [21], SMAI [20], GDR [12], P-Net [40], FCDD [22], SCADN [36], PSVDD [37], SPADE [10], KDAD [30], Loc-Glo [34]. With pure normal samples, ADTR stably outperforms the best baseline, SPADE [10], by 1.2%. With merely simple synthetic anomalies, the performance of ADTR+ is further improved by 0.3%.

Quantitative results of anomaly detection are shown in Tab. 2. Our approach is compared with GANomaly [2], ArNet [14], SPADE [10], SCADN [36], PSVDD [37], TS [5], KDAD [30]. ADTR considerably exceeds all baseline methods ($\geq$ 3.9%) with only normal samples. The performance of ADTR+ is improved by 0.5% with simple synthetic anomalies.

4.3 Anomaly Detection on CIFAR-10

To further validate the anomaly detection ability, we evaluate our model in the unsupervised one-class classification task of CIFAR-10 [18].

Setup. The setup is the same as that in Sec. 4.2 except the followings. First, the sizes of the image and feature map are 32×32 and 8×8, respectively. Second, in anomaly-available case, the model is trained with the image-level loss, $\mathcal{L}_{img}$, in Eq. (9), where α and k are selected as 0.003 and 20, respectively.

Quantitative results on CIFAR-10 are shown in Tab. 3. The competitors include: KDE [7], VAE [3], LSA [1], AnoGAN [31], DSVDD [28], OCGAN [26], GradCon [19], GT [15], TS [5], Loc-Glo [34], KDAD [30]. ADTR surpasses KDAD [30] by a great margin (7.5%) when training in normal-sample-only case. In

Table 2: **Anomaly detection results under image-level AUROC metric on MVTec-AD [4].**

	Texture					Object										
	Carp.	Grid	Leat.	Tile	Wood	Bott.	Cable	Caps.	Haze.	Meta.	Pill	Screw	Toot.	Tran.	Zipp.	Mean
GANomaly [2]	69.9	70.8	84.2	79.4	83.4	89.2	75.7	73.2	78.5	70.0	74.3	74.6	65.3	79.2	74.5	76.2
ArNet [14]	70.6	88.3	86.2	73.5	92.3	94.1	83.2	68.1	85.5	66.7	78.6	**100**	**100**	84.3	87.6	83.9
SPADE [10]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	85.5
SCADN [36]	50.4	98.3	65.9	79.2	96.8	95.7	85.6	76.5	83.3	62.4	81.4	83.1	98.1	86.3	84.6	81.8
PSVDD [37]	98.6	90.3	76.7	92.9	94.6	92.0	90.9	**94.0**	86.1	81.3	97.8	**100**	91.5	96.5	**97.9**	92.1
TS [5]	95.3	**98.7**	93.4	95.8	95.5	96.7	82.3	92.8	91.4	94.0	86.7	87.4	98.6	83.6	95.8	92.5
KDAD [30]	79.3	78.0	95.1	91.6	94.3	99.4	89.2	80.5	98.4	73.6	82.7	83.3	92.2	85.6	93.2	87.7
ADTR(ours)	**100**	97.5	**100**	**100**	**99.9**	**100**	**92.5**	93.1	**100**	**94.9**	92.1	94.0	93.1	97.6	95.8	96.4
ADTR+(ours)	**100**	97.8	**100**	**100**	**99.9**	**100**	**92.5**	92.5	99.9	94.5	**93.3**	94.2	93.9	**98.0**	97.0	**96.9**

Table 3: **Anomaly detection results under image-level AUROC metric on CIFAR-10 [18].**

	Airplane	Automobile	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Mean
KDE [7]	65.8	52.0	65.7	49.7	72.7	49.6	75.8	56.4	68.0	54.0	61.0
VAE [3]	63.4	44.2	64.0	49.7	74.3	51.5	74.5	52.7	67.4	41.6	58.3
LSA [1]	73.5	58.0	69.0	54.2	76.1	54.6	75.1	53.5	71.7	54.8	64.1
AnoGAN [31]	67.1	54.7	52.9	54.5	65.1	60.3	58.5	62.5	75.8	66.5	61.8
DSVDD [28]	61.7	65.9	50.8	59.1	60.9	65.7	67.7	67.3	75.9	73.1	64.8
OCGAN [26]	75.7	53.1	64.0	62.0	72.3	62.0	72.3	57.5	82.0	55.4	65.7
GradCon [19]	76.0	59.8	64.8	58.6	73.3	60.3	68.4	56.7	78.4	67.8	66.4
GT [15]	76.2	84.8	77.1	73.2	82.8	84.8	82.0	88.7	89.5	83.4	82.3
TS [5]	78.9	84.9	73.4	74.8	85.1	79.3	89.2	83.0	86.2	84.8	82.0
Loc-Glo [34]	79.1	70.3	67.5	56.1	73.9	63.8	73.2	67.4	81.4	72.2	70.5
KDAD [30]	90.5	90.4	80.0	77.0	86.7	91.4	89.0	86.8	91.5	88.9	87.2
ADTR(ours)	94.1	97.4	92.3	89.0	93.2	94.4	97.4	95.8	96.3	96.7	94.7
ADTR+(ours)	**96.2**	**98.0**	**94.5**	**91.7**	**95.1**	**95.6**	**98.0**	**97.1**	**98.0**	**96.9**	**96.1**

anomaly-available case, the performance of ADTR+ is further improved by 1.4% with the help of external irrelevant dataset, reflecting the effectiveness of the designed image-level loss function, $\mathcal{L}_{img}$.

4.4 Ablation Study

Extensive ablation studies with pixel-level AUROC metric are conducted on anomaly localization task of MVTec-AD [4].

Attention and auxiliary query embedding. As shown in Tab. 4a, a CNN revised from ResNet [17] is firstly included as the baseline of the reconstruction model. (1) The replacement of the attention layer is a concatenation followed by projection. If we remove the attention layer (w/o Attn) from the transformer, the performance shows no obvious superiority to CNN. (2) Without the auxiliary query embedding (w/o Query), meaning that only the encoder embedding is input to the decoder, the performance is even worse than CNN. (3) Equipped with both attention and auxiliary query embedding (Attn+Query), transformer stably outperforms CNN by 2.8%. This proves our assertion in Sec. 3.2 that the auxiliary query embedding in attention layer helps prevent transformer from reconstructing anomalies well.

Table 4: **Ablation study** on (a) attention & auxiliary query embedding, (b) reconstructing pixels *vs.* features, (c) backbone, and (d) multi-scale features under pixel-level AUROC metric on anomaly localization of MVTec-AD [4].

(a) Attention & auxiliary query embedding

	CNN	w/o Attn	w/o Query	Attn+Query
Pixel AUROC	94.4	94.8	94.2	**97.2**

(b) Reconstructing pixels *vs.* features

	Pixels	Features
Pixel AUROC	91.3	**97.2**

(c) Backbone

	Res-18	Res-34	Efficient-B0	Efficient-B4
Pixel AUROC	95.3	95.7	96.4	**97.2**

(d) Multi-scale features

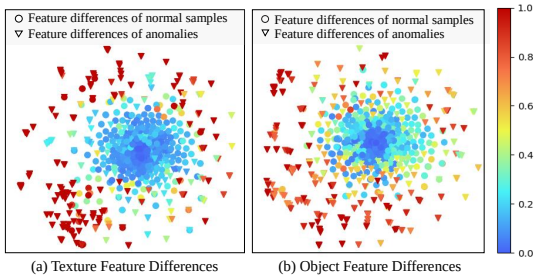
	Last-layer	Multi-scale
Pixel AUROC	96.0	**97.2**

Reconstructed target. In Tab. 4b, reconstructing features surpasses pixel values substantially, indicating that the features extracted by pre-trained backbone are more distinguishable for normal samples and anomalies than raw pixels.

Backbone and multi-scale features. (1) As shown in Tab. 4c, four different backbones all achieve quite good performance, reflecting that our method could cooperate with different types of backbones. (2) In Tab. 4d, multi-scale features obviously outperform last-layer feature, because multi-scale features contain different levels of receptive fields thus are sensitive to different anomalies.

4.5 Visualization of Feature Difference Vectors

We visualize the feature difference vectors $\boldsymbol{d}(:, u)$ in Eq. (2) to better interpret our approach. Specifically, we randomly sample 600 feature difference vectors (normal : anomaly = 1:1) from MVTec-AD [4]. Then t-SNE is utilized to visualize the high dimensional vectors in a 2D space, as shown in Fig. 5. Firstly, normal samples and anomalies are mostly colored with blue and red, respectively, indicating good anomaly detection ability. Secondly, normal samples are well clustered, and there is a wide gap between the normal samples and anomalies. These observations indicate that our approach brings a large generalization gap between normal samples and anomalies.



(a) Texture Feature Differences (b) Object Feature Differences

Fig. 5: **visualization of feature difference vectors** by t-SNE. Circles and triangles respectively represent normal samples and anomalies. The color map indicates the predicted anomaly possibility. Our method brings large generalization gap between normal samples and anomalies.

5 Conclusion

In this paper, we propose anomaly detection transformer to utilize a transformer to reconstruct pre-trained features. First, the pre-trained features contain dis-

tinguishable semantic information. Second, the adoption of transformer prevents reconstructing anomalies well such that anomalies could be detected easily once the reconstruction fails. Our method brings a large generalization gap between normal samples and anomalies. Moreover, we propose novel loss functions to extend our approach from normal-sample-only case to anomaly-available case with both image-level labeled and pixel-level labeled anomalies, further improving the performance. Our approach achieves the state-of-the-art performance on anomaly detection benchmarks including MVTec-AD and CIFAR-10.

References

1. Abati, D., Porrello, A., Calderara, S., Cucchiara, R.: Latent space autoregression for novelty detection. In: CVPR (2019)
2. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training. In: ACCV (2018)
3. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE (2015)
4. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: MVTec AD: a comprehensive real-world dataset for unsupervised anomaly detection. In: CVPR (2019)
5. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In: CVPR. pp. 4183–4192 (2020)
6. Bergmann, P., Lwe, S., Fauser, M., Sattlegger, D., Steger, C.: Improving unsupervised defect segmentation by applying structural similarity to autoencoders. In: International Conference on Computer Vision Theory and Applications (2019)
7. Bishop, C.M.: Pattern recognition and machine learning. springer (2006)
8. Borghesi, A., Bartolini, A., Lombardi, M., Milano, M., Benini, L.: Anomaly detection using autoencoders in high performance computing systems. In: AAAI (2019)
9. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
10. Cohen, N., Hoshen, Y.: Sub-image anomaly detection with deep pyramid correspondences. arXiv preprint arXiv:2005.02357 (2020)
11. Defard, T., Setkov, A., Loesch, A., Audigier, R.: PaDim: A patch distribution modeling framework for anomaly detection and localization. In: ICPR (2021)
12. Dehaene, D., Frigo, O., Combrexelle, S., Eline, P.: Iterative energy-based projection on a normal data manifold for anomaly localization. In: ICLR (2019)
13. Dehaene, D., Frigo, O., Combrexelle, S., Eline, P.: Iterative energy-based projection on a normal data manifold for anomaly localization. In: ICLR (2020)
14. Fei, Y., Huang, C., Jinkun, C., Li, M., Zhang, Y., Lu, C.: Attribute restoration framework for anomaly detection. IEEE Transactions on Multimedia (2020)
15. Golan, I., El-Yaniv, R.: Deep anomaly detection using geometric transformations. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) NIPS (2018)
16. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Hengel, A.v.d.: Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: ICCV (2019)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)

18. Krizhevsky, A.: Learning multiple layers of features from tiny images. Master's thesis, University of Tront (2009)
19. Kwon, G., Prabhushankar, M., Temel, D., AlRegib, G.: Backpropagated gradient representations for anomaly detection. In: ECCV (2020)
20. Li, Z., Li, N., Jiang, K., Ma, Z., Wei, X., Hong, X., Gong, Y.: Superpixel masking and inpainting for self-supervised anomaly detection. In: BMVC (2020)
21. Liu, W., Li, R., Zheng, M., Karanam, S., Wu, Z., Bhanu, B., Radke, R.J., Camps, O.: Towards visually explaining variational autoencoders. In: CVPR (2020)
22. Liznerski, P., Ruff, L., Vandermeulen, R.A., Franks, B.J., Kloft, M., Müller, K.: Explainable deep one-class classification. In: ICLR (2021)
23. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
24. Mishra, P., Verk, R., Fornasier, D., Piciarelli, C., Foresti, G.L.: VT-ADL: A vision transformer network for image anomaly detection and localization. In: International Symposium on Industrial Electronics (2021)
25. Park, H., Noh, J., Ham, B.: Learning memory-guided normality for anomaly detection. In: CVPR (2020)
26. Perera, P., Nallapati, R., Xiang, B.: OCGAN: One-class novelty detection using GANs with constrained latent representations. In: CVPR (2019)
27. Pirnay, J., Chai, K.: Inpainting transformer for anomaly detection. arXiv preprint arXiv:2104.13897 (2021)
28. Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: ICML (2018)
29. Sabokrou, M., Khalooei, M., Fathy, M., Adeli, E.: Adversarially learned one-class classifier for novelty detection. In: CVPR (2018)
30. Salehi, M., Sadjadi, N., Baselizadeh, S., Rohban, M.H., Rabiee, H.R.: Multiresolution knowledge distillation for anomaly detection. In: CVPR (2021)
31. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International conference on information processing in medical imaging (2017)
32. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: ICML (2019)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NIPS (2017)
34. Wang, S., Wu, L., Cui, L., Shen, Y.: Glancing at the patch: Anomaly localization with global and local feature comparison. In: CVPR (2021)
35. Xia, Y., Zhang, Y., Liu, F., Shen, W., Yuille, A.L.: Synthesize then compare: Detecting failures and anomalies for semantic segmentation. In: ECCV (2020)
36. Yan, X., Zhang, H., Xu, X., Hu, X., Heng, P.A.: Learning semantic context from normal samples for unsupervised anomaly detection. In: AAAI (2021)
37. Yi, J., Yoon, S.: Patch SVDD: Patch-level SVDD for anomaly detection and segmentation. In: ACCV (2020)
38. Yunseung, L., Pilsung, K.: AnoViT: Unsupervised anomaly detection and localization with vision transformer-based encoder-decoder. arXiv preprint arXiv:2203.10808 (2022)
39. Zaheer, M.Z., Lee, J.h., Astrid, M., Lee, S.I.: Old is gold: Redefining the adversarially learned one-class classifier training paradigm. In: CVPR (2020)
40. Zhou, K., Xiao, Y., Yang, J., Cheng, J., Liu, W., Luo, W., Gu, Z., Liu, J., Gao, S.: Encoding structure-texture relation with p-net for anomaly detection in retinal images. In: ECCV (2020)

Appendix

A More Details

Backbone. We use the ImageNet pre-trained EfficientNet-B4 [32] [4] as the backbone. The features from layer1 to layer5 of EfficientNet-B4 [32] have the channel of 24, 32, 56, 160, 448, respectively. Here we define "layer" as the combination of stages that have the same size of features. The 5 features are resized to the same size and concatenated together to form a 720-channel feature map. For MVTec-AD [4], the image size and the feature size are set as 512×512 and 32×32, respectively. Therefore, a feature map with the shape of $32 \times 32 \times 720$ is obtained. For CIFAR-10 [18], the image size is 32×32, which is quite small. Thus the size of the feature map is set relatively large (with the output stride of 4), so an $8 \times 8 \times 720$ feature map is obtained.

Transformer. A 1×1 convolution is applied firstly to the feature map to reduce the channel from 720 to 256. Then the feature map is split to separate feature tokens. For MVTec-AD [4] and CIFAR-10 [18], there are 1024 and 64 feature tokens with the channel of 256, respectively. The position embedding is a learned embedding with the same size as the input feature tokens.

The transformer encoder follows the standard architecture in [33] with 4 layers. Each layer consists of a multi-head self-attention layer, a feed forward layer, and a shortcut connection with layer normalization. The head number in attention is 8. The architecture of the feed forward layer is shown as follows.

Layer	Input	FC1	Relu	FC2
Output Size	256	1024	1024	256

Besides, the position embedding is added in each self-attention layer rather than only in the first layer to keep more position information.

The transformer decoder also has 4 decoder layers. Each layer is composed of 2 parts, the self-attention part and the cross-attention part. The self-attention part includes a multi-head self-attention layer and a shortcut connection with layer normalization. The cross-attention part consists of a multi-head cross-attention layer, a feed forward layer, and a shortcut connection with layer normalization. The head number in both attention layers is set as 8. The architecture of the feed forward layer is the same as that in the transformer encoder. Also, the position embedding is added in all attention layers. The query embedding is a learned embedding with the same size as the input feature tokens.

The outputs of the transformer have the same size as the inputs (1024×256 for MVTec-AD, 64×256 for CIFAR-10). Then a 1×1 convolution is applied to increase the channel from 256 to 720. After reshape, we obtain the reconstructed feature map ($32 \times 32 \times 720$ for MVTec-AD, $8 \times 8 \times 720$ for CIFAR-10).

[4] We use the EfficientNet-B4 checkpoint in https://github.com/lukemelas/EfficientNet-PyTorch/releases/download/1.0/efficientnet-b4-6ed6700e.pth

Training configurations on MVTec-AD. In *normal-sample-only case*, the backbone is frozen. The transformer is trained with $\mathcal{L}_{norm}$ in Eq. (3) for 500 epochs with batch size 16. AdamW optimizer [23] with weight decay 1×10^{-4} is used. The learning rate is set as 1×10^{-4} initially, and dropped by 0.1 after 400 epochs. In *anomaly-available case*, the trained model in *normal-sample-only case* is firstly loaded. The transformer is trained with $\mathcal{L}_{px}$ in Eq. (6) for 300 epochs. α in Eq. (6) is set as 0.003. The learning rate is initially set as 1×10^{-4}, and dropped by 0.1 after 200 epochs.

Training configurations on CIFAR-10. In *normal-sample-only case*, the details are the same as those in **MVTec-AD** except the image size and feature size described in **Backbone**. For more efficient training, the batch size is set as 128. In *anomaly-available case*, the same implementations as **MVTec-AD** are adopted except the followings. Considering that the anomalies are image-level labeled in CIFAR-10 case, the transformer is trained with $\mathcal{L}_{img}$ in Eq. (8), where α and k are selected as 0.003 and 20, respectively.

B More Visualization Results

Qualitative results on MVTec-AD are provided. These categories include: carpet (Fig. A1), grid (Fig. A2), leather (Fig. A3), tile (Fig. A4), wood (Fig. A5), bottle (Fig. A6), cable (Fig. A7), capsule (Fig. A8), hazelnut (Fig. A9), metal nut (Fig. A10), pill (Fig. A11), screw (Fig. A12), toothbrush (Fig. A13), transistor (Fig. A14), and zipper (Fig. A15). Our approach could detect different kinds of anomalies in all categories with quite high localization accuracy. The performance of the proposed approach keeps stable in all these categories with various anomaly types, demonstrating strong generalization ability and robustness. Specifically, for both quite small anomalies (e.g. the second column in Fig. A8) and quite large anomalies (e.g. the ninth column in Fig. A4), both single-kind anomalies (e.g. the second column in Fig. A3) and multi-kind combined anomalies (e.g. the last column in Fig. A5), both texture or color disorder (e.g. the second column in Fig. A1) and misplacement (e.g. the last column in Fig. A14), our approach could effectively detect all anomalies.

Qualitative results on CIFAR-10 are given. These categories include: airplane (Fig. A16), automobile (Fig. A17), bird (Fig. A18), cat (Fig. A19), deer (Fig. A20), dog (Fig. A21), frog (Fig. A22), horse (Fig. A23), ship (Fig. A24), and truck (Fig. A25). Our approach could successfully detect various kinds of anomalies. Also, high anomaly scores mainly center on the anomaly objects rather than the backgrounds, which indicates that our approach detects anomalies based on the understanding of semantic features. In particular, even for anomalies that are very similar to normal samples, like the "truck" category when "automobile" category serves as normal samples (e.g. the sixth column in Fig. A17), the "dog" category when "cat" category serves as normal samples (e.g. the last column in Fig. A19), the "horse" category when "deer" category serves as normal samples (e.g. the tenth column in Fig. A20), our approach still successfully distinguishes these anomalies from normal samples.

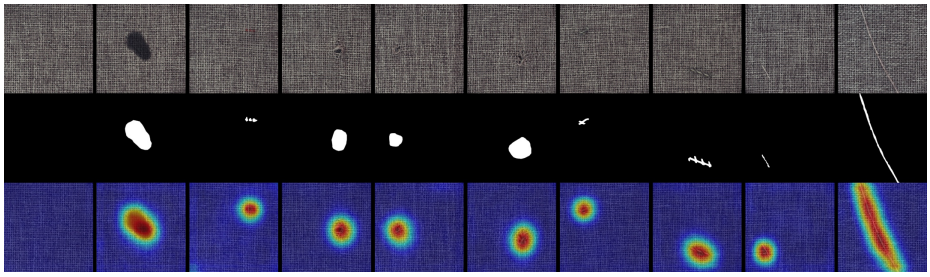


Fig. A1: Anomaly detection results of carpet on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

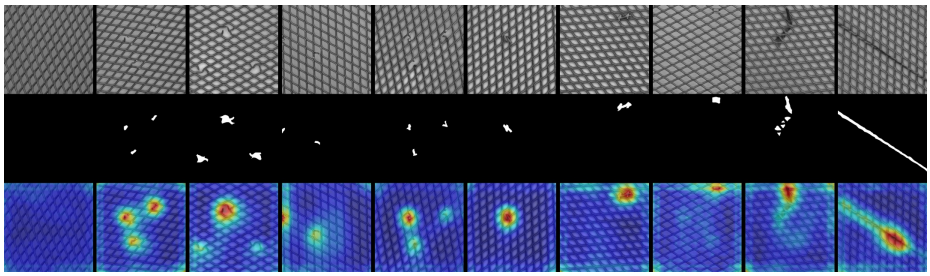


Fig. A2: Anomaly detection results of grid on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

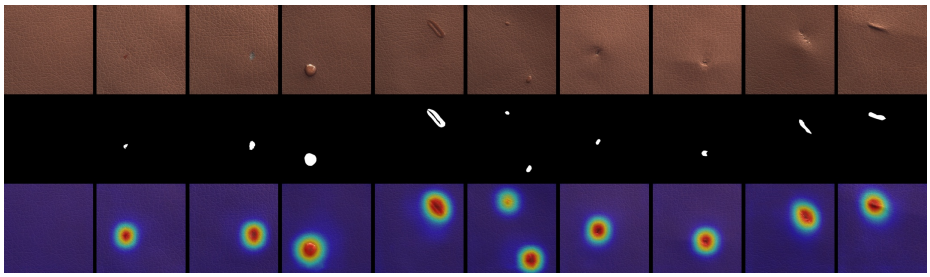


Fig. A3: Anomaly detection results of leather on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

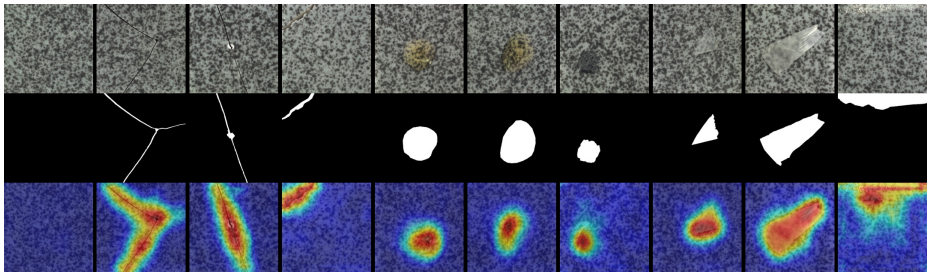


Fig. A4: Anomaly detection results of tile on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

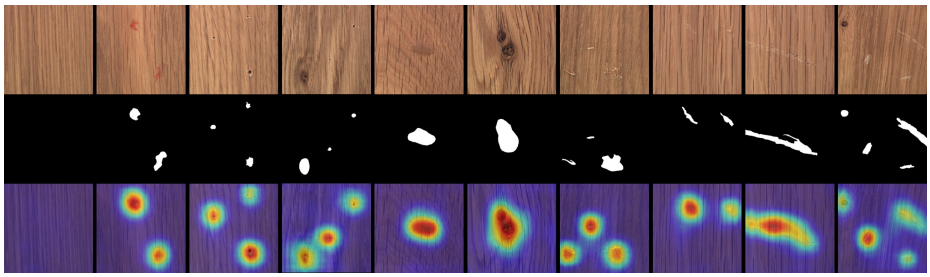


Fig. A5: Anomaly detection results of wood on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

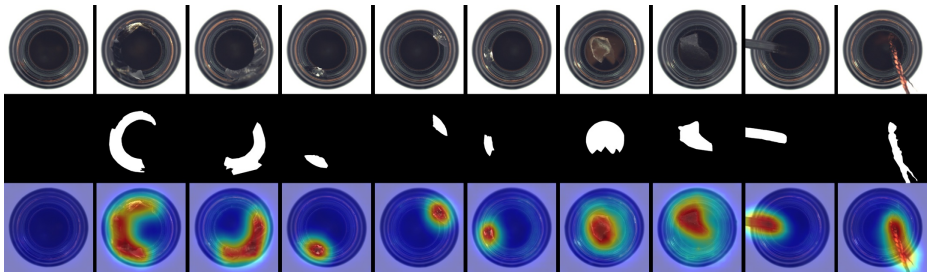


Fig. A6: Anomaly detection results of bottle on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

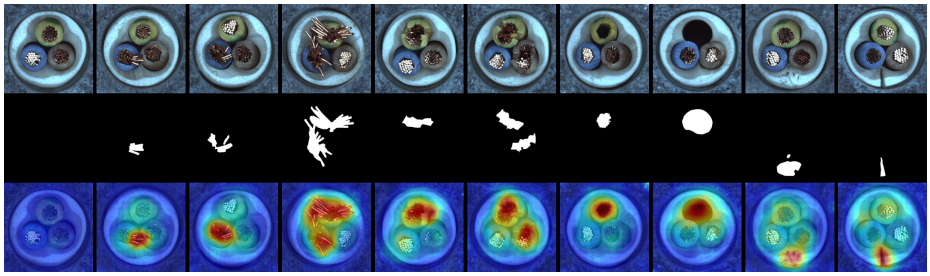


Fig. A7: Anomaly detection results of cable on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

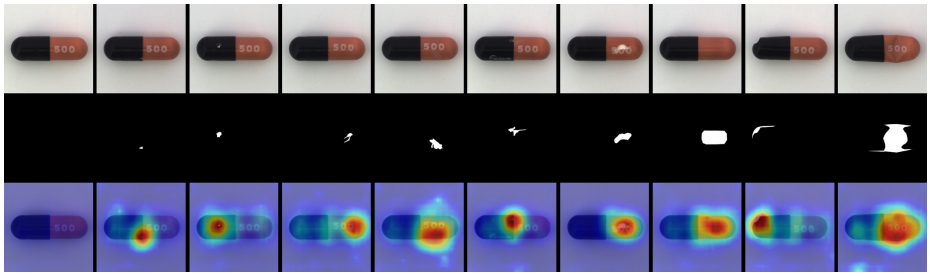


Fig. A8: Anomaly detection results of capsule on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

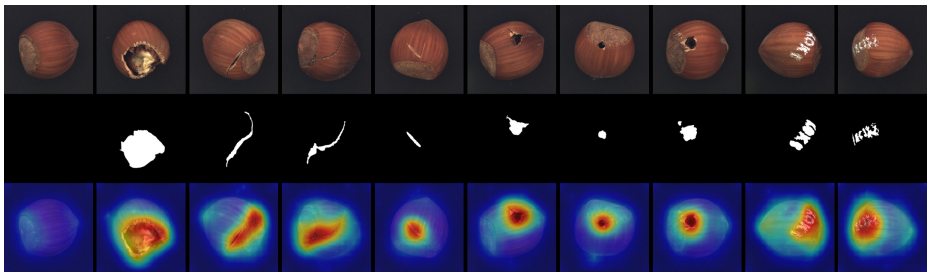


Fig. A9: Anomaly detection results of hazelnut on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

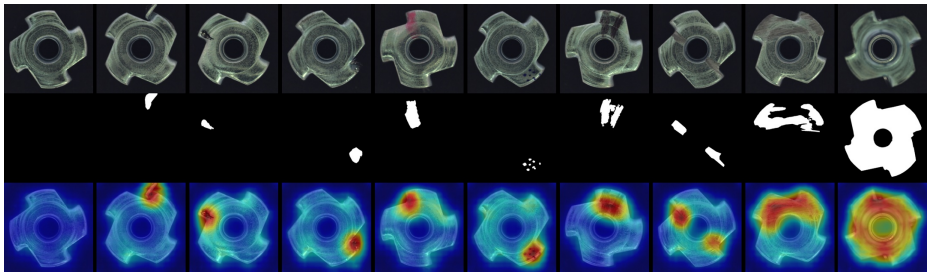


Fig. A10: Anomaly detection results of metal nut on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

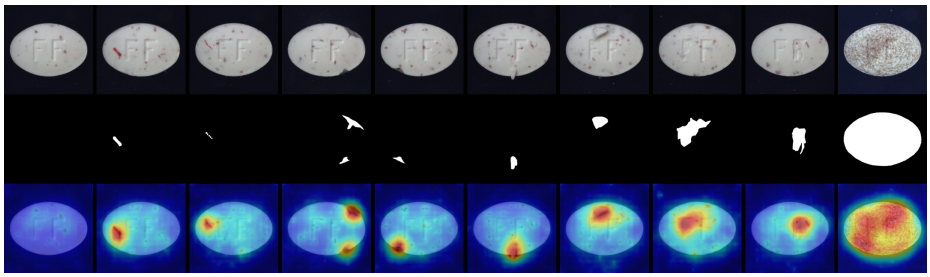


Fig. A11: Anomaly detection results of pill on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

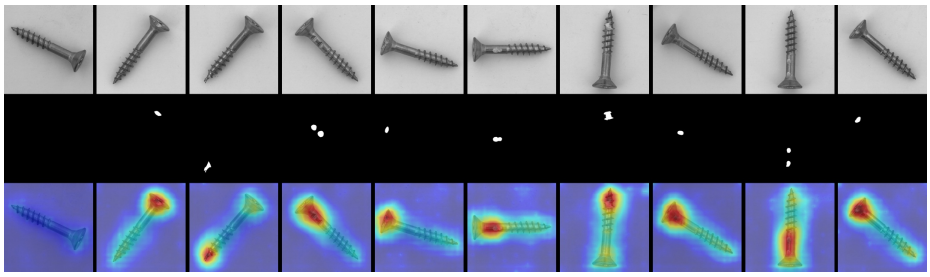


Fig. A12: Anomaly detection results of screw on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

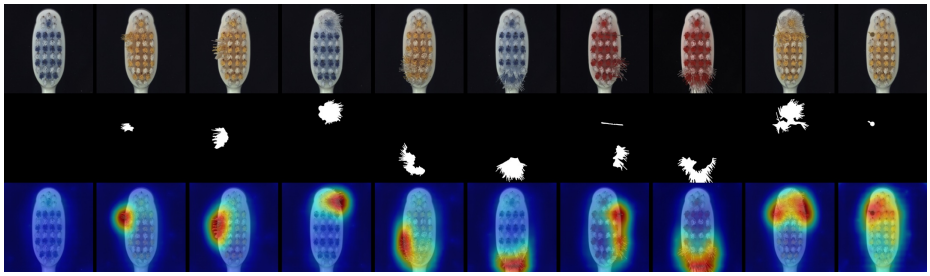


Fig. A13: Anomaly detection results of toothbrush on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

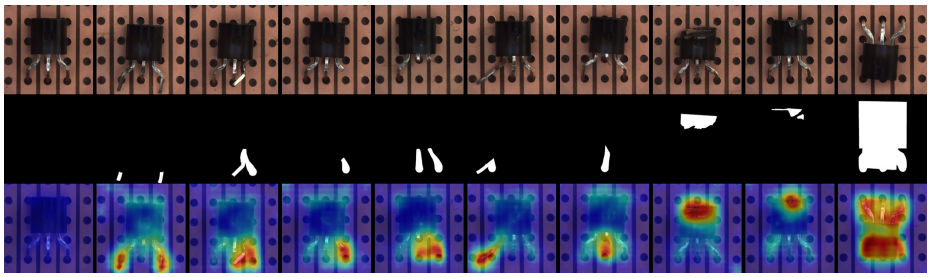


Fig. A14: Anomaly detection results of transistor on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

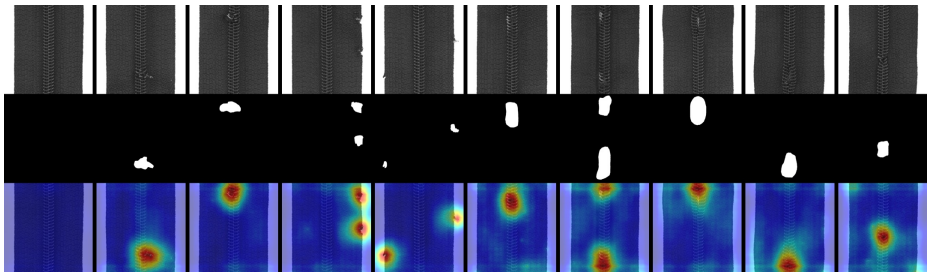


Fig. A15: Anomaly detection results of zipper on MVTec-AD. From top to down: samples, ground-truth, and the anomaly score maps of ADTR. The first column is the normal sample.

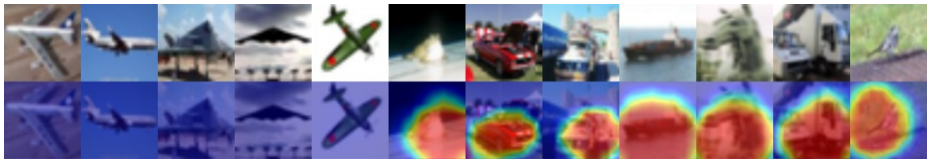


Fig. A16: Anomaly detection results of airplane on CIFAR-10. From top to down: samples and the anomaly score maps of ADTR. Images from the first column to the fifth column are normal samples.

Fig. A17: Anomaly detection results of automobile on CIFAR-10. From top to down: samples and the anomaly score maps of ADTR. Images from the first column to the fifth column are normal samples.

Fig. A18: Anomaly detection results of bird on CIFAR-10. From top to down: samples and the anomaly score maps of ADTR. Images from the first column to the fifth column are normal samples.

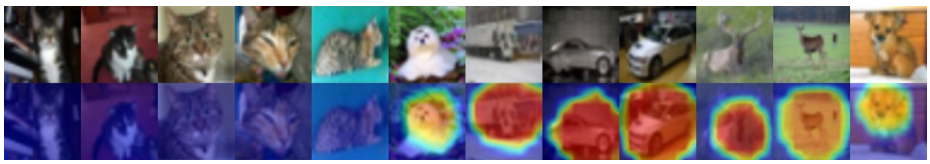


Fig. A19: Anomaly detection results of cat on CIFAR-10. From top to down: samples and the anomaly score maps of ADTR. Images from the first column to the fifth column are normal samples.

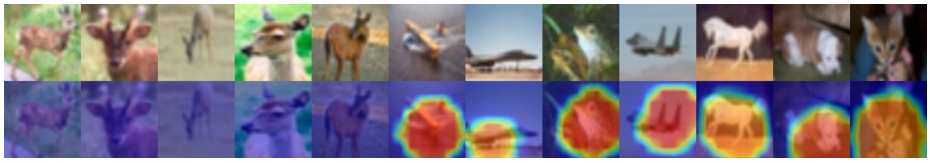


Fig. A20: Anomaly detection results of deer on CIFAR-10. From top to down: samples and the anomaly score maps of ADTR. Images from the first column to the fifth column are normal samples.

Fig. A21: Anomaly detection results of dog on CIFAR-10. From top to down: samples and the anomaly score maps of ADTR. Images from the first column to the fifth column are normal samples.

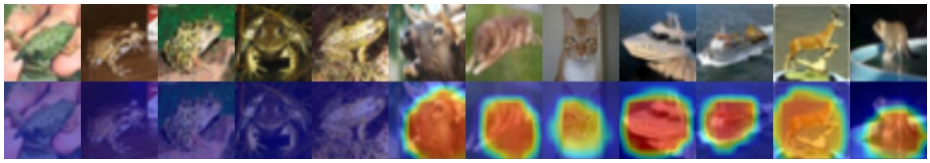


Fig. A22: Anomaly detection results of frog on CIFAR-10. From top to down: samples and the anomaly score maps of ADTR. Images from the first column to the fifth column are normal samples.

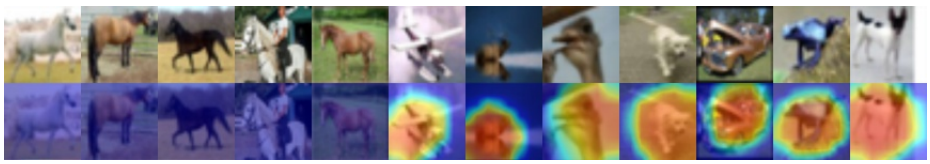


Fig. A23: Anomaly detection results of horse on CIFAR-10. From top to down: samples and the anomaly score maps of ADTR. Images from the first column to the fifth column are normal samples.

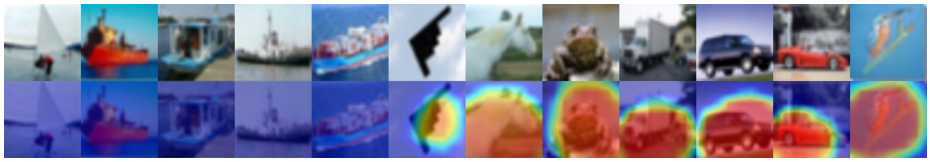


Fig. A24: Anomaly detection results of ship on CIFAR-10. From top to down: samples and the anomaly score maps of ADTR. Images from the first column to the fifth column are normal samples.

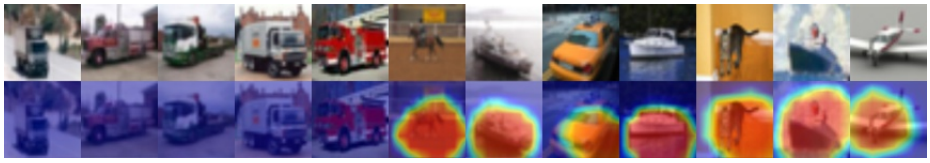


Fig. A25: Anomaly detection results of truck on CIFAR-10. From top to down: samples and the anomaly score maps of ADTR. Images from the first column to the fifth column are normal samples.