

# One-for-More: Continual Diffusion Model for Anomaly Detection

Xiaofan Li<sup>1</sup>, Xin Tan<sup>1</sup>, Zhuo Chen<sup>2</sup>, Zhizhong Zhang<sup>1</sup>, Ruixin Zhang<sup>4</sup>, Rizen Guo<sup>5</sup>, Guanna Jiang<sup>6</sup>, Yulong Chen<sup>3</sup>, Yanyun Qu<sup>2</sup>, Lizhuang Ma<sup>1,3†</sup>, Yuan Xie<sup>1†</sup>

<sup>1</sup>East China Normal University <sup>2</sup>Xiamen University

<sup>3</sup>Shanghai Jiao Tong University <sup>4</sup>Tencent YouTu Lab

<sup>5</sup>Tencent WeChatPay Lab <sup>33</sup>CATL

{funzi}@stu.ecnu.edu.cn, {lzma, yxie}@cs.ecnu.edu.cn

## Abstract

With the rise of generative models, there is a growing interest in unifying all tasks within a generative framework. Anomaly detection methods also fall into this scope and utilize diffusion models to generate or reconstruct normal samples when given arbitrary anomaly images. However, our study found that the diffusion model suffers from severe “faithfulness hallucination” and “catastrophic forgetting”, which can’t meet the unpredictable pattern increments. To mitigate the above problems, we propose a continual diffusion model that uses gradient projection to achieve stable continual learning. Gradient projection deploys a regularization on the model updating by modifying the gradient towards the direction protecting the learned knowledge. But as a double-edged sword, it also requires huge memory costs brought by the Markov process. Hence, we propose an iterative singular value decomposition method based on the transitive property of linear representation, which consumes tiny memory and incurs almost no performance loss. Finally, considering the risk of “over-fitting” to normal images of the diffusion model, we propose an anomaly-masked network to enhance the condition mechanism of the diffusion model. For continual anomaly detection, ours achieves first place in 17/18 settings on MVTec and VisA. Code is available at <https://github.com/FuNz-0/One-for-More>

## 1. Introduction

Anomaly detection (AD) has a wide array of applications in both medical and industrial fields. Traditional methods typically operate on a one-to-one paradigm, where a customized model is trained for a specific category or a kind of industrial product. The one-for-one mode severely limits the generalization of the model. Instead, the one-for-all paradigm proposes that training a single model can perform anomaly

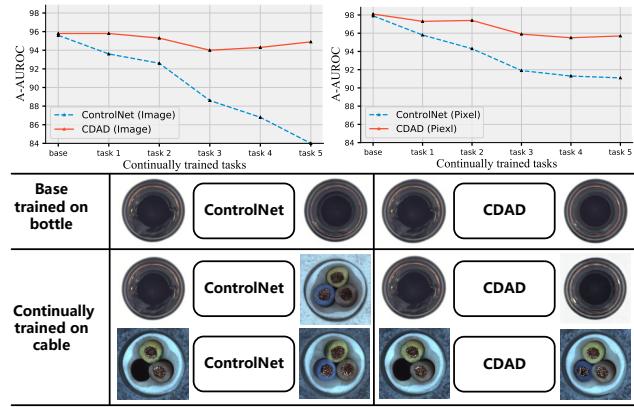


Figure 1. **Top:** Results of ControlNet and CDAD on image/pixel-level anomaly detection as the number of tasks on the MVTec continually increases. **Bottom:** “faithfulness hallucination” Problem in ControlNet for continuous anomaly detection.

detection across all classes. However, in real-world scenarios, the pattern increments are unpredictable, necessitating the model’s ability to learn continuously, which we refer to as the on-for-more paradigm. This paper focuses on the task of continuous anomaly detection and proposes a **Continual Diffusion model for Anomaly Detection (CDAD)**.

The diffusion model (DM) has been widely used in anomaly detection [10, 14, 40, 46]. Current DM-based methods generate the normal images conditioned on anomaly images through the image-to-image mechanism. The difference between the input and output of the diffusion model is used as the score of anomaly detection, which requires the diffusion model to reconstruct anomaly areas of the input image. However, our study found that the image-to-image diffusion model suffers from “catastrophic forgetting”[20] in continuously generating diverse samples. Taking ControlNet [45] as an example, as shown in Figure 1 (top), the performance on the base task (10 classes in MVTec [4]) is excellent but deteriorates seriously as the

†Corresponding author

model continues to be trained on new tasks (new classes). Furthermore, as shown in Figure 1 (bottom), the ControlNet [45], after training in the latest class, suffers from “faithfulness hallucination” [19]. Specifically, the model’s output becomes inconsistent with previously trained samples, leading to a failure in detecting anomalies. In this paper, we mitigate the above problem from two aspects: 1) use a stable parameter update strategy, and 2) strengthen the conditional mechanism of image-to-image generation.

In this paper, we propose a continual diffusion model framework that projects the updating gradient of new tasks onto a subspace orthogonal to the significant representation of previous tasks [27, 32], thus eliminating the effect of the update gradient on the previous feature space. Since the significant representation is calculated through singular value decomposition (SVD) on the matrix unfolded by intermediate features, the gradient projection operation requires a lot of memory. However, the Markov-based diffusion process significantly increases the memory overhead. Taking U-Net of stable diffusion [28] as an example, computing the significant representation of **10** images consumes almost an additional **157 GB** of memory (see Table 1), and more images need more memory. To address this issue, we propose an iterative singular value decomposition (iSVD) method based on linear transitivity theoretical property. Through the online iterative operation, iSVD only needs about **17 GB** memory consumption to calculate the significant representation of **any number** of images. This paper demonstrates and validates the effectiveness of iSVD from both theoretical and experimental perspectives.

Another issue of DM-based anomaly detection methods [14, 46] is that these methods tend to “over-fitting” normal samples rather than focusing on “reconstructing” anomalous regions, which weakens the role of the image-to-image conditioning mechanism and aggravates the hallucination problem in continuous learning. To this end, we propose the anomaly-masked network. In it, we use a CNN to encode the input image and match its size with the input of U-Net, and we also design a non-local encoder using a transformer [37] structure to perceive the global information. The neighbor-masked self-attention [42] and anomaly-masked loss are used to mask out abnormal features, which preserves normal features output by the local encoder. As a result, the diffusion model pays more attention to the reconstruction of abnormal regions while avoiding its overfitting to generate normal images. Figure 1 illustrates our great advantages, compared with ControlNet [45], CDAD can well overcome the hallucination problem and effectively alleviate the forgetting of continuous anomaly detection.

To summarize, the main contributions of this paper are:

- We propose a continual diffusion framework by modifying the gradient to alleviate the issues of forgetting and hallucination caused by continual learning.

- Iterative singular value decomposition is proposed, which can greatly reduce the memory consumption of the gradient projection.
- We propose the anomaly-masked network, which can enhance the condition mechanism of image-to-image generation to avoid the side effect of “over-fitting” in anomaly detection.
- For image-level and pixel-level anomaly detection, our method achieves first place in 17/18 continual anomaly detection settings on MVTec [4] and VisA [48].

## 2. Related Work

**Diffusion models** have achieved significant advancements in the field of image generation [16, 28, 31, 35, 45]. They are also widely used in anomaly detection tasks, with some methods [10, 14, 40, 46] utilizing diffusion models to reconstruct anomalous regions of an image. Additionally, some approaches [17, 18] use diffusion models to generate various types of defects, which then assist in training anomaly detection models.

**One-for-One Anomaly Detection** is the classical anomaly detection framework, which trains a customized model for each class to detect anomalies. The feature embedding methods [2, 7, 8, 29, 30, 41] extract the patch features of the normal image and store them as memory, then the distance between the test image features and the memory is used as anomaly scores. The knowledge distillation methods [3, 11, 33, 38, 47] compute anomaly scores by the difference between the teacher network and the student network. The reconstruction methods [9, 12, 43, 46] train the network to reconstruct abnormal regions into normal regions, and then calculate the anomaly score by the difference. In addition, some methods Chen et al. [6], Liu et al. [24], Schlüter et al. [34], Zavrtanik et al. [43] assist the model in identifying anomalies by generating anomalies. The one-for-one paradigm has achieved almost saturated performance, but the above methods are not suitable for multi-class generalization.

**One-for-All Anomaly Detection** is a recently emerging multi-class anomaly detection paradigm, which aims to train a model that can detect multiple classes. UniAD [42] is the pioneer method, which uses transformer architecture to reconstruct features and proposes neighbor-masked attention to alleviate the “identical shortcut” problem. Subsequently, many multi-class anomaly detection methods [13, 15, 25] have been proposed, which are based on the paradigm of feature reconstruction. DiAD [14] introduces the latent diffusion model [28] into multi-class anomaly detection, and proposes the semantic-guided network as the condition mechanism for the generation process. The above methods require all classes to be trained simultaneously, and their performance degrades as the number of classes increases.

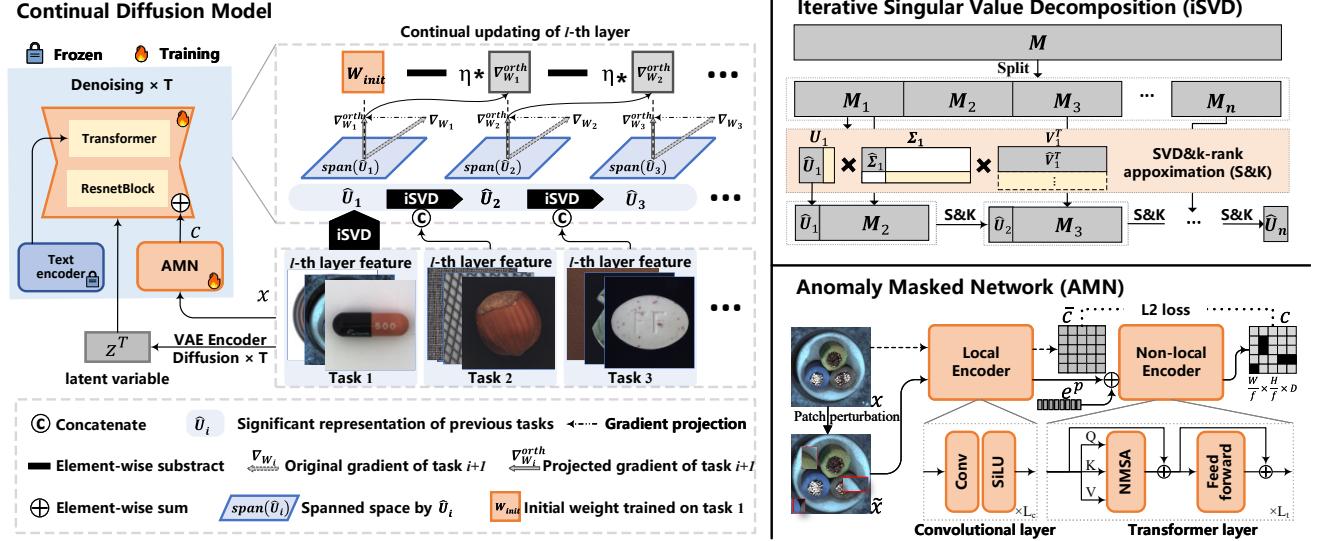


Figure 2. Illustration of CDAD. Left is the overall framework of the continual diffusion model. Right is the proposed iterative singular value decomposition and anomaly-masked network.

**One-for-More Anomaly Detection** is the focus of this paper, which enables models to continuously learn anomaly detection for new categories without backtracking on previous tasks. DNE [22] and UCAD [23] propose using an additional memory bank to store the knowledge of old tasks. The most direct issue is that as tasks increase, the memory bank gradually expands. IUF [36] introduces a semantic compression loss to retain the knowledge of previous tasks, however, when one incremental step has many classes, it will still cause a large forgetting of the previous tasks.

### 3. Method

#### 3.1. Overview

An overview of our proposed CDAD is illustrated in Figure 2, our goal is to design a continuously updated diffusion model via task-by-task training to generate or reconstruct various kinds of normal images and then measure the difference between the generated normal image and the test image, to determine whether there exists a defect. The continual diffusion model is based on a pre-trained variational auto-encoder (VAE) and controls the denoising process conditioned on input images. Note that the input of the text encoder is a space character. The continuous update is achieved by modifying the gradient with the direction that has less impact on previous tasks, such that we don't require previous samples and there is no additional inference cost. However, this projection process requires huge memory overhead of computing significant representation, so the iterative singular value decomposition (iSVD) is proposed. In addition, the anomaly-masked network is proposed masking out anomaly features while retaining normal

features to serve as conditions for the U-Net.

During the inference phase, a testing image  $x_{test}$  with unknown categories is first encoded into latent space variable by VAE encoder  $z^0 = \mathcal{E}(x_{test})$  and perform the diffusion process  $z^T = Diffusion(z^0)$ . Then  $x_{test}$  is used as the input of AMN to control denoising processes  $\hat{z} = Denoise(z^T, x_{test})$ . Finally, the image is reconstructed by VAE decoder  $\tilde{x}_{test} = \mathcal{D}(\hat{z})$ . Besides, the original image also reconstructed  $\hat{x}_{test} = \mathcal{D}(\mathcal{E}(x_{test}))$ . For anomaly localization and detection, we use pre-trained CNNs  $\phi(\cdot)$  to extract intermediate-layer feature maps, the distance of  $\hat{x}_{test}$  and  $\tilde{x}_{test}$  in position  $(h, w)$  of  $l$ -th layer feature maps can be calculated as:

$$\mathcal{M}_{h,w}^l(\hat{x}_{test}, \tilde{x}_{test}) = \|\phi^l(\hat{x}_{test}) - \phi^l(\tilde{x}_{test})\|_2^2. \quad (1)$$

Then, the anomaly score map for anomaly localization is:

$$S = \sum_{l \in \mathbb{L}} Upsample(\mathcal{M}^l(\hat{x}_{test}, \tilde{x}_{test})) \quad (2)$$

Where  $\mathbb{L}$  is the set of intermediate layers and  $Upsample(\cdot)$  is a bilinear interpolation that can upsample  $\mathcal{M}^l$  from different layers to the same size. The anomaly score is the max value of  $S$  for image-level detection.

#### 3.2. Continual Diffusion Model

The proposed continual diffusion model (CDM) uses a pre-trained VAE to project the image into the latent space for the diffusion and denoising processes. Refer to latent diffusion model [28], the objective function is given by:

$$\mathcal{L}_{CDM} = \mathbb{E}_{\mathcal{E}(x), \tilde{x}, \epsilon \sim \mathcal{N}(0, I), t} \|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(\tilde{x}))\|_2^2, \quad (3)$$

where  $\epsilon_\theta$  is U-net,  $\tau_\theta$  is the anomlay-masked network,  $\mathcal{E}(\cdot)$  is the encoder of VAE,  $z_t$  is the encoded latent variable after  $t$  step diffusion,  $\tilde{x}$  is the condition of U-net. Concretely,  $\tilde{x}$  is the original image perturbed by random patches.

Our study found that the diffusion model suffers from “catastrophic forgetting” during continual learning. As shown in Figure 2, the U-Net is mainly composed of ResnetBlock and Transformer, including the CNN and MLP layers. Since MLP and CNN can be represented as matrix multiplication, we can summarize the forgetting principle from the perspective of keeping the previous feature space unchanged after the gradient update. Taking layer  $l$  for example and omitting it, denote that  $\mathbf{X}_{pre(i)} \in \mathbb{R}^{n \times d_1}$  is the input feature for  $l$ -th layer of previous  $i$  tasks,  $\mathbf{X}_i \in \mathbb{R}^{m \times d_1}$  is the input feature of task  $i$ ,  $\mathbf{W}_{pre} \in \mathbb{R}^{d_1 \times d_2}$  is previous network parameters and  $\mathbf{O}_{pre} \in \mathbb{R}^{n \times d_2}$  is old output of previous tasks, which is obtained as:

$$\mathbf{O}_{pre(i)} = \mathbf{X}_{pre(i)} \mathbf{W}_{pre(i)}, \quad (4)$$

After training on the  $i + 1$  task, the network parameters are updated, and the output of previous  $i$  tasks is as follows:

$$\hat{\mathbf{O}}_{pre(i)} = \mathbf{X}_{pre(i)} \mathbf{W}_{i+1} = \mathbf{O}_{pre(i)} - \eta \mathbf{X}_{pre(i)} \nabla_{\mathbf{W}_i}, \quad (5)$$

where  $\eta \neq 0$  is the learning rate,  $\nabla_{\mathbf{W}_i} \in \mathbb{R}^{d_1 \times d_2}$  is the updating gradient of the task  $i + 1$ . After updating,  $\hat{\mathbf{O}}_{pre} \neq \mathbf{O}_{pre}$  which leads to the occurrence of forgetting. To solve this problem, as illustrated in Figure 2 the gradient projection is introduced to project  $\nabla_{\mathbf{W}_{i+1}}$  into an orthogonal space of  $span(\mathbf{X}_{pre(i)})$ . To obtain this orthogonal space, we first calculate the k-rank column basis  $\hat{\mathbf{U}}_i \in \mathbb{R}^{d_1 \times k}$  of the previous  $i$  tasks as follows:

$$\hat{\mathbf{U}}_i = \begin{cases} \hat{\mathbf{C}}((\mathbf{X}_i)^T), & i = 0, \\ \hat{\mathbf{C}}([\hat{\mathbf{U}}_{i-1}, (\mathbf{X}_i)^T]), & i > 0, \end{cases} \quad (6)$$

where  $[\cdot]$  denotes concatenate operation,  $\hat{\mathbf{C}}(\mathbf{M})$  denotes k-rank column basis of  $\mathbf{M}$ , calculated by SVD followed k-rank approximation. As shown in Figure 2, for stable continual learning, CDM projects the original gradient of the task  $i + 1$  onto the orthogonal space of  $span(\hat{\mathbf{U}}_i)$ :

$$\nabla_{\mathbf{W}_i}^{orth} = \nabla_{\mathbf{W}_i} - (\hat{\mathbf{U}}_i)(\hat{\mathbf{U}}_i)^T \nabla_{\mathbf{W}_i}, \quad (7)$$

where  $\mathbf{X}_{pre(i)} \nabla_{\mathbf{W}_i}^{orth} \approx \mathbf{0}$ . The initial task is updated with the original gradient, and the incremental tasks are updated with the projected gradient. Thus, we can deduce that:

$$\hat{\mathbf{O}}_{pre(i)} = \mathbf{X}_{pre(i)} \mathbf{W}_i - \eta \mathbf{X}_{pre(i)} \nabla_{\mathbf{W}_i}^{orth} \approx \mathbf{O}_{pre(i)}, \quad (8)$$

where the updated model on the new task does not perturb the output space of the previous task, thus ensuring the model retains the old knowledge.

---

### Algorithm 1 SVD followed by k-rank approximation

---

```
# M: Feature matrix, d x m
# th: Threshold
def S_and_K(M, th):
    # U: d x d, S: d, Vh: m x m
    U, S, Vh = torch.linalg.svd(M)
    total = (S ** 2).sum()
    ratio = (S ** 2) / total
    k = (torch.cumsum(ratio) < th).sum()
    U_hat = U[:, :k]
    return U_hat
```

---

### 3.3. Iterative Singular Value Decomposition

The traditional SVD method used in Eq (6) needs the global calculation to calculate the significant representation, thus it requires enough memory to store intermediate features. However, the Markov-based denoising process greatly increases the size of intermediate features, according to our statistics, it requires almost 157 GB of memory to compute significant representation for only ten images (see Table 1), which is an unaffordable memory cost. Therefore, we propose an iterative SVD (iSVD) method, which can be used to calculate the significant representation of a large matrix and retain the original properties with a small memory overhead. Specifically, we denote the input matrix of Eq (6) as  $\mathbf{M} \in \mathbb{R}^{d \times \Lambda}$ , where  $d$  and  $\Lambda$  denote the dimension and number of features and  $\Lambda \gg d$ . As illustrated in Figure 3.3,  $\mathbf{M}$  is split into  $\{\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n\}$ , where  $\mathbf{M}_i \in \mathbb{R}^{d \times m}$ ,  $m = \frac{\Lambda}{n}$ . The significant representation  $\hat{\mathbf{U}}_1 \in \mathbb{R}^{d \times k}$  of  $\mathbf{M}_1$  is calculated by SVD followed by its k-rank approximation according to the threshold,  $\gamma_{th}$ :

$$\begin{aligned} \mathbf{M}_1 &= \mathbf{U}_1 \boldsymbol{\Sigma}_1 \mathbf{V}_1^T, \mathbf{U}_1 \in \mathbb{R}^{d \times d}, \boldsymbol{\Sigma}_1 \in \mathbb{R}^{d \times m}, \\ (\mathbf{M}_1)_k &= \hat{\mathbf{U}}_1 \hat{\boldsymbol{\Sigma}}_1 \hat{\mathbf{V}}_1^T, \hat{\mathbf{U}}_1 \in \mathbb{R}^{d \times k}, \hat{\boldsymbol{\Sigma}}_1 \in \mathbb{R}^{k \times k}, \\ \text{s.t. } \|(\mathbf{M}_1)_k\|_F^2 &\geq \gamma_{th} \|\mathbf{M}_1\|_F^2, \end{aligned} \quad (9)$$

where  $\|\cdot\|_F^2$  denotes F-norm,  $\mathbf{U}_1 = [u^1, u^2, \dots, u^d], u^i \in \mathbb{R}^d$ , and the significant representation  $\hat{\mathbf{U}}_1 = [u^1, u^2, \dots, u^k]$ , selected by the  $k$  vectors with highest singular values in  $\mathbf{U}_1$ , the pytorch implementation is shown in Algorithm 1. Then,  $\hat{\mathbf{U}}_1$  is concatenated with  $\mathbf{M}_2$  to continue S&K operation iteratively. Finally, the overall significant representation  $\hat{\mathbf{U}}_n$  is obtained to calculate the orthogonal gradient of the new task. Next, we will prove that  $\hat{\mathbf{U}}_n$  obtained through iSVD is a significant representation of  $\mathbf{M}$  and give the memory saving rating.

**Lemma 1** linear transitivity: Given matrices  $\mathbf{S}_1$ ,  $\mathbf{S}_2$ , and  $\mathbf{S}_3$ , if  $\mathbf{S}_1$  can be linearly expressed by  $\mathbf{S}_2$ , and  $\mathbf{S}_2$  can be linearly expressed by  $\mathbf{S}_3$ , then  $\mathbf{S}_1$  can be linearly expressed by  $\mathbf{S}_3$ .

**Proof 1** For ease of understanding, we ignore the approximate information loss, assuming that the k-rank approxi-

mation picks all vectors with nonzero singular values:

$$\mathbf{M}_1 = \mathbf{U}_1 \Sigma_1 \mathbf{V}^T = [\hat{\mathbf{U}}_1 \mathbf{U}_1^0] [\begin{array}{cc} \hat{\Sigma}_1 & 0 \\ 0 & 0 \end{array}] \mathbf{V}_1^T, \quad (10)$$

where  $\hat{\mathbf{U}}_1 \in \mathbb{R}^{d \times k_1}$ ,  $\hat{\Sigma}_1 \in \mathbb{R}^{k_1 \times k_1}$ ,  $\text{rank}(\mathbf{M}_1) = k_1$ , and  $\hat{\mathbf{U}}_1$  is an orthonormal basis of  $\mathbf{M}_1$ , thus  $\mathbf{M}_1$  can be linearly expressed by  $\hat{\mathbf{U}}_1$ .

$$[\hat{\mathbf{U}}_1 \mathbf{M}_2] = \mathbf{U}_2 \Sigma_2 \mathbf{V}_2^T = [\hat{\mathbf{U}}_2 \mathbf{U}_2^0] [\begin{array}{cc} \hat{\Sigma}_2 & 0 \\ 0 & 0 \end{array}] \mathbf{V}_2^T, \quad (11)$$

where  $\hat{\mathbf{U}}_2 \in \mathbb{R}^{d \times k_2}$ ,  $\hat{\Sigma}_2 \in \mathbb{R}^{k_2 \times k_2}$ ,  $\text{rank}([\hat{\mathbf{U}}_1; \mathbf{M}_2]) = k_2$ , similarly,  $\hat{\mathbf{U}}_1$  and  $\mathbf{M}_2$  can be linearly expressed by  $\hat{\mathbf{U}}_2$ . According to Lemma 1,  $\mathbf{M}_1$  can also be linearly expressed by  $\hat{\mathbf{U}}_2$ .

By the same reasoning, it can be proved that  $\mathbf{M} = [\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n]$  can be linearly expressed by  $\hat{\mathbf{U}}_n$ , thus  $\hat{\mathbf{U}}_n$  is a significant representation of  $\mathbf{M}$ .

**Memory saving rate.** Theoretically, the memory saving rate of iSVD over SVD is about  $\frac{n^2 - 1}{n^2}$  (see supplementary). Considering memory sharing and other factors, there are some changes in practice. Table 1 shows the actual memory usage of SVD and iSVD for computing the significant representation of ten images. When  $n = 10$  the memory is saved by 89.4%. In this paper, we set  $n$  to be the image number of the previous task. Therefore, the memory consumption of iSVD does not expand as the number of images increases.

	<b>SVD</b>		<b>iSVD</b>	
<i>n</i>	1	2	5	10
Memory	157.3GB	69.9GB	30.3GB	<b>16.7GB</b>
Saving rate	0%	55.6%	80.7%	89.4%

Table 1. Memory consumption and memory saving rate of iSVD with different number of split blocks  $n$ .

### 3.4. Anomaly-Masked Network

The image-to-image diffusion models in anomaly detection face the problems of "identical shortcuts" and "overfitting". To address the above two issues, we propose the anomaly-masked conditional network (AMN) to control anomaly-to-normal generation. As shown in Figure 2, AMN comprises local and non-local encoders. The local encoder is CNNs with SiLU activation, which is used to encode the local features. Given an input image  $x \in \mathbb{R}^{H \times W \times 3}$ ,  $H$  and  $W$  denote its height and width, the image is initially encoded by local encoder  $\bar{c} = E_{lo}(x)$ , where  $\bar{c} \in \mathbb{R}^{\frac{H}{f} \times \frac{W}{f} \times D}$ , which downsamples by a factor of  $f$  and the shape of  $\bar{c}$  is consistent with the input shape of U-Net. The non-local encoder is composed of the transformer encoder layer and

is used to mask out abnormal regions through the perception of non-local features. The self-attention module is replaced by neighbor-masked self-attention (NMSA) [42], which aims to enhance the role of non-local features in feature encoding. Concretely, the locally encoded feature map  $\bar{c}$  is added with the sine positional embedding  $e_p$  and then processed through the non-local encoder  $c = E_{no}(\bar{c} + e_p)$ , where  $c \in \mathbb{R}^{\frac{H}{f} \times \frac{W}{f} \times D}$  is the reprocessed  $\bar{c}$ . Patch perturbation is introduced to make the non-local encoder mask out abnormal features and retain normal features. We perturb the original image  $x$  to obtain abnormal images  $\tilde{x}$ , both of which are input to AMN and constrained by anomaly-masked loss:

$$\mathcal{L}_{amn} = \mathbb{E}_{x, \tilde{x}} \|E_{no}(E_{lo}(\tilde{x}) + e_p) - E_{lo}(x)\|_2^2, \quad (12)$$

which can mask out the anomaly region and retain normal region features. So that the denoising process pays more attention to the reconstruction of anomaly regions.

## 4. Experiments

**Dataset.** In this paper, the datasets we use are MVTec [4] and VisA [48]. Both datasets contain multiple classes. In particular, MVTec contains 15 classes with  $700^2 - 900^2$  pixels per image, and VisA contains 12 classes with roughly  $1.5K \times 1K$  pixels per image. The training set contains only normal images, while the test set contains normal images and anomaly images with image-level and pixel-level annotations. These two datasets contain intricate instances that are widely acknowledged for their real-world applicability.

**Protocol.** Referring to IUF[36], we conducted comparative experiments between our approach and the other methods under different continual anomaly detection settings. Specifically, in MVTec-AD [4], we performed the subsequent four settings:  $14 - 1$  with 1 step,  $10 - 5$  with 1 step,  $3 \times 5$  with 5 steps, and  $10 - 1 \times 5$  with 5 steps. In VisA [48], we established three settings:  $11 - 1$  with 1 step,  $8 - 4$  with 1 step, and  $8 - 1 \times 4$  with 4 steps, . In addition, we completed the cross-dataset continuous anomaly detection, including the MVTec to VisA increment and the VisA to MVTec increment.

**Evaluation metrics.** We follow the literature [4] in reporting the Area Under the Receiver Operation Characteristic (AUROC) for both image-level and pixel-level anomaly detection. To measure the performance of the model in continuous learning, referring to DNE [22], we calculated the average AUROC (A-AUROC) and the forgetting measure (FM) for  $N$  continual steps.

**Implementation details.** CDAD is based on the pre-trained Stable Diffusion with a pre-trained VAE, and the VAE does not need further fine-tuning. The input images have all resized of  $256 \times 256$ , and the input to the text encoder is a space character. For the Anomaly-Masked Network, the

Method	14 – 1 with 1 Step		10 – 5 with 1 Step		3 × 5 with 5 Steps		10 – 1 × 5 with 5 Steps	
	A-AUROC (↑)	FM (↓)	A-AUROC (↑)	FM (↓)	A-AUROC (↑)	FM (↓)	A-AUROC (↑)	FM (↓)
UniAD[42]	85.7 / 89.6	18.3 / 13.3	86.7 / 91.5	14.9 / 10.6	81.3 / 88.7	7.4 / 10.6	76.6 / 82.3	21.1 / 17.3
UniAD + EWC[20]	92.8 / 95.4	4.1 / 1.9	90.5 / 93.6	7.3 / 4.2	79.6 / 89.0	9.5 / 10.1	89.6 / 93.8	5.4 / 3.6
UniAD + SI[44]	85.7 / 89.5	18.4 / 13.4	84.1 / 88.3	20.2 / 17.0	81.9 / 88.5	7.0 / 10.8	77.9 / 82.0	19.5 / 17.7
UniAD + MAS[1]	85.8 / 89.6	18.1 / 13.3	86.8 / 91.0	14.9 / 11.6	81.5 / 89.0	7.2 / 10.2	77.9 / 82.0	19.5 / 17.7
UniAD + LVT[39]	80.4 / 86.0	29.1 / 20.6	87.1 / 90.6	14.1 / 12.3	80.4 / 88.6	8.6 / 10.6	78.2 / 88.3	19.1 / 16.1
UCAD*[23]	93.8 / 95.7	1.8 / 0.3	88.7 / 93.1	5.2 / <u>3.7</u>	84.8 / 90.1	10.3 / 9.2	91.2 / 94.0	6.3 / 3.0
IUF[36]	96.0 / 96.3	<u>1.0</u> / 0.6	92.2 / 94.4	9.3 / 4.3	84.2 / 91.1	10.0 / 8.4	94.2 / 95.1	3.2 / 1.0
DiAD†[14]	93.5 / 96.5	1.7 / 0.1	91.9 / 93.9	3.2 / 3.9	80.5 / 89.0	11.8 / <u>7.2</u>	83.3 / 92.7	12.2 / 3.6
ControlNet†[45]	92.6 / 96.4	3.6 / <u>0.1</u>	92.7 / 93.8	<u>2.9</u> / 4.0	79.2 / 89.6	13.9 / 7.3	82.6 / 91.8	13.5 / 4.5
<b>CDAD†</b>	<b>96.4 / 96.6</b>	<b>-0.57 / -0.04</b>	<b>94.2 / 95.3</b>	<b>2.05 / 2.40</b>	<b>89.1 / 92.2</b>	<b>3.69 / 4.65</b>	<b>94.9 / 95.7</b>	<b>1.01 / 0.74</b>

Table 2. Image-level/pixel-level results of our method on MVTec under 4 continual anomaly detection settings. The best and second-best results are marked in **bold** and underline. † indicates DM-based methods, and \* indicates memory-limited.

Method	11 – 1 with 1 Step		8 – 4 with 1 Step		8 – 1 × 4 with 4 Steps	
	A-AUROC (↑)	FM (↓)	A-AUROC (↑)	FM (↓)	A-AUROC (↑)	FM (↓)
UniAD[42]	75.0 / 92.1	22.4 / 11.4	78.1 / 94.0	14.7 / 8.4	72.2 / 90.8	16.6 / 9.2
UniAD + EWC[20]	78.7 / 95.4	14.9 / 4.8	80.5 / 95.4	10.0 / 5.3	72.3 / 92.3	16.5 / 7.3
UniAD + SI[44]	78.1 / 92.0	16.9 / 11.5	69.8 / 88.5	9.2 / 8.3	69.8 / 88.5	19.8 / 12.0
UniAD + MAS[1]	75.4 / 91.8	21.5 / 11.9	72.1 / 90.6	14.1 / 8.4	72.1 / 90.6	16.7 / 9.4
UniAD + LVT[39]	77.5 / 92.3	17.3 / 10.9	70.8 / 91.4	13.4 / 8.1	70.8 / 94.4	18.3 / 8.4
UCAD*[23]	85.9 / 94.5	2.7 / 0.6	79.9 / 94.2	8.9 / 4.8	78.8 / 93.5	10.4 / <u>5.4</u>
IUF[36]	<u>87.3</u> / <b>97.6</b>	<u>2.4</u> / 1.8	<u>80.1</u> / <u>95.4</u>	9.8 / 6.8	<u>79.8</u> / <u>95.0</u>	<u>9.8</u> / 6.8
DiAD†[14]	70.2 / 91.1	5.6 / 2.2	67.4 / 91.7	7.7 / 1.6	65.0 / 84.9	11.2 / 8.8
ControlNet†[45]	82.4 / 92.3	2.6 / 0.2	76.3 / 91.8	<u>5.8</u> / 0.4	70.0 / 88.7	13.8 / 5.8
<b>CDAD†</b>	<b>88.3 / 97.2</b>	<b>-1.28 / -0.06</b>	<b>85.3 / 97.1</b>	<b>1.39 / 0.29</b>	<b>83.4 / 95.8</b>	<b>3.79 / 1.49</b>

Table 3. Image-level/pixel-level results of our method on VisA under 3 continual anomaly detection settings. The best and second-best results are marked in **bold** and underline. † indicates DM-based methods, and \* indicates memory-limited.

Method	MVTec-VisA		VisA-MVTec	
	A-AUROC (↑)	FM (↓)	A-AUROC (↑)	FM (↓)
IUF	82.4 / 92.5	11.09 / 4.88	74.3 / 89.7	16.74 / <u>5.54</u>
DiAD	<u>85.5</u> / <u>93.0</u>	<u>10.24</u> / <u>4.15</u>	78.1 / <u>92.2</u>	12.54 / 5.89
ControlNet	85.3 / 91.7	10.35 / 4.73	<u>78.9</u> / 86.2	<u>12.07</u> / 10.68
<b>CDAD</b>	<b>90.0 / 94.9</b>	<b>4.71 / 1.78</b>	<b>84.2 / 94.1</b>	<b>6.90 / 3.45</b>

Table 4. Image-level/pixel-level results on cross-dataset continual anomaly detection.

depth of both the local encoder and the non-local encoder is 8. In the continuous learning process, the initial task is trained for 500 epochs, and each incremental task is trained for 100 epochs. During the inference phase, the Denoising Diffusion Implicit Model [35] (DDIM) is employed as the sampler with a default of 10 steps, and the pre-trained CNN we used to extract features is a ResNet50 pre-trained on ImageNet.

## 4.1. Comparison Results

The image/pixel-level experimental results of CDAD and current methods on continual anomaly detection are recorded in Table 2 3, where UniAD[42] and DiAD[14] are multi-class anomaly detection methods, EWC[20], SI[44], MAS[1] and LVT[39] are traditional continual learning method. IUF[36] and UCAD[23] are incremental frameworks for continual anomaly detection. We also replicate the results of DiAD[14] and ControlNet[45] on continual anomaly detection. As illustrated in Table 2 3, after incorporating traditional continuous learning methods, the results of UniAD showed some improvement but were still below the SOTA level, the DM-based methods achieve good results when the number of steps is small, while deteriorate and suffer “catastrophic forgetting” when the number of steps is larger. Compared with the above methods, IUF has achieved better results overall, but in some tasks, such as setting 3 on MVTec and setting 6 on visa, it still suffers from forgetting problems. Regarding our method, CDAD achieved all the best results on MVTec, especially on set-

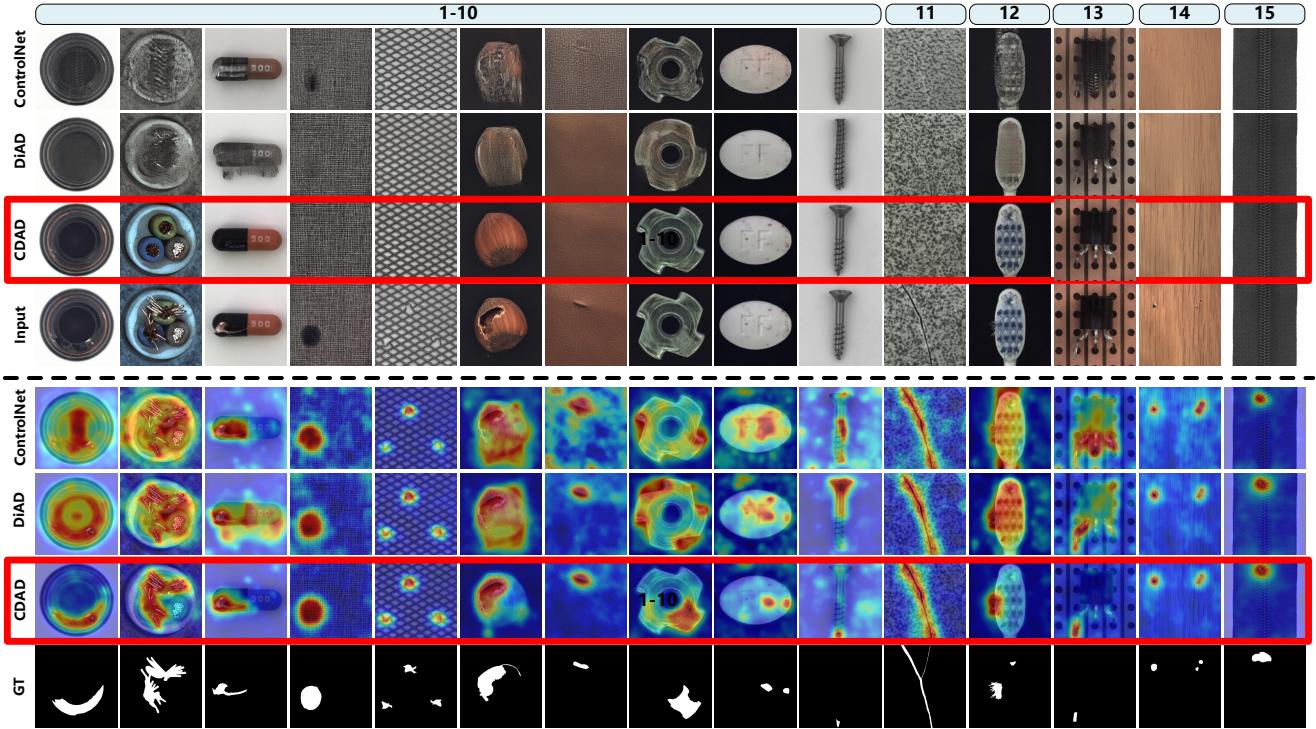


Figure 3. Qualitative comparison results under setting 4 of MVTec, the numbers represent continual training steps. The **top** is the reconstruction result of the anomaly image, and the **bottom** is the heat map of the anomaly location.

ting 3, which increased by 4.3/1.1 compared to the second place, and the forgetting rate was also significantly reduced by 6.31/3.75. On the VisA dataset, in addition to the pixel level result of setting 5, CDAD achieves the best results on other settings and improves 5.2/1.7 and 3.6/0.8 compared to the SOTA level of setting 6 and setting 7, respectively. The recorded results indicate that CDAD exhibits excellent anomaly detection performance and a significant anti-forgetting capability. Notably, in MVTec setting 1 and VisA setting 5, CDAD not only mitigates the issue of forgetting but also demonstrates enhanced performance on previous tasks, which suggests that CDAD can improve the overall generalization of the model with a limited number of incremental tasks.

## 4.2. Cross-Dataset Continual Learning

Table 4 shows the results of cross-dataset continual anomaly detection, MVTec–VisA represents base training on MVTec followed by incremental training on VisA. Due to the memory efficiency, CDAD enables continual learning on large-scale datasets. As recorded, CDAD outperforms the second-best method by 4.5/1.9 and 5.3/1.9 on two cross-dataset settings, respectively. In addition, CDAD also achieves the optimal cross-dataset forgetting rate, which indicates its excellent stability for large-scale incremental training.

## 4.3. Visualization Results

Figure 3 shows the quantitative results. It can be seen that on the last task, all methods can show excellent anomaly reconstruction and anomaly location effects, however, ControlNet [45] and DiAD [14] have serious illusion on the reconstruction results of the previous task, and the results of anomaly localization also collapse, especially on the object class. Our method shows strong stability in continuous learning, which shows excellent anomaly reconstruction and localization effects not only in the latest task but also in previous tasks. This indicates that our method effectively mitigates the hallucination problem caused by continual learning in diffusion models.

## 4.4. Ablation Study

We verify the impact of different modules on the overall performance of CDAD under setting 4 on MVTec [4]. These include the continual diffusion model (CDM) and anomaly-masked network (AMN). To better verify the effectiveness of our method, we use ControlNet [45] as the baseline for comparison. Results of the ablation study are recorded in Table 5.

**Continual Diffusion Model.** The continual diffusion model (CDM) is proposed to better retain the model’s knowledge of previous tasks during continual learning,

Step	1-10	11	12	13	14	15		FM
ControlNet	<b>95.8 / 97.8</b>	92.4 / 95.8	93.2 / 96.2	87.4 / 91.0	86.9 / 91.2	82.6 / 91.8		13.46 / 4.46
w/o AMN	94.7 / 96.3	94.9 / 96.3	94.7 / 96.3	93.8 / 94.0	94.0 / 93.6	94.3 / 94.0		<u>1.69 / 1.55</u>
w/o CDM	<b>95.8 / 98.1</b>	93.2 / 96.6	93.1 / 96.4	87.9 / 92.4	88.3 / 91.4	90.6 / 93.5		5.36 / 2.55
CDAD	<b>95.8 / 98.1</b>	<b>95.8 / 97.3</b>	<b>95.3 / 97.4</b>	<b>94.0 / 95.5</b>	<b>94.3 / 95.1</b>	<b>94.9 / 95.7</b>		<b>1.01 / 0.74</b>

Table 5. Ablation study results (image-level/pixel-level A-AUROC ( $\uparrow$ ) and FM ( $\downarrow$ ) on setting 4 of MVTec.

thereby alleviating the problem of forgetting. As shown in Table 5, after removing CDM, the performance of the model in the new task gradually decreases with the increase of steps, and the final forgetting rate also increases by 4.35/1.81. However, compared with ControlNet, the forgetting rate of the model without CMD (only use AMN) decreased by 8.1/1.9, which is a significant improvement.

**Anomaly-Masked Network.** The anomaly-masked network (AMN) is proposed as a condition mechanism for the denoising process. In image-to-image tasks, the anomalous region in the input image is masked to retain the normal region, allowing the diffusion model to focus more effectively on reconstructing the abnormal area. As illustrated in Table 5, without AMN, the model’s results have a 1.1/1.8 decrease on the initial step and a 0.6/1.7 decrease after 5 steps of continuous learning. and finally. Interestingly, the use of AMN reduces the forgetting rate of the Model by 0.68/0.81, suggesting that AMN not only enhances the performance of diffusion models in anomaly detection but also better combats forgetting.

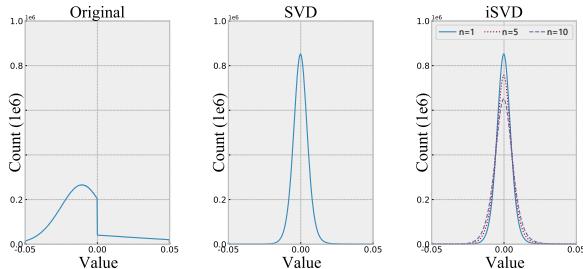


Figure 4. Value distribution of original matrix. Value distribution after projected by SVD and iSVD.  $n$  is the number of split blocks.

#### 4.5. Validation Experiments of iSVD

We complete the experiments to verify the effectiveness of iSVD. Specifically, we expand an intermediate feature map of U-Net into a matrix, using SVD and iSVD to project this matrix into its orthogonal space. Figure 4 illustrates the value distribution of the matrix before and after projection, the matrix size is  $1280 \times 50000$  and the threshold for k-rank approximation is 0.98. It can be seen that the distribution of features is relatively uniform before projection, and after SVD projection, the distribution of values is concentrated around 0. iSVD shows almost the same performance as

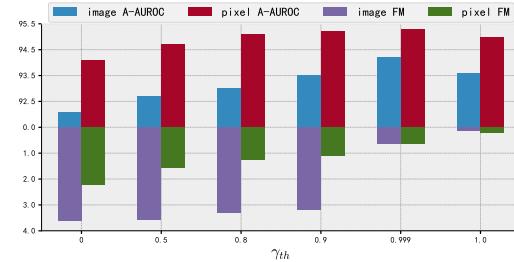


Figure 5. Impact of  $\gamma_{th}$  on A-AUROC ( $\uparrow$ ) and FM ( $\downarrow$ ) under setting 2 of MVTec.

SVD, which verifies that the significant representation calculated by iSVD is effective. In addition, we also find that with the increase of the number of split blocks, the effect of iSVD becomes worse, this is because the k-rank approximate is performed at each iteration, which loses part of the information. However, continuous learning can not project gradients 100% into the orthogonal space of the previous task, which will harm the model’s plasticity to new tasks, so the information loss of iSVD is acceptable.

#### 4.6. Hyper-parameter Analysis

We complete the effect of the threshold  $\gamma_{th}$  of k-rank approximation.  $\gamma_{th}$  is the threshold for selecting the significant representation in the unitary matrix, used to control the amount of information retained from previous tasks. Figure 4 illustrates the effect of  $\gamma_{th}$  under setting 2 of MVTec. As shown, when  $\gamma_{th} = 0$ , the model experiences significant forgetting and performance degradation. As  $\gamma_{th}$  increases, both the forgetting rate and performance improve. When  $\gamma_{th} = 1.0$ , although the forgetting rate is minimal, performance drops due to the model retaining all column bases from previous tasks, which interferes with learning new tasks. Even when  $\gamma_{th} = 0.999$ , the selected vectors are only a small subset of the unitary matrix, preserving the core knowledge of previous tasks while maintaining the model’s plasticity to new tasks. Therefore, we set  $\gamma = 0.999$ .

#### 5. Conclusion

In this paper, we propose a continual diffusion model for anomaly detection termed CDAD, which continually learns anomaly detection for new tasks. First, we introduce a continual diffusion model framework that uses significant

representation to project the training gradient of the new task to a direction having less influence on the previous tasks, to ensure the stability of the model. Second, we propose the iterative singular value decomposition, which greatly alleviates the memory pressure of computing the significant representation for diffusion models. Finally, an anomaly-masked network is proposed as a control mechanism for the diffusion model. The experimental results show that both the anomaly detection performance and anti-forgetting capability of our method are state-of-the-art.

## References

- [1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision*, pages 139–154, 2018. [6](#)
- [2] Jaehyeok Bae, Jae-Han Lee, and Seyun Kim. Pni: industrial anomaly detection using position and neighborhood information. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6373–6383, 2023. [2](#)
- [3] Kilian Batzner, Lars Heckler, and Rebecca König. Efficientad: Accurate visual anomaly detection at millisecond-level latencies. *arXiv preprint arXiv:2303.14535*, 2023. [2](#)
- [4] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019. [1, 2, 5, 7](#)
- [5] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4183–4192, 2020. [1](#)
- [6] Qiyu Chen, Huiyuan Luo, Chengkan Lv, and Zhengtao Zhang. A unified anomaly synthesis strategy with gradient ascent for industrial anomaly detection and localization. *arXiv preprint arXiv:2407.09359*, 2024. [2](#)
- [7] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020. [2](#)
- [8] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2021. [2](#)
- [9] Zheng Fang, Xiaoyang Wang, Haocheng Li, Jiejie Liu, Qiguai Hu, and Jimin Xiao. Fastrecon: Few-shot industrial anomaly detection via fast feature reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17481–17490, 2023. [2](#)
- [10] Matic Fučka, Vitjan Zavrtanik, and Danijel Skočaj. Transfusion—a transparency-based diffusion model for anomaly detection. In *Proceedings of the European conference on computer vision*, pages 91–108. Springer, 2025. [1, 2](#)
- [11] Zhihao Gu, Liang Liu, Xu Chen, Ran Yi, Jiangning Zhang, Yabiao Wang, Chengjie Wang, Annan Shu, Guannan Jiang, and Lizhuang Ma. Remembering normality: Memory-guided knowledge distillation for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16401–16409, 2023. [2](#)
- [12] Hewei Guo, Liping Ren, Jingjing Fu, Yuwang Wang, Zhizheng Zhang, Cuiling Lan, Haoqian Wang, and Xinwen Hou. Template-guided hierarchical feature restoration for anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6447–6458, 2023. [2](#)
- [13] Haoyang He, Yuhu Bai, Jiangning Zhang, Qingdong He, Hongxu Chen, Zhenye Gan, Chengjie Wang, Xiangtai Li, Guanzhong Tian, and Lei Xie. Mambaad: Exploring state space models for multi-class unsupervised anomaly detection. *arXiv preprint arXiv:2404.06564*, 2024. [2](#)
- [14] Haoyang He, Jiangning Zhang, Hongxu Chen, Xuhai Chen, Zhishan Li, Xu Chen, Yabiao Wang, Chengjie Wang, and Lei Xie. A diffusion-based framework for multi-class anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8472–8480, 2024. [1, 2, 6, 7, 3](#)
- [15] Liren He, Zhengkai Jiang, Jinlong Peng, Liang Liu, Qiang-gang Du, Xiaobin Hu, Wenbing Zhu, Mingmin Chi, Yabiao Wang, and Chengjie Wang. Learning unified reference representation for unsupervised multi-class anomaly detection. *arXiv preprint arXiv:2403.11561*, 2024. [2](#)
- [16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. [2](#)
- [17] Jie Hu, Yawen Huang, Yilin Lu, Guoyang Xie, Guannan Jiang, Yefeng Zheng, and Zhichao Lu. Anomalyxfusion: Multi-modal anomaly synthesis with diffusion. *arXiv preprint arXiv:2404.19444*, 2024. [2](#)
- [18] Teng Hu, Jiangning Zhang, Ran Yi, Yuzhen Du, Xu Chen, Liang Liu, Yabiao Wang, and Chengjie Wang. Anomalydiffusion: Few-shot anomaly image generation with diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8526–8534, 2024. [2](#)
- [19] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023. [2](#)
- [20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. [1, 6](#)
- [21] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 9664–9674. Computer Vision Foundation / IEEE, 2021. [1](#)

- [22] Wujin Li, Jiawei Zhan, Jinbao Wang, Bzhong Xia, Bin-Bin Gao, Jun Liu, Chengjie Wang, and Feng Zheng. Towards continual adaptation in industrial anomaly detection. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2871–2880, 2022. 3, 5, 1
- [23] Jiaqi Liu, Kai Wu, Qiang Nie, Ying Chen, Bin-Bin Gao, Yong Liu, Jinbao Wang, Chengjie Wang, and Feng Zheng. Unsupervised continual anomaly detection with contrastively-learned prompt. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3639–3647, 2024. 3, 6, 2
- [24] 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*, pages 20402–20411, 2023. 2
- [25] Ruiying Lu, YuJie Wu, Long Tian, Dongsheng Wang, Bo Chen, Xiyang Liu, and Ruimin Hu. Hierarchical vector quantized transformer for multi-class unsupervised anomaly detection. *Advances in Neural Information Processing Systems*, 36:8487–8500, 2023. 2
- [26] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 577–582. 2023. 1
- [27] Jingyang Qiao, Xin Tan, Chengwei Chen, Yanyun Qu, Yong Peng, Yuan Xie, et al. Prompt gradient projection for continual learning. In *The Twelfth International Conference on Learning Representations*, 2023. 2
- [28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3
- [29] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2022. 2
- [30] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, pages 1906–1915. IEEE, 2021. 2
- [31] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 2
- [32] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*, 2021. 2
- [33] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H Rohban, and Hamid R Rabiee. Multiresolution knowledge distillation for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14902–14912, 2021. 2
- [34] Hannah M Schlüter, Jeremy Tan, Benjamin Hou, and Bernhard Kainz. Natural synthetic anomalies for self-supervised anomaly detection and localization. In *Proceedings of the European conference on computer vision*, pages 474–489. Springer, 2022. 2, 1
- [35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 6
- [36] Jiaqi Tang, Hao Lu, Xiaogang Xu, Ruizheng Wu, Sixing Hu, Tong Zhang, Tsz Wa Cheng, Ming Ge, Ying-Cong Chen, and Fugee Tsung. An incremental unified framework for small defect inspection. In *European Conference on Computer Vision*, pages 307–324. Springer, 2025. 3, 5, 6, 2
- [37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2
- [38] Guodong Wang, Shumin Han, Errui Ding, and Di Huang. Student-teacher feature pyramid matching for anomaly detection. *arXiv preprint arXiv:2103.04257*, 2021. 2
- [39] Zhen Wang, Liu Liu, Yiqun Duan, Yajing Kong, and Dacheng Tao. Continual learning with lifelong vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 171–181, 2022. 6
- [40] Hang Yao, Ming Liu, Haolin Wang, Zhicun Yin, Zifei Yan, Xiaopeng Hong, and Wangmeng Zuo. Glad: Towards better reconstruction with global and local adaptive diffusion models for unsupervised anomaly detection. *arXiv preprint arXiv:2406.07487*, 2024. 1, 2
- [41] 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. 2
- [42] 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*, 35:4571–4584, 2022. 2, 5, 6
- [43] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draem: a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8330–8339, 2021. 2
- [44] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017. 6
- [45] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1, 2, 6, 7, 3
- [46] Xinyi Zhang, Naiqi Li, Jiawei Li, Tao Dai, Yong Jiang, and Shu-Tao Xia. Unsupervised surface anomaly detection with diffusion probabilistic model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6782–6791, 2023. 1, 2
- [47] Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. Destseg: Segmentation guided denoising

- student-teacher for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3914–3923, 2023. [2](#)
- [48] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *Proceedings of the European conference on computer vision*, pages 392–408. Springer, 2022. [2](#), [5](#), [1](#)

# One-for-More: Continual Diffusion Model for Anomaly Detection

## Supplementary Material

### A. Experimental details

**Data pre-processing.** We employ the data pre-processing pipeline specified in DiAD [14] for both the MVTec [4] and VisA [48] datasets to mitigate potential train-test discrepancies. This involves channel-wise standardization using precomputed mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225] after normalizing each RGB image to [0, 1].

**Patch perturbation.** We adopt the method proposed by NSA [34] for patch perturbation on the original images. The NSA method builds upon the Cut-paste technique [21] and enhances it by incorporating the Poisson image editing method [26] to alleviate the discontinuities caused by pasting image patches. The cut-paste method is commonly used in the anomaly detection domain to generate simulated anomalous images. It involves randomly cropping a patch from one image and pasting it onto a random location in another image, thus creating a simulated anomaly. The Poisson-based pasting method seamlessly blends the cloned object from one image into another by solving Poisson partial differential equations, thereby better simulating a realistic anomalous region. In this paper, the number of patches is set as a random value from 1 to 4, and the patch size is a random value from 0.03 to 0.4 of the original image size. The visualization of patch perturbation is shown in Figure 6.

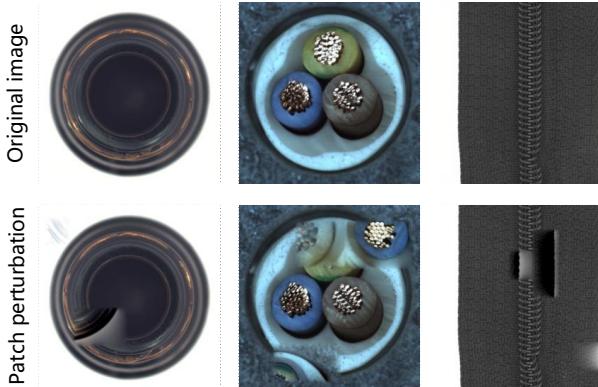


Figure 6. Qualitative results of logical anomaly detection.

**Evaluation metrics.** We follow the literature [4] in reporting the Area Under the Receiver Operation Characteristic (AUROC) for both image-level and pixel-level anomaly detection. To measure the performance of the model in continuous learning, referring to DNE [22], we calculated the average AUROC (A-AUROC) and the forgetting measure

(FM) for  $N$  continual steps. Specially, we define  $A_{N,i}^{\text{pix}}$  and  $A_{N,i}^{\text{img}}$  as the test AUROC of task  $i$  after training on task  $N$ .

$$\text{A-AUROC} = \begin{cases} \frac{1}{N} \sum_{i=1}^{N-1} A_{N,i}^{\text{pix}}, \\ \frac{1}{N} \sum_{i=1}^{N-1} A_{N,i}^{\text{img}} \end{cases}, \quad (13)$$

$$\text{FM} = \begin{cases} \frac{1}{N-1} \sum_{i=1}^{N-1} \max_{b \in \{1, \dots, N-1\}} (A_{b,i}^{\text{pix}} - A_{N,i}^{\text{pix}}), \\ \frac{1}{N-1} \sum_{i=1}^{N-1} \max_{b \in \{1, \dots, N-1\}} (A_{b,i}^{\text{img}} - A_{N,i}^{\text{img}}) \end{cases} \quad (14)$$

In addition to the results for the AUROC documented in the body of the paper, We also supplement the image-level Precision-Recall (AUPR) results and pixel-level Per-region-overlap (PRO) [4, 5] results. Referring to Equation (13) and Equation (14), we calculate A-AUPR, A-PRO, and their FM to evaluate our method. The results are shown in Table 6 7 9 10. Our method still achieves an advanced level in the above metrics.

### B. Memory Analysis of iSVD

In Section 3.2, considering the storage of the original matrix and  $\mathbf{U}$ ,  $\Sigma$ ,  $\mathbf{V}$  during SVD, the memory overhead of SVD is  $d\Lambda + d^2 + \Lambda^2 + \min(d, \Lambda)$ , while iSVD uses a memory overhead of  $d(m+k) + d^2 + (m+k)^2 + \min(d, m+k)$ . It is known that  $\Lambda \gg d$ ,  $m > d > k$  and  $\Lambda = mn$ , thus, the memory saving rate of iSVD is about:

$$\begin{aligned} & \frac{d\Lambda + d^2 + \Lambda^2 + d - [d(m+k) + d^2 + (m+k)^2 + d]}{d\Lambda + d^2 + \Lambda^2 + d} \\ &= \frac{d\Lambda + \Lambda^2 - d(m+k) - (m+k)^2}{\Lambda^2} / \frac{d\Lambda + d^2 + \Lambda^2 + d}{\Lambda^2} \\ &= \frac{d}{\Lambda} + 1 - \frac{d(m+k)}{\Lambda^2} - \frac{(m+k)^2}{\Lambda^2} \\ &\approx 1 - \frac{m^2 + 2mk + k^2}{m^2 n^2} \\ &= 1 - \frac{1}{n^2} - \frac{2k}{m\Lambda} - \frac{k^2}{\Lambda^2} \approx \frac{n^2 - 1}{n^2}. \end{aligned} \quad (15)$$

In practice, the actual memory saving rate differs from the theoretical value due to factors such as memory sharing. Taking the intermediate features of ten images as an example, Figure 7 shows the actual and theoretical memory saving rate of splitting the feature matrix into  $n$  blocks for iSVD. Although there are some differences between the theoretical value and the actual value, the general trend is consistent.

In this paper, we determine the number of blocks  $n$  according to the number of images of the old task. Specifically, we first sample the old task dataset, randomly retain

Method	14 – 1 with 1 Step		10 – 5 with 1 Step		3 × 5 with 5 Steps		10 – 1 × 5 with 5 Steps	
	A-AUPR (↑)	FM (↓)	A-AUPR (↑)	FM (↓)	A-AUPR (↑)	FM (↓)	A-AUPR (↑)	FM (↓)
UCAD* [23]	95.8	0.26	95.0	0.98	93.1	2.02	95.5	0.07
IUF [36]	97.8	0.25	95.4	1.92	91.1	2.86	95.3	0.16
ControlNet [45]	97.2	1.55	96.7	1.76	86.7	6.40	89.0	7.43
DiAD [14]	97.4	0.71	96.4	1.85	89.1	4.31	91.4	4.83
<b>CDAD</b>	<b>98.4</b>	<b>0.08</b>	<b>98.3</b>	<b>0.55</b>	<b>95.8</b>	<b>1.88</b>	<b>98.4</b>	<b>0.02</b>

Table 6. Image-level A-AUPR of our method on MVTec under 4 continual anomaly detection settings. The best and second-best results are marked in **bold** and underline. \* indicates memory limited.

Method	14 – 1 with 1 Step		10 – 5 with 1 Step		3 × 5 with 5 Steps		10 – 1 × 5 with 5 Steps	
	A-PRO (↑)	FM (↓)	A-PRO (↑)	FM (↓)	A-PRO (↑)	FM (↓)	A-PRO (↑)	FM (↓)
UCAD [23]	86.3	1.16	80.7	2.89	71.1	7.48	80.8	1.19
IUF [36]	88.6	<u>0.62</u>	85.0	3.22	<u>72.9</u>	<u>5.79</u>	<u>84.3</u>	2.41
ControlNet [45]	88.5	1.75	85.8	4.70	71.5	10.0	77.9	7.11
DiAD [14]	88.9	0.85	<u>87.4</u>	3.91	72.1	9.23	83.1	2.93
<b>CDAD</b>	<b>89.8</b>	<b>0.47</b>	<b>88.9</b>	<b>2.57</b>	<b>83.8</b>	<b>4.05</b>	<b>89.2</b>	<b>1.16</b>

Table 7. Pixel-level A-PRO of our method on MVTec under 4 continual anomaly detection settings. The best and second-best results are marked in **bold** and underline. \* indicates memory limited.

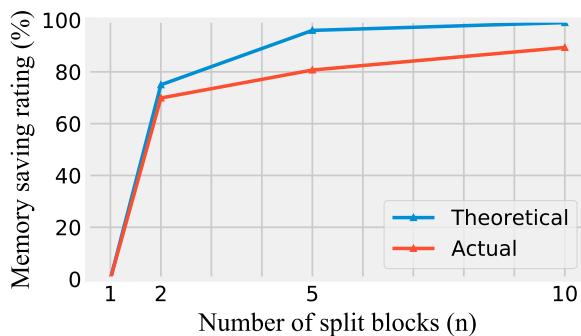


Figure 7. The theoretical and actual values of the memory saving ratio when the different number of split blocks is used.

10% of the images, and then group according to the number of sampled images (denoted as  $N_{img}$ ). The number of groups in iSVD is  $n = \frac{N_{img}}{e}$ . In this paper,  $e$  is set to 1 by default, that is, the intermediate features of each image are divided into a separate group for iSVD operation.

In addition, we analyze the impact of different  $e$  on the model and the time and memory overhead. Table 8 records, for setting different  $e$ , the anomaly detection results of our model on MVTec setting 2, as well as the time and memory overhead for computing the significant representation of the old task. When  $e$  is set to different values, the anomaly detection results and forgetting rate of the model will not

	A-AUROC	FM	Memory	Times
e=1	94.2 / 95.3	2.05 / 2.40	16.7GB	33.5h
e=2	94.1 / 95.4	2.10 / 2.32	30.3GB	28.3h
e=4	94.4 / 95.6	1.95 / 2.21	57.9GB	22.6h
e=6	94.5 / 95.7	1.91 / 2.23	85.9GB	20.2h

Table 8. The impact of different  $e$  on the model and the time and memory overhead of iSVD under MVTec Setting 2.

have much influence. Although we discussed in Section 4.5 that the large number of split blocks will affect the performance of iSVD, it will not impair its representation ability of core information, so it can still ensure the continuous learning ability of the model. Table 8 also shows that with the increase of  $e$ , the memory consumption increases, but the time cost decreases, which indicates that although iSVD can greatly alleviate the pressure of memory, it will bring extra time cost.

## C. Qualitative Results

We supplement the qualitative results on MVTec and Visa datasets, which show the localization image reconstruction results and anomaly localization results for the seven tasks, respectively, as shown in Figure 8–14. Our method not only overcomes the “faithfulness hallucination” problem of the diffusion model but also shows excellent anomaly localization results.

Method	11 – 1 with 1 Step		8 – 4 with 1 Step		8 – 1 × 4 with 4 Steps	
	A-AUROC ( $\uparrow$ )	FM ( $\downarrow$ )	A-AUROC ( $\uparrow$ )	FM ( $\downarrow$ )	A-AUROC ( $\uparrow$ )	FM ( $\downarrow$ )
UCAD* [23]	88.1	0.29	83.2	<u>5.17</u>	82.9	<u>2.16</u>
IUF [36]	<b>91.6</b>	<u>-0.04</u>	<u>83.4</u>	<u>7.51</u>	<u>83.0</u>	<u>6.87</u>
ControlNet [45]	85.2	2.38	78.8	6.25	72.4	4.56
DiAD [14]	74.9	5.42	70.1	12.29	59.5	9.55
<b>CDAD</b>	<u>89.4</u>	<b>-0.77</b>	<b>85.3</b>	<b>3.1</b>	<b>84.7</b>	<b>1.83</b>

Table 9. Image-level A-AUPR of our method on VisA under 3 continual anomaly detection settings. The best and second-best results are marked in **bold** and underline. \* indicates memory limited.

Method	11 – 1 with 1 Step		8 – 4 with 1 Step		8 – 1 × 4 with 4 Steps	
	A-PRO ( $\uparrow$ )	FM ( $\downarrow$ )	A-PRO ( $\uparrow$ )	FM ( $\downarrow$ )	A-PRO ( $\uparrow$ )	FM ( $\downarrow$ )
UCAD* [23]	80.4	2.02	<u>72.4</u>	7.46	<u>70.5</u>	9.83
IUF [36]	82.0	<u>1.04</u>	<u>63.9</u>	20.8	<u>57.0</u>	23.95
ControlNet [45]	62.3	2.45	61.0	<u>1.81</u>	51.7	10.38
DiAD [14]	69.9	4.33	67.7	8.29	55.0	11.74
<b>CDAD</b>	<b>81.6</b>	<b>-0.22</b>	<b>78.9</b>	<b>1.59</b>	<b>77.7</b>	<b>1.66</b>

Table 10. Pixel-level A-PRO of our method on VisA under 3 continual anomaly detection settings. The best and second-best results are marked in **bold** and underline. \* indicates memory limited.

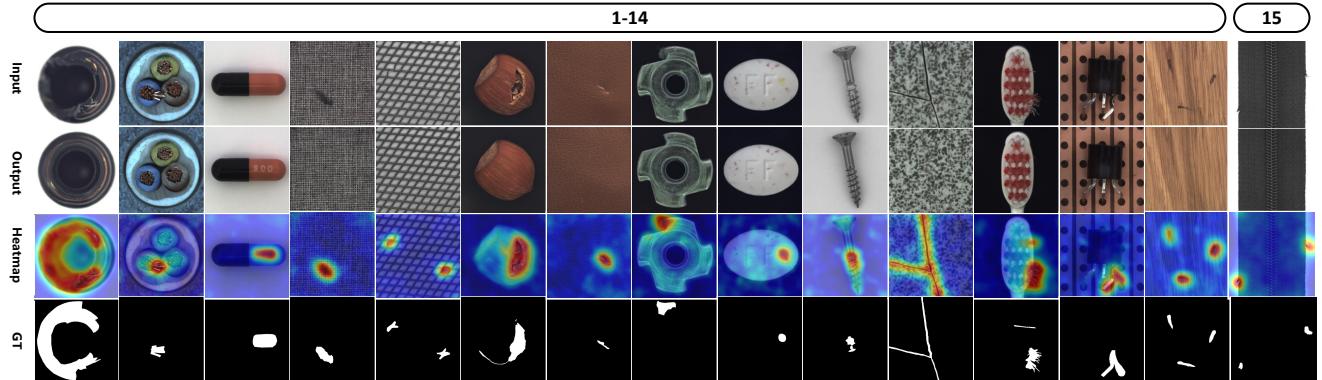


Figure 8. Qualitative comparison results under setting 1 of MVTec, the numbers represent continual training classes.

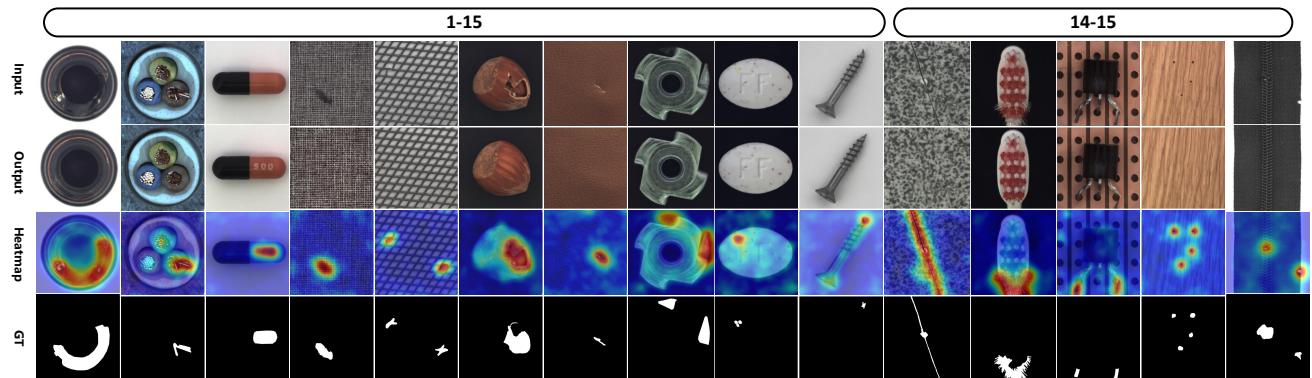


Figure 9. Qualitative comparison results under setting 2 of MVTec, the numbers represent continual training classes.

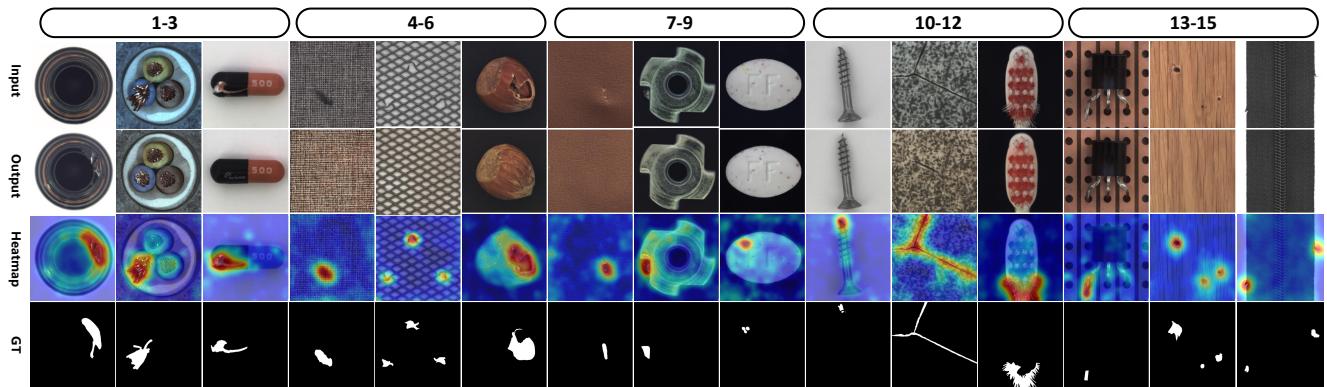


Figure 10. Qualitative comparison results under setting 3 of MVTec, the numbers represent continual training classes.

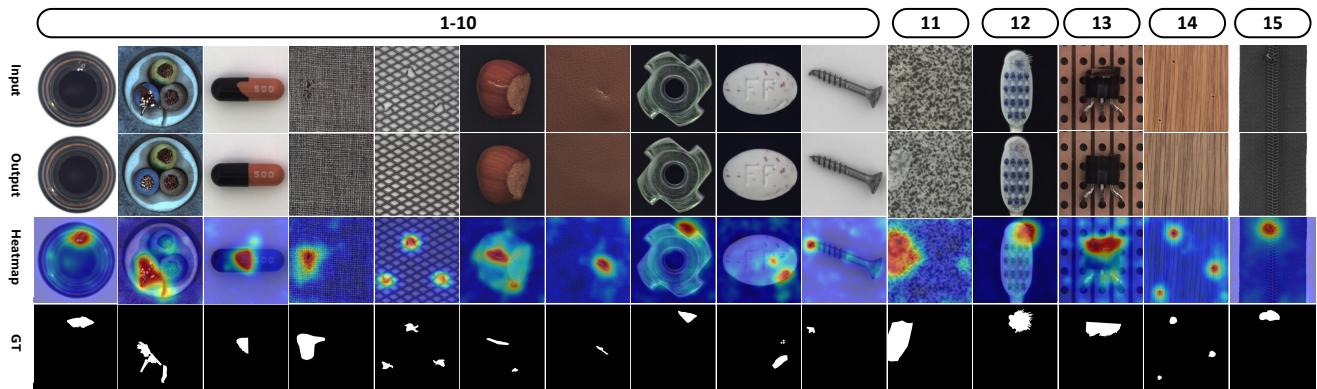


Figure 11. Qualitative comparison results under setting 4 of MVTec, the numbers represent continual training classes.

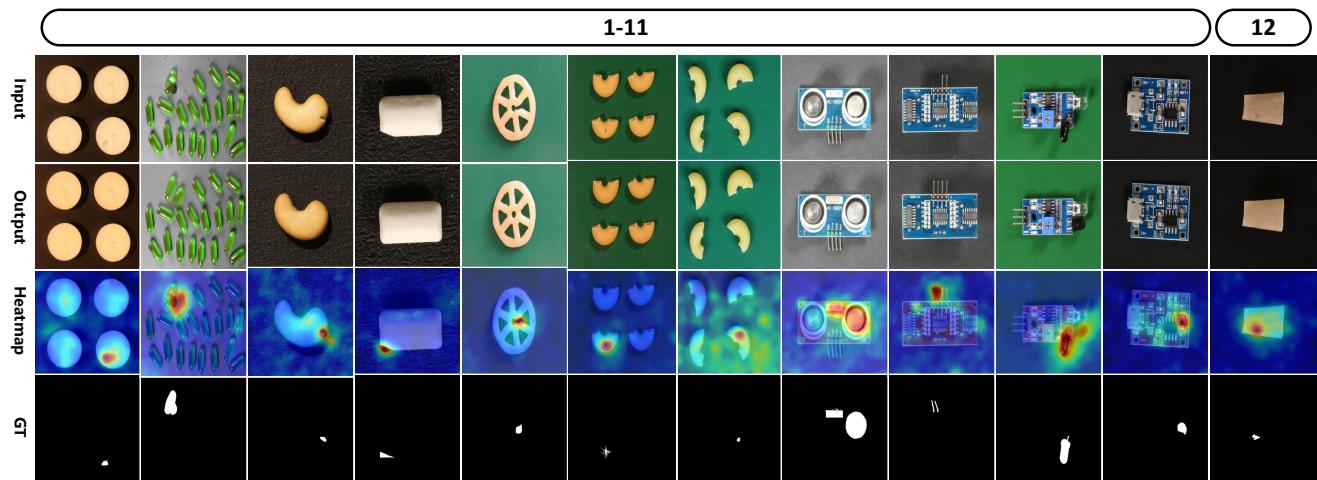


Figure 12. Qualitative comparison results under setting 5 of VisA, the numbers represent continual training classes.

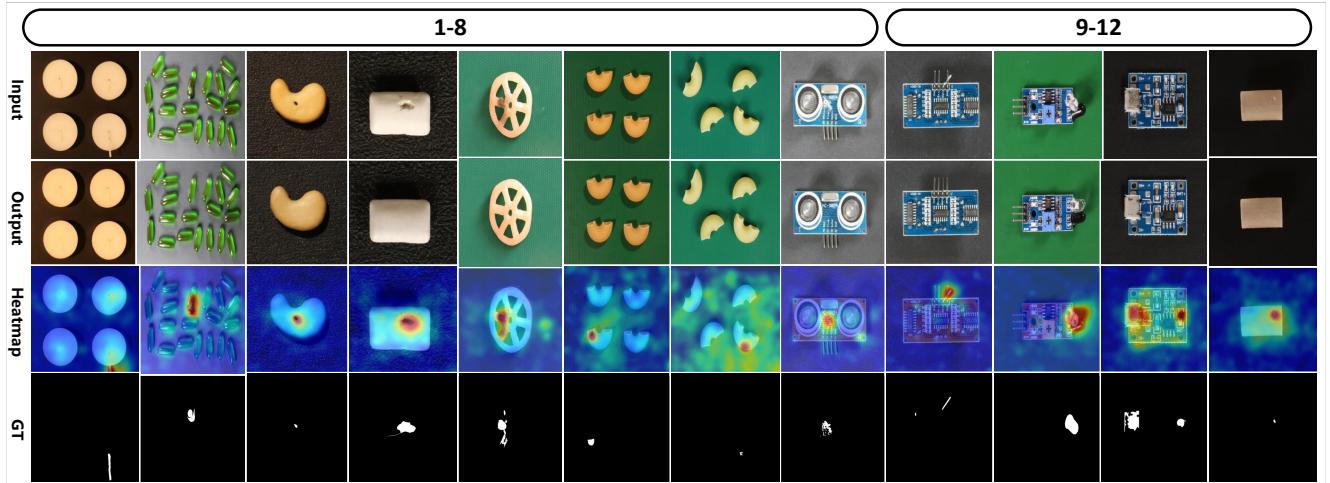


Figure 13. Qualitative comparison results under setting 6 of VisA, the numbers represent continual training classes.

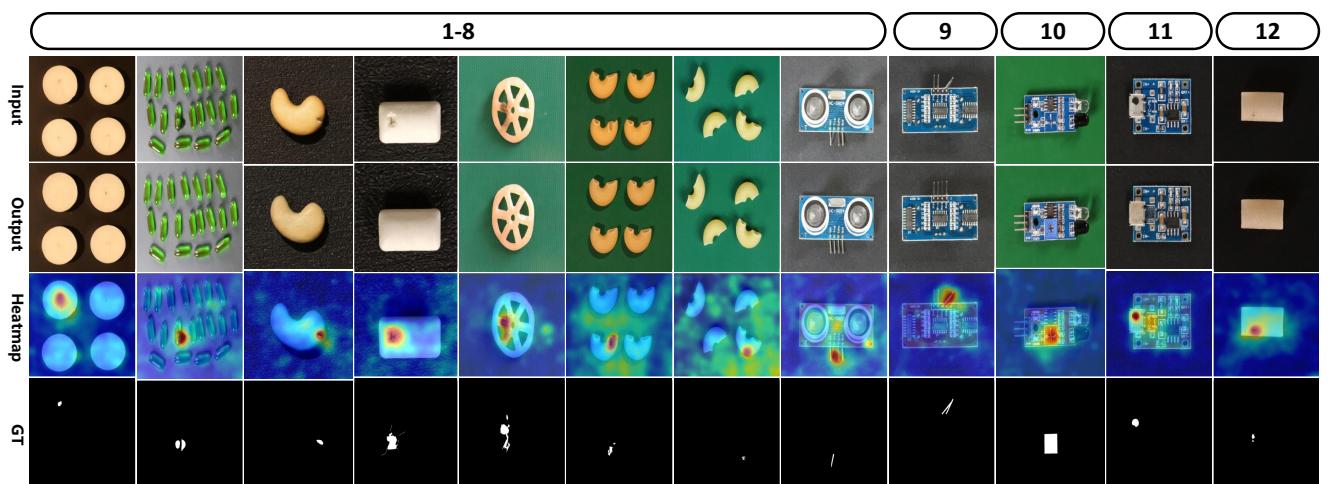


Figure 14. Qualitative comparison results under setting 7 of VisA, the numbers represent continual training classes.