

# AnomalyDiffusion: Few-Shot Anomaly Image Generation with Diffusion Model

Teng Hu<sup>1\*</sup>, Jiangning Zhang<sup>2\*</sup>, Ran Yi<sup>1†</sup>, Yuzhen Du<sup>1</sup>, Xu Chen<sup>2</sup>, Liang Liu<sup>2</sup>, Yabiao Wang<sup>2</sup>, Chengjie Wang<sup>1,2</sup>

<sup>1</sup>Shanghai Jiao Tong University <sup>2</sup>YouTu Lab, Tencent

{hu-teng, ranyi, Haaaaaaaaa}@sjtu.edu.cn;  
{vtzhang, cxxuchen, leoneliu, caseywang, jasoncjwang}@tencent.com;

## Abstract

Anomaly inspection plays an important role in industrial manufacture. Existing anomaly inspection methods are limited in their performance due to insufficient anomaly data. Although anomaly generation methods have been proposed to augment the anomaly data, they either suffer from poor generation authenticity or inaccurate alignment between the generated anomalies and masks. To address the above problems, we propose *AnomalyDiffusion*, a novel diffusion-based few-shot anomaly generation model, which utilizes the strong prior information of latent diffusion model learned from large-scale dataset to enhance the generation authenticity under few-shot training data. Firstly, we propose Spatial Anomaly Embedding, which consists of a learnable anomaly embedding and a spatial embedding encoded from an anomaly mask, disentangling the anomaly information into anomaly appearance and location information. Moreover, to improve the alignment between the generated anomalies and the anomaly masks, we introduce a novel Adaptive Attention Re-weighting Mechanism. Based on the disparities between the generated anomaly image and normal sample, it dynamically guides the model to focus more on the areas with less noticeable generated anomalies, enabling generation of accurately-matched anomalous image-mask pairs. Extensive experiments demonstrate that our model significantly outperforms the state-of-the-art methods in generation authenticity and diversity, and effectively improves the performance of downstream anomaly inspection tasks. The code and data are available in <https://github.com/sjtuplayer/anomalydiffusion>.

## 1 Introduction

In recent years, industrial anomaly inspection algorithms, *i.e.*, anomaly detection, localization, and classification, plays a crucial role in industrial manufacture (Duan et al. 2023). However, in real-world industrial production, the anomaly samples are very few, posing a significant challenge for anomaly inspection (Fig. 1-top). To mitigate the issue of few anomaly data, existing anomaly inspection mostly relies on unsupervised learning methods that only use normal samples (Zavrtanik, Kristan, and Skočaj 2021; Li et al. 2021), or few-shot supervised learning methods (Zhang et al. 2023a). Although these methods perform well in anomaly detection,

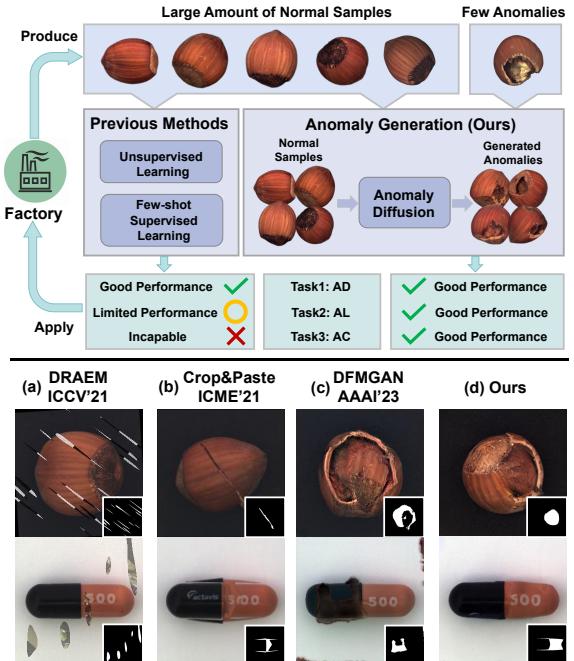


Figure 1: **Top:** Our model generates extensive anomaly data, which supports the downstream Anomaly Detection (AD), Localization (AL) and Classification (AC) tasks, while previous methods mainly rely on unsupervised learning or few-shot supervised learning due to the limited anomaly data; **Bottom:** Generated anomaly results on hazelnut-crack and capsule-squeeze of our model and existing anomaly generation methods, where our results are the most authentic.

they have limited performance in anomaly localization and cannot handle anomaly classification.

To cope with the problem of scarce anomaly samples, researchers propose anomaly generation methods to supplement the anomaly data, which can be divided into two types: **1) The model-free methods** randomly crop and paste patches from existing anomalies or anomaly texture dataset onto normal samples (Li et al. 2021; Lin et al. 2021; Zavrtanik, Kristan, and Skočaj 2021). But such methods exhibit poor authenticity in the synthesized data (Fig. 1-bottom-a/b). **2) The GAN-based methods** (Zhang et al. 2021; Niu et al. 2020;

\*Equal contributions.

†Corresponding author.

Duan et al. 2023) utilize Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) to generate anomalies, but most of them require a large amount of anomaly samples for training. The only few-shot generation model DFM-GAN (Duan et al. 2023) employs StyleGAN2 (Karras et al. 2020) pretrained on normal samples, and then performs domain adaption with a few anomaly samples. But the generated anomalies are not accurately aligned with the anomaly masks (Fig. 1-bottom-c). To sum up, the existing anomaly generation methods either fail to generate authentic anomalies or accurately-aligned anomalous image-mask pairs by learning from few-shot anomaly data, which limits their improvement in the downstream anomaly inspection tasks.

To address the above issues, we propose *AnomalyDiffusion*, a novel anomaly generation method based on the diffusion model, which generates anomalies onto the input normal samples with the anomaly masks. By leveraging the strong prior information of a pretrained LDM (Rombach et al. 2022) learned from large-scale dataset (Schuhmann et al. 2021), we can extract better anomaly representation using only a few anomaly images and boost the generation authenticity and diversity. To generate anomalies with specified type and locations, we propose *Spatial Anomaly Embedding*, which disentangles anomaly information into an anomaly embedding (a learned textual embedding to represent the appearance type of anomaly) and a spatial embedding (encoded from an anomaly mask to indicate the locations). By disentangling anomaly location from appearance, we can generate anomalies in any desired positions, which enables producing a large amount of anomalous image-mask pairs for the downstream tasks. Moreover, we propose an *Adaptive Attention Re-weighting Mechanism* to allocate more attention to the areas with less noticeable generated anomalies, which dynamically adjusts the cross-attention maps based on disparities between the generated images and input normal samples during the diffusion inference stage. This adaptive mechanism results in accurately aligned generated anomaly images and anomaly masks, which greatly facilitates downstream anomaly localization tasks.

Extensive qualitative and quantitative experiments and comparisons demonstrate that our *AnomalyDiffusion* outperforms state-of-the-art anomaly generation models in terms of generation authenticity and diversity. Moreover, our generated anomaly images can be effectively applied to downstream anomaly inspection tasks, yielding a pixel-level **99.1% AUROC** and **81.4% AP** score in anomaly localization on MVTec (Bergmann et al. 2019). The main contribution of this paper can be summarized as follows:

- We propose *AnomalyDiffusion*, a few-shot diffusion-based anomaly generation method, which disentangles anomalies into anomaly embedding (for anomaly appearance) and spatial embedding (for anomaly location), and generates authentic and diverse anomaly images.
- We design *Adaptive Attention Re-weighting Mechanism*, which adaptively allocates more attention to the areas with less noticeable generated anomalies, improving the alignment between the generated anomalies and masks.
- Extensive experiments demonstrate the superiority of

our model over the state-of-the-art competitors, and our generated anomaly data effectively improves the performance of downstream anomaly inspection tasks, which will be released to facilitate future research.

## 2 Related Work

### 2.1 Generative Models

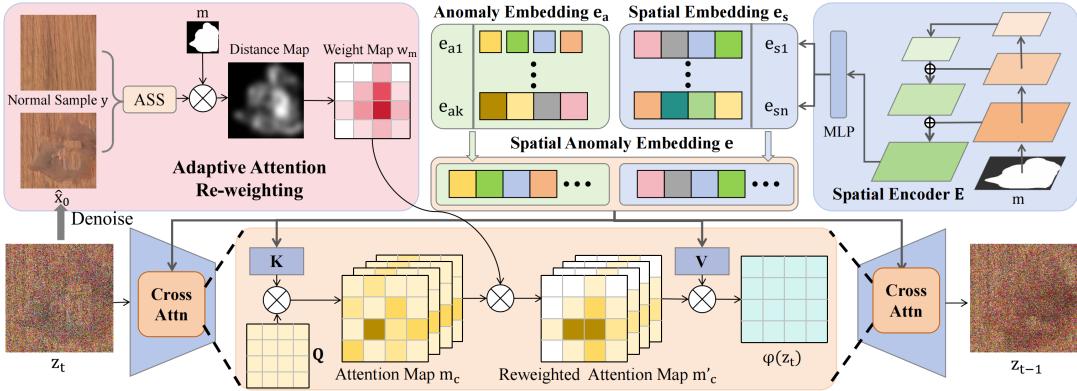
**Generative models.** VAEs (Kingma and Welling 2013) and GANs (Goodfellow et al. 2014) have achieved great progress in image generation. Recently, diffusion model (Nichol and Dhariwal 2021) demonstrates a more enhanced potential in generating images in a wide range of domains. Latent diffusion model (LDM) (Rombach et al. 2022) further improves the generation ability through compression of the diffusion space and obtains strong prior information by training on LAION dataset (Schuhmann et al. 2021).

**Few-shot image generation.** Few-shot image generation aims to generate diverse images with limited training data. Early methods propose modifying network weights (Mo, Cho, and Shin 2020), using various regularization techniques (Li et al. 2020) and data augmentation (Tran et al. 2021) to prevent overfitting. To deal with the extremely limited data (less than 10), recent works (Ojha et al. 2021; Wang et al. 2022; Hu et al. 2023) introduce cross-domain consistency losses to keep the generated distribution. Textual Inversion (Gal et al. 2022) and Dreambooth (Ruiz et al. 2023) encode a few images into the textual space of a pretrained LDM, but cannot control the generated locations accurately.

### 2.2 Anomaly Inspection

**Anomaly inspection.** The anomaly inspection task consists of anomaly detection, localization and classification. Some existing methods (Schlegl et al. 2017, 2019; Liang et al. 2023) rely on image reconstruction, comparing the differences between reconstructed images and anomaly images to achieve anomaly detection and localization. Moreover, deep feature modeling-based methods (Lee, Lee, and Song 2022; Cao et al. 2022; Roth et al. 2022; Gu et al. 2023; Wang et al. 2023) build a feature space for input images and then compare the differences between features to detect and localize anomalies. Additionally, some supervised learning-based methods (Zhang et al. 2023a) utilize a small number of anomaly samples to enhance the anomaly localization capabilities. Some studies conduct zero-/few-shot AD without using or with only a small number of anomaly samples (Jeong et al. 2023; Cao et al. 2023; Chen, Han, and Zhang 2023; Chen et al. 2023; Zhang et al. 2023b; Huang et al. 2022). Although these methods have shown promising results in anomaly detection, their performance in anomaly localization is still limited due to the lack of anomaly data.

**Anomaly generation.** The scarcity of anomaly data has sparked research interest in anomaly generation. DRAEM (Zavrtanik, Kristan, and Skočaj 2021), Cut-Paste (Li et al. 2021), Crop-Paste (Lin et al. 2021) and PRN (Zhang et al. 2023a) crop and paste unrelated textures or existing anomalies into normal sample. But they either generate less realistic anomalies or have limited generated



**Figure 2: Overall framework of our AnomalyDiffusion:** 1) The *Spatial Anomaly Embedding*  $e$ , consisting of an anomaly embedding  $e_a$  (a learned textual embedding to represent anomaly appearance type) and a spatial embedding  $e_s$  (encoded from an input anomaly mask  $m$  to indicate anomaly locations), serves as the text condition to guide the anomaly generation process; 2) The *Adaptive Attention Re-weighting Mechanism* computes the weight map  $w_m$  based on the difference between the denoised image  $\hat{x}_0$  and the input normal sample  $y$ , and adaptively reweights the cross-attention map  $m_c$  by the weight map  $w_m$  to help the model focus more on the less noticeable anomaly areas during the denoising process.

diversity. The GAN-based model SDGAN (Niu et al. 2020) and Defect-GAN (Zhang et al. 2021), generate anomalies on normal samples by learning from anomaly data. But they require a large amount of anomaly data and cannot generate anomaly mask. DFMGAN (Duan et al. 2023) transfers a StyleGAN2 (Karras et al. 2020) pretrained on normal samples to anomaly domain, but lacks generation authenticity and accurate alignment between generated anomalies and masks. In contrast, our model incorporates spatial anomaly embedding and adaptive attention re-weighting mechanism, which can generate anomalous image-mask pairs with great diversity and authenticity.

### 3 Method

Our *AnomalyDiffusion* aims to generate a large amount of anomaly data aligned with anomaly masks, by learning from a few anomaly samples. The inputs to our model include an anomaly-free sample  $y$  and an anomaly mask  $m$ , and the output is an image with anomalies generated in the mask area, while the remaining region is consistent with the input anomaly-free sample.

As shown in Fig. 2, our *AnomalyDiffusion* is developed based on Latent Diffusion Model (Rombach et al. 2022). To disentangle the anomaly location information from anomaly appearance, we propose Spatial Anomaly Embedding  $e$ , which consists of an anomaly embedding  $e_a$  (for anomaly appearance) and a spatial embedding  $e_s$  (for anomaly location). Moreover, to enhance the alignment between the generated anomalies and given masks, we introduce an Adaptive Attention Re-weighting Mechanism, which helps the model to allocate more attention to the areas with less noticeable generated anomalies (Fig. 3(c)).

Specifically, the anomaly embedding  $e_a$  provides the anomaly appearance type information, with one  $e_a$  corresponding to a certain type of anomaly (*e.g.*, hazelnut-crack, capsule-squeeze), which is learned by our masked textual inversion (Sec. 3.2). And the spatial embedding  $e_s$  provides

the anomaly location information, which is encoded from the input anomaly mask  $m$  by a spatial encoder  $E$  (shared among all anomalies). By combining the anomaly embedding  $e_a$  with spatial embedding  $e_s$ , the spatial anomaly embedding  $e$  contains both the anomaly appearance and spatial information, which serves as the text condition in the diffusion model to guide the generation process. With the spatial anomaly embedding as condition, given a normal sample, we generate an anomaly image with the blended diffusion process (Avrahami, Lischinski, and Fried 2022):

$$x_{t-1} = p_\theta(x_{t-1}|x_t, e) \odot m + q(y_{t-1}|y_0) \odot (1 - m), \quad (1)$$

where  $x_t$  is the generated anomaly image at timestep  $t$ ,  $y_0$  is the input normal sample,  $m$  is the anomaly mask, and  $q(\cdot)$  and  $p_\theta(\cdot)$  are the forward and backward process in diffusion as illustrated in Sec. 3.1.

#### 3.1 Preliminaries

Denoising diffusion probabilistic models (DDPM) (Ho, Jain, and Abbeel 2020) has achieved significant success in image generation tasks. It employs a forward process to add noise into the data and then learns denoising during the backward process, thereby accomplishing the fitting of the training data distribution. With the training image  $x_0$ , the forward process  $q(\cdot)$  in diffusion model is formulated as:

$$\begin{aligned} q(x_1, \dots, x_T | x_0) &:= \prod_{t=1}^T q(x_t | x_{t-1}), \\ q(x_t | x_{t-1}) &:= \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t \mathbf{I}), \end{aligned} \quad (2)$$

where  $\beta_t$  is the variance at timestep  $t$ .

The backward process is approximated by predicting the mean  $\mu_\theta(x_t, t)$  and variance  $\Sigma_\theta(x_t, t)$  (set as a constant in DDPM) of a Gaussian distribution iteratively by:

$$p_\theta(x_{t-1} | x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \quad (3)$$

Textual inversion (Gal et al. 2022) utilizes a pre-trained Latent Diffusion Model to extract the shared content information in few-shot input samples by optimizing text embeddings. With the refined text embeddings as condition  $c$ , textual inversion can generate novel images  $x_0$  with similar contents of input images by:

$$x_0 = \prod_{t=1}^T p_\theta(x_{t-1}|x_t, c), x_T \sim \mathcal{N}(0, 1). \quad (4)$$

### 3.2 Spatial Anomaly Embedding

**Disentangle spatial information from anomaly appearance.** We aim at controllable anomaly generation with specified anomaly type and location. A direct solution is to control anomaly type by textual embedding learned from textual inversion (Gal et al. 2022), and control anomaly location by the input mask. However, textual inversion tends to capture the location of anomalies along with the anomaly type information, which results in the generated anomalies only distributed in specific locations. To address the issue, we propose to disentangle the textual embedding into two parts, where one part (the spatial embedding  $e_s$ ) is directly encoded from the anomaly mask to indicate the anomaly location, leaving the rest (the anomaly embedding  $e_a$ ) to only learn anomaly type information. We name our decomposed textual embedding as Spatial Anomaly Embedding.

**Anomaly embedding** is a learned textual embedding that represents the anomaly appearance type information. Different from textual inversion method that learns the features of the entire image, in anomaly generation, our model only needs to focus on anomaly areas, without requiring information of the entire image. Therefore, we introduce *masked textual inversion*, where we mask out irrelevant background and normal regions of the anomaly image, and only the anomaly regions are visible to the model. We initialize the anomaly embedding  $e_a$  with  $k$  tokens and optimize it using the masked diffusion loss:

$$\mathcal{L}_{dif} = \|m \odot (\epsilon - \epsilon_\theta(z_t, t, \{e_a, e_s\}))\|_2^2, \quad (5)$$

where  $\epsilon \sim \mathcal{N}(0, 1)$  and  $z_t$  is the noised latent code of the input image  $x$  at timestep  $t$ .

**Spatial embedding.** To provide accurate spatial information of the anomaly locations, we introduce a spatial encoder  $E$  that encodes the input anomaly mask  $m$  into spatial embedding  $e_s$ , which is in the form of textual embedding and contains precise location information from the mask. Specifically, we input the anomaly mask into ResNet-50 (He et al. 2016) to extract the image features in different layers and fuse them together by Feature Pyramid Networks (Lin et al. 2017). Finally, several fully-connected networks are employed to map the fused features into textual embedding space, with each network predicting one text token, thereby outputting the final spatial embedding  $e_s$  with  $n$  tokens.

**Overall training framework.** For each anomaly type  $i$ , we employ an anomaly embedding  $e_{a,i}$  to extract its appearance information, while all anomaly categories share a common spatial encoder  $E$ . For a set of image-mask pairs  $(x_i, m_i)$  in the training data, we first input anomaly mask  $m_i$  into spatial

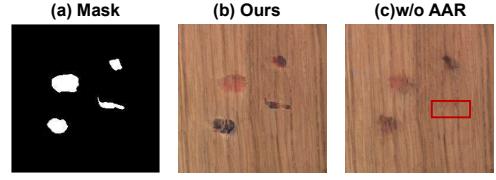


Figure 3: Comparison between the models w/ (Ours) and w/o Adaptive Attention Re-weighting (AAR). The model w/o AAR cannot generate anomalies to fill the entire mask.

encoder  $E$  to obtain the spatial embedding  $e_s = E(m_i)$ . Then, we concatenate the anomaly embedding  $e_{a,i}$  and the spatial embedding  $e_s$  together to obtain our spatial anomaly embedding  $e = \{e_a, e_s\}$ . Finally, the concatenated textual embedding  $e$  is used as the text condition to the diffusion model, and the training process can be formulated as:

$$e_a^*, E^* = \arg \min_{e_a, E} \mathbb{E}_{z \sim \mathcal{E}(x_i), m_i, \epsilon, t} \mathcal{L}_{dif}. \quad (6)$$

where  $\mathcal{E}(\cdot)$  is the image encoder of latent diffusion model and  $\epsilon \sim \mathcal{N}(0, 1)$ .

### 3.3 Adaptive Attention Re-Weighting

With the spatial anomaly embedding  $e$ , we can use it as the text condition to guide the generation of anomaly images by Eq. (1). However, the generated anomaly images sometimes fail to fill the entire mask, especially when there are multiple anomaly regions in the mask or when the mask has irregular shapes (Fig. 3-a/c). In such cases, the generated anomalies are usually not well aligned with the mask, which limits the improvement in downstream anomaly localization task. To address this problem, we propose an adaptive attention re-weighting mechanism, which allocates more attention to the areas with less noticeable generated anomalies during the denoising process, thereby facilitating better alignment between the generated anomalies and the anomaly masks.

**Adaptive attention weight map.** Specifically, at the  $t$ -th denoising step, we calculate the corresponding  $\hat{x}_0 = D(p_\theta(\hat{z}_0|z_t, e))$  (where  $D$  is the decoder of LDM). Then, we calculate the pixel-level difference between  $\hat{x}_0$  and the normal sample  $y$  within the mask  $m$ . Based on the difference, we calculate the weight map  $w_m$  by the Adaptive Scaling Softmax (ASS) operation:

$$w_m = \|m\|_1 \cdot \text{Softmax}(f(\|m \odot y - m \odot \hat{x}_0\|_2^2)), \quad (7)$$

where  $f(x) = \frac{1}{x}$  when  $x \neq 0$  and  $f(x) = -\infty$  otherwise. For the regions within the mask that are similar to normal samples, the generated anomalies in these regions are less noticeable. To enhance the anomaly generation effects, these regions are assigned higher weights by Eq. (7) and allocated with more attention by attention re-weighting.

**Attention re-weighting.** We employ the weight map  $w_m$  to adaptively control the cross-attention, in order to guide our model to focus more on the areas with less noticeable generated anomalies. In our cross-attention calculation, Query is calculated from the latent code  $z_t$ , and Key and Value are calculated from our spatial anomaly embedding  $e$ :

$$Q = W_Q^{(i)} \cdot \varphi_i(z_t), K = W_K^{(i)} \cdot e, V = W_V^{(i)} \cdot e, \quad (8)$$

Category	DiffAug		CDC		Crop-Paste		SDGAN		Defect-GAN		DFMGAN		Ours	
	IS ↑	IC-L ↑	IS ↑	IC-L ↑	IS ↑	IC-L ↑	IS ↑	IC-L	IS ↑	IC-L ↑	IS ↑	IC-L ↑	IS ↑	IC-L ↑
bottle	<u>1.59</u>	0.03	1.52	0.04	1.43	0.04	1.57	0.06	1.39	0.07	<b>1.62</b>	<u>0.12</u>	1.58	<b>0.19</b>
cable	1.72	0.07	<u>1.97</u>	0.19	1.74	<u>0.25</u>	1.89	0.19	1.70	0.22	1.96	<u>0.25</u>	<b>2.13</b>	<b>0.41</b>
capsule	1.34	0.03	1.37	0.06	1.23	0.05	<u>1.49</u>	0.03	<b>1.59</b>	0.04	<b>1.59</b>	<u>0.11</u>	<b>1.59</b>	<b>0.21</b>
carpet	1.19	0.06	<b>1.25</b>	0.03	1.17	0.11	1.18	0.11	<u>1.24</u>	0.12	1.23	<u>0.13</u>	1.16	<b>0.24</b>
grid	1.96	0.06	1.97	0.07	2.00	0.12	1.95	0.10	<u>2.01</u>	0.12	1.97	<u>0.13</u>	<b>2.04</b>	<b>0.44</b>
hazel_nut	1.67	0.05	<u>1.97</u>	0.05	1.74	0.21	1.85	0.16	1.87	0.19	1.93	<u>0.24</u>	<b>2.13</b>	<b>0.31</b>
leather	<u>2.07</u>	0.06	1.80	0.07	1.47	0.14	2.04	0.12	<b>2.12</b>	0.14	2.06	<u>0.17</u>	1.94	<b>0.41</b>
metal nut	<u>1.58</u>	0.29	1.55	0.04	1.56	0.15	1.45	0.28	1.47	<u>0.30</u>	1.49	<b>0.32</b>	<b>1.96</b>	<u>0.30</u>
pill	1.53	0.05	1.56	0.06	1.49	0.11	<u>1.61</u>	0.07	<u>1.61</u>	0.10	<b>1.63</b>	<u>0.16</u>	<u>1.61</u>	<b>0.26</b>
screw	1.10	0.10	1.13	0.11	1.12	<u>0.16</u>	1.17	0.10	<u>1.19</u>	0.12	1.12	0.14	<b>1.28</b>	<b>0.30</b>
tile	1.93	0.09	2.10	0.12	1.83	0.20	<u>2.53</u>	0.21	2.35	<u>0.22</u>	2.39	<u>0.22</u>	<b>2.54</b>	<b>0.55</b>
toothbrush	1.33	0.06	1.63	0.06	1.30	0.08	1.78	0.03	<b>1.85</b>	0.03	<u>1.82</u>	<u>0.18</u>	1.68	<b>0.21</b>
transistor	1.34	0.05	1.61	0.13	1.39	0.15	<b>1.76</b>	0.13	1.47	0.13	<u>1.64</u>	<u>0.25</u>	1.57	<b>0.34</b>
wood	2.05	0.30	2.05	0.03	1.95	0.23	2.12	0.25	<u>2.19</u>	0.29	2.12	<u>0.35</u>	<b>2.33</b>	<b>0.37</b>
zipper	<u>1.30</u>	0.05	<u>1.30</u>	0.05	1.23	0.11	1.25	0.10	1.25	0.10	1.29	<b>0.27</b>	<b>1.39</b>	<u>0.25</u>
Average	1.58	0.09	1.65	0.07	1.51	0.14	1.71	0.13	1.69	0.15	1.72	<u>0.20</u>	<b>1.80</b>	<b>0.32</b>

Table 1: **Comparison on IS and IC-LPIPS on MVTec dataset.** Our model generates the most high-quality and diverse anomaly data, achieving the best IS and IC-LPIPS. **Bold** and underline represent optimal and sub-optimal results, respectively.

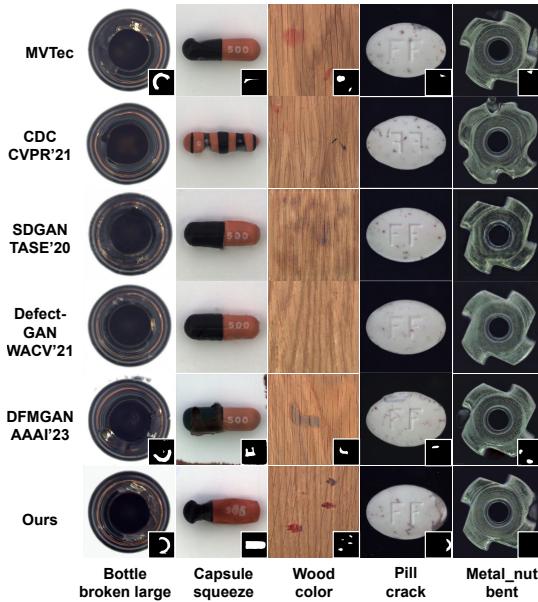


Figure 4: **Comparison on the generation results on MVTec.** Our model generates high quality anomaly images that are accurately aligned with the anomaly masks.

where  $\varphi_i$  is the intermediate representation of the U-Net ( $\epsilon_\theta$ ) and the  $W^{(i)}$ 's are the learnable projection matrices. The cross-attention calculation process is then formulated as  $Attn(Q, K, V) = m_c \cdot V$ , where  $m_c = Softmax(\frac{QK^T}{\sqrt{d}})$  is the cross-attention map.

Considering the cross-attention map  $m_c$  controls the generated layout and effects, where higher attention leads to stronger generation effects (Hertz et al. 2022), we reweight the cross-attention map by our weight map:  $m'_c = m_c \odot w_m$ . The new cross-attention map  $m'_c$  focuses more on the areas with less noticeable generated anomalies, thereby enhancing the alignment accuracy between the generated anomalies and the input anomaly masks. The re-weighted cross attention is formulated as  $RW-Attn(Q, K, V) = m'_c \cdot V$ .

### 3.4 Mask Generation

Recall that our model requires anomaly masks as inputs. However, the number of real anomaly masks in the training datasets is very few, and the mask data lacks diversity even after augmentation, which motivates us to generate more anomaly masks by learning the real mask distribution. We employ textual inversion to learn a mask embedding  $e_m$ , which can be used as text condition to generate extensive anomaly masks. Specifically, we initialize the mask embedding  $e_m$  as  $k'$  random tokens and optimize it by:

$$e_m^* = \arg \min_{e_m} \mathbb{E}_{z \sim \mathcal{E}(m), \epsilon, t} \left[ \| \epsilon - \epsilon_\theta(z_t, t, e_m) \|_2^2 \right]. \quad (9)$$

With the learned mask embedding, we can generate extensive anomaly masks for each type of anomaly.

## 4 Experiments

### 4.1 Experiment Settings

**Dataset.** we conduct experiments on the widely used MVTec (Bergmann et al. 2019) dataset. We employ one-third of the anomaly data with the lowest ID numbers as the training set, reserving the remaining two-thirds for testing.

**Implementation details.** We assign  $k = 8$  tokens for anomaly embedding,  $n = 4$  tokens for spatial embedding, and  $k' = 4$  tokens for mask embedding. For each type of anomaly, we generate 1000 anomalous image-mask pairs for the downstream anomaly inspection tasks. More details are recorded in the supplementary material.

**Metric. 1) For generation**, due to the limited anomaly data, FID (Heusel et al. 2017) and KID (Bińkowski et al. 2018) are not reliable since the overfitted model tends to yield better scores (best) (Duan et al. 2023). Therefore, we employ Inception Score (IS), which is independent of the given anomaly data, for a direct assessment of generation quality; we also introduce Intra-cluster pairwise LPIPS distance (IC-LPIPS) (Ojha et al. 2021) to measure the generation diversity. **2) for anomaly inspection**, we utilize AUROC, Average Precision (AP), and the **F<sub>1</sub>-max** score to evaluate the accuracy of anomaly detection and localization.

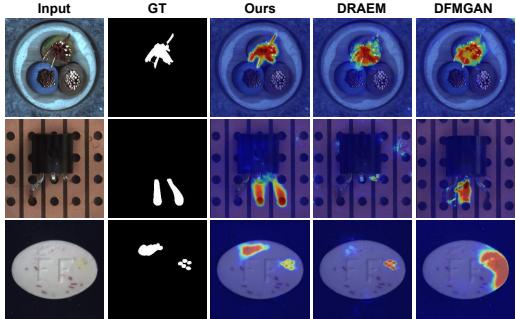


Figure 5: **Quantitative anomaly localization comparison** with an U-Net trained on the data generated by DRAEM, DFMGAN and our model. It shows that our model achieves the best anomaly localization results.

Category	DRAEM			PRN			DFMGAN			Ours		
	AUC	AP	$F_1$ -max	AUC	AP	$F_1$ -max	AUC	AP	$F_1$ -max	AUC	AP	$F_1$ -max
bottle	96.7	80.2	74.0	97.5	76.4	71.3	98.9	90.2	83.9	99.4	94.1	87.3
cable	80.3	21.8	28.3	94.5	64.4	61.0	97.2	81.0	75.4	99.2	90.8	83.5
capsule	76.2	25.5	32.1	95.6	45.7	47.9	79.2	26.0	35.0	98.8	57.2	59.8
carpet	92.6	43.0	41.9	96.4	69.6	65.6	90.6	33.4	38.1	98.6	81.2	74.6
grid	99.1	<b>59.3</b>	<b>58.7</b>	98.9	<b>58.6</b>	<b>58.9</b>	75.2	14.3	20.5	98.3	52.9	54.6
hazelnut	98.8	73.6	68.5	98.0	73.9	68.2	99.7	<b>95.2</b>	<b>89.5</b>	99.8	<b>96.5</b>	<b>90.6</b>
leather	98.5	67.6	65.0	99.4	58.1	54.0	98.5	68.7	66.7	99.8	<b>79.6</b>	71.0
metal nut	96.9	84.2	74.5	97.9	93.0	87.1	99.3	<b>98.1</b>	<b>94.5</b>	99.8	98.7	94.0
pill	95.8	45.3	53.0	98.3	55.5	72.6	81.2	67.8	72.6	99.8	<b>97.0</b>	<b>90.8</b>
screw	91.0	30.1	35.7	94.0	47.7	49.8	58.8	2.2	5.3	97.0	<b>51.8</b>	<b>50.9</b>
tile	98.5	93.2	<b>87.8</b>	98.5	91.8	84.4	<b>99.5</b>	<b>97.1</b>	<b>91.6</b>	99.2	<b>93.9</b>	86.2
toothbrush	93.8	29.5	28.4	96.1	46.4	46.2	96.4	<b>75.9</b>	<b>72.6</b>	99.2	<b>76.5</b>	73.4
transistor	76.5	31.7	24.2	94.9	68.6	68.4	96.2	<b>81.2</b>	<b>77.0</b>	99.3	<b>92.6</b>	85.7
wood	98.8	<b>87.8</b>	<b>80.9</b>	96.2	74.2	67.4	95.3	70.7	65.8	98.9	84.6	74.5
zipper	93.4	65.4	64.7	98.4	<b>79.0</b>	<b>73.7</b>	92.9	65.6	64.9	99.4	<b>86.0</b>	<b>79.2</b>
Average	92.2	54.1	53.1	96.9	66.2	64.7	90.0	62.7	62.1	99.1	<b>81.4</b>	76.3

Table 2: **Comparison on pixel-level anomaly localization on MVTec dataset** by training an U-Net on the generated data from DRAEM, PRN, DFMGAN and our model.

Category	DRAEM			PRN			DFMGAN			Ours		
	AUC	AP	$F_1$ -max	AUC	AP	$F_1$ -max	AUC	AP	$F_1$ -max	AUC	AP	$F_1$ -max
bottle	99.3	<b>99.8</b>	<b>98.9</b>	94.9	98.4	94.1	99.3	<b>99.8</b>	<b>97.7</b>	<b>99.8</b>	<b>99.9</b>	<b>98.9</b>
cable	72.1	83.2	79.2	86.3	92.0	84.0	95.9	<b>97.8</b>	<b>93.8</b>	<b>100</b>	<b>100</b>	<b>100</b>
capsule	93.2	<b>98.7</b>	94.0	84.9	95.8	94.3	92.8	98.5	<b>94.5</b>	<b>99.7</b>	<b>99.9</b>	<b>98.7</b>
carpet	95.3	<b>98.7</b>	93.4	92.6	97.8	92.1	67.9	87.9	87.3	<b>96.7</b>	<b>98.8</b>	<b>94.3</b>
grid	<b>99.8</b>	<b>99.9</b>	<b>98.8</b>	96.6	98.9	95.0	73.0	90.4	85.4	98.4	99.5	98.7
hazelnut	<b>100</b>	<b>100</b>	<b>100</b>	93.6	96.0	94.1	99.9	<b>100</b>	99.0	99.8	<b>99.9</b>	98.9
leather	<b>100</b>	<b>100</b>	<b>100</b>	99.1	99.7	97.6	99.9	<b>100</b>	99.2	<b>100</b>	<b>100</b>	<b>100</b>
metal_nut	97.8	99.6	97.6	97.8	99.5	96.9	99.3	<b>99.8</b>	<b>99.2</b>	<b>100</b>	<b>100</b>	<b>100</b>
pill	94.4	98.9	<b>95.8</b>	88.8	97.8	93.2	68.7	91.7	91.4	98.0	<b>99.6</b>	97.0
screw	88.5	96.3	<b>89.3</b>	84.1	94.7	87.2	22.3	64.7	85.3	<b>96.8</b>	<b>97.9</b>	<b>95.5</b>
tile	<b>100</b>	<b>100</b>	<b>100</b>	91.1	<b>96.9</b>	<b>89.3</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>
toothbrush	99.4	99.8	<b>97.6</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>
transistor	79.6	80.5	71.1	88.2	<b>88.9</b>	84.0	90.8	<b>92.5</b>	<b>88.9</b>	<b>100</b>	<b>100</b>	<b>100</b>
wood	<b>100</b>	<b>100</b>	<b>100</b>	77.5	92.7	86.7	<b>98.4</b>	<b>99.4</b>	<b>98.8</b>	<b>98.4</b>	<b>99.4</b>	<b>98.8</b>
zipper	<b>100</b>	<b>100</b>	<b>100</b>	98.7	99.7	97.6	99.7	<b>99.9</b>	<b>99.4</b>	<b>99.9</b>	<b>100</b>	<b>99.4</b>
Average	94.6	<b>97.0</b>	94.4	91.6	96.6	92.4	87.2	94.8	94.7	<b>99.2</b>	<b>99.7</b>	<b>98.7</b>

Table 3: **Comparison on image-level anomaly detection.**

## 4.2 Comparison in Anomaly Generation

**Baseline.** The compared anomaly generation methods can be classified into 2 groups: **1)** the models (Crop&Paste (Lin et al. 2021), DRAEM (Zavrtanik, Kristan, and Skočaj 2021), PRN (Zhang et al. 2023a) and DFMGAN (Duan et al. 2023)) that can generate anomalous image-mask pairs, which are employed to compare anomaly detection and localization; **2)** the models (DiffAug (Zhao et al. 2020), CDC (Ojha et al. 2021), Crop&Paste, SDGAN (Niu et al. 2020), DefectGAN (Zhang et al. 2021) and DFMGAN) that can generate specific anomaly types, which are employed to compare anomaly generation quality and classification.

Category	DiffAug	CDC	Crop&Paste	SDGAN	Defect-GAN	DFMGAN	Ours
bottle	48.84	38.76	52.71	48.84	53.49	<b>56.59</b>	<b>90.70</b>
cable	21.36	39.06	32.81	21.36	<b>45.31</b>	<b>67.19</b>	
capsule	34.67	28.89	32.89	30.22	32.00	<b>37.23</b>	<b>66.67</b>
carpet	35.48	25.27	27.96	21.50	29.03	<b>47.31</b>	<b>58.06</b>
grid	28.33	35.83	28.33	30.83	27.50	<b>40.83</b>	<b>42.50</b>
hazelnut	65.28	54.86	59.03	43.75	61.11	<b>81.94</b>	<b>85.42</b>
leather	40.74	43.38	34.39	38.10	42.33	<b>49.73</b>	<b>61.90</b>
metalnut	58.85	48.44	<b>59.89</b>	44.27	56.77	<b>64.58</b>	59.38
pill	<b>29.86</b>	21.88	26.74	20.49	28.47	29.52	<b>59.38</b>
screw	25.10	32.92	28.81	26.75	28.81	<b>37.45</b>	<b>48.15</b>
tile	59.65	48.54	68.42	42.69	26.90	<b>74.85</b>	<b>84.21</b>
transistor	38.09	29.76	41.67	32.14	35.72	<b>52.38</b>	<b>60.71</b>
wood	41.27	28.57	47.62	30.95	24.60	<b>49.21</b>	<b>71.43</b>
zipper	22.76	14.63	26.42	21.54	18.70	<b>27.64</b>	<b>69.51</b>
Average	39.31	35.06	40.55	32.43	34.77	<b>49.61</b>	<b>66.09</b>

Table 4: **Comparison on anomaly classification accuracy** trained on the generated data by the anomaly generation models with a ResNet-18.

**Anomaly generation quality.** We compare our model with DiffAug, CDC, Crop&Paste, SDGAN, DefectGAN and DFMGAN on anomaly generation quality and diversity in Tab. 1. Since DRAEM and PRN crop random textures to imitate anomalies, we cannot compute IC-LPIPS for them. For each anomaly category, we allocate one-third of the anomaly data for training and generate 1000 anomalies to compute IS and IC-LPIPS. It demonstrates that our model generates anomalies with both the highest quality and diversity.

Moreover, we exhibit the generated anomalies in Fig. 4. It can be seen that our model excels in producing high-quality authentic anomalies that accurately align with their corresponding masks. In contrast, CDC yields visually perplexing outcomes, particularly for structural anomaly categories like capsule-squeeze. SDGAN and DefectGAN yield poor outputs, frequently encountering difficulties in generating anomalies such as pill-crack. The state-of-the-art model DFMGAN sometimes struggles to produce authentic anomalies and fails to keep the alignment between the generated anomalies and masks, as shown in metal nut-bent. More results are presented in supplementary material.

**Anomaly generation for anomaly detection and localization.** We compare the performance of our approach with existing anomaly generation methods in downstream anomaly detection and localization. Due to the inability of DiffAug and SDGAN to generate anomaly masks, we only compare our method with Crop&Paste, DRAEM, PRN and DFMGAN. For each method, we generate 1000 images per anomaly category and train an U-Net (Ronneberger, Fischer, and Brox 2015) alongside normal samples for anomaly localization. The localization outcomes are aggregated using average pooling to derive confidence scores for image-level anomaly detection (the same as DREAM). We compute pixel-level metrics including AUROC, AP,  $F_1$ -max. The results, as presented in Tab. 2, illustrate that our model outperforms other anomaly generation models at most conditions. Furthermore, we also evaluate image-level AUROC, AP, and  $F_1$ -max scores in Tab. 3. It demonstrates our model has the best anomaly detection performance compared to other methods. We also compare the qualitative results on anomaly localization in Fig. 5, which shows our superior performance in localizing the anomalies.

Category	Unsupervised							Supervised			
	KDAD	CFLOW	DRAEM	SSPCAB	CFA	RD4AD	PatchCore	DevNet	DRA	PRN	Ours
bottle	94.7/50.5	98.8/49.9	99.1/88.5	98.9/88.6	98.9/50.9	98.8/51.0	97.6/75.0	96.7/67.9	91.7/41.5	<b>99.4</b> /92.3	99.3/ <b>94.1</b>
cable	79.2/11.6	98.9/72.6	94.8/61.4	93.1/52.1	98.4/79.8	98.8/77.0	96.8/65.9	97.9/67.6	86.1/34.8	98.8/78.9	<b>99.2</b> / <b>90.8</b>
capsule	96.3/ 9.9	<b>99.5</b> /64.0	97.6/47.9	90.4/48.7	<b>98.9</b> / <b>71.1</b>	99.0/60.5	98.6/46.6	91.1/46.6	88.5/11.0	98.5/62.2	98.8/57.2
carpet	91.5/45.8	<b>99.7</b> /67.0	96.3/62.5	92.3/49.1	99.1/47.7	99.4/46.0	98.7/65.0	94.6/19.6	98.2/54.0	99.0/ <b>82.0</b>	98.6/81.2
grid	89.0/ 7.6	99.1/ <b>87.8</b>	99.5/53.2	<b>99.6</b> /58.2	98.6/82.9	98.0/75.4	97.2/23.6	90.2/44.9	86.2/28.6	98.4/45.7	98.3/52.9
hazelnut	95.0/34.2	97.9/67.2	99.5/88.1	99.6/94.5	98.5/80.2	94.2/57.2	97.6/55.2	76.9/46.8	88.8/20.3	99.7/93.8	<b>99.8</b> / <b>96.5</b>
leather	98.2/26.7	99.2/ <b>91.1</b>	98.8/68.5	97.2/60.3	96.2/60.9	96.6/53.5	98.9/43.4	94.3/66.2	97.2/ 5.1	99.7/69.7	<b>99.8</b> /79.6
metal nut	81.7/30.6	98.8/78.2	<b>98.7</b> /91.6	99.3/95.1	98.6/74.6	97.3/53.8	97.5/86.6	93.3/57.4	80.3/30.6	99.7/98.0	<b>99.8</b> / <b>98.7</b>
pill	90.1/23.1	98.9/60.3	<b>97.7</b> /44.8	96.5/48.1	98.8/67.9	98.4/58.1	<b>97.0</b> /75.9	98.9/79.9	79.6/22.1	99.5/91.3	<b>99.8</b> / <b>97.0</b>
screw	95.4/ 5.9	98.8/45.7	<b>99.7</b> / <b>72.9</b>	99.1/62.0	98.7/61.4	99.1/51.8	98.7/34.2	66.5/21.1	51.0/ 5.1	97.5/44.9	97.0/51.8
tile	78.6/26.7	98.0/86.7	99.4/96.4	99.2/96.3	98.6/92.6	97.4/78.2	94.9/56.0	88.7/63.9	91.0/54.4	<b>99.6</b> / <b>96.5</b>	99.2/93.9
toothbrush	95.6/20.0	99.1/56.9	97.3/49.2	97.5/38.9	98.4/61.7	99.0/63.1	97.6/37.1	96.3/52.4	74.5/ 4.8	<b>99.6</b> / <b>78.1</b>	99.1/76.5
transistor	76.0/25.9	98.8/40.6	92.2/56.0	85.3/36.5	98.6/82.9	<b>99.6</b> /50.3	91.8/66.7	55.2/ 4.4	79.3/11.2	98.4/85.6	99.3/ <b>92.6</b>
wood	88.3/24.7	98.9/47.2	97.6/81.6	97.2/77.1	97.6/25.6	<b>99.3</b> /39.1	95.7/54.3	93.1/47.9	82.9/21.0	97.8/82.6	98.9/ <b>84.6</b>
zipper	95.1/30.5	96.5/63.9	98.6/73.6	98.1/78.2	95.9/53.9	<b>99.7</b> /52.7	98.5/63.1	92.4/53.1	96.8/42.3	98.8/77.6	99.4/ <b>86.0</b>
Average	89.6/24.9	98.7/65.3	97.7/69.0	96.2/65.5	98.3/66.3	98.3/57.8	97.1/56.6	86.4/49.3	84.8/25.7	99.0/78.6	<b>99.1</b> / <b>81.4</b>

Table 5: **Comparison on pixel-level anomaly localization (AUROC/AP)** between the simple U-Net trained on our generated dataset and the existing anomaly detection methods with their official codes or pre-trained models.

SAE	Method Masked $\mathcal{L}$	AAR	Metric		
			AUROC	AP	$F_1$ -max
✓			81.3	31.1	46.5
✓	✓		90.3	51.2	60.7
✓	✓	✓	95.0	64.9	68.8
✓	✓	✓	95.5	67.5	68.9
			<b>99.1</b>	<b>81.4</b>	<b>76.3</b>

Table 6: **Ablation study** on our spatial anomaly embedding (SAE), masked diffusion loss (Masked  $\mathcal{L}$ ) and adaptive attention re-weighting mechanism (AAR).

**Anomaly generation for anomaly classification.** To further validate the generation quality of our model, we employ the generated anomalies to train a downstream anomaly classification model. Specifically, we adopt the experiment setting in DFMGAN, which trains a ResNet-34 (He et al. 2016) on the generated dataset and test the classification accuracy on the remaining shared test dataset. The comparison results are shown in Tab. 4. It can be seen that our model outperforms all other models in almost all types of components and the average accuracy (**66.09%**) surpasses that of the second-ranked DFMGAN (49.61%) by a margin of **16.48%**.

### 4.3 Comparison with Anomaly Detection Models

To further validate the efficacy of our model, we conduct a comparative experiment with the state-of-the-art anomaly detection methods CFLow (Gudovskiy, Ishizaka, and Kozuka 2022), DRAEM (Zavrtanik, Kristan, and Skočaj 2021), CFA (Lee, Lee, and Song 2022), RD4AD (Deng and Li 2022), PatchCore (Roth et al. 2022), DevNet (Pang et al. 2021), DRA (Ding, Pang, and Shen 2022) and PRN (Zhang et al. 2023a). We employ their official codes or pre-trained models and evaluate them on the same testing dataset that we use. It is worth noting that due to the absence of the open-source code for PRN, we utilize the data provided in its paper. The comparison results on pixel-level AUROC and AP are presented in Tab. 5. It can be seen that although our model is only a simple U-Net, with the help of our generated anomaly data, it has a good performance in anomaly

localization with the highest AP of **81.4%** and AUROC of **99.1%**, indicating the profound significance of our generated data for downstream anomaly inspection tasks.

### 4.4 Ablation Study

We evaluate the effectiveness of our components: spatial anomaly embedding (SAE), masked diffusion loss (Masked  $\mathcal{L}$ ) and adaptive attention re-weighting mechanism (AAR). Not that the models without SAE employ only an anomaly embedding trained by textual inversion. We train 5 models: **1**) with none of these components; **2**) only SAE; **3**) SAE + masked  $\mathcal{L}$ ; **4**) masked  $\mathcal{L}$  + AAR and **5**) the full model (ours). We employ these models to generate 1000 anomalous image-mask pairs and train an U-Net for anomaly localization. We compare the pixel-level localization results in Tab. 6. It demonstrates that the omission of any of the proposed modules leads to a noticeable decline in the model’s performance on anomaly localization, which validates the efficacy of the proposed modules. For more experiments, please refer to the supplementary material.

## 5 Conclusion

In this paper, we propose *Anomalydiffusion*, a novel anomaly generation model which generates anomalous image-mask pairs. We disentangle anomaly information into anomaly appearance and location information represented by anomaly embedding and spatial embedding in the textual space of LDM. Moreover, we also introduce an adaptive attention re-weighting mechanism, which helps our model focus more on the areas with less noticeable generated anomalies, thus improving the alignment between the generated anomalies and masks. Extensive experiments show that our model outperforms the existing anomaly generation methods and our generated anomaly data effectively improves the performance of the downstream anomaly inspection tasks. In future work, we would explore the application of a more potent diffusion model to enhance the resolution of the generated anomalies, which could further improve the performance.

## Acknowledgments

This work was supported by National Natural Science Foundation of China (62302297, 72192821, 62272447), Young Elite Scientists Sponsorship Program by CAST (2022QNRC001), Shanghai Sailing Program (22YF1420300), Beijing Natural Science Foundation (L222117), the Fundamental Research Funds for the Central Universities (YG2023QNB17), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Science and Technology Commission (21511101200), CCF-Tencent Open Research Fund (RAGR20220121).

## References

- Avrahami, O.; Lischinski, D.; and Fried, O. 2022. Blended diffusion for text-driven editing of natural images. In *CVPR*, 18208–18218.
- Bergmann, P.; Fauser, M.; Sattlegger, D.; and Steger, C. 2019. MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection. In *CVPR*, 9592–9600.
- Bińkowski, M.; Sutherland, D. J.; Arbel, M.; and Gretton, A. 2018. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*.
- Cao, Y.; Wan, Q.; Shen, W.; and Gao, L. 2022. Informative knowledge distillation for image anomaly segmentation. *Knowledge-Based Systems*, 248: 108846.
- Cao, Y.; Xu, X.; Sun, C.; Cheng, Y.; Du, Z.; Gao, L.; and Shen, W. 2023. Segment Any Anomaly without Training via Hybrid Prompt Regularization. *arXiv preprint arXiv:2305.10724*.
- Chen, X.; Han, Y.; and Zhang, J. 2023. A Zero-/Few-Shot Anomaly Classification and Segmentation Method for CVPR 2023 VAND Workshop Challenge Tracks 1&2: 1st Place on Zero-shot AD and 4th Place on Few-shot AD. *arXiv preprint arXiv:2305.17382*.
- Chen, X.; Zhang, J.; Tian, G.; He, H.; Zhang, W.; Wang, Y.; Wang, C.; Wu, Y.; and Liu, Y. 2023. CLIP-AD: A Language-Guided Staged Dual-Path Model for Zero-shot Anomaly Detection. *arXiv preprint arXiv:2311.00453*.
- Deng, H.; and Li, X. 2022. Anomaly detection via reverse distillation from one-class embedding. In *CVPR*, 9737–9746.
- Ding, C.; Pang, G.; and Shen, C. 2022. Catching both gray and black swans: Open-set supervised anomaly detection. In *CVPR*, 7388–7398.
- Duan, Y.; Hong, Y.; Niu, L.; and Zhang, L. 2023. Few-Shot Defect Image Generation via Defect-Aware Feature Manipulation. In *AAAI*, volume 37, 571–578.
- Gal, R.; Alaluf, Y.; Atzmon, Y.; Patashnik, O.; Bermano, A. H.; Chechik, G.; and Cohen-Or, D. 2022. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *NIPS*, 27.
- Gu, Z.; Liu, L.; Chen, X.; Yi, R.; Zhang, J.; Wang, Y.; Wang, C.; Shu, A.; Jiang, G.; and Ma, L. 2023. Remembering Normality: Memory-guided Knowledge Distillation for Unsupervised Anomaly Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16401–16409.
- Gudovskiy, D.; Ishizaka, S.; and Kozuka, K. 2022. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 98–107.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.
- Hertz, A.; Mokady, R.; Tenenbaum, J.; Aberman, K.; Pritch, Y.; and Cohen-Or, D. 2022. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*.
- Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NIPS*, 30.
- Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.
- Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.
- Hu, T.; Zhang, J.; Liu, L.; Yi, R.; Kou, S.; Zhu, H.; Chen, X.; Wang, Y.; Wang, C.; and Ma, L. 2023. Phasic Content Fusing Diffusion Model with Directional Distribution Consistency for Few-Shot Model Adaption. In *ICCV*, 2406–2415.
- Huang, C.; Guan, H.; Jiang, A.; Zhang, Y.; Spratling, M.; and Wang, Y.-F. 2022. Registration based few-shot anomaly detection. In *ECCV*, 303–319. Springer.
- Jeong, J.; Zou, Y.; Kim, T.; Zhang, D.; Ravichandran, A.; and Dabeer, O. 2023. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19606–19616.
- Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and improving the image quality of stylegan. In *CVPR*, 8110–8119.
- Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Lee, S.; Lee, S.; and Song, B. C. 2022. CfA: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *IEEE Access*, 10: 78446–78454.
- Li, C.-L.; Sohn, K.; Yoon, J.; and Pfister, T. 2021. Cutpaste: Self-supervised learning for anomaly detection and localization. In *CVPR*, 9664–9674.
- Li, Y.; Zhang, R.; Lu, J.; and Shechtman, E. 2020. Few-shot image generation with elastic weight consolidation. *arXiv preprint arXiv:2012.02780*.
- Liang, Y.; Zhang, J.; Zhao, S.; Wu, R.; Liu, Y.; and Pan, S. 2023. Omni-frequency channel-selection representations for unsupervised anomaly detection. *IEEE Transactions on Image Processing*.

- Lin, D.; Cao, Y.; Zhu, W.; and Li, Y. 2021. Few-shot defect segmentation leveraging abundant defect-free training samples through normal background regularization and crop-and-paste operation. In *ICME*, 1–6. IEEE.
- Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *CVPR*, 2117–2125.
- Mo, S.; Cho, M.; and Shin, J. 2020. Freeze the discriminator: a simple baseline for fine-tuning gans. *arXiv preprint arXiv:2002.10964*.
- Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *ICML*, 8162–8171. PMLR.
- Niu, S.; Li, B.; Wang, X.; and Lin, H. 2020. Defect image sample generation with GAN for improving defect recognition. *IEEE Transactions on Automation Science and Engineering*, 17(3): 1611–1622.
- Ojha, U.; Li, Y.; Lu, J.; Efros, A. A.; Lee, Y. J.; Shechtman, E.; and Zhang, R. 2021. Few-shot image generation via cross-domain correspondence. In *CVPR*, 10743–10752.
- Pang, G.; Ding, C.; Shen, C.; and Hengel, A. v. d. 2021. Explainable deep few-shot anomaly detection with deviation networks. *arXiv preprint arXiv:2108.00462*.
- Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*, 10684–10695.
- Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III* 18, 234–241. Springer.
- Roth, K.; Pemula, L.; Zepeda, J.; Schölkopf, B.; Brox, T.; and Gehler, P. 2022. Towards total recall in industrial anomaly detection. In *CVPR*, 14318–14328.
- Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; and Aberman, K. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 22500–22510.
- Schlegl, T.; Seeböck, P.; Waldstein, S. M.; Langs, G.; and Schmidt-Erfurth, U. 2019. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Medical image analysis*, 54: 30–44.
- Schlegl, T.; Seeböck, P.; Waldstein, S. M.; Schmidt-Erfurth, U.; and Langs, G. 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, 146–157. Springer.
- Schuhmann, C.; Vencu, R.; Beaumont, R.; Kaczmarczyk, R.; Mullis, C.; Katta, A.; Coombes, T.; Jitsev, J.; and Komatsuzaki, A. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.
- Tran, N.-T.; Tran, V.-H.; Nguyen, N.-B.; Nguyen, T.-K.; and Cheung, N.-M. 2021. On data augmentation for gan training. *IEEE Transactions on Image Processing*, 30: 1882–1897.
- Wang, Y.; Peng, J.; Zhang, J.; Yi, R.; Wang, Y.; and Wang, C. 2023. Multimodal Industrial Anomaly Detection via Hybrid Fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8032–8041.
- Wang, Y.; Yi, R.; Tai, Y.; Wang, C.; and Ma, L. 2022. Ctl-gan: Few-shot artistic portraits generation with contrastive transfer learning. *arXiv preprint arXiv:2203.08612*.
- Zavrtanik, V.; Kristan, M.; and Skočaj, D. 2021. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *ICCV*, 8330–8339.
- Zhang, G.; Cui, K.; Hung, T.-Y.; and Lu, S. 2021. Defect-GAN: High-fidelity defect synthesis for automated defect inspection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2524–2534.
- Zhang, H.; Wu, Z.; Wang, Z.; Chen, Z.; and Jiang, Y.-G. 2023a. Prototypical residual networks for anomaly detection and localization. In *CVPR*, 16281–16291.
- Zhang, J.; Chen, X.; Xue, Z.; Wang, Y.; Wang, C.; and Liu, Y. 2023b. Exploring Grounding Potential of VQA-oriented GPT-4V for Zero-shot Anomaly Detection. *arXiv preprint arXiv:2311.02612*.
- Zhao, S.; Liu, Z.; Lin, J.; Zhu, J.-Y.; and Han, S. 2020. Differentiable augmentation for data-efficient gan training. *NIPS*, 33: 7559–7570.

## A Overview

This supplementary material consists of:

- Details of the data augmentation method (Sec. B).
- More implementation details (Sec. C).
- More ablation studies (Sec. D).
- Comparison between our Spatial Anomaly Embedding and Prompt-to-Prompt (Sec. E).
- More qualitative comparison results with the anomaly generation methods (Sec. F).
- More quantitative comparison results with the anomaly generation methods (Sec. G).

## B Data Augmentation

Due to the limited number of samples for each anomaly category, typically less than 10 images are available for training. This constraint makes it challenging for our anomaly embedding to completely eliminate spatial information, as it still tends to generate anomalies at the positions observed in the training images. Additionally, when the training data for the spatial encoder is scarce, the model becomes susceptible to overfitting, making it difficult to accurately generate anomalies at the correct positions.

To address these issues, we employ a data augmentation approach during training. For paired image-mask data, we perform random cropping, translation, and rotation on both the image and its corresponding mask. By recording the maximum and minimum coordinates of the anomaly region in the image, we ensure that the anomaly remains within the image during data augmentation. This data augmentation process effectively disrupts the spatial information within the training data, causing the anomaly embedding to lose its focus on recording positions and instead concentrate solely on the anomaly appearance. Simultaneously, the spatial encoder benefits from having enough augmented data for training, boosting its ability in position encoding.

## C Implementation Details

### C.1 Training Details

**Training spatial anomaly embedding.** For each anomaly type, an anomaly embedding  $e_a$  is assigned, while a shared spatial encoder  $E$  is employed across all anomaly categories. Each anomaly embedding  $e_a$  is composed of 8 tokens, and the spatial embedding  $e_s$  comprises 4 tokens. The batch size is set at 4, and the learning rate is 0.005. During each training iteration, we randomly sample 4 anomalous image-mask pairs from all the anomaly categories. We train all the anomaly embedding and spatial encoder at the same time for 300K iterations in 3 days on an NVIDIA GeForce RTX 3090 24GB GPU.

**Training mask embedding.** For each anomaly type, we assign a mask embedding  $e_m$  for it. To enhance the diversity of the generated masks, each mask embedding consists of 2 tokens, preventing it from overfitting. Furthermore, with a batch size of 4 and a learning rate of 0.005, each mask embedding is trained for 30K iterations.

**Mask Generation.** With the trained mask embedding, we input it as a text condition to guide the generation process of latent diffusion model (Rombach et al. 2022). Specifically, we employ the classifier-free guidance (Ho and Salimans 2022) to generate masks:

$$\hat{\epsilon}_\theta(x_t | e_m) = \epsilon_\theta(x_t) + s \cdot (\epsilon_\theta(x_t, e_m) - \epsilon_\theta(x_t)), \quad (10)$$

where  $s$  is set 5 (the same as Textual Inversion).

## C.2 Metrics

In the quantitative experiments, we employ the following metrics to measure the model performance.

- **Inception Score (IS)** quantifies the quality and diversity of generated images by computing the exponential of the negative of the KL divergence between the marginal distribution of generated images and the conditional distribution of class labels predicted by an Inception model. A higher IS score represents a better generation quality and diversity.
- **Intra-cluster Pairwise LPIPS Distance (IC-LPIPS)** (Ojha et al. 2021) clusters the generated images into  $k$  groups based on LPIPS distance to  $k$  target samples, and then compute the average mean LPIPS distances to corresponding target samples within each cluster. A higher IC-LPIPS indicates a better generation diversity.
- **Area Under the Receiver Operating Characteristic (AUROC)** measures the performance of a binary classification model by evaluating its ability to distinguish between true positive and false positive rates across different probability thresholds. A higher AUROC means better anomaly detection and localization performance.
- **Average Precision (AP, which is also known as PR-AUC)** assesses the precision-recall curve for a classification model, calculating the average precision of the model across different recall levels, providing a summary of its overall performance. A higher AP means better anomaly detection and localization results.
- **F<sub>1</sub>-max** is a variant of the  $F_1$  score that maximizes both precision and recall by selecting the threshold that yields the highest  $F_1$  score when evaluating a binary classification model. A higher  $F_1$ -max represents better anomaly detection and localization results

## D More Ablation Studies

### D.1 Ablation on Spatial Anomaly Embedding

We aim to seek a text embedding that guides the latent diffusion model in generating anomalies within a given anomaly mask. However, textual inversion tends to capture the location of anomalies along with the anomaly type information, which results in the generated anomalies only distributed in specific locations. Therefore, we propose spatial anomaly embedding  $e$ , consisting of an anomaly embedding  $e_a$  (for appearance) and a spatial embedding  $e_s$  (for location), which disentangles the spatial information from anomaly appearance. To further validate this theory, we directly employ

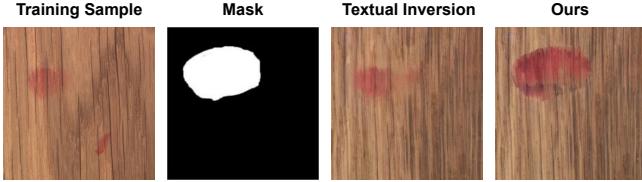


Figure 6: **Comparison results between Textual Inversion (Anomaly Embedding only) and our model (Spatial Anomaly Embedding).** The generated result of Textual Inversion tends to generate anomalies at the location as the same as the training sample.

Anomaly Rate	10%	20%	<b>30% (Ours)</b>	40%	50%
AUROC $\uparrow$	96.2	98.1	<b>99.1</b>	99.0	98.7
AP $\uparrow$	64.2	75.5	<b>81.4</b>	81.1	80.0

Table 7: **ablation study on the rate of anomalies.**

text inversion to train an anomaly embedding and use the trained embedding to generate anomalies with a given mask by blended latent diffusion (the same as our generation process). The generated results are shown in Fig. 6. It can be seen that the generated result by Textual Inversion tends to generate anomalies at the location the same as the training sample, which limits its application in anomaly generation where anomalies can be located at arbitrary positions.

## D.2 Ablation on the rate of anomalies

We conduct additional experiments with the rate of anomalies as 10%, 20%, 30% (**Ours**), 40%, and 50% and test the performance on anomaly localization measured by AUROC and AP. The performance of anomaly localization is shown in Tab. 7. It can be seen that AP decreases quickly when the anomaly rate falls below 30%. This is attributed to the limited availability of training data for most categories, often comprising only 1-2 instances, making it challenging for the model to capture the anomaly information. Conversely, when the rate exceeds 30%, the model performance is similar. This indicates that our model can effectively learn sufficient anomaly information without a heavy reliance on an abundance of training samples.

## D.3 Ablation on the hyperparameters

We conduct ablation studies on the length of the anomaly embedding  $l_a$  and spatial embedding  $l_s$  in Tab. 8. Specifically, we train models with different  $l_a$  and  $l_s$  and then employ their generated data to train a UNet to localize the anomalies. It can be seen that when increasing  $l_s$  and  $l_a$ , the total parameter number of the model rises, but the final performance in the downstream anomaly localization task is similar, which demonstrates that our model is not sensitive to the hyperparameters.

## D.4 Ablation on SAE and AAR

We conduct more ablation studies on the effectiveness of our spatial anomaly embedding (SAE) and adaptive attention re-weighting mechanism (AAR) by adding SAE and

Model	AUROC $\uparrow$	AP $\uparrow$	$F_1$ -max $\uparrow$	PRO $\uparrow$
$L_s = 4, l_a = 8$ ( <b>Ours</b> )	<b>99.1</b>	<b>81.4</b>	<b>76.3</b>	<b>94.0</b>
$L_s = 4, l_a = 16$	98.8	80.6	75.1	93.2
$L_s = 8, l_a = 8$	99.0	80.9	75.8	93.5
$L_s = 8, l_a = 16$	<b>99.1</b>	81.2	75.9	93.8

Table 8: **Ablation study on the anomaly embedding  $l_a$  and spatial embedding  $l_s$**



Figure 7: **Ablation study on SAE and AAR.**

AAR separately. We compare our model with 3 models: 1) w/o AAR&SAE, 2) AAR only, and 3) SAE only in generating glue anomalies to the leather in the Fig. 7. It shows that the model without AAR&SAE cannot generate authentic anomalies or fill anomaly mask. While adding SAE improves anomaly authenticity, it doesn't fill the mask. Moreover, incorporating AAR fills the mask but sacrifices authenticity. In contrast, our model (SAE + AAR) effectively generates authentic anomalies filling the mask.

## E Comparison with Prompt-to-Prompt

Prompt-to-Prompt (Hertz et al. 2022) proposed a method that allows modifying generated images by altering corresponding text descriptions. For instance, when transforming "a cat sits on the street" to "a dog sits on the street," Prompt-to-Prompt replaces the cross-attention map of "dog" with that of "cat", which transforms the cat in the original image into a dog while maintaining nearly unchanged content in other regions, achieving controlled image generation with specific positions. However, Prompt-to-Prompt requires a text corresponding to the original image for generation, which is unavailable in anomaly generation.

It seems that Prompt-to-prompt presents a potential solution for controlling generation positions through the manipulation of cross-attention maps. A direct solution is to resize the mask  $m$  to match and substitute the cross-attention map  $m_c$  of anomaly embedding, thus controlling the generation location. However, even though the new cross-attention map  $m'_c$  seemingly dictates anomaly location, it could conflict with the values  $V$  in the cross-attention module. Since  $V$  is designed for the original cross-attention map  $m_c$ , the semantic information of  $V$  in the newly enforced mask  $m'_c$  might not align with the semantics in original mask  $m_c$ , consequently leading to a unstable generated results.

To verify it, we conduct reconstruction experiments on real anomalies, comparing the results of Textual Inversion + Prompt-to-Prompt with our approach (Spatial Anomaly Embedding). Specifically, we sample a real anomaly image  $I$  as ground truth and mask out the anomaly parts for generation. The results are shown in Fig. 8. Textual Inversion + Prompt-to-Prompt can not generate anomalies as authentic

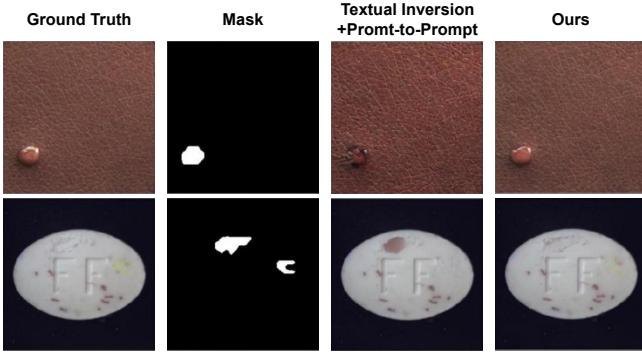


Figure 8: Comparison results between Textual Inversion + Prompt-to-Prompt and our model (Spatial Anomaly Embedding) in anomaly generation.

Method	Metric			
	AUROC	AP	$F_1$ -max	PRO
Textual Inversion+Prompt-to-Prompt	91.2	55.1	64.4	73.5
Ours	<b>99.1</b>	<b>81.4</b>	<b>76.4</b>	<b>94.0</b>

Table 9: Comparison with Textual Inversion + Prompt-to-Prompt on anomaly localization.

as ours. And its generated anomalies are quite different from the ground truth, indicating that replacing the cross-attention map directly cannot generate satisfying anomalies. Moreover, we also conduct a quantitative experiment, where we generate anomalous image-mask pairs to support the downstream anomaly localization task. We follow the experiment settings in the main paper, in which we train an U-Net on the generated data and compare the localization accuracy. The results are recorded in Tab. 9. Our model outperforms Textual Inversion + Prompt-to-Prompt significantly.

## F More qualitative experiments

We give a more comprehensive comparison with the existing anomaly generation methods DiffAug (Zhao et al. 2020), CDC (Ojha et al. 2021), Crop&Paste, SDGAN (Niu et al. 2020), Defect-GAN (Zhang et al. 2021) and DFM-GAN (Duan et al. 2023). We exhibit the generation results of all the anomaly generation methods across all components in Fig. 9. Our model demonstrates remarkable proficiency in generating high-quality, authentic anomalies that are precisely aligned with the corresponding masks. In contrast, Crop&Paste exhibits limited diversity in generating various anomaly types. DiffAug displays evident overfitting tendencies towards the training samples (the image in the lower-right corner). CDC yields visually perplexing results, particularly for structural anomaly categories like capsule-squeeze. SDGAN and DefectGAN yield poor outputs, frequently encountering challenges in generating anomalies such as pill-crack. The state-of-the-art model DFMGAN occasionally struggles to create authentic anomalies and fails in maintaining alignment between the generated anomalies

and masks, as observed in the case of metal nut-bent. In comparison, our model generates anomalies with the highest diversity and authenticity, and the generated anomalies align with the masks accurately, which can effectively support the downstream anomaly inspection tasks.

## G More Quantitative experiments

**More comparison with the anomaly generation models.** In this section, we provide supplementary experiments to complement those presented in the main paper. Specifically, in addition to the methods covered in the main paper, we include Crop&Paste (Lin et al. 2021) for comparison and we additionally introduce the Per Region Overlap (PRO) metric to provide a more comprehensive evaluation on anomaly localization. The experiment settings are the same as that in the main paper, where we train an U-net on the generated anomaly data. The pixel-level anomaly localization results are shown in Tab. 10 and the image-level anomaly detection results are shown in Tab. 11. The quantitative results demonstrate that our model outperforms all the other anomaly generation methods in terms of both anomaly localization and detection, indicating our good anomaly generation quality and diversity.

**More comparison with the anomaly localization models.** In this section, we further compare the anomaly detection methods with  $F_1$ -max score on anomaly localization. The results are shown in Table 12. It can be seen that our model achieves the best performance in anomaly localization with  $F_1$ -max.

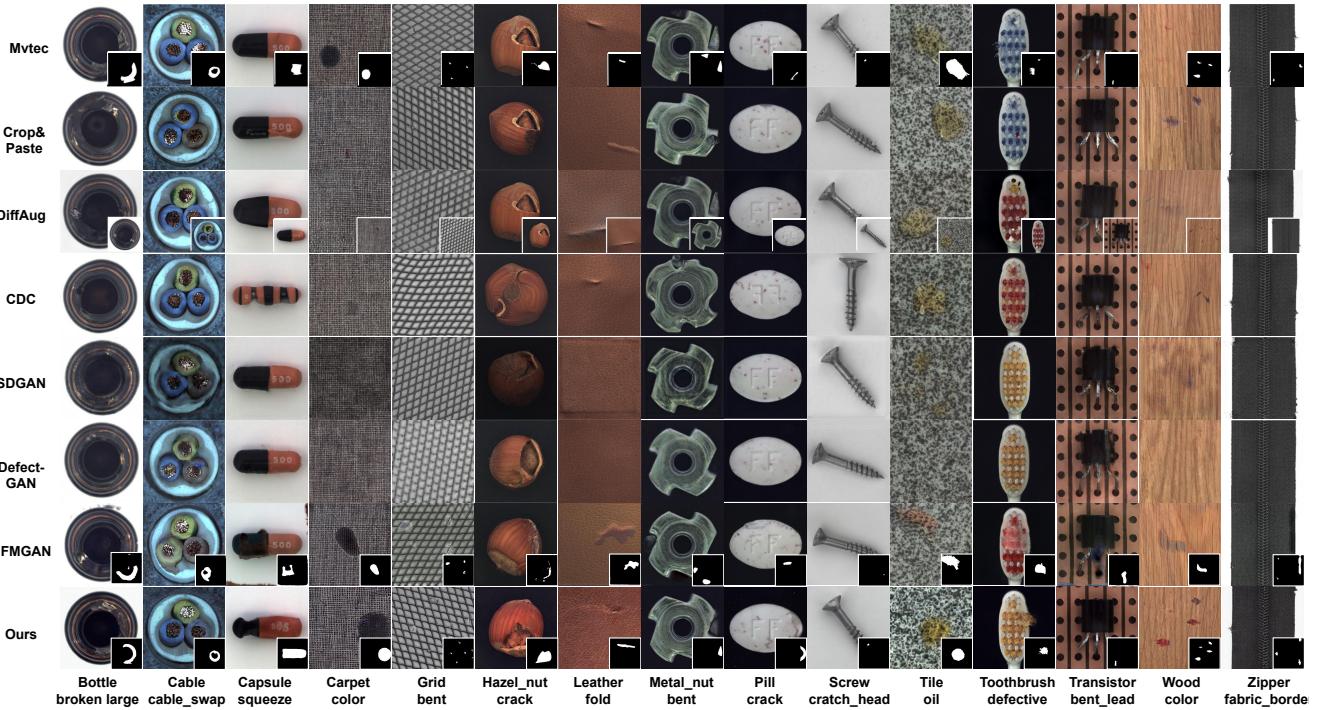


Figure 9: **Qualitative comparison on the anomaly generation quality.** Note that the generated anomalies by DiffAug is the same as the training samples (images in lower-right corner)

Category	Crop&Paste				DRAEM				PRN				DFMGAN				Ours			
	AUC	AP	$F_1$ -max	PRO	AUC	AP	$F_1$ -max	PRO	AUC	AP	$F_1$ -max	PRO	AUC	AP	$F_1$ -max	PRO	AUC	AP	$F_1$ -max	PRO
bottle	94.5	67.4	63.5	77.8	96.7	80.2	74.0	91.2	97.5	76.4	71.3	88.5	98.9	90.2	83.9	91.7	<b>99.4</b>	<b>94.1</b>	87.3	94.3
cable	96.0	75.3	69.3	<u>87.1</u>	80.3	21.8	28.3	58.2	94.5	64.4	61.0	79.7	<u>97.2</u>	81.0	75.4	84.9	<b>99.2</b>	<b>90.8</b>	83.5	95.0
capsule	95.3	49.2	<u>51.1</u>	89.5	76.2	25.5	32.1	81.1	95.6	45.7	47.9	89.7	79.2	26.0	35.0	66.1	<b>98.8</b>	<u>57.2</u>	<u>59.8</u>	95.4
carpet	83.7	36.6	39.7	62.9	92.6	43.0	41.9	80.0	<u>96.4</u>	<u>69.6</u>	65.6	<u>90.6</u>	90.6	33.4	38.1	76.5	<b>98.6</b>	<u>81.2</u>	74.6	91.6
grid	84.7	13.1	22.4	70.2	<b>99.1</b>	<b>59.3</b>	<u>58.7</u>	<b>95.8</b>	98.9	<u>58.6</u>	<b>58.9</b>	<b>95.8</b>	75.2	14.3	20.5	52.3	98.3	52.9	54.6	<u>92.3</u>
hazelnut	88.5	38.0	42.8	74.1	98.8	73.6	68.5	95.9	98.0	73.9	68.2	92.7	99.7	<u>95.2</u>	89.5	96.4	<b>99.8</b>	<b>96.5</b>	<u>90.6</u>	<b>97.1</b>
leather	<u>97.5</u>	76.0	<u>70.8</u>	95.7	98.5	67.6	65.0	96.7	<u>99.4</u>	58.1	54.0	<u>97.5</u>	98.5	68.7	66.7	96.0	<b>99.8</b>	<b>79.6</b>	<u>71.0</u>	<u>98.2</u>
metal nut	96.3	84.2	74.0	67.2	96.9	84.2	74.5	90.4	97.9	93.0	87.1	85.0	99.3	<u>98.1</u>	<b>94.5</b>	88.0	<b>99.8</b>	<b>98.7</b>	94.0	<b>94.8</b>
pill	81.5	17.8	24.3	57.4	95.8	45.3	53.0	83.7	<u>98.3</u>	55.5	<u>72.6</u>	88.2	81.2	<u>67.8</u>	72.6	56.5	<b>99.8</b>	<b>97.0</b>	<u>90.8</u>	<u>97.3</u>
screw	93.4	31.2	36.0	<b>83.9</b>	91.0	30.1	35.7	78.1	<u>94.0</u>	<u>47.7</u>	49.8	<u>83.8</u>	58.8	2.2	5.3	41.8	<b>97.0</b>	<b>51.8</b>	<u>50.9</u>	80.3
tile	94.0	79.3	74.5	79.2	98.5	93.2	<u>87.8</u>	95.3	98.5	91.8	84.4	91.3	<b>99.5</b>	<b>97.1</b>	<b>91.6</b>	<b>97.5</b>	99.2	<u>93.9</u>	86.2	<u>96.1</u>
toothbrush	89.3	30.9	34.6	66.6	93.8	29.5	28.4	75.1	96.1	46.4	46.2	83.1	96.4	<u>75.9</u>	<u>72.6</u>	74.3	<b>99.2</b>	<b>76.5</b>	<u>73.4</u>	<u>91.4</u>
transistor	85.9	52.5	52.1	64.5	76.5	31.7	24.2	54.3	94.9	68.6	68.4	<u>70.0</u>	96.2	<u>81.2</u>	77.0	65.5	<b>99.3</b>	<b>92.6</b>	<u>85.7</u>	<u>96.2</u>
wood	84.0	45.7	48.0	57.9	<u>98.8</u>	<b>87.8</b>	<b>80.9</b>	<b>94.7</b>	96.2	74.2	67.4	82.1	95.3	70.7	65.8	89.9	<b>98.9</b>	<u>84.6</u>	74.5	<u>94.3</u>
zipper	94.8	47.6	51.4	83.4	93.4	65.4	64.7	84.6	<u>98.4</u>	<u>79.0</u>	73.7	93.7	92.9	65.6	64.9	83.0	<b>99.4</b>	<b>86.0</b>	<u>79.2</u>	<u>96.3</u>
Average	90.4	48.4	49.4	74.3	92.2	54.1	53.1	83.1	96.9	66.2	64.7	87.4	90.0	62.7	62.1	76.3	<b>99.1</b>	<b>81.4</b>	<u>76.3</u>	<u>94.0</u>

Table 10: **Comparison on the pixel-level anomaly localization** with AUC, AP,  $F_1$ -max and PRO metrics by training an U-Net on the generated datasets produced by Crop&Paste, DRAEM, PRN, DFMGAN and our model. **Bold** and underline represent optimal and sub-optimal results, respectively.

Category	Crop&Paste			DRAEM			PRN			DFMGAN			Ours		
	AUC	AP	$F_1$ -max	AUC	AP	$F_1$ -max	AUC	AP	$F_1$ -max	AUC	AP	$F_1$ -max	AUC	AP	$F_1$ -max
bottle	85.4	95.1	90.9	99.3	99.8	<b>98.9</b>	94.9	98.4	94.1	99.3	99.8	97.7	<b>99.8</b>	<b>99.9</b>	<b>98.9</b>
cable	93.3	96.1	91.6	72.1	83.2	79.2	86.3	92.0	84.0	95.9	97.8	93.8	<b>100</b>	<b>100</b>	<b>100</b>
capsule	77.1	94.1	90.4	93.2	98.7	94.0	84.9	95.8	94.3	92.8	98.5	94.5	<b>99.7</b>	<b>99.9</b>	<b>98.7</b>
carpet	57.7	84.3	87.3	95.3	98.7	93.4	92.6	97.8	92.1	67.9	87.9	87.3	<b>96.7</b>	<b>98.8</b>	<b>94.3</b>
grid	83.0	94.1	87.6	<b>99.8</b>	<b>99.9</b>	<b>98.8</b>	96.6	98.9	95.0	73.0	90.4	85.4	98.4	99.5	98.7
hazelnut	68.8	85.0	78.0	<b>100</b>	<b>100</b>	<b>100</b>	93.6	96.0	94.1	99.9	<b>100</b>	99.0	99.8	99.9	98.9
leather	91.9	97.5	90.9	<b>100</b>	<b>100</b>	<b>100</b>	99.1	99.7	97.6	99.9	<b>100</b>	99.2	<b>100</b>	<b>100</b>	<b>100</b>
metal nut	92.2	98.1	93.3	97.8	99.6	97.6	97.8	99.5	96.9	99.3	99.8	99.2	<b>100</b>	<b>100</b>	<b>100</b>
pill	51.7	87.1	91.4	94.4	98.9	95.8	88.8	97.8	93.2	68.7	91.7	91.4	<b>98</b>	<b>99.6</b>	<b>97</b>
screw	59.3	81.9	86.0	88.5	96.3	89.3	84.1	94.7	87.2	22.3	64.7	85.3	<b>96.8</b>	<b>97.9</b>	<b>95.5</b>
tile	73.8	91.1	83.8	<b>100</b>	<b>100</b>	<b>100</b>	91.1	96.9	89.3	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>
toothbrush	81.2	91.0	88.9	99.4	99.8	97.6	<b>100</b>	<b>100</b>	<b>100</b>						
transistor	85.9	81.8	80.0	79.6	80.5	71.4	88.2	88.9	84.0	90.8	92.5	88.9	<b>100</b>	<b>100</b>	<b>100</b>
wood	49.5	81.2	86.6	<b>100</b>	<b>100</b>	<b>100</b>	77.5	92.7	86.7	98.4	99.4	98.8	98.4	99.4	98.8
zipper	59.4	82.8	88.9	<b>100</b>	<b>100</b>	<b>100</b>	98.7	99.7	97.6	99.7	99.9	99.4	99.9	<b>100</b>	99.4
Average	74.0	89.4	87.7	94.6	97.0	94.4	91.6	96.6	92.4	87.2	94.8	94.7	<b>99.2</b>	<b>99.7</b>	<b>98.7</b>

Table 11: **Comparison on the image-level anomaly detection** with AUC, AP and  $F_1$ -max metrics by training an U-Net on the generated datasets produced by Crop&Paste, DRAEM, PRN, DFMGAN and our model.

$F_1$ -max	KDAD	CFLOW	DREAM	SSPCAB	CFA	RD4AD	PatchCore	DevNet	DRA	Ours
bottle	50.9	9.5	83.0	80.3	75.9	82.1	78.6	64.6	53.5	<b>87.3</b>
cable	18.3	10.5	58.5	51.2	76.3	65.2	68.5	54.9	55.3	<b>83.5</b>
capsule	15.1	6.9	48.9	49.5	57.0	60.4	56.7	38.7	47.7	<b>60.8</b>
carpet	54.2	3.5	60.0	47.1	48.3	67.8	67.9	52.3	42.3	<b>74.6</b>
grid	10.9	3.2	56.3	58.4	32.2	<b>59.9</b>	49.1	42.9	50.1	54.6
hazelnut	37.5	3.9	80.6	88.9	61.4	70.0	68.1	22.4	47.2	<b>90.6</b>
leather	30.4	3.9	63.2	58.1	53.8	67.2	54.7	32.1	19.8	<b>71.0</b>
metal nut	34.2	30.7	84.4	87.8	87.1	77.0	86.0	65.2	64.6	<b>94.0</b>
pill	29.9	17.6	62.6	46.5	79.5	63.7	73.5	22.8	45.5	<b>90.8</b>
screw	8.3	0.9	<b>66.9</b>	63.8	37.8	58.7	47.2	14.8	0.7	50.9
tile	27.8	26.6	<b>90.8</b>	88.5	77.8	71.8	69.4	69.9	61.4	86.2
toothbursh	25.1	4.7	47.5	37.2	62.1	58.7	63.8	35.1	22.6	<b>73.4</b>
transistor	26.7	19.9	55.2	34.8	76.3	59.8	64.2	28.9	33.2	<b>85.7</b>
wood	25.2	10.0	<b>75.1</b>	68.7	48.6	61.3	60.3	51.9	49.9	74.5
zipper	26.8	4.5	68.2	73.7	65.8	69.4	70.0	45.6	56.9	<b>79.2</b>
average	28.1	10.4	66.7	62.3	61.4	66.2	65.2	42.8	43.4	<b>76.4</b>

Table 12: **Comparison on anomaly localization with  $F_1$ -max.**