

Few-Shot Anomaly-Driven Generation for Anomaly Classification and Segmentation

Guan Gui^{1,*}, Bin-Bin Gao^{1,*□}
Jun Liu¹, Chengjie Wang^{1,2,□}, and Yunsheng Wu¹

¹ Tencent YouTu Lab

² Shanghai Jiao Tong University

{guiguan}@smail.nju.edu.cn, {csgaobb,junsenselee}@gamil.com,
{jasoncjwang,simonwu}@tencent.com

Abstract. Anomaly detection is a practical and challenging task due to the scarcity of anomaly samples in industrial inspection. Some existing anomaly detection methods address this issue by synthesizing anomalies with noise or external data. However, there is always a large semantic gap between synthetic and real-world anomalies, resulting in weak performance in anomaly detection. To solve the problem, we propose a few-shot Anomaly-driven Generation (AnoGen) method, which guides the diffusion model to generate realistic and diverse anomalies with only a few real anomalies, thereby benefiting training anomaly detection models. Specifically, our work is divided into three stages. In the first stage, we learn the anomaly distribution based on a few given real anomalies and inject the learned knowledge into an embedding. In the second stage, we use the embedding and given bounding boxes to guide the diffusion model to generate realistic and diverse anomalies on specific objects (or textures). In the final stage, we propose a weakly-supervised anomaly detection method to train a more powerful model with generated anomalies. Our method builds upon DRAEM and DesTSeg as the foundation model and conducts experiments on the commonly used industrial anomaly detection dataset, MVTec. The experiments demonstrate that our generated anomalies effectively improve the model performance of both anomaly classification and segmentation tasks simultaneously, *e.g.*, DRAEM and DesTSeg achieved a 5.8% and 1.5% improvement in AU-PR metric on segmentation task, respectively. The code and generated anomalous data are available at <https://github.com/gaobb/AnoGen>.

1 Introduction

Anomaly detection has wide real-world application scenarios, *e.g.*, manufacturing quality inspection and medical out-of-distribution detection. However, the

* indicates equal contribution (G. Gui and B.-B. Gao). This research was done when G. Gui was an intern at YouTu Lab, Tencent, under the supervision of B.-B. Gao.

□ indicates corresponding authors.

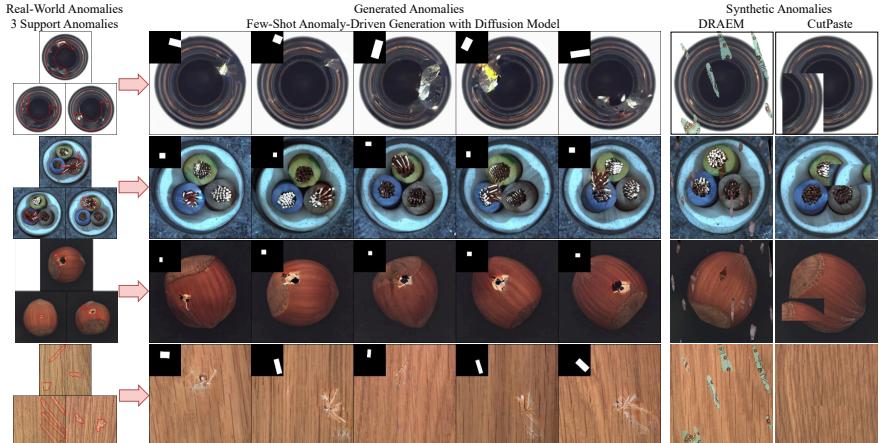


Fig. 1: Comparisons of real anomalies (left column) and generated anomalies with ours (middle column) and other methods (right column). Given *a few images of a real anomaly concept*, our AnoGen is able to generate more *realistic and diverse anomalies* through learning a pre-trained diffusion model compared to the existing synthetic methods such as DRAEM and CutPaste. Meanwhile, our generated anomalies are spatially controllable because of a given mask (*e.g.*, bounding box), which will benefit downstream anomaly detection tasks, *i.e.*, classification and segmentation.

extreme scarcity of anomaly data in the real world makes anomaly detection tasks (including image-level classification and pixel-level segmentation) highly challenging.

Faced with the fact of rare anomaly data, several works propose unsupervised learning methods to eliminate the need for anomaly data. For example, [1, 11] estimate the multivariate Gaussian distribution of normal images, [33] creates a large memory bank to store the features of normal images, and [12, 38, 48] train a reconstruction network to compare the difference between reconstructed output and original image or feature extracted from a pre-trained model. While these methods have achieved satisfactory performance in anomaly classification tasks, unfortunately, due to the lack of discriminative guidance from anomaly data, they still perform poorly in anomaly segmentation tasks.

To perform anomaly segmentation models better, some recent works, such as DRAEM [47], CutPaste [20], and SimpleNet [24], propose to artificially synthesize anomalies to train discriminative models. Specifically, DRAEM mixes an external texture dataset and normal images to synthesize anomalies, CutPaste crops an image's region and randomly pastes it to another region, and SimpleNet adds noise to the feature map to simulate anomalies. These synthesized anomalies have indeed proven beneficial for discriminative models, leading to superior performance in anomaly segmentation tasks. However, the drawback is that the synthesized anomalies are based on additional datasets or noise, which results in a significant semantic gap compared to real anomalies. This raises the question:

is it possible to create realistic and diverse anomaly images that are semantically consistent with real-world anomaly concepts, thereby further enhancing these discriminative models?

Fortunately, generative models have indeed achieved remarkable progress in creating realistic and diverse images. GANs [29] are trained to generate images by pitting a generative network against a discriminative network in an adversarial fashion. Although GANs are efficient in generating images with good perceptual quality, they are difficult to optimize and capture the full data distribution. Recently, Diffusion Models (DM) [45] have further surpassed GANs in image generation, which learn the distribution of images through a process of addnoising and denoising. However, these powerful generative model still requires a large number of training images, which raises the question: *how to generate realistic and diverse anomaly images with the DM only a few real anomaly images are available?*

To solve the above problems, we propose a few-shot anomaly-driven generation method, which aims to generate realistic and diverse anomalies under the guidance of only a few real-world anomalies. Borrowing concepts in few-shot learning [43], we call these real-world anomalies “support anomalies”. Considering that the number of support anomalies is very limited (1 or 3), it is not possible to optimize millions of parameters in the DM. Instead, we use a pre-trained diffusion model and then freeze its all parameters and only optimize an embedding vector that contains only a few hundred parameters. After training, this embedding vector is able to represent the distribution of given support anomalies, which guides the diffusion model to generate realistic and diverse anomalies as shown in Figure 1.

In the generation process, we provide a mask (a bounding box) condition to control the position and size of the anomaly region. This mask also serves as a ground truth for downstream anomaly detection tasks, *i.e.*, discriminative anomaly segmentation. Previous work [14] attempted to generate anomalies with GANs, while failing on downstream tasks due to the absence of labels. Specifically, we employ the generated images to two discriminative models: DRAEM [47] and DeSTSeg [49]. Instead of using accurate masks in DRAEM and DeSTSeg, we have to take bounding box masks as supervision for training DRAEM and DeSTSeg. To this end, we propose a weakly-supervised learning [50] version built on DRAEM and DeSTSeg, where high-confidence normal predictions within the box region will be filtered out to alleviate their interference for model training.

It is worth noting that a concurrent study, AnomalyDiffusion [17], shares similar concepts to ours. It also generates more anomalies with a small number of real anomalies, but their implementations are different. First, AnomalyDiffusion is more complex, it trains a mask generation network, which significantly increases computational costs. In contrast, we only learn a 768-parameter embedding. Then, AnomalyDiffusion utilizes prior knowledge of anomaly masks to constrain its generated shape. This potentially limits the diversity of generated anomalies. While we do not impose such constraints and thus retain the diver-

sity of generated anomaly. Finally, to alleviate the problem of inaccurate masks, AnomalyDiffusion uses adaptive attention re-weighting to fill the mask region. However, it still cannot completely solve it. We propose a weak supervision method, which is more effective in addressing this issue, making our approach more robust and generalizable.

We conduct experiments on the commonly used industrial anomaly detection dataset, MVTec [6]. With the help of our generated images equipped with the proposed weakly supervised anomaly detection method, we successfully improved the anomaly detection performance of both DRAEM (from 67.4% to 76.6% in P-AUPR) and DeSTSeg (from 73.2% to 78.1% in P-AUPR) models. In a word, our contributions can be summarized as:

- We propose a few-shot anomaly-driven generation method guiding a diffusion model to generate realistic and diverse anomalies with a few real-world anomalies. These generated anomalies are consistent with real-world anomalies in semantics.
- We propose a bounding-box-guided anomaly generation process, which not only allows for control over the position and size of the anomaly regions but also provides bounding box supervision for discriminative anomaly detection models.
- Based on bounding box supervision, we propose a simple weakly-supervised anomaly detection method built on two discriminative anomaly models, DRAEM and DeSTSeg. The experiments on MVTec show that we successfully improve their performance both on anomaly classification and segmentation tasks.

2 Related Work

Conditional Diffusion Models. The diffusion model [16] has achieved remarkable success in image generation, leading to a surge of research interest in generating images that meet specific user expectations under given conditions. For instance, [13] focuses on generating images corresponding to class labels. Other methods such as [23, 28, 37] allow users to provide text and generate images that align with the provided text. [8, 40] utilize reference images to generate images with similar style and structure. [7, 21] refer to the given layout conditions, which ensure that the elements in the image have the expected relative positional structure. Additionally, [5, 26] enables users to generate corresponding images based on the user’s sketches. However, these methods are mainly applied to natural images and generate semantically correct images without being able to generate specific objects.

There is a huge gap between industrial anomalies and natural images. In this paper, we aim to generate special anomalies based on a given anomaly concept. In [36, 39], they can generate a specific concept with given reference images, but they focus on generating the entire object. However, the industrial anomalies are often only a small part of the object. In addition, to control the specific area of the generated anomalies, we use the inpainting mode, which has been explored in [3, 25].

Furthermore, the use of diffusion models to generate images has proven to be beneficial for various downstream image recognition tasks. For example, [41] employs generated images as data augmentation, [4] generates images to enhance training ImageNet classification model, and [2] generates skin images for disease classification task. *As far as we know, we are the first to utilize diffusion models to generate images and assist in anomaly detection.*

Industrial Anomaly Detection. Unsupervised anomaly detection models are generally based on embedding or reconstruction to learn normal distribution. In embedding-based methods, [1, 11] use a pre-trained network to extract features from images and estimate their multivariate Gaussian distribution. [35] fits the normalizing flow into a distribution in the form of a product of Gaussian and Dirac distributions. [33] maintains a huge memory bank to save the extracted features. [19, 31] propose feature adaptation for adapting targeted datasets. Reconstruction-based methods [32, 48] encourage a model to learn the masked areas. [27] proposes a robust GAN-inversion to store any images. [46] uses a layer-wised query transformer to reconstruct original features preventing a shortcut issue.

Discriminator-based models often achieve superior performance in anomaly segmentation tasks and they require synthetic anomaly images to train the discriminator. [47] utilizes an external texture dataset to construct anomalies, while [20] randomly crops image patches and pastes them onto other parts to create anomalous images. [24] directly perturbs the feature map to simulate the features of anomalous images. However, these synthetic anomalous images have a significant distribution gap with real-world anomalous ones, which is not conducive to training anomaly detection models in practical applications. *We expect the generated images to be as similar as possible to real anomalies, making the model more robust.*

3 Preliminaries

Image Generation with Diffusion Models. The diffusion model (DM) is a probabilistic model for learning data distribution, which can reconstruct diverse samples from noise. The DM regards the process of addnosing and denosing as a Markov chain of length T , and learns the data distribution through a continuous process of addnosing and denoising. The optimization objective of DM is to predict noise from the noisy image, which can be expressed as:

$$L_{DM} = \mathbb{E}_{x, \epsilon \sim \mathcal{N}(0, 1), t} [\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2], \quad (1)$$

where t uniformly sampled from $[1, \dots, T]$, ϵ is the noise sampled by a Gaussian distribution, x_t is the noisy version during the addnoising process, and $\epsilon_\theta(x_t, t)$ is the noise predicted by the network during the denoising process.

Conditional Diffusion Models. Given a random noise, DM can generate diverse images through iterative denoising. However, the semantics of the generated image are uncontrollable. To solve the problem, the Latent Diffusion Model (LDM) proposes to use a condition y to control the denoising process. LDM first

transforms the images into a latent space, then injects the condition y into the model through a cross-attention module, allowing the model to generate images corresponding to y . The optimization objective of LDM can be simplified as:

$$L_{LDM} = \mathbb{E}_{\varepsilon(x), \epsilon, t, y} [||\epsilon - \epsilon_\theta(\varepsilon(x), t, \tau_\theta(y))||_2^2], \quad (2)$$

$\varepsilon(\cdot)$ is a auto-encoder and is used to transform x into the latent space. The condition y could be a text, a class label, etc., and $\tau_\theta(\cdot)$ is the corresponding encoder.

Intuitively, one might think that training an expert model $\tau_\theta(\cdot)$ to encode the industrial prompt, such as “scratch”, would be sufficient for generating anomaly images. However, this approach becomes impractical due to the scarcity of anomaly images. Similarly, fine-tuning the LDM is not feasible for the same reason. To overcome these challenges and reduce the reliance on anomaly images, we propose a scheme that directly utilizes embeddings as conditions. By doing so, we only need to learn an embedding with a small number of parameters, typically a few hundred. This approach allows us to generate anomaly images effectively while mitigating the limitations imposed by the scarcity of anomaly data.

Discriminative Anomaly Detection Model. A discriminative model typically consists of two modules: reconstruction and discrimination. The reconstruction module is responsible for reconstructing anomalous (normal) images into normal (itself) ones, and then the discrimination module predicts the segmentation map of the anomalous regions based on the difference between the reconstructed output and the original input. To formalize the process, let’s use DRAEM as an example to explain. Assuming I_n denotes a normal image, and the corresponding synthetic anomalous image is I_a . We denote the reconstructed output of I_a or I_n as I_r , then the optimization objective for reconstructive sub-network is

$$L_{rec} = \lambda L_{SSIM}(I, I_r) + l_2(I, I_r), \quad (3)$$

where λ is a weight balancing between SSIM [44] loss and l_2 loss. Then, a normal image I_n (or its synthetic version I_a) and the corresponding reconstruction version I_r are considered as the input of the discriminative sub-network, and the optimization objective of the discriminative sub-network is

$$L_{seg} = L_{Focal}(M, \hat{M}), \quad (4)$$

where \hat{M} is the predicted segmentation map, M is the ground truth mask of I_n (or I_r) and L_{Focal} is the Focal Loss [22].

It can be seen that both reconstruction and discrimination modules require anomalous images I_a . In DRAEM and DeSTSeg, they create anomalies by blending normal images with an external dataset, DTD [9], while CutPaste creates anomalies by randomly pasting image patches. As mentioned earlier, these synthetic anomaly images are semantically inconsistent with real-world images. Therefore, our goal is to generate semantically consistent images to further enhance discriminator-based models.

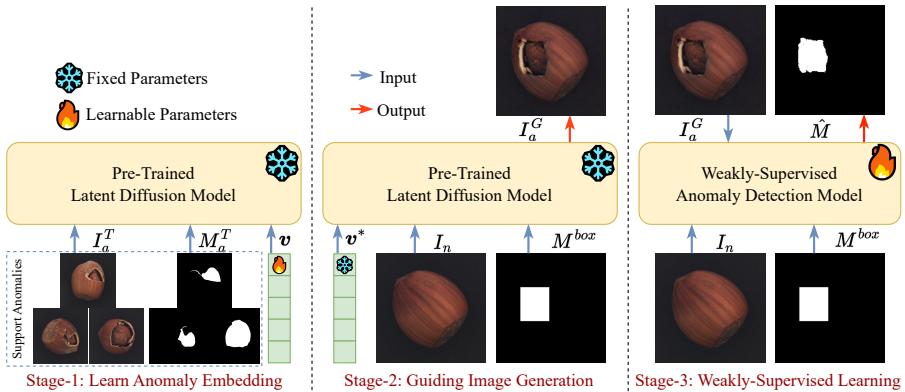


Fig. 2: Pipeline of our work, and it consists of three stages. In the first stage, we learn an embedding vector v with few support anomalies (I_a^T , M_a^T) based on a pre-trained Latent Diffusion Model (LDM) fixing all parameters, where the number of real-world anomalous images I_a^T is only 1 or 3, and M_a^T is the corresponding ground-truth masks. In the second stage, given a normal image I_n and a bounding box mask M^{box} , we use the learned embedding v^* to guide the LDM to generate anomalous image I_a^G . In the third stage, we use the normal image I_n , bounding box mask M^{box} , and generated image I_a^G to train a weakly-supervised anomaly detection model for image-level classification and pixel-level segmentation.

4 Methodology

Our method consists of three stages, as shown in Figure 2. The first stage learns embeddings based on a few support anomalies and their ground-truth segmentation maps. In the second stage, anomalies are generated by leveraging the learned embeddings, given objects (or textures) and bounding boxes as guidance. In the third stage, a weakly supervised anomaly detection model is trained using anomalies and bounding box supervision.

4.1 Stage-1: Learn Anomaly Embedding

When dealing with a limited number of anomalous images, it becomes impractical to optimize a diffusion model with millions of parameters. However, the optimization process becomes much easier when working with an embedding that consists of only a few hundred parameters. Hence, our choice is to focus on learning an embedding that effectively captures the semantic characteristics of real anomalies, rather than relying on fine-tuning a complex model.

Given that predicting noise in LDM involves learning the data distribution, we can leverage the loss associated with noise prediction to gain insights into the distribution of real anomalies. Specifically, we first initialize an embedding v to replace the condition embedding $\tau_\theta(y)$ in LDM, and then optimize it with

Equ 2. It can be denoted as follows:

$$\mathbf{v}^* = \arg \min_{\mathbf{v}} L_{LDM}(I_a^T, t, \mathbf{v}), \quad (5)$$

where I_a^T is a few (*e.g.*, 1 or 3) real-world anomalous images. Similar to the conditioning mechanism in LDM, we insert embedding \mathbf{v} into intermediate layers of the UNet in LDM implementing with cross-attention. Instead of learning the entire network, we initiate the embedding and freeze the parameters of a pre-trained LDM model. This allows us to solely update the embedding during the addnoising and denoising process. By doing so, the learned embedding captures distribution about the provided real-world anomalies, which subsequently guides the image generation in the subsequent stage.

In certain cases, the anomaly region within an image is typically a small fraction of the overall object (*e.g.*, “*a hole in a carpet*”). Training the model on the entire image may result in a learned data distribution that is biased towards the object itself (*e.g.*, “*carpet*”) rather than focusing on the anomaly (*e.g.*, “*hole*”). To address this concern, we incorporate the segmentation mask of the abnormal image to guide the loss function. We denote M_a^T as the segmentation mask of I_a^T , then the modified LDM loss can be described as

$$L'_{LDM} = \mathbb{E}_{\varepsilon(x), \epsilon, t, \mathbf{v}} [||(\epsilon - \epsilon_\theta(\varepsilon(I_a^T), t, \mathbf{v})) \odot M_a^T||_2^2], \quad (6)$$

There are some similar works [15, 18] to this approach, where they also learn the specified object by optimizing an embedding. However, these works focus on learning the entire object, whereas our approach emphasizes capturing the local details of the object through a mask-guided loss. We demonstrate the impact of these two approaches on anomaly generation in the following experiments.

4.2 Stage-2: Guiding Anomaly Generation

Our generation objective is to ensure that the generated images exhibit both semantic and spatial controllability. On the semantic level, we aim to generate images that are consistent with real-world examples, maintaining consistency in terms of objects or textures (*e.g.* “*bottle*”) and the type of anomaly (*e.g.* “*broken*”). On the spatial level, we strive to control the position and size of the anomaly region by providing bounding boxes. By achieving both semantic and spatial controllability, we can generate images that closely align with our desired specifications.

To achieve the above goals, we adopt an inpainting technique inspired by [3]. Specifically, we randomly sample a normal image I_n from the training set as the input image and employ a bounding box mask M^{box} to regulate the location and size of the generated anomaly. The embedding \mathbf{v}^* will be frozen and injected into the image as a condition through the cross-attention module, thus generating the expected anomaly. For the inference image at each step in the denoising process, the area within the box will be retained, and the area outside the box will be replaced by the noisy version of I_n . By doing so, we can control the generated

anomalies to be located in specified areas of the input image while leaving other areas untouched. We can represent this process as:

$$z_t = z_n^t \odot (1 - M^{box}) + z_t' \odot M^{box}, \quad (7)$$

where z_n^t is the addnosing version of $\epsilon(I_n)$ at t step, and z_t' is the denoising version of z_{t+1} . $\epsilon(\cdot)$ transforms I_n into the latent space. After the denoising process, the latent variable z_0 is passed through a decoder to produce anomalous image I_a^G . Since the diffusion model generates images from random noise, they inherently possess a certain degree of diversity. However, to augment this diversity further, we introduce ground boxes with arbitrary positions and sizes.

4.3 Stage-3: Weakly-Supervised Anomaly Detection

Existing discriminative models are typically trained on precise masks. However, it is not suitable for our bounding box supervision since not all pixels within the box are necessarily anomalous. If we directly use the entire box as supervision for the anomaly region, it would mistakenly classify normal pixels as anomalous ones and thus damage the model performance in the anomaly segmentation task. To accommodate the bounding box supervision, we design a weakly supervised loss for anomaly detection.

Let $\hat{p}_{(i,j)}$ represent the predicted normal probability at the location (i,j) . We use a threshold τ to distinguish confident predictions:

$$\delta_{(i,j)} = \begin{cases} 1, & \text{if } \hat{p}_{(i,j)} \geq \tau \\ 0, & \text{otherwise} \end{cases}, \quad (8)$$

where $\hat{p}_{(i,j)} = 1 - \hat{M}_{(i,j)}$. For high-confidence normal pixels within the box region, we set their loss to 0, thereby reducing the confliction of possible normal pixels within the bounding box. For all pixels out of the bounding box, we use the same segmentation loss L_{seg} as Eq. 4. Therefore, the overall weakly-supervised loss of discriminative sub-network is

$$L'_{seg} = M^{box} \odot (1 - \delta) \odot L_{seg} + (1 - M^{box}) \odot L_{seg}, \quad (9)$$

where M^{box} is the given box mask in anomaly generation.

5 Experiment

5.1 Implementation Details

Datasets. MVTec [6] is a widely used industrial anomaly detection dataset that contains 10 objects and 5 textures, each with 1-8 types of anomaly and a few anomalous images, totaling 73 types of anomalies and 1258 anomalous images. For each type of anomaly, we generate 4 anomalous images for each object (or texture) in the training set, obtaining an anomaly dataset of 70,760 images.

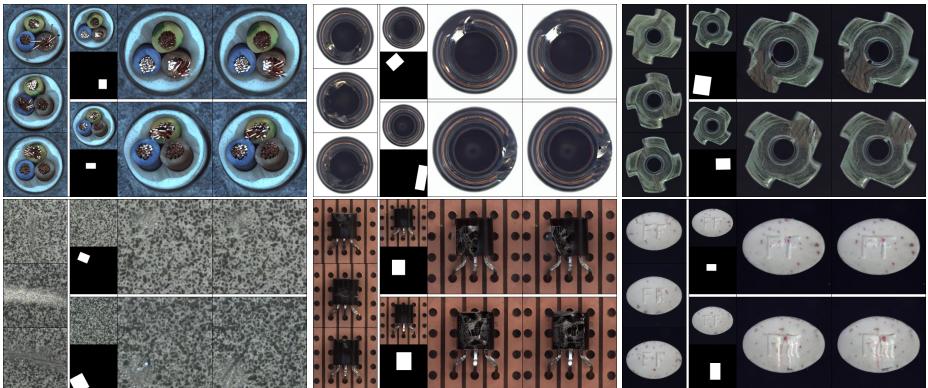


Fig. 3: We show six sets of images, in each set, the first column is the support anomalies (only 3 images), and the second column is the object (or texture) sampled from the training set with a randomly generated bounding box mask, the third and fourth columns are the generated anomalous images.

Learning embeddings and generated images. We use the pre-trained LDM [42] without any parameter fine-tuning. The model uses the text encoder in the CLIP [30] to obtain text embeddings, thus we use the word “defect” through the text encoder to obtain the initialized embedding v (dimension is 768). In the stage of learning v , we randomly select 3 anomalous images from the real anomalies to be the support anomalies. We train 6000 iterations with a learning rate of 0.005. During the stage of generating anomalous images, we randomly create 2 masks for each object (or texture) and generate 2 anomalous images using each mask.

Bounding-Box Generation. In order to enhance the rationality of the generated anomalies, we impose certain constraints on the bounding box. Specifically, we control position to ensure at least 50% IoU between the bounding box and the foreground region obtained with GrabCut [34]. Then, we set hyper-parameters to control the size, *e.g.*, the hyper-parameters of hazelnut-hole are [0.1, 0.5], indicating that the bounding box is between 10% and 50% of the image width or height. Based on these above constraints, we randomly generate bounding boxes to enhance the diversity of generated anomalies.

Anomaly detection task. We use DRAEM and DeSTSeg as baseline models. Apart from the hyper-parameter $\tau = 0.9$, we keep the other hyper-parameters consistent with the original paper to ensure fair comparisons. When sampling a training batch, we randomly sample the original synthetic anomalies and our generated anomalies with a probability of 0.5. For evaluating the performance, we compared the AU-PR and AU-ROC metrics for both anomaly classification and anomaly segmentation tasks. It is worth noting that due to the severe imbalance between normal and anomalous pixels, the AU-PR metric on the segmentation task may better measure the performance of the model [10]. We focus on comparing with the DRAEM and DeSTSeg to validate that our generated anomalous

Table 1: Anomaly classification comparisons (image-level AU-ROC / AU-PR) and anomaly segmentation comparisons (pixel-level AU-ROC / AU-PR) on MVTec. The results of DRAEM and DeSTSeg are reported by running their official code. To ensure a fair comparison, we ran DRAEM, DeSTSeg and our AnoGen under the same environment, while keeping the hyper-parameters consistent with the original paper.

Metrics	CS-FLow	PaDim	PatchCore	RD4AD	DRAEM	AnoGen	DeSTSeg	AnoGen
Image	AU-ROC	97.5	91.2	97.8	98.7	97.1	98.7 (1.6 ↑)	98.3
	AU-PR	97.7	94.2	98.8	97.8	98.5	99.5 (1.0 ↑)	99.4
Pixel	AU-ROC	93.4	96.9	97.5	93.9	96.8	98.1 (1.3 ↑)	98.2
	AU-PR	59.6	48.5	61.7	55.4	67.4	73.2 (5.8 ↑)	76.6
								98.8 (0.5 ↑)
								78.1 (1.5 ↑)

images can benefit the model train better. In addition, we also compared with other unsupervised anomaly detection models: PaDim [11], PatchCore [33], CS-Flow [35] and RD4AD [12].

5.2 Qualitative Analysis for Anomaly Generation

In Figure 3, we present a selection of generated anomalies (Additional generated images are shown in the supplementary materials). Our generated anomalies effectively meet our expectations, despite the presence of only 3 real anomalies in the support set. They exhibit similarity to real-world anomalies and demonstrate diversity. Moreover, by applying box conditions, we gain control over the position and size of the anomaly region, enabling spatial controllability.

5.3 Quantitive Comparisons on Anomaly Detection

We evaluate the effectiveness of our generated images by assessing the model’s performance on anomaly detection tasks. We focus on whether our generated images can directly improve the model’s performance. As shown in Table 1, both DRAEM and DeSTSeg consistently exhibit improvements across all metrics. For example, we observe a 1.6% enhancement in image-level AU-ROC for DRAEM and a 0.5% improvement for DeSTSeg. Similarly, in terms of pixel-level AU-PR, DRAEM and DeSTSeg show improvements of 5.8% and 1.5% respectively. This demonstrates that incorporating real-world anomaly distribution guidance can be beneficial for the model, especially in the anomaly segmentation task. Then, when compared to unsupervised learning models, it becomes evident that discriminator-based models outperform in anomaly segmentation tasks. This finding reveals the guidance of anomalies is of great significance for anomaly segmentation tasks. These conclusions strongly support the crucial importance of providing anomalies that are consistent with real-world anomalies for the performance of anomaly segmentation models, which aligns perfectly with the purpose of our work.

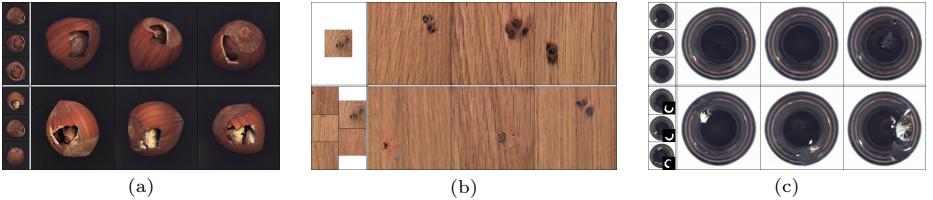


Fig. 4: The ablation study (visualization of the generated images). (a) Comparison of different support anomalies, the number of images is fixed to 3. (b) Comparison of different numbers of support anomalies. (c) Comparison between mask guide loss and non-mask guide loss during embedding learning.

Table 2: The ablation study (model performance on anomaly detection task).

(a) Effects of different support anomalies.		(b) Effects of the number of support anomalies.		(c) Effects of mask-guided learning loss.	
support set		k-shot	1	3	5
Image	AU-ROC	98.7	98.5		
	AU-PR	99.5	99.1		
Pixel	AU-ROC	98.1	97.7		
	AU-PR	73.2	71.4		

		leaning embedding			mask	non-mask
		AU-ROC	AU-PR			
Image	AU-ROC	98.7	97.8			
	AU-PR	99.5	98.7			
Pixel	AU-ROC	98.1	97.6			
	AU-PR	73.2	70.5			

6 Ablation Study

6.1 Anomalous Images Generation

Effects of different support anomalies. We analyze the impact of different support anomalies, and the results are presented in Figure 4a and Table 2a. In Figure 4 (a), both sets of abnormal images are associated with the object of “*crack hazelnut*”. However, the instances in the support set are inconsistent, with one set (the second row) revealing the *white kernel*, causing the generated images to also tend to reveal the *white kernel*. Despite this inconsistency in instances, the semantics (“*crack hazelnut*”) remain consistent and correct. As a result, the impact on model performance is minimal, as shown in Table 2a. This indicates that different support sets can affect the generated instances, but the semantics remain aligned with the intended expectations.

Effects of the number of support anomalies. We also analyzed to assess the impact of different numbers of support anomalies. As depicted in Figure 4b, when only one image is available, the learned distribution tends to be biased towards the specific features of that image. Consequently, the generated images lack diversity and generalization. This is reflected in Table 2b, where the performance on anomaly detection tasks slightly decreases. When using more images (5 images) to learn the embeddings, the diversity and generalization of the generated images significantly improve, as confirmed by the specific metrics in Table 2b for anomaly detection tasks. Considering that the performance using 3 support anomalies is comparable to using 5 support anomalies (0.1% difference

Table 3: Ablation study of τ .

τ	1.0	0.95	0.90	0.80	
Image	AU-ROC	98.5	98.5	98.7	98.2
	AU-PR	99.5	99.4	99.5	99.2
Pixel	AU-ROC	98.0	98.2	98.1	97.1
	AU-PR	68.9	71.1	73.2	65.4

Table 4: Ablation study of anomalies.

Anomalies	CutPaste	DRAEM	Ours	AU-PR
DRAEM_CutPaste	✓			59.8
DRAEM_Original		✓		67.4
DRAEM_Ours			✓	68.3
DRAEM_CutPaste+Ours	✓		✓	69.0
DRAEM_DRAEM+Ours		✓	✓	73.2

on image AU-ROC and 0.2% difference on pixel AU-PR), we have opted for 3 images to reduce the demand for anomaly data.

Mask-Guided embedding learning. We emphasize the importance of incorporating the ground truth of support anomalies when learning embeddings. This is to ensure that the learning target is biased towards the anomaly region rather than the object (or texture). As shown in Figure 4c, without the guidance of the mask, the distribution represented by the embeddings is biased towards the entire object (“bottle”), failing to generate anomaly (“broken”). At the same time, the performance of anomaly detection tasks is also severely affected.

6.2 Anomaly Detection with Generated Anomalies

Confidence threshold τ . In the weakly supervised anomaly detection model, we use a threshold τ to filter out interference from high-confidence normal pixels within the bounding box. As shown in Table 3, τ has a significant impact on segmentation tasks. For example, when the $\tau = 0.9$, the pixel-level AU-PR reaches 73.2%, whereas when $\tau = 1.0$ and $\tau = 0.8$, the performance decreases to 68.9% and 65.4%, respectively. This is because if the threshold is too low, the model will be forced to classify more normal pixels within the box region as anomaly pixels. Conversely, if the threshold is too high, the model will ignore learning anomalous pixels. Based on our experimentation, we tend to prefer setting the threshold to 0.9 or 0.95, as it strikes a balance between capturing anomalous pixels and avoiding misclassification of normal pixels within the bounding box.

Training with different anomalies. We also investigate the impact of the training model with different anomalies. As shown in Table 4, both DRAEM-based synthetic anomalies and our generated anomalies can achieve good performance on pixel AU-PR. The performance of the model trained solely on DRAEM-based anomalies or our anomalies is comparable (67.4% and 68.3%, respectively). Furthermore, when these anomalies are used together, a significant improvement can be observed, reaching 73.2%. This can be attributed to the fact that support anomalies may not fully represent the distribution of the test set, resulting in weaker predictive capabilities for unseen images. On the other hand, DRAEM-based anomalies, which incorporate a large amount of out-of-distribution data, can enhance the model’s ability to predict unseen images. Therefore, training the model jointly on both DRAEM-based anomalies and our anomalies proves to be a better choice.

The number of generated anomaly images. In our study, we generate $N = 4$ for each object (or texture), and Figure 5 illustrates the impact on the model with different values of N . As N increases, the model benefits from having more anomalies. When $N < 3$, it is evident that increasing the number of anomalies is beneficial for the model. However, it is important to consider that generating more anomalies requires additional time and resources. Taking into account the trade-off between model performance and the cost of image generation, we ultimately chose $N = 4$ as the optimal value.

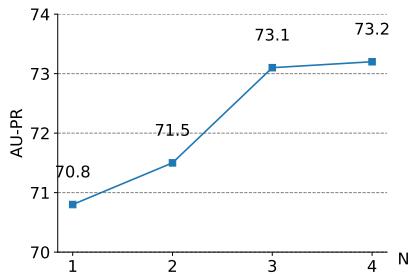


Fig. 5: Ablation study of N .

7 Conclusion

In this work, we have identified and addressed two key challenges in anomaly detection. The first challenge is the scarcity of available real-world abnormal images, which makes it difficult to train anomaly detection models effectively. The second challenge is that synthetic anomalies used in previous methods are unrealistic and have a significant semantic gap compared to real-world anomalies. To address these challenges, we propose an anomaly-driven generation method (AnoGen) to generate a large number of real and diverse abnormal images. Since real abnormal images are scarce, our generation process is driven by few-shot learning, requiring only three real abnormal images. By applying these generated images to DRAEM and DeSTSeg, we achieved promising results on the widely used MVTec anomaly detection dataset. In particular, our generated images have significantly improved the performance on anomaly segmentation task, as evidenced by the AU-PR metrics. This demonstrates the effectiveness of our approach in addressing the above challenges in anomaly detection and highlights the potential of our method for enhancing anomaly detection in real-world scenarios. Overall, our approach uses a small number of real anomaly images to significantly improve model performance, and since a small number of real anomaly images are present in many real-world scenarios, this is of great significance for many practical applications of industrial anomaly detection.

Limitation. The generated anomaly images enjoy free annotations but in bounding boxes form. Therefore, pixel-level model training needs the efforts of weakly supervised methods, which inevitably introduces additional hyperparameters. In our future work, we will explore how to obtain more precise annotations when generating anomaly images so that they can be more easily and widely used in anomaly detection.

References

1. Ahuja, N.A., Ndiour, I., Kalyanpur, T., Tickoo, O.: Probabilistic modeling of deep features for out-of-distribution and adversarial detection. In: NeurIPS (2019) **2**, **5**
2. Akrout, M., Gyepesi, B., Holló, P., Poór, A., Kincső, B., Solis, S., Cirone, K., Kawahara, J., Slade, D., Abid, L., et al.: Diffusion-based data augmentation for skin disease classification: Impact across original medical datasets to fully synthetic images. In: MICCAI (2023) **5**
3. Avrahami, O., Fried, O., Lischinski, D.: Blended latent diffusion. In: CVPR (2022) **4**, **8**
4. Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., Fleet, D.J.: Synthetic data from diffusion models improves imagenet classification. In: TMLR (2023) **5**
5. Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Kreis, K., Aittala, M., Aila, T., Laine, S., Catanzaro, B., et al.: ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. arXiv:2211.01324 (2022) **4**
6. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: MV Tec AD: A comprehensive real-world dataset for unsupervised anomaly detection. In: CVPR (2019) **4**, **9**
7. Chai, S., Zhuang, L., Yan, F.: Layoutdm: Transformer-based diffusion model for layout generation. In: CVPR (2023) **4**
8. Choi, J., Kim, S., Jeong, Y., Gwon, Y., Yoon, S.: Ilvr: Conditioning method for denoising diffusion probabilistic models. ICCV (2021) **4**
9. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: CVPR (2014) **6**
10. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: ICML (2006) **10**
11. Defard, T., Setkov, A., Loesch, A., Audigier, R.: Padim: a patch distribution modeling framework for anomaly detection and localization. In: ICPR (2021) **2**, **5**, **11**
12. Deng, H., Li, X.: Anomaly detection via reverse distillation from one-class embedding. In: CVPR (2022) **2**, **11**
13. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. In: NeurIPS (2021) **4**
14. Duan, Y., Hong, Y., Niu, L., Zhang, L.: Few-shot defect image generation via defect-aware feature manipulation. In: AAAI (2023) **3**
15. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. In: ICLR (2023) **8**
16. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020) **4**
17. Hu, T., Zhang, J., Yi, R., Du, Y., Chen, X., Liu, L., Wang, Y., Wang, C.: Anomalydiffusion: Few-shot anomaly image generation with diffusion model. In: AAAI (2024) **3**
18. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. In: CVPR (2023) **8**
19. Lee, S., Lee, S., Song, B.C.: Cfα: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. IEEE Access **10** (2022) **5**
20. Li, C.L., Sohn, K., Yoon, J., Pfister, T.: Cutpaste: Self-supervised learning for anomaly detection and localization. In: CVPR (2021) **2**, **5**

21. Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., Li, C., Lee, Y.J.: Gligen: Open-set grounded text-to-image generation. In: CVPR (2023) **4**
22. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017) **6**
23. Liu, X., Park, D.H., Azadi, S., Zhang, G., Chopikyan, A., Hu, Y., Shi, H., Rohrbach, A., Darrell, T.: More control for free! image synthesis with semantic diffusion guidance. In: WACV (2023) **4**
24. Liu, Z., Zhou, Y., Xu, Y., Wang, Z.: Simplenet: A simple network for image anomaly detection and localization. In: CVPR (2023) **2, 5**
25. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: CVPR (2022) **4**
26. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. In: ECCV (2021) **4**
27. Mou, S., Gu, X., Cao, M., Bai, H., Huang, P., Shan, J., Shi, J.: Rgi: robust gan-inversion for mask-free image inpainting and unsupervised pixel-wise anomaly detection. In: ICLR (2022) **5**
28. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In: ICML (2021) **4**
29. Pan, Z., Yu, W., Yi, X., Khan, A., Yuan, F., Zheng, Y.: Recent progress on generative adversarial networks (gans): A survey. IEEE access **7** (2019) **3**
30. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021) **10**
31. Reiss, T., Cohen, N., Bergman, L., Hoshen, Y.: Panda: Adapting pretrained features for anomaly detection and segmentation. In: CVPR (2021) **5**
32. Ristea, N.C., Madan, N., Ionescu, R.T., Nasrollahi, K., Khan, F.S., Moeslund, T.B., Shah, M.: Self-supervised predictive convolutional attentive block for anomaly detection. In: CVPR (2022) **5**
33. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. In: CVPR (2022) **2, 5, 11**
34. Rother, C., Kolmogorov, V., Blake, A.: “GrabCut” interactive foreground extraction using iterated graph cuts. TOG **23**(3) (2004) **10**
35. Rudolph, M., Wehrbein, T., Rosenhahn, B., Wandt, B.: Fully convolutional cross-scale-flows for image-based defect detection. In: WACV (2022) **5, 11**
36. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: CVPR (2023) **4**
37. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. In: NeurIPS (2022) **4**
38. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: IPMI (2017) **2**
39. Shi, J., Xiong, W., Lin, Z., Jung, H.J.: Instantbooth: Personalized text-to-image generation without test-time finetuning. In: CVPR (2024) **4**

40. Singh, V., Jandial, S., Chopra, A., Ramesh, S., Krishnamurthy, B., Balasubramanian, V.N.: On conditioning the input noise for controlled image generation with diffusion models. In: CVPRW (2022) **4**
41. Trabucco, B., Doherty, K., Gurinas, M., Salakhutdinov, R.: Effective data augmentation with diffusion models. arXiv preprint arXiv:2302.07944 (2023) **5**
42. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: CVPR (2018) **10**
43. Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M.: Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys **53**(3) (2020) **3**
44. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP **13**(4) (2004) **6**
45. Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., Yang, M.H.: Diffusion models: A comprehensive survey of methods and applications. ACM Computing Surveys **56**(4) (2023) **3**
46. You, Z., Cui, L., Shen, Y., Yang, K., Lu, X., Zheng, Y., Le, X.: A unified model for multi-class anomaly detection. In: NeurIPS (2022) **5**
47. Zavrtanik, V., Kristan, M., Skočaj, D.: Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In: ICCV (2021) **2, 3, 5**
48. Zavrtanik, V., Kristan, M., Skočaj, D.: Reconstruction by inpainting for visual anomaly detection. Pattern Recognition **112**, 107706 (2021) **2, 5**
49. Zhang, X., Li, S., Li, X., Huang, P., Shan, J., Chen, T.: Destseg: Segmentation guided denoising student-teacher for anomaly detection. In: CVPR (2023) **3**
50. Zhou, Z.H.: A brief introduction to weakly supervised learning. National Science Review **5**(1) (2018) **3**