AnomalyPainter: Vision-Language-Diffusion Synergy for Realistic and Diverse Unseen Industrial Anomaly Synthesis

Zhangyu Lai*[1], Yilin Lu*[1], Xinyang Li[1], Jianghang Lin[1], Yansong Qu[1], Ming Li,[2] Liujuan Cao[1†]

[1]Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, Xiamen University, 361005, P.R. China.
[2] Shandong Inspur Database Technology Co., Ltd
{laizhangyu, yilinlu}@stu.xmu.edu.cn, caoliujuan@xmu.edu.cn

Abstract

Visual anomaly detection is limited by the lack of sufficient anomaly data. While existing anomaly synthesis methods have made remarkable progress, achieving both realism and diversity in synthesis remains a major obstacle. To address this, we propose AnomalyPainter, a novel framework that breaks the diversity-realism trade-off dilemma through synergizing Vision Language Large Model (VLLM), Latent Diffusion Model (LDM), and our newly introduced texture library Tex-9K. Tex-9K is a professional texture library containing 75 categories and 8,792 texture assets crafted for diverse anomaly synthesis. Leveraging VLLM's general knowledge, reasonable anomaly text descriptions are generated for each industrial object and matched with relevant diverse textures from Tex-9K. These textures then guide the LDM via ControlNet to paint on normal images. Furthermore, we introduce Texture-Aware Latent Init to stabilize the natural-image-trained ControlNet for industrial images. Extensive experiments show that AnomalyPainter outperforms existing methods in realism, diversity, and generalization, achieving superior downstream performance.

Introduction

Anomaly detection plays a crucial role in practical applications such as industrial quality control (Liu et al. 2024) and medical anomaly detection (Huang et al. 2024). However, real-world anomaly samples are often exceedingly rare, presenting significant challenges for anomaly detection tasks, including image-level classification and pixel-level segmentation. While numerous works (Sun et al. 2025; Hu et al. 2024; Duan et al. 2023; Zhang, Xu, and Zhou 2024) have been proposed to synthesize anomaly samples, these synthetic anomalies often lack realism or require extensive training data, limiting their real-world applicability.

Some methods such as DRAEM (Zavrtanik, Kristan, and Skočaj 2021), CutPaste (Li et al. 2021), and Realnet (Zhang, Xu, and Zhou 2024), synthesize diverse unseen anomalies by cropping and pasting patches from existing anomalies or texture datasets onto normal samples. However, this random operation often results in unreasonable anomaly content and abrupt transitions at anomaly boundaries, significantly reducing the realism. The distribution of these synthesized anomaly samples, abstractly represented as red scatter points, is similar to that shown in Figure 1 (a). These red scatter points tend to fall within the unrealistic anomaly distribution, distant from the feature space of normal samples, which are abstractly represented as blue scatter points.

Another methods like DFMGAN (Duan et al. 2023), AnoGen (Gui et al. 2024), and AnoDiff (Hu et al. 2024) use generative models to learn anomaly patterns for generation. However, the samples synthesized by these methods require sufficient and representative normal or abnormal samples for training. As a result, they tend to overfit the anomaly samples in the training data, failing to capture the diverse unseen anomalies that may appear in real-world objects. Similar to Figure 1 (b), the red scatter points representing these synthesized anomaly samples with similar features, tend to cluster in certain regions in the realistic anomaly distribution.

In short, existing methods face the diversity-realism tradeoff. To address the dilemma, we propose AnomalyPainter, a novel framework that synergizes the VLLM and LDM with our proposed texture library Tex-9K to simulate the formation of real anomalies via texture variations without training.

Specifically, we construct Tex-9K, a texture library with appropriate texture density, designed for diverse anomaly synthesis. It contains 75 categories and 8,792 professional texture image assets. Leveraging the general knowledge of VLLM, for each industrial object, it generates reasonable and diverse anomaly text descriptions. These descriptions retrieve the most relevant textures from Tex-9K, which serve as anomaly pattern conditions to guide the LDM via ControlNet's (Zhang, Rao, and Agrawala 2023) edge-mask control for inpainting normal images. Since ControlNet is trained on natural images, it performs unstably on industrial images. Texture-Aware Latent Init is then introduced to stabilize ControlNet's handling, ensuring that normal and texture images blend clearly in the latent space as the initial denoising point, enabling the LDM to achieve precise denoising performance. In short, our novel AnomalyPainter paints reasonable content (via VLLM) with diverse anomaly patterns (via Tex-9K) on normal images while ensuring soft transitions (via LDM) at anomaly boundaries, ultimately synthesizing unseen diverse and realistic anomaly samples that exhibit an ideal distribution, as shown in Figure 1(c).

*These authors contributed equally.
†Corresponding Author

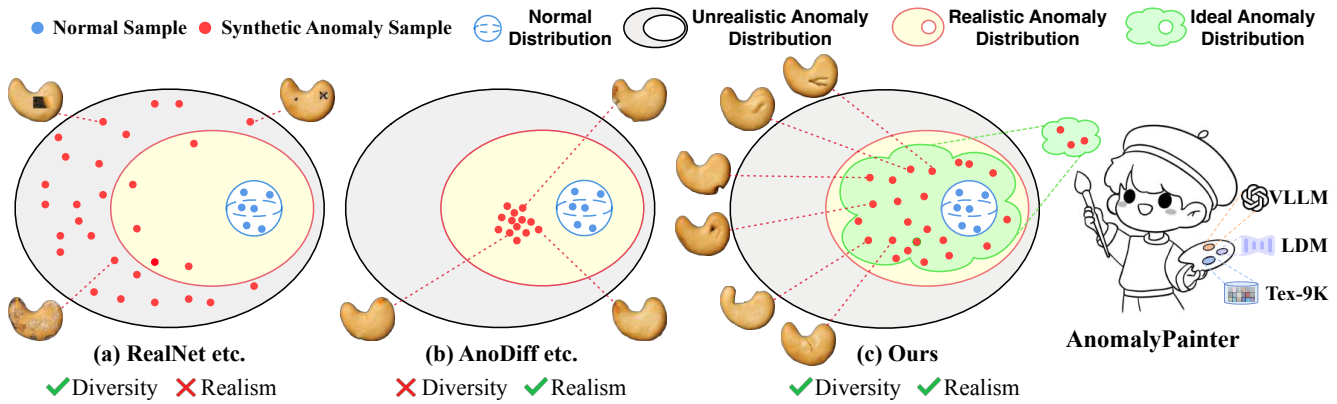

Figure 1: **Conceptual illustration.** The blue hypersphere represents the normal distribution; realistic anomalies should be close, while unrealistic ones should be farther. Anomaly samples synthesized by different methods show different distributions.

Our contributions are threefold:

- We propose AnomalyPainter, a novel synthesis framework for realistic and diverse unseen anomalies. Users are able to supply arbitrary normal images of objects along with anomaly descriptions to synthesize realistic and diverse anomaly samples without any training.
- We propose Tex-9K, a professional texture library containing 75 categories and 8,792 assets, designed for broadened texture diversity. Texture-Aware Latent Init is proposed to stabilize ControlNet's edge-mask control, effectively translating the diverse texture assets provided by Tex-9K into realistic and diverse anomaly samples.
- Extensive experiments show our synthesized anomaly samples outperform current SOTA synthesis methods and effectively boost downstream anomaly detection.

Related Work

Anomaly Synthesis

Anomaly synthesis has become an essential technique to support anomaly detection, especially when real anomaly samples are scarce. some methods (Li et al. 2021; Zavrtanik, Kristan, and Skočaj 2021) crop and paste random patches onto normal samples. RealNet (Zhang, Xu, and Zhou 2024) further employs a generative model to apply noise to normal images and then denoise them to obtain an anomaly dataset, but ultimately pastes part of the anomaly back using a random mask. While these methods can produce diverse unseen anomaly samples, the synthesized samples often exhibit unrealistic. Another methods (Hu et al. 2024; Duan et al. 2023) use generative models to learn anomaly patterns and generate anomaly samples. However, as they rely on substantial amounts of normal and abnormal data to model dataset-specific distributions, their application becomes impractical in data-limited scenarios. Moreover, these approaches can only generate samples resembling those in the training set—i.e., seen anomalies—while failing to synthesize unseen ones. Recent concurrent work, AnomalyAny (Sun et al. 2025), which is guided solely by text descriptions, lacks the precision needed for complex anomalies due to overly broad language space. In contrast, AnomalyPainter constrains the language solution space to the real-world image space, enhancing the fidelity of unseen anomaly synthesis.

Anomaly Detection

Supervised detection has made great progress (Lin et al. 2025, 2024; Yue et al. 2024; Ye et al. 2025; Guo et al. 2025). While the scarcity of anomaly samples has made unsupervised detection the dominant paradigm, where the objective is to model the distribution of normal data and detect anomalies as outliers (Liu et al. 2023). These methods, however, rely on sufficient normal samples to capture the underlying distribution. Given the scarcity of representative normal samples across diverse product variations, few-shot detection has drawn growing interest (Lu et al. 2023; You et al. 2022). Current few-shot methods leverage external knowledge from CLIP to compute similarities between data samples and the normal or abnormal text prompts (Gu et al. 2024; Jeong et al. 2023a; Li et al. 2024b). Yet, relying solely on normal samples for training remains problematic, as it provides no direct understanding of anomaly distributions, highlighting the need for realistic anomaly samples.

Latent Diffusion Models

Recent advances in LDMs, such as Stable Diffusion (Rombach et al. 2022), have greatly improved image generation and boost many downstream applications (Qu et al. 2025a; Li et al. 2024a; Qu et al. 2025b; Li et al. 2025; Dai et al. 2025; Qu et al. 2024). However, pure text-based control struggles to capture complex scene requirements, leading to the development of various diffusion model plugins (Hu et al. 2022; Ruiz et al. 2023) for more precise control. Among these, ControlNet (Zhang, Rao, and Agrawala 2023) excels in structural control by integrating additional signals like edge maps. While many pre-trained ControlNet models are available, they are typically fine-tuned on natural images, leading to instability when applied to industrial objects. Training-free image composition methods (Lu, Liu, and Kong 2023; Liu, Huang, and Xu 2024) improve denoising by guiding attention or blending latents for cross-domain

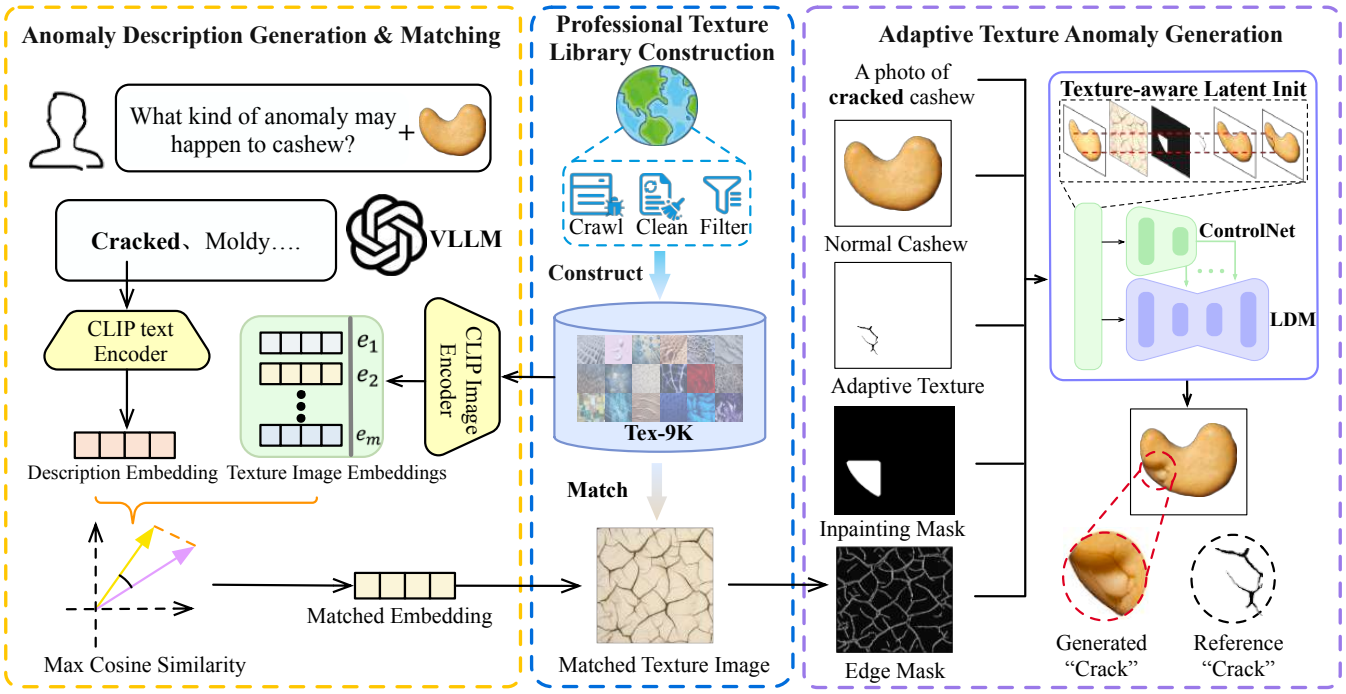

Figure 2: **Overview.** Our framework synthesizes diverse and realistic anomaly samples in three steps: **Middle: Professional Texture Library Construction** constructs Tex-9K, a texture library with 8,792 assets, designed to provide diverse textures crafted for anomaly synthesis. **Left: Anomaly Description Generation and Matching** utilizes VLLM to generate reasonable anomaly descriptions for each industrial object and matches them with relevant textures from Tex-9K using cosine similarity. **Right: Adaptive Texture Anomaly Generation** utilizes Texture-Aware Latent Init to stabilize ControlNet's edge-mask control for LDM's high-realism inpainting, ensuring the seamless integration of textures into normal industrial object images.

synthesis. However, these methods insert whole objects into images, struggling to blend textures with industrial objects.

Method

Empirically, we consider that the formation of real industrial anomaly samples is usually constrained by the physical properties of objects, which can be understood using the general knowledge of VLLMs. Under various potential random circumstances, texture variations related to the object may manifest in the image. To effectively simulate this and achieve diverse and realistic anomaly synthesis, we propose AnomalyPainter, which is implemented in three key steps: Professional Texture Library Construction, Anomaly Description Generation and Matching, and Adaptive Texture Anomaly Generation. The overview is shown in Figure 2.

Preliminaries

Latent Diffusion Models consist of two key components: an auto-encoder (Kingma and Welling 2014) and a latent denoising network. The autoencoder establishes a bi-directional mapping from the space of the original data to a low-resolution latent space: $z = \mathcal{E}(x), x = \mathcal{D}(z)$, where $\mathcal{E}$ and $\mathcal{D}$ are the encoder and decoder respectively. The latent denoising network ϵ_θ is trained to denoise noisy latent given a specific timestep t and textual prompt embedding p. The diffusion process adopts the standard formulation DDIM (Song, Meng, and Ermon 2020), which comprises a forward add-noise diffusion and a backward denoising process. During noise addition, the noisy latent representation at a specified timestep t is obtained as: $z_t = \sqrt{\bar{\alpha}_t}\mathcal{E}(x) + \sqrt{1-\bar{\alpha}_t}\epsilon$, $\bar{\alpha}_t$ is a monotonically decreasing noise schedule and $\epsilon \sim \mathcal{N}(0,1)$ is random noise. By continuously denoising the random noise z_T with textual prompt embedding p through predicting noise $\epsilon_\theta(z_t, t, p)$, we can derive a fully denoised latent z_0. Then, the final clean latent z_0 is passed through the latent decoder $\mathcal{D}$ to generate the high-resolution image $x_0 = \mathcal{D}(z_0)$.

Professional Texture Library Construction

The motivation for constructing Tex-9K stems from the potential mismatch between our visual intuition for text descriptions and the understanding embedded in CLIP, which is pre-trained on large-scale image-text data from the web. For instance, when searching for images of the description 'cracked' online, not all results correspond to the clear crack textures required for fine-grained anomaly synthesis. Similarly, existing texture libraries, such as DTD (Cimpoi et al. 2014), often contain overly complex textures that are unsuitable for this purpose. To smoothly align text descriptions with suitable visual concepts for anomaly synthesis and provide more diverse textures, we expanded existing texture datasets by collecting additional texture images on the In-

ternet and defining 75 commonly used texture categories.

The construction of our texture library can be divided into three parts: (a) Crawling: We use a web crawler to collect corresponding images from the Internet legally for each texture category. Data from existing texture datasets are also incorporated into the corresponding categories in our library. (b) Cleaning: The Canny operator (Canny 1986) is applied to each image to extract the edge mask, and images with excessively dense or sparse textures edge are automatically discarded. (c) Filtering: The remaining data is manually screened to retain images with clear textures and remove potentially harmful or inappropriate content. Ultimately, Tex-9K which retains 8,792 images, serves as the texture library to provide texture assets crafted for anomaly synthesis. See *Appendix.A* for more details.

Anomaly Description Generation & Matching

We use VLLM's general knowledge to generate reasonable anomaly descriptions, which is detailed in Alg 1.

Specifically, let $O = \{o_1, o_2, \dots\}$ denote the set of industrial object categories. We first preprocess the entire Tex-9K by encoding all texture images $\mathcal{X} = \{x_{\text{tex},1}, x_{tex_2}, \dots, x_{tex_m}\}$ through the CLIP image encoder. This generates static Texture Images Embeddings $e_{\mathcal{X}} = \{e_1, e_2, \dots, e_m\}$, which are cached for persistent reuse. For each possible industrial object $o_i \in O$, we select a normal image $x_N^{o_i}$ corresponding to the object o_i, then pose a carefully designed question Q_{o_i} for o_i with the prompt template. We will detail the template in *Appendix.B*. By querying VLLMs with Q_{o_i} and $x_N^{o_i}$, we obtain the anomaly description answer $A_{o_i} = \{d_1, d_2, \dots, d_k\}$, where each $d \in A_{o_i}$ is a description. The CLIP text encoder then encodes d into description embedding e_d. After computing cosine similarity between e_d and each $e_i \in e_{\mathcal{X}}$, the max one is taken as matched embedding e_{match}, $e_{\text{match}} \in e_{\mathcal{X}}$. The corresponding $x_{\text{match}} \in \mathcal{X}$ is then used as the matched texture image.

Taking the VisA dataset as an example, let $O = \{\text{candle}, \text{cashew}, \dots\}$ denote a collection of 12 object types. If $o_i = \text{cashew}$, we select a normal cashew image as $x_N^{o_i}$. A designed question Q_{o_i} can be regarded as "What kind of anomaly may happen to cashew?". By leveraging the powerful general knowledge, VLLMs (e.g. GPT-4V) may respond $A_{o_i} = \{\text{cracked, moldy...}\}$. Take $d = \text{cracked}$ as an example, the embedding of text "cracked" will be used to match the most similar texture image in Tex-9K.

Adaptive Texture Anomaly Generation

After obtaining the matched texture image, we propose Adaptive Texture Anomaly Generation, which seamlessly integrates the matched texture into a normal industrial image, creating a realistic anomaly sample, detailed in Alg 2.

Specifically, we first generate an adaptive texture image $x_\wp$ based on the matched texture image and the normal industrial object image x_N (dropping the superscript for simplicity), along with an inpainting mask M_{in} that indicates the region where the anomaly content is to be generated. We discuss how to get M_{in} and $x_\wp$ later in **Mask Generation**.

To realistically express texture variations in the normal image x_N in region M_{in}, we pioneer the application of ControlNet, which offers powerful edge-mask control ability. Since ControlNet is trained on natural images, it often becomes unstable on industrial data. To address this, we introduce Texture-Aware Latent Initialization (TALI), enhancing its reliability in industrial anomaly synthesis (Figure 3).

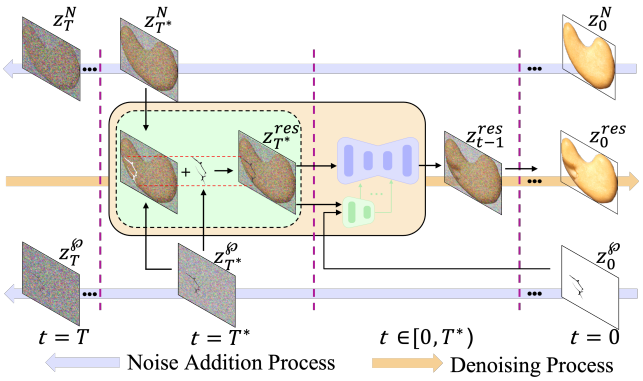

Figure 3: Texture-Aware Latent Initialization blends normal image latent z^N and adaptive texture latent $z^\wp$ at a later timestep T^* to get $z_{T^*}^{\text{res}}$ as the start point for better result.

Texture-aware Latent Init. The core idea of TALI is to blend the normal industrial image x_N with the adaptive texture image $x_\wp$ in the latent space as the starting point for denoising. In fact, this part can be regarded as an image composition task, which aims to harmonize x_N and $x_\wp$ to generate the composite anomaly image x_{res}. Unlike traditional image composition methods, such as TF-ICON (Lu, Liu, and Kong 2023), which typically invert x_N and $x_\wp$ into their corresponding noisy latent representations z_T^N and $z_T^\wp$ at a predefined timestep T, we choose to begin at a later timestep T^*. This choice helps avoid excessive stylistic disconnection between the generated anomalous part of x_{res} and the normal industrial image x_N, while ensuring anomalous part of x_{res} remains geometrically stable and consistent with $x_\wp$. Mathematically, we choose $0 < T^* < T$ and use $z_{T^*}^{\text{res}}$ as the starting point for denoising:

$$\mathbf{z}_{T^*}^{res} = \mathbf{z}_{T^*}^N \odot (\mathbf{1} - \mathbf{M}_\wp^z) + \mathbf{z}_{T^*}^\wp \odot \mathbf{M}_\wp^z, \tag{1}$$

where $\mathbf{M}_\wp^z$ is the segmentation mask of $x_\wp$ in latent space.

After initialization, the pre-trained denoising network ϵ_θ with ControlNet will preserve the layout structure of $z_{T^*}^{res}$ during $t \in [0, T^*]$, while gradually harmonizing the anomalous texture with the normal image in the inpainting mask M_{in} region. The denoising process is achieved through the removal of the estimated noise $\epsilon_\theta(z_t^{res}, t, M_{in}, x_\wp, P)$, where M_{in} defines the region of the original image that can be inpainted, $x_\wp$ serves as the ControlNet condition, P is an object-specific text prompt embedding. The text prompt is formulated as: "A photo of [d] [o_i]", where, for example, d could be "cracked" generated by VLLM, and o_i refers to an industrial category such as "cashew". When the denoising reaches $t = 0$, the decoder $\mathcal{D}$ is used to decode and obtain $x^{res} = \mathcal{D}(z_0^{res})$. We detail T^* selection in **Ablation Study**.

Mask Generation. The previous anomaly synthesis methods rely on arbitrary inpainting masks, such as Perlin

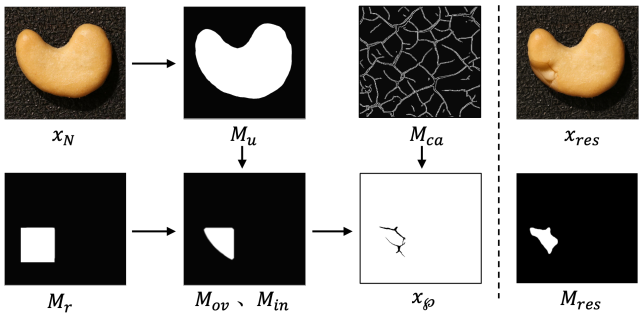

Figure 4: **Left**: An example of generated inpainting mask M_{in} and corresponding adaptive texture $x_\wp$. **Right**: An example of generated anomaly result and the refined mask.

noise (Zhang, Xu, and Zhou 2024) or random masks (Hu et al. 2024), which often result in anomaly being placed on the background of industrial objects, reducing realism. To address this, we propose the following strategy for generating a more effective inpainting mask: (a) Randomly generate a rectangular mask M_r. (b) Compute the intersection $M_{\text{ov}} = M_r \odot M_u$ between M_r and foreground M_u (object segmentation (Qin et al. 2020) in normal image). Proceed only if the overlap $Area(M_{\text{r}} \odot M_u) > \text{thresh}_1$. (c) Compute the intersection between M_{ov} and M_{ca} (Canny edges (Canny 1986) of matched texture). If the overlap $Area(M_{ov} \odot M_{ca}) > \text{thresh}_2$, we accept M_{ov} as the inpainting mask M_{in}, and use the corresponding texture region as the anomaly texture $x_\wp$. The final inpainting mask is obtained as $M_{\text{in}} = M_r \odot M_u$, and adaptive texture $x_\wp$ is generated by applying morphological operations to $M_{\text{in}} \odot M_{ca}$ to ensure connectivity, detailed in *Appendix.C*. This strategy generates a random and diverse mask while ensuring valid texture. We present an example of a generated mask and its corresponding adaptive texture image in Figure 4 (left).

Mask Refine. Since the inpainting mask M_{in} covers a large area, the LDMs' generation process is relatively unconstrained, often resulting in anomalies appearing only in certain regions. A coarse refinement is then applied by computing the SSIM (Wang et al. 2004) difference map between the original image x_N and the generated anomaly image x_{res} within the inpainting region M_{in}, yielding a per-pixel similarity score, i.e. $\text{ScoreMap}_{\text{ssim}} = \text{SSIM}(x_N \odot M_{\text{in}}, x_{\text{res}} \odot M_{\text{in}})$. We then compute the mean similarity $\text{thresh}_{\text{mean}} = \text{Average}(\text{ScoreMap}_{\text{ssim}})$ within the region and retain pixels with similarity scores exceeding this threshold, i.e. $M_{\text{res}} = \text{ScoreMap}_{\text{ssim}} > \text{thresh}_{\text{mean}}$. Figure 4 (right) shows an example of generated anomaly with refined mask.

Algorithm 1: Anomaly Description Generation & Matching

Data: $O = \{o_1, o_2, \dots\}$, $\mathcal{X} = \{x_{\text{tex}_1}, x_{\text{tex}_2}, \dots, x_{\text{tex}_m}\}$
Cache: $e_{\mathcal{X}} = \{e_1, e_2, \dots, e_m\} \leftarrow \text{CLIP}_{\text{image}}(\mathcal{X})$
Input: $o_i \in O$, a normal image $x_N^{o_i}$
Output: Matched descriptions and texture images for o_i

Pose question $Q_{o_i} = \text{Template}(o_i)$
Obtain anomaly descriptions $A_{o_i} = \text{VLLM}(Q_{o_i}, x_N^{o_i})$
for description d in $A_{o_i} = \{d_1, d_2, \dots, d_k\}$ **do**
    Obtain description embedding $e_d \leftarrow \text{CLIP}_{\text{text}}(d)$
    Obtain matched embedding

$$e_{\text{match}} \leftarrow \arg\max_{e_i \in e_{\mathcal{X}}} \cos\langle e_d, e_i \rangle$$

    Select the corresponding texture image $x_{\text{match}} \in \mathcal{X}$
end for
return All descriptions and matched texture images

Algorithm 2: Adaptive Texture Anomaly Generation

Input: x_N, x_{match}, T^*, P.
Cache: $M_u \leftarrow \text{Seg}(x_N)$,$M_{\text{ca}} \leftarrow \text{Canny}(x_{\text{match}})$
Output: Generated anomaly image x_{res} and mask M_{res}.

$M_{\text{in}}, x_\wp = \textbf{Mask Generation}(M_u, M_{\text{ca}})$
TALI:
$z_N, z_\wp = \mathcal{E}(x_N), \mathcal{E}(x_\wp)$
Sample Noise $\epsilon \sim \mathcal{N}(0, 1)$
Add Noise $z_{T^*}^N \leftarrow \sqrt{\bar{\alpha}_{T^*}} z_N + \sqrt{1 - \bar{\alpha}_{T^*}}\epsilon$
Add Noise $z_{T^*}^\wp \leftarrow \sqrt{\bar{\alpha}_{T^*}} z_\wp + \sqrt{1 - \bar{\alpha}_{T^*}}\epsilon$
Blend latents $z_{T^*}^{\text{res}} \leftarrow z_{T^*}^N \odot (1 - M_\wp^z) + z_{T^*}^\wp \odot M_\wp^z$
Denoise with ControlNet:
for $t \leftarrow T^*$ **downto** 1 **do**
    $z_{t-1}^{res} \leftarrow \text{DDIM}\big(z_t^{\text{res}}, \epsilon_\theta(z_t^{\text{res}}, t, M_{\text{in}}, x_\wp, P)\big)$.
end for
$x_{\text{res}} \leftarrow \mathcal{D}(z_0^{\text{res}})$.
$M_{\text{res}} = \textbf{Mask Refine}(x_{\text{N}}, x_{\text{res}}, M_{\text{in}})$
return $x_{\text{res}}, M_{\text{res}}$.

Experiments

Experimental Settings

Dataset. We conduct extensive experiments on the MVTec AD (Bergmann et al. 2019) and VisA (Zou et al. 2022) datasets. MVTec AD consists of 15 categories with 5,354 images with pixel-level annotations. VisA contains 12 categories with 10,821 images with annotations. Since VisA contains categories with complex structures and multi-object images, it poses greater challenges than MVTec. Therefore, we tend to show more comparisons in VisA.

Evaluation Metrics. We use Inception Score (IS) and intra-cluster pairwise LPIPS distance (IL) to assess quality and diversity. We validate the effectiveness of our synthesized samples by training anomaly detection frameworks (Gu et al. 2024) following AnomalyAny. We employ five metrics to evaluate the detection performance: image-level and pixel-level Area Under the Receiver Operating Characteristic, denoted as AUC-I and AUC-P, respectively, the image-level and pixel-level max-F1 scores, denoted as F1-I and PF1, and Per-Region-Overlap, denoted as PRO.

Implementation Details. AnomalyPainter is implemented using the HuggingFace Diffusers library (von Platen et al. 2022), built on the Stable Diffusion XL 1.0 (SDXL) (Podell et al. 2024) model and ControlNet-Canny, with a CLIP model utilizing a ViT-B/32 backbone. We adopt GPT-4V as our VLLM. Following (Lu, Liu, and Kong 2023), we use a 20-step DDIM sampler, while starting denoising at $T^* = 16$.

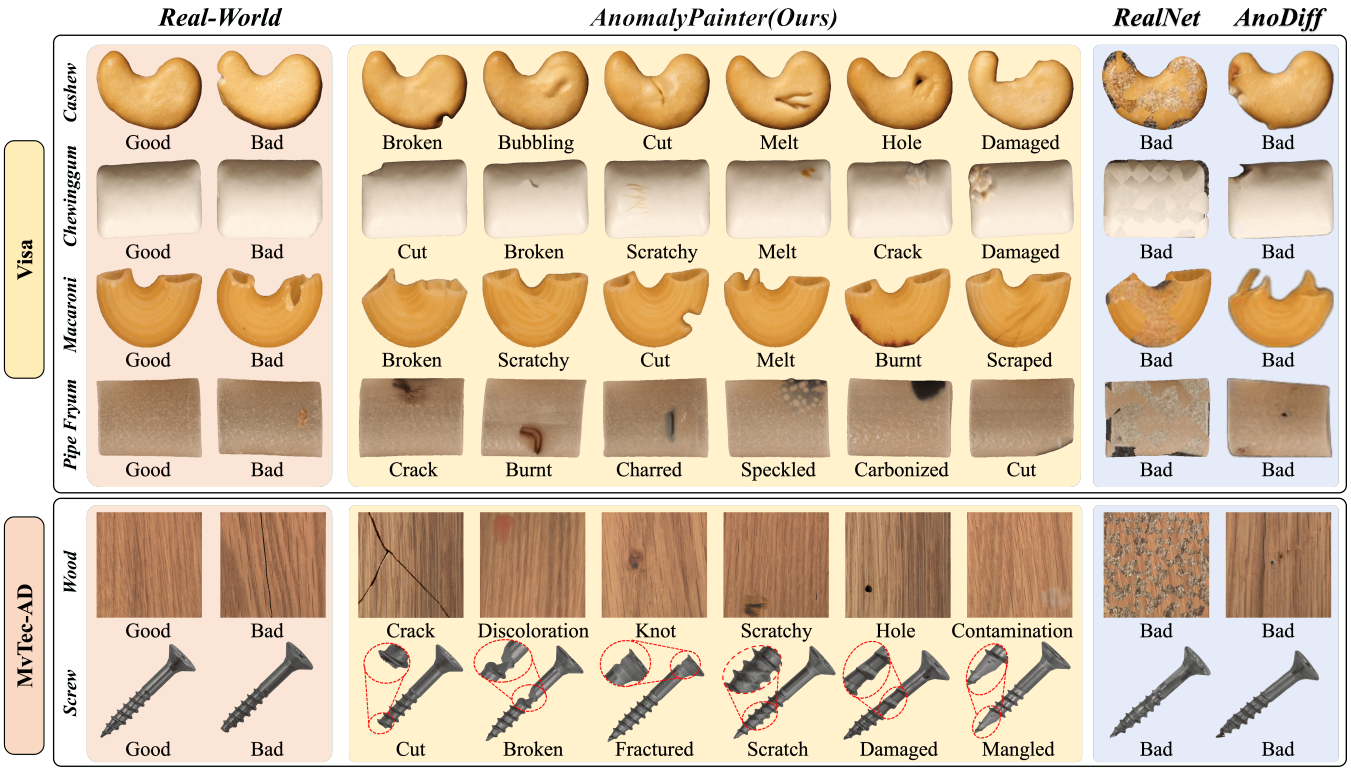

Figure 5: Qualitative comparison shows our method generates diverse and realistic anomaly data across various industrial objects and datasets. Ours outperforms representative methods like AnoDiff and RealNet.

Category	RealNet		AnoDiff		Ours	
	IS	IL	IS	IL	IS	IL
candle	1.33	0.27	1.33	0.18	**1.65**	**0.29**
capsules	1.65	**0.44**	1.32	0.33	**1.67**	0.39
cashew	1.65	0.31	1.29	0.27	**1.83**	**0.36**
chewinggum	1.69	0.37	1.27	0.36	**1.77**	**0.43**
fryum	1.35	0.22	1.13	0.18	**1.69**	**0.28**
macaroni1	**1.75**	**0.22**	1.50	0.21	1.73	0.21
macaroni2	1.78	0.32	1.61	0.24	**1.92**	**0.41**
pcb1	1.47	0.33	1.22	**0.35**	**1.49**	0.34
pcb2	1.37	**0.33**	**1.56**	0.30	1.53	0.31
pcb3	1.26	0.19	1.21	0.23	**1.51**	**0.26**
pcb4	1.35	**0.30**	1.27	0.28	**1.50**	0.27
pipe fryum	1.53	0.22	1.34	0.22	**1.69**	**0.36**
Average	1.52	0.29	1.34	0.26	**1.67**	**0.33**

Table 1: Quantitative comparison on IS and IL on VisA.

Comparison in Anomaly Generation

Qualitative Comparison. We present anomaly images synthesized by RealNet and AnoDiff on the VisA and MVTec datasets, as shown in Fig 5. Although RealNet can generate anomaly images with noticeable defects, the results often appear visually unrealistic and confusing. AnoDiff struggles with effective anomaly generation, as it maps different anomaly features into the same embedding, leading to feature confusion, lack of diversity. Notably, RealNet and AnoDiff rely on available anomalous training data, while AnomalyPainter synthesizes without training and generalizes to unseen object types and anomalies during inference.

Quantitative Comparison. We report IS and IL metrics on the VisA in Table 1. The results show that Ours outperforms AnoDiff and RealNet in both diversity and realism. Notably AnoDiff trains on 1/3 anomaly data from the test set.

Comparison in Anomaly Detection

We further evaluate our method under the challenging 1-shot anomaly detection scenario following AnomalyAny, where only one single normal sample is accessible and no abnormal samples are available. Following AnomalyGPT detection framework, we condition on a single normal image to generate 100 random anomaly samples. Tab. 2 presents the comparison results between our AnomalyPainter and existing few-shot anomaly detection methods, including two full-shot methods PaDiM (Defard et al. 2021),PatchCore (Roth et al. 2021) in the few-shot settings, and three CLIP-based few-shot methods WinCLIP+ (Jeong et al. 2023b), AnomalyGPT, PromptAD (Li et al. 2024b). As can be seen, our generated anomaly samples consistently outperform other alternatives in the majority of the metrics. To further compare with other anomaly synthesis methods under the data scarcity scenarios, we follow the same 1-shot setup and substitute the synthetic training data with anomaly samples generated by alternative methods. For DRAEM, NSA (Schlüter

Methods	MVTec AD					VisA				
	AUC-I	F1-I	AUC-P	F1-P	PRO	AUC-I	F1-I	AUC-P	F1-P	PRO
PaDiM	76.6±3.1	88.2±1.1	89.3±0.9	40.2±2.1	73.3±2.0	62.8±5.4	75.3±1.2	89.9±0.8	17.4±1.7	64.3±2.4
PatchCore	83.4±3.0	90.5±1.5	92.0±1.0	50.4±2.1	79.7±2.0	79.9±2.9	81.7±1.6	95.4±0.6	38.0±1.9	80.5±2.5
WinCLIP+	93.1±2.0	93.7±1.1	95.2±0.5	55.9±2.7	87.1±1.2	83.8±4.0	83.1±1.7	96.4±0.4	41.3±2.3	85.1±2.1
AnomalyGPT	94.1±1.1	-	95.3±0.1	-	-	87.4±0.8	-	96.2±0.1	-	-
PromptAD	94.6±1.7	-	95.9±0.5	-	-	86.9±2.3	-	96.7±0.4	-	-
AnomalyPainter (Ours)	**96.5±0.6**	**95.8±0.3**	**96.7±0.3**	**60.1±0.4**	**92.5±0.5**	**92.6±0.5**	**87.7±0.4**	**97.9±0.3**	**45.9±0.6**	**93.5±0.4**

Table 2: Comparison of 1-shot anomaly detection on MVTec AD and VisA datasets (5 runs).

Methods	MVTec AD					VisA				
	AUC-I	F1-I	AUC-P	F1-P	PRO	AUC-I	F1-I	AUC-P	F1-P	PRO
AnoDiff	94.4±0.3	94.4±0.2	95.3±0.5	57.3±3.0	92.2±1.0	-	-	-	-	-
DRAEM	93.6±0.3	94.2±0.4	95.1±0.1	56.0±0.9	91.8±0.1	86.0±0.7	83.0±0.9	97.5±0.1	42.6±0.7	92.6±0.6
NSA	94.0±0.5	94.2±0.3	95.1±0.1	56.1±0.5	91.8±0.2	86.2±2.0	83.1±1.2	97.4±0.1	40.8±0.5	92.3±0.3
RealNet	92.7±0.7	93.6±0.3	95.1±0.1	56.3±1.3	91.7±0.1	86.0±1.4	82.9±1.1	97.5±0.2	41.9±1.8	92.8±0.3
AnomalyAny	94.9±0.4	94.7±0.4	95.4±0.2	57.3±0.0	91.9±0.0	89.7±0.8	85.8±0.5	97.7±0.4	43.2±0.4	92.5±0.1
AnomalyPainter (Ours)	**96.5±0.6**	**95.8±0.3**	**96.7±0.3**	**60.1±0.4**	**92.5±0.5**	**92.6±0.5**	**87.7±0.4**	**97.9±0.3**	**45.9±0.6**	**93.5±0.4**

Table 3: Comparisons of 1-shot anomaly detection performance with different anomaly generation methods on MVTec AD and VisA. Since AnoDiff uses test anomalies for training and causes data leakage, it is excluded from ranking. Results are averaged over 5 runs. The best results are in **bold**, and second-best are underlined.

ADGM	TALI	ControlNet	IS	IL	AUC-I	AUC-P
✓	-	-	1.58	0.25	87.4	97.3
✓	✓	-	1.62	0.29	91.6	97.6
✓	-	✓	1.63	0.28	91.1	97.6
-	✓	✓	1.53	0.26	90.9	97.4
✓	✓	✓	**1.67**	**0.33**	**92.6**	**97.9**

Table 4: Ablation on VisA shows that removing any component degrades performance.

et al. 2022), RealNet, and AnomalyAny, we condition on the same single normal sample as for our method and generate 100 random anomaly samples for training. We report results in Tab 3, showing that AnomalyPainter's generated anomalies achieve the best detection performance.

Ablation Study

We evaluate the effectiveness of our components: Anomaly Description Generation and Matching (ADGM), Texture-Aware Latent Initialization (TALI) and ControlNet in Adaptive Anomaly Texture Generation. We design 5 different combinations shown in Table. 4. First, with only ADGM module, our method degrades to a crop-and-paste method, resulting in a significant decrease in realism with IL $= 0.25$. Next, we introduce TALI and ControlNet separately, both of which improve the realism, with IL increasing to 0.29 and 0.28. When TALI and ControlNet are combined, the realism improves further, with IL reaching 0.33, which also performs best in downstream detection tasks. We also run the full TALI and ControlNet experiment without the ADGM module, using random textures as guidance. As expected, random textures often lead to mismatched anomalies, reducing realism with IL decreasing to 0.26.

T^* **Selection**. Following TF-ICON, we adopt the common practice of setting $T = 20$. To find best T^* later than T, we experiment with the influence of different choices of T^* with quantitative results in Table 5. A typical qualitative visualization on wood is also shown in Figure 6. It is clear that too large step (e.g. $T^* = 20$) may cause the inpainting intensity to be too strong, while too small step (e.g. $T^* = 12$) can result in an overly strong initialization binding, making the anomaly content appear excessively unnatural.

Choice	T^*=20	T^*=18	T^*=16	T^*=14	T^*=12
IS/IL	1.64 / 0.30	**1.69** / 0.32	1.67 / **0.33**	1.65 / 0.31	1.59 / 0.32
AUC-I/AUC-P	91.6 / 97.7	92.2 / 97.8	**92.6 / 97.9**	91.9 / 97.7	91.7 / 97.8

Table 5: Quantitative Ablation with different T^*.

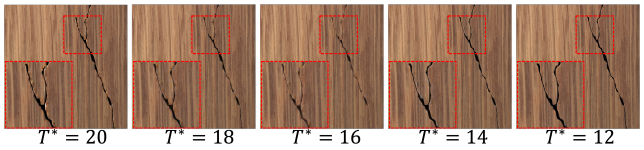

Figure 6: "A photo of cracked wood" with differet T^*.

Conclusion

We propose AnomalyPainter, a novel framework that generates diverse and realistic unseen anomalies by integrating the generative power of VLLMs and LDMs with our newly introduced Tex-9K. It provides an open-world anomaly synthesis capability, eliminating the need for prior industrial anomaly samples, and future work could focus on more sophisticated text-driven control and prompt engineering to further enhance flexibility. Overall, AnomalyPainter provides a strong foundation for scalable anomaly detection and opens new directions for industry and research.

Acknowledgments

This work was supported by the National Science Fund for Distinguished Young Scholars (No.62025603 and No.62525605), National Natural Science Foundation of China (No. U21B2037, U22B2051, No. U23A20383, No. 62176222, No. 62176226, No. 62272401, No. 62576300).

References

Bergmann, P.; Fauser, M.; Sattlegger, D.; and Steger, C. 2019. MVTec AD–A comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9592–9600.

Canny, J. 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6): 679–698.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; ; and Vedaldi, A. 2014. Describing Textures in the Wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Dai, S.; Qu, Y.; Li, Z.; Li, X.; Zhang, S.; and Cao, L. 2025. Training-Free Hierarchical Scene Understanding for Gaussian Splatting with Superpoint Graphs. *arXiv preprint arXiv:2504.13153*.

Defard, T.; Setkov, A.; Loesch, A.; and Audigier, R. 2021. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International conference on pattern recognition*, 475–489. Springer.

Duan, Y.; Hong, Y.; Niu, L.; and Zhang, L. 2023. Few-shot defect image generation via defect-aware feature manipulation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 571–578.

Gu, Z.; Zhu, B.; Zhu, G.; Chen, Y.; Tang, M.; and Wang, J. 2024. Anomalygpt: Detecting industrial anomalies using large vision-language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 1932–1940.

Gui, G.; Gao, B.-B.; Liu, J.; Wang, C.; and Wu, Y. 2024. Few-shot anomaly-driven generation for anomaly classification and segmentation. In *European Conference on Computer Vision*, 210–226. Springer.

Guo, Y.; Hu, J.; Qu, Y.; and Cao, L. 2025. WildSeg3D: Segment Any 3D Objects in the Wild from 2D Images. *arXiv preprint arXiv:2503.08407*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Hu, T.; Zhang, J.; Yi, R.; Du, Y.; Chen, X.; Liu, L.; Wang, Y.; and Wang, C. 2024. Anomalydiffusion: Few-shot anomaly image generation with diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 8526–8534.

Huang, C.; Jiang, A.; Feng, J.; Zhang, Y.; Wang, X.; and Wang, Y. 2024. Adapting visual-language models for generalizable anomaly detection in medical images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11375–11385.

Jeong, J.; Zou, Y.; Kim, T.; Zhang, D.; Ravichandran, A.; and Dabeer, O. 2023a. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19606–19616.

Jeong, J.; Zou, Y.; Kim, T.; Zhang, D.; Ravichandran, A.; and Dabeer, O. 2023b. WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19606–19616.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations (ICLR) 2014, Banff, AB, Canada*.

Li, C.-L.; Sohn, K.; Yoon, J.; and Pfister, T. 2021. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9664–9674.

Li, X.; Lai, Z.; Xu, L.; Qu, Y.; Cao, L.; Zhang, S.; Dai, B.; and Ji, R. 2024a. Director3d: Real-world camera trajectory and 3d scene generation from text. *Advances in Neural Information Processing Systems*, 37: 75125–75151.

Li, X.; Yi, C.; Lai, J.; Lin, M.; Qu, Y.; Zhang, S.; and Cao, L. 2025. SynergyAmodal: Deocclude Anything with Text Control. *arXiv preprint arXiv:2504.19506*.

Li, X.; Zhang, Z.; Tan, X.; Chen, C.; Qu, Y.; Xie, Y.; and Ma, L. 2024b. PromptAD: Learning Prompts with only Normal Samples for Few-Shot Anomaly Detection. *arXiv preprint arXiv:2404.05231*.

Lin, J.; Hu, Y.; Shen, J.; Shen, Y.; Cao, L.; Zhang, S.; and Ji, R. 2025. What You Perceive Is What You Conceive: A Cognition-Inspired Framework for Open Vocabulary Image Segmentation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 2841–2850.

Lin, J.; Shen, Y.; Wang, B.; Lin, S.; Li, K.; and Cao, L. 2024. Weakly supervised open-vocabulary object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 3404–3412.

Liu, J.; Huang, T.; and Xu, C. 2024. Training-Free Composite Scene Generation for Layout-to-Image Synthesis. In *European Conference on Computer Vision*, 37–53. Springer.

Liu, J.; Xie, G.; Wang, J.; Li, S.; Wang, C.; Zheng, F.; and Jin, Y. 2024. Deep industrial image anomaly detection: A survey. *Machine Intelligence Research*, 21(1): 104–135.

Liu, Z.; Zhou, Y.; Xu, Y.; and Wang, Z. 2023. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20402–20411.

Lu, R.; Wu, Y.; Tian, L.; Wang, D.; Chen, B.; Liu, X.; and Hu, R. 2023. Hierarchical Vector Quantized Transformer for Multi-class Unsupervised Anomaly Detection. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Lu, S.; Liu, Y.; and Kong, A. W.-K. 2023. TF-ICON: Diffusion-Based Training-Free Cross-Domain Image Composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2294–2305.

Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2024. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. In *The Twelfth International Conference on Learning Representations*.

Qin, X.; Zhang, Z.; Huang, C.; Dehghan, M.; Zaiane, O.; and Jagersand, M. 2020. U2-Net: Going Deeper with Nested U-Structure for Salient Object Detection. volume 106, 107404.

Qu, Y.; Chen, D.; Li, X.; Li, X.; Zhang, S.; Cao, L.; and Ji, R. 2025a. Drag Your Gaussian: Effective Drag-Based Editing with Score Distillation for 3D Gaussian Splatting. *arXiv preprint arXiv:2501.18672*.

Qu, Y.; Dai, S.; Li, X.; Lin, J.; Cao, L.; Zhang, S.; and Ji, R. 2024. Goi: Find 3d gaussians of interest with an optimizable open-vocabulary semantic-space hyperplane. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 5328–5337.

Qu, Y.; Dai, S.; Li, X.; Wang, Y.; Shen, Y.; Cao, L.; and Ji, R. 2025b. DeOcc-1-to-3: 3D De-Occlusion from a Single Image via Self-Supervised Multi-View Diffusion. *arXiv preprint arXiv:2506.21544*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10684–10695.

Roth, K.; Pemula, L.; Zepeda, J.; Schölkopf, B.; Brox, T.; and Gehler, P. 2021. Towards Total Recall in Industrial Anomaly Detection. arXiv:2106.08265.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 22500–22510.

Schlüter, H. M.; Tan, J.; Hou, B.; and Kainz, B. 2022. Natural synthetic anomalies for self-supervised anomaly detection and localization. In *European Conference on Computer Vision*, 474–489. Springer.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Sun, H.; Cao, Y.; Dong, H.; and Fink, O. 2025. Unseen Visual Anomaly Generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 25508–25517.

von Platen, P.; Patil, S.; Lozhkov, A.; Cuenca, P.; Lambert, N.; Rasul, K.; Davaadorj, M.; Nair, D.; Paul, S.; Berman, W.; Xu, Y.; Liu, S.; and Wolf, T. 2022. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Ye, K.; Luan, Y.; Chen, Z.; Meng, G.; Dai, P.; and Cao, L. 2025. RIS-LAD: A Benchmark and Model for Referring Low-Altitude Drone Image Segmentation. arXiv:2507.20920.

You, Z.; Cui, L.; Shen, Y.; Yang, K.; Lu, X.; Zheng, Y.; and Le, X. 2022. A Unified Model for Multi-class Anomaly Detection. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Yue, P.; Lin, J.; Zhang, S.; Hu, J.; Lu, Y.; Niu, H.; Ding, H.; Zhang, Y.; Jiang, G.; Cao, L.; et al. 2024. Adaptive selection based referring image segmentation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 1101–1110.

Zavrtanik, V.; Kristan, M.; and Skočaj, D. 2021. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 8330–8339.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding Conditional Control to Text-to-Image Diffusion Models.

Zhang, X.; Xu, M.; and Zhou, X. 2024. RealNet: A feature selection network with realistic synthetic anomaly for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16699–16708.

Zou, Y.; Jeong, J.; Pemula, L.; Zhang, D.; and Dabeer, O. 2022. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*, 392–408. Springer.