One-to-More: High-Fidelity Training-Free Anomaly Generation with Attention Control

Supplementary Material

Overview

This supplementary material provides additional details and results complementing the main paper. Specifically:

- Sec. A describes the algorithmic pipeline of O2MAG for anomaly generation.
- Sec. B presents additional self-attention feature analyses, providing further interpretability for our TriAG moudle of self-attention editing design.
- Sec. C supplies more experimental details and computation consumption.
- Sec. D provides additional experimental results on zero-shot cross-class anomaly generation.
- Sec. E presents qualitative visual results for the ablation of each component.
- Sec. F demonstrates the performance of our O2MAG on the VisA dataset under the same settings.
- Sec. G demonstrates the performance of our O2MAG on the Real-IAD dataset under one-reference settings.
- Sec. H reports more qualitative and quantitative results of anomaly image generation on MVTec-AD dataset.
- Sec. I analyzes the limitations of our O2MAG and discusses potential directions for future improvements.

A. Algorithmic Description of O2MAG

Algorithm 1 outlines the O2MAG pipeline for synthesizing anomalous samples. Given a reference anomaly image–mask pair $(I_{\mathrm{R}}, M_{\mathrm{R}})$, a normal image I_{N}, and a target anomaly mask M_{T}, our goal is to generate an anomaly image $\hat{I}$ that preserves the normal appearance outside M_{T} while producing realistic, text-consistent defects within M_{T}. We first align text conditioning with anomaly semantics via the Anomaly-Guided Optimization module (Sec. 4.2), which refines the input prompt to yield an anomaly-aligned text embedding. The pipeline then runs three parallel diffusion branches: a *reference–anomaly* branch and a *normal–background* branch that process noised versions of I_{R} and I_{N}, respectively, and a *target–anomaly* branch initialized with the noised latent of I_{N} to better retain structure and background. During sampling, we manipulate attention across the three branches at selected timesteps and attention layers using the **EDIT** function (Eq. (10)). Our attention editing mechanism comprises (i) *Tri-branch Attention Grafting* (TriAG) within self-attention, which injects anomalous cues from the reference branch and background cues from the normal branch into the target branch, and (ii) *Dual-Attention Enhancement* (DAE), which amplifies both cross-attention and self-attention responses. Together, these operations yield anomalies faithful to the target distribution and improve downstream detection performance.

B. Analysis on Self-attention for Anomaly Generation

As shown in Sec. 4.1, we apply principal component analysis (PCA) to the self-attention maps of anomalous images and visualize the top three principal components, where regions with similar structure are rendered in similar colors. We further present the query (Q), key (K), and the resulting self-attention maps $\boldsymbol{A} = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)$ to elucidate TriAG's self-attention grafting mechanism in Fig. 7.

As shown in Fig. 7, early layers exhibit attention aligned with the image's semantic layout, grouping regions by object parts and capturing coarse structure. Deeper layers progressively shift toward higher-frequency content, revealing fine-grained, patch-level texture cues. The spatial resolution follows the U-Net path—downsampling from 64×64 to 8×8, then upsampling back to 64×64. Because the down and middle block provides limited informative self-attention in our setting, we omit it and perform self-attention grafting primarily on the up block (layers 10–16).

C. More Experimental Details

C.1. More implementation details

Data preparation. We augment the normal images in the MVTec-AD training set (approximately 200–400 per class; toothbrush has only 60) using translations, flips, and rotations. For each category, we expand the training dataset of normal images to 1000 via data augmentation. For orientation-sensitive categories (e.g., zipper, capsule), we use flips only to preserve canonical semantics. This lightweight augmentation, together with our method's excellent preservation of normal background regions, may partly account for the lower IC-L compared with some baselines. Even so, our method still produces realistic and diverse anomalies. We adopt anomaly masks generated by AnomalyDiffusion as the target masks. Because AnomalyDiffusion occasionally yields empty masks (all-zero), we generate 1,200 masks and retain 1,000 valid ones after filtering out the empty cases.

Experimental Details and Hyperparameters. We deploy the pre-trained Stable Diffusion v1.5 [30] with 50 denoising steps and classifier-free guidance [16] set to 7.5 for anomaly synthesis without additional training.

Algorithm 1 Pipeline of Anomaly Generation in O2MAG

Input: A reference image–mask pair $(I_{\mathrm{R}}, M_{\mathrm{R}})$, a normal image I_{N}, a target anomaly mask M_{T}, and an anomaly text prompt y.
Output: Generated anomaly image $\hat{I}$.

$e^{\star} \leftarrow \mathrm{TextEncoder}(y) - \nabla_e L_{\mathrm{recon}}(I_{\mathrm{R}}, \hat{I}_{\mathrm{R}})$ ▷ AGO
$\{\boldsymbol{Z}_T^{\mathrm{ref}}, \boldsymbol{Z}_{T-1}^{\mathrm{ref}}, \ldots, \boldsymbol{Z}_0^{\mathrm{ref}}\} \leftarrow \mathrm{Inversion}(I_{\mathrm{R}})$
$\{\boldsymbol{Z}_T^{\mathrm{nor}}, \boldsymbol{Z}_{T-1}^{\mathrm{nor}}, \ldots, \boldsymbol{Z}_0^{\mathrm{nor}}\} \leftarrow \mathrm{Inversion}(I_{\mathrm{N}})$
$\boldsymbol{Z}_T^{\mathrm{tar}} \leftarrow \boldsymbol{Z}_T^{\mathrm{nor}}$
for $t = T, T-1, \ldots, 1$ **do**
  $\{\boldsymbol{Q}_R, \boldsymbol{K}_R, \boldsymbol{V}_R\} \leftarrow \epsilon_\theta(\boldsymbol{Z}_t^{\mathrm{ref}}, t)$ ▷ reference-anomaly diffusion branch
  $\{\boldsymbol{Q}_N, \boldsymbol{K}_N, \boldsymbol{V}_N\} \leftarrow \epsilon_\theta(\boldsymbol{Z}_t^{\mathrm{nor}}, t)$ ▷ normal-image diffusion branch
  $\{\boldsymbol{Q}_T, \boldsymbol{K}_T, \boldsymbol{V}_T\} \leftarrow \epsilon_\theta(\boldsymbol{Z}_t^{\mathrm{tar}}, t)$ ▷ target-anomaly diffusion branch
  $\boldsymbol{Attn}^{\star} \leftarrow \mathbf{EDIT}(\{\boldsymbol{Q}_T, \boldsymbol{K}_R, \boldsymbol{V}_R\}, \{\boldsymbol{Q}_T, \boldsymbol{K}_N, \boldsymbol{V}_N\}, M_{\mathrm{R}}, M_{\mathrm{T}})$ ▷ edit attention as Eq.(10) (DAE and TriAG)
  $\varepsilon \leftarrow \epsilon_\theta(\boldsymbol{Z}_t^{\mathrm{tar}}, t, \mathbf{Attn}^{\star}, e^{\star})$ ▷ noise prediction
  $\boldsymbol{Z}_{t-1}^{\mathrm{tar}} \leftarrow \mathrm{SampleDDIM}(\boldsymbol{Z}_t^{\mathrm{tar}}, \varepsilon, t)$ ▷ DDIM step
end for
$\hat{I} \leftarrow \mathrm{Decode}(\boldsymbol{Z}_0^{\mathrm{tar}})$
return $\hat{I}$

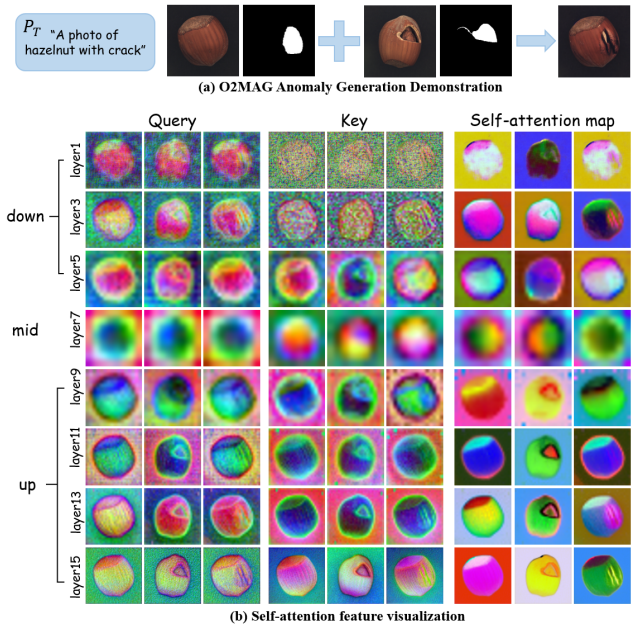

Figure 7. (a) The Demonstration of O2MAG Anomaly Generation, and (b) Self-attention feature maps visualized at the 30th denoising step. In each triplet of subfigures, the maps correspond (left→right) to the normal image, the reference anomaly image, and the generated image.

(1) **Additional Analysis on TriAG:** As illustrated in Fig. 8, images in anomaly datasets exhibit relatively simple structure and their layout emerge early during denoising. Accordingly, we enable self-attention editing from the $5th$ denoising step. Consistent with Sec. 4.1 and Appendix B, we perform attention grafting on the U-Net layers 10-16, which carry fine-grained, patch-level texture information.

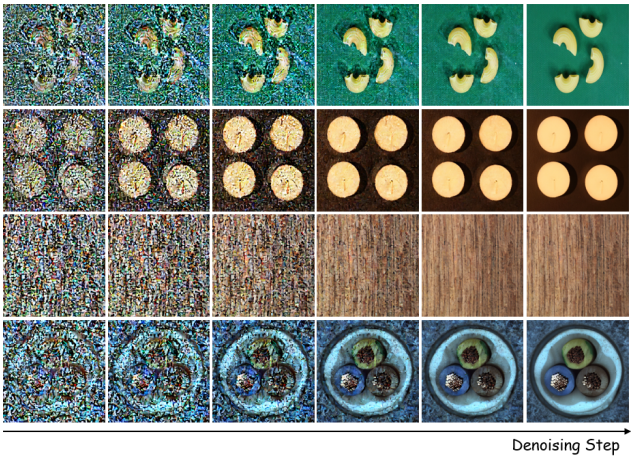

Figure 8. The intermediate reconstruction during the iterative denoising process. For both texture and object categories, the global image layout is already formed in the early denoising steps.

(2) **Additional Analysis on AGO:** During AGO, we optimize the anomaly text embedding for 500 steps on the 64×64 diffusion stage using Adam with a fixed learning rate of 3×10^{-3}. As illustrated in Fig. 4 of the main paper, optimization guides embeddings toward the target distribution. We set hyperparameters based on the reconstruction quality of the optimized prompt embeddings while accounting for computational cost, thereby balancing generation fidelity and optimization overhead. As illustrated in Fig. 9, our 500-step setting avoids the under-optimization of low steps while maintaining a substantial safety margin against the rigid reconstruction observed at 1000 steps.

Additionally, we explore incorporating data augmentation during the optimization stage. Specifically, we apply

Table 7. Comparison between the original and augmented settings.

Setting	AUC-I	AP-I	F1-I	AUC-P	AP-P	F1-P
Origin	99.9	100.0	99.0	99.8	96.2	90.1
Augment	99.9	99.9	98.9	99.8	95.9	91.0

random rotation and translation to the reference anomaly images, augmenting each reference sample into five variants. We then optimize the corresponding anomaly text embeddings. Subsequently, during the generation process conditioned on the real reference images, we randomly sample an augmented image along with its optimized embedding to guide the generation. Due to the inherent similarity in object appearance and defect characteristics within anomaly datasets, the model still primarily focuses on learning normal background features and foreground anomalies even after augmentation. Consequently, the underlying appearance remains essentially unchanged, meaning that the overall diversity is not significantly increased. Instead, the primary role of this augmentation module is to enhance the realism of the generated samples, ensuring they better align with the true distribution of real-world anomaly data. The comparison results on hazelnut of MVTec-AD are presented in Tab. 7.

(3) **Additional Analysis on DAE:** The self-attention bias γ is set to 1.1, the temperature τ_{fg} to 0.7, and the cross-attention upweight to $C = 100$. Timestamp $\tau_s \in (5, 50)$ for self-attention enhancement and $\tau_c \in (20, 40)$ for cross-attention enhancement. We selected DAE hyperparameters via empirical validation to handle the trade-off between sensitivity and stability. While a sufficiently large amplification factor C is essential for manifesting anomalies in small masks, Fig. 10 reveals a wide safety margin, with artifacts only emerging when $C > 10,000$. Similarly, γ and τ_{fg} perform robustly across broad ranges.

Critically, we employ the **same** hyperparameters across all datasets (MVTec-AD, VisA, Real-IAD) without per-dataset tuning, validating the **robust generalizability** of our method.

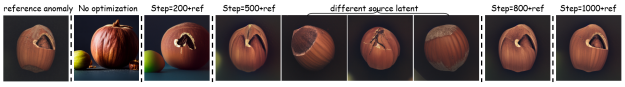

Figure 9. Analysis of the optimization step in AGO.

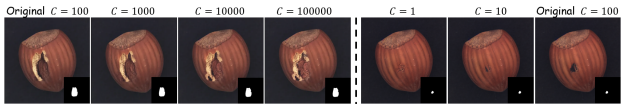

Figure 10. Analysis of the scaling factor C in DAE.

In addition, we use a unified simple text prompt template, "*A photo of a [cls] with a [anomaly_type]*", to specify the object class and anomaly type, which serves as the positive prompt. We further introduce negative prompts during synthesis. Technically, the positive prompt steers the diffusion process toward images consistent with its description, while the negative prompt pushes it away from the specified attributes. For anomaly types that are semantically related in the dataset, we construct antonym-like phrases as negative prompts. For example, for anomalies such as *crack*, *scratch*, and *rough* in MVTec-AD, we adopt phrases describing normal appearance (e.g., "no crack", "no cut", "no scratch", "smooth surface") as negative prompts, encouraging the model to generate anomalies that deviate from the normal appearance.

Table 8. Comparison of training cost and inference speed.

Methods	**Training Overall Time**	**Inference Time (per image)**
DFMGAN [9]	464 hours	0.9 s
AnomalyDiffusion [18]	310 hours	7.4 s
DualAnoDiff [20]	197 hours	32.4 s
SeaS [8]	124 hours	10.7 s
AnomalyAny [41]	0 hours	120s
O2MAG (ours)	0 hours	28s

C.2. Resource requirement and computation consumption

We conduct anomaly synthesis on NVIDIA RTX 5880 Ada (48 GB) GPUs for each product category, which may use about 30G memory. Table 13 reports runtime for training-based methods (DFMGAN, AnomalyDiffusion, DualAnoDiff, SeaS) and training-free methods (AnomalyAny and our O2MAG). DFMGAN, DualAnoDiff, and SeaS need to train a separate model per category, which increases both training time and storage. In addition, DFMGAN and AnomalyDiffusion operates at 256×256 resolution, whereas the other methods synthesize 512×512 images. Within the training-free setting, AnomalyAny requires about 120 s per image at inference, while our O2MAG needs only 28 s per image, achieving a $\approx 4.3\times$ speedup.

D. Additional Zero-Shot Generation Details

As discussed in Sec. 3.2, we evaluate the zero-shot generation capability of SeaS, AnomalyAny, and O2MAG. In this setting, the generation model only accesses normal images from the target category and anomalous images from other categories without access to target category anomalies. In Fig. 6, we transfer *wood–hole* anomalous attributes to synthesize *hazelnut–hole*. We further test other cross-class pairs including *hazelnut–crack*→*tile–crack*, *pill–scratch*→*metal_nut–scratch*, and *leather–color*→*wood–color*. Additional visualizations are shown in Fig. 11. Because self-attention features encode both texture and appearance information, attention-based feature transfer can cause appearance leakage into the target images, yielding cut-and-paste-like pseudo anomalies rather than seamlessly integrated defects.

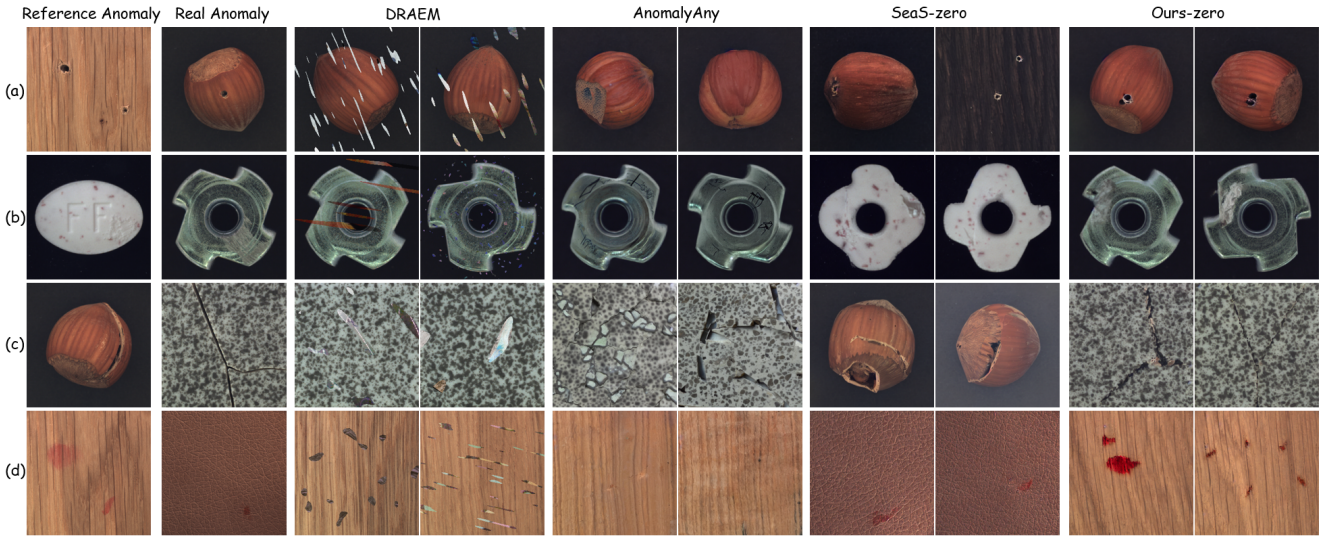

Figure 11. Anomaly generation under zero-shot settings. We aim to transfer anomalous features of the same anomaly type from "Reference Anomaly" images of other categories to the target category, synthesizing realistic defects that are consistent with "Real Anomaly".

E. More Ablation Studies

We ablate the three components of O2MAG —TriAG, DAE, and AGO. TriAG is kept active as the backbone in all variants, while DAE and AGO are removed individually and jointly to isolate their contributions. Additional anomaly-generation results are provided in Fig. 12.

Images to the left of the divider show the reference anomalous image, the normal image, and the target anomaly mask used for synthesizing new anomalies. In the *hazelnut–hole* case, our method can use one same reference anomaly image to guide the synthesis of more realistic defects. The visualization results indicate that the TriAG backbone effectively captures realistic anomaly semantics, but the synthesized anomalies may do not fully occupy the target anomaly mask, which limits improvements in downstream anomaly localization. To address this issue, we introduce the DAE module to enhance the visibility of anomalies. AGO further provides text-level guidance for anomaly synthesis, injecting more realistic anomalous textures and benefiting downstream classification performance. Finally, by combining attention editing for image-level anomaly semantics with text-level anomalous prompt optimization, our method can synthesize realistic and diverse anomalies, thereby improving downstream anomaly detection performance.

Mask acquisition & Impact. We also investigate the impact of mask sources on our synthesis process. Standard approaches generate masks via heuristics (e.g., Perlin noise in DRAEM) or learn mask distributions from few-shot samples. Aligning with these protocols, we utilize diverse mask sources for synthesis: specifically, AnomalyDiffusion masks for MVTec-AD and SeaS for VisA/Real-IAD.

Furthermore, we evaluate the effect of four mask sources on the *hazelnut-crack* category. For the Perlin noise setting, we randomly generate masks following the DRAEM approach and subsequently compute the intersection with the foreground mask of the normal image. The Nano Banana setting utilizes masks synthesized by Google's current image generation model. Experiments on *hazelnut-crack* confirm our robustness against these diverse mask sources (Table 9).

Table 9. Robustness to different mask source for our method.

Mask Source	AUROC-I	AP-I	F1-I	AUROC-P	AP-P	F1-P	Pro
Perlin noise	100	100	100	99.8	95.5	89.8	94.0
Nano Banana	100	100	100	99.9	97.3	92.1	95.1
AnomalyDiffusion	100	100	100	99.9	96.3	90.6	95.2
SeaS	100	100	100	99.6	94.8	89.3	95.9

F. Experiments on VisA Dataset

To validate the effectiveness of our approach, we additionally evaluate downstream anomaly detection and localization on VisA using anomalies synthesized by our method.

VisA dataset. The VisA dataset comprises 12 object categories spanning three domains. Its anomalous images exhibit a broad spectrum of defects, ranging from surface imperfections—such as scratches, dents, stains, and cracks—to logical anomalies, including misalignment and missing parts.

Experimental settings. We use VisA dataset divided by SeaS according to defect categories, while dividing the good data in the original dataset into train and good. We use SeaS to generate masks for anomaly synthesis, since AnomalyDiffusion often produces empty masks on VisA.

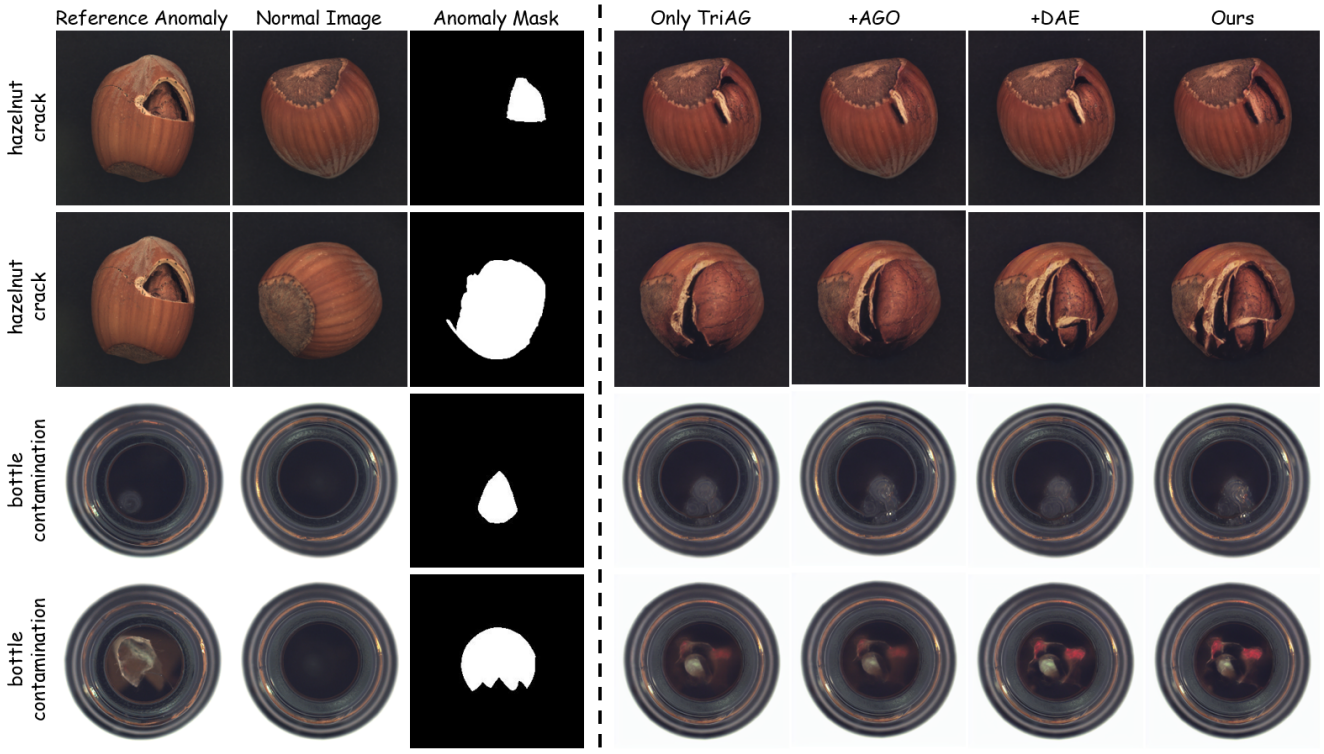

Figure 12. Qualitative ablation of TriAG, AGO, and DAE for anomaly realism and localization.

Following the MVTec-AD setup, we designate the first one-third images of VisA as reference images for anomaly synthesis and use the remaining two-thirds to evaluate downstream anomaly detection and localization. We evaluate three training-based anomaly generation baselines on VisA—AnomalyDiffusion, DualAnoDiff, and SeaS. We also include the training-free AnomalyAny for comparison. For each method, we synthesize 1,000 image–mask pairs and combine them with the first one-third of real anomalous images to train a U-Net segmentation model under the same experimental protocol as in MVTec-AD. Because detailed prompts for VisA and generation details in AnomalyAny are not publicly available, we instead report its best VisA detection performance from the paper under the *full-shot* setting.[1]

Anomaly image generation quality. Fig. 13 presents qualitative results on representative VisA categories synthesized by different methods. Since AnomalyAny cannot generate precise anomaly masks, it is excluded from this comparison. As shown in Fig. 13, AnomalyDiffusion still fails to correctly synthesize small anomalies on VisA. DualAnoDiff produces anomalous samples that deviate from the real data distribution, often corrupting the background and generating unrealistic objects. Using the same masks from SeaS, we observe that SeaS does not consistently produce mask-filling anomalies within the masked regions, which limits the improvement on downstream detection. In contrast, our method not only generates realistic anomalies but also faithfully fills the anomaly masks.

[1]The full-shot setting in AnomalyAny uses all normal images and generates 3-5 anomalous images conditioned on each normal image for training detection model.

Table 10. Comparison of different methods on VisA. Bold and underlined text indicate the best and second-best results, respectively.

Category	AnomalyDiffusion [18]		DualAnoDiff [20]		SeaS [8]		Ours	
	KID↓	IC-L↑	KID↓	IC-L↑	KID↓	IC-L↑	KID↓	IC-L↑
capsules	103.91	0.56	126.19	**0.64**	111.32	0.61	**20.53**	0.54
candle	183.21	0.17	180.22	**0.38**	47.65	0.10	**41.91**	0.12
cashew	78.22	0.38	52.46	**0.48**	10.37	0.35	**10.29**	0.35
chewinggum	39.63	0.34	29.76	**0.43**	**24.19**	0.29	33.36	0.33
fryum	238.41	0.27	102.00	**0.41**	82.20	0.24	**24.93**	0.18
macaroni1	153.18	0.22	207.27	**0.40**	152.04	0.22	**44.92**	0.19
macaroni2	178.19	0.35	221.05	**0.50**	150.64	0.40	**26.83**	0.35
pcb1	77.68	0.31	50.66	**0.40**	**29.91**	0.31	36.03	0.29
pcb2	81.66	0.29	36.04	**0.38**	30.80	0.30	**10.91**	0.27
pcb3	52.12	0.23	81.88	**0.35**	83.62	0.25	**8.63**	0.21
pcb4	34.29	0.29	28.68	**0.37**	12.31	0.25	**9.82**	0.24
pipe_fryum	76.04	0.21	47.70	**0.40**	23.59	0.19	**9.02**	0.23
Average	108.04	0.30	96.99	**0.43**	63.22	0.29	**23.10**	0.27

Anomaly generation for anomaly detection and localization. We report image-level and pixel-level anomaly detection results on VisA in Table 14 and Table 15, respectively. In addition, Table 11 compares our method with AnomalyAny. Since AnomalyAny does not report AP, we only compare AUROC and F1-max scores. Our method is

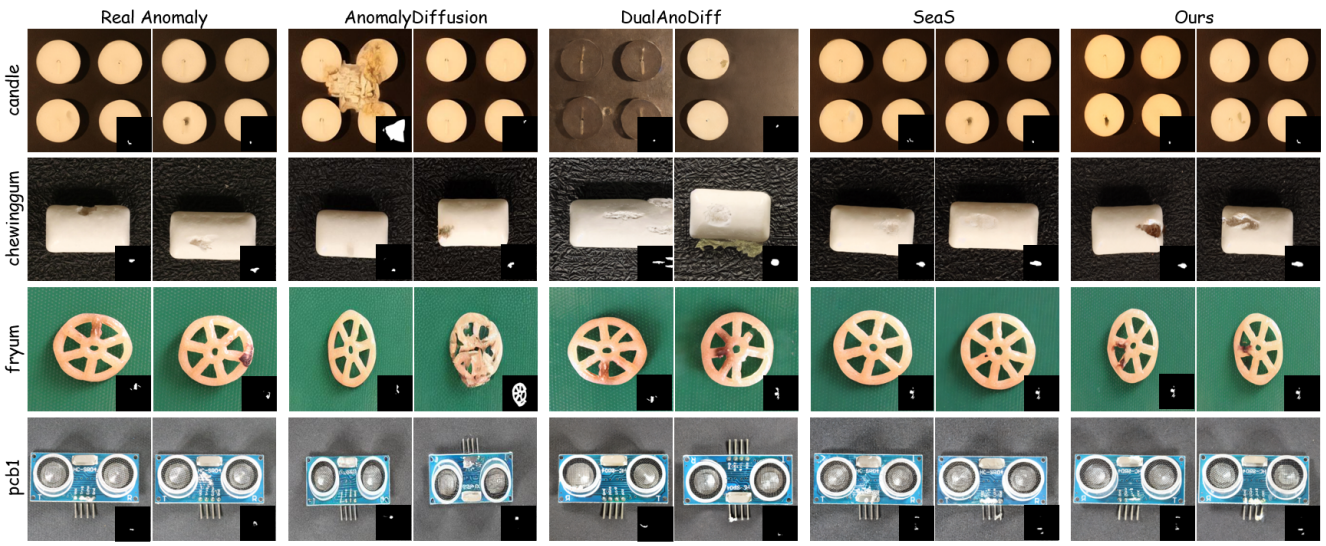

Figure 13. Qualitative comparison of generated results on VisA. The sub-image in the lower right corner is anomaly mask.

Table 11. Comparison to training-free AnomalyAny on VisA.

Category	AnomalyAny				Ours			
	I-AUC	I-F1	P-AUC	P-F1	I-AUC	I-F1	P-AUC	P-F1
candle	95.6	90.0	99.3	40.1	94.9	86.1	98.8	42.4
capsules	96.2	93.8	99.1	60.1	89.6	83.2	99.2	68.7
cashew	97.4	94.5	99.2	70.4	94.0	91.7	99.9	92.3
chewinggum	98.7	97.0	99.5	75.3	98.5	94.8	99.7	78.1
fryum	98.4	97.5	97.4	53.6	94.4	91.5	98.6	70.8
macaroni1	95.3	88.5	99.5	36.4	99.1	95.5	99.9	49.7
macaroni2	84.7	79.3	99.6	28.9	83.4	74.2	98.5	30.9
pcb1	95.9	91.8	98.8	41.8	93.7	82.9	99.7	83.4
pcb2	94.1	88.2	98.2	40.3	96.3	89.8	94.8	53.9
pcb3	95.9	90.4	97.5	52.9	97.1	92.9	98.2	64.6
pcb4	99.4	96.6	98.4	46.5	98.7	95.0	99.1	62.5
pipe_fryum	98.4	95.9	99.1	58.7	83.5	82.7	99.8	78.2
Average	**95.8**	**91.9**	98.7	50.4	93.6	88.4	**98.9**	**64.6**

slightly inferior to DualAnoDiff at the image level, as DualAnoDiff introduces a dedicated anomaly branch to learn foreground defects. Nevertheless, despite the challenges posed by small anomalies discussed in Sec. I, our approach still achieves the best performance at the pixel level.

G. Experiments on Real-IAD Dataset

Additionally, we evaluate our method's robustness under the one-reference setting using the Real-IAD dataset. This large-scale dataset provides five-view images encompassing 30 categories and 111 anomaly types, with a total of over 150K images. Each type roughly has 120 anomalous images. We use the first image from the top view as the reference and employ SeaS masks to control for spatial variation. We report both downstream AD performance using a trained U-Net in Table 12 and resource usage in Table 13. Training Time is computed on a single 48GB NVIDIA RTX 5880 GPU. Measured on a single NVIDIA RTX 5880 GPU, training-based methods incur high cumulative costs by requiring separate models per category (i.e., 30 distinct models for Real-IAD). Conversely, our training-free method outperforms them, relying on a frozen SD model to generalize across datasets.

Table 12. One-reference AD results on Real-IAD (trained U-Net).

Method	AUROC-I	AP-I	F1-I	AUROC-P	AP-P	F1-P	Pro
DualAnoDiff	72.5	77.8	75.8	91.3	25.7	31.9	75.7
SeaS	76.7	80.4	77.2	87.7	31.1	36.5	76.9
Ours	**79.3**	**84.0**	**80.4**	**93.6**	**37.3**	**41.9**	**82.9**

Table 13. Comparison of runtime and memory cost.

Methods	**Training Time (single GPU)**	**Inference Time (per image)**	**GPU Memory**
DualAnoDiff	119 hours	23 s	12G
SeaS	160 hours	11 s	24G
O2MAG Ours	0 hours	28 s	32G

H. Additional Results

H.1. More anomaly detection and localization Results

Tables 16 and 17 report detailed image-level and pixel-level metrics on MVTec-AD for each category, where we first generate 1,000 anomalous images per category and then combine them with the first one-third of real anomalous images in MVTec-AD dataset to train a U-Net segmentation model for downstream anomaly detection and localization. Our method achieves the best performance on both image-level and pixel-level anomaly detection tasks.

H.2. More anomaly generation results

We conduct a comprehensive qualitative comparison between our generated results and existing anomaly image synthesis methods. Since the GAN-based method DFM-

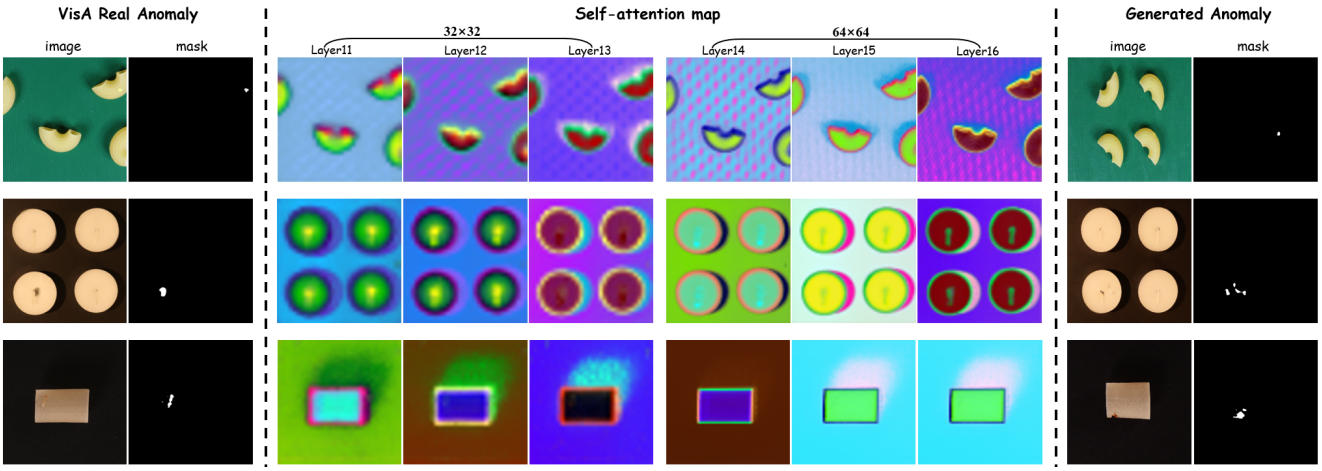

Figure 14. Limitations of Self-Attention for Tiny Anomaly Generation. The middle columns show the three leading principal components of the self-attention matrix for the real reference anomaly on the left at the 30th denoising step, while the right columns present our synthesized anomalous images and the corresponding anomaly masks. For small anomalous regions, the anomaly is almost invisible in the self-attention feature maps spanned by the top three PCA components, which makes it challenging to synthesize anomalies within small masks.

Table 14. Comparison on a trained U-Net segmentation model for **image-level** anomaly detection and localization on **VisA** dataset.

Category	AnomalyDiffusion			DualAnoDiff			SeaS			Ours		
	AUC-I	AP-I	F1-I	AUC-I	AP-I	F1-I	AUC-I	AP-I	F1-I	AUC-I	AP-I	F1-I
candle	95.1	95.3	90.1	**96.7**	**95.6**	**90.4**	78.7	77.9	68.5	94.9	93.3	86.1
capsules	**91.9**	**93.8**	**85.9**	91.0	93.0	85.7	84.7	89.0	78.6	89.6	91.9	83.2
cashew	95.2	96.0	92.2	**99.2**	**99.4**	**96.4**	94.8	95.2	92.4	94.0	94.6	91.7
chewinggum	94.4	97.2	93.1	**99.6**	**99.7**	**97.8**	98.9	99.3	95.0	98.5	99.0	94.8
fryum	81.3	87.2	79.2	86.7	90.7	84.1	91.2	94.4	86.7	**94.4**	**96.7**	**91.5**
macaroni1	94.6	92.7	86.5	96.8	95.6	90.3	98.8	98.4	92.9	**99.1**	**98.9**	**95.5**
macaroni2	67.3	48.9	41.0	**91.1**	**89.9**	**78.3**	86.3	86.0	77.6	83.4	83.1	74.2
pcb1	90.9	88.4	79.5	**96.2**	**96.0**	90.6	95.7	95.8	**92.1**	93.7	91.9	82.9
pcb2	94.8	92.9	88.3	95.8	95.8	**90.4**	94.5	94.5	88.1	**96.3**	**96.0**	89.8
pcb3	92.8	90.8	81.8	95.0	94.3	90.2	**99.0**	**98.6**	**93.8**	97.1	97.1	92.9
pcb4	94.8	92.1	82.3	98.1	**98.4**	95.0	**98.9**	95.9	**97.2**	98.7	97.2	95.0
pipe_fryum	88.3	91.2	85.5	**95.2**	**96.9**	**90.1**	92.0	94.3	86.5	83.5	87.8	82.7
Average	90.1	88.9	82.1	**95.1**	**95.4**	**89.9**	92.8	93.3	87.5	93.6	94.0	88.4

GAN yields relatively limited visual fidelity, we primarily compare against diffusion-based approaches, with additional visualizations provided in Figs. 17–31. The leftmost column shows real anomalous and normal images from the training set, the middle columns show anomalies synthesized by training-based methods (AnomalyDiffusion, DualAnoDiff, SeaS), and the rightmost column presents the training-free AnomalyAny and our O2MAG. Because AnomalyAny does not release its official prompts, we adopt the same prompt template as ours, "*A photo of a [cls] with a [anomaly_type]*", and use GPT-5 to expand them into detailed text prompts for generation.

From these visualizations, we observe that AnomalyDiffusion generally preserves the normal background appearance but sometimes fails to synthesize clearly visible defects. DualAnoDiff and SeaS produce more pronounced anomalies, yet these often deviate from the true data distribution, and SeaS further introduces synthesis artifacts, which in turn limit downstream anomaly classification. In contrast, training-free methods better preserve the appearance of normal regions. However, AnomalyAny lacks guidance from real anomalous images and cannot reliably capture rich anomaly semantics from text alone. As a result, it often fails to capture the spatial relationship be-

Table 15. Comparison on a trained U-Net segmentation model for **pixel-level** anomaly detection and localization on **VisA** dataset.

Category	AnomalyDiffusion			DualAnoDiff			SeaS			Ours		
	AUC-P	AP-P	F1-P	AUC-P	AP-P	F1-P	AUC-P	AP-P	F1-P	AUC-P	AP-P	F1-P
candle	**99.2**	24.7	33.0	98.9	**48.8**	**49.7**	96.5	29.9	37.5	98.8	39.8	42.4
capsules	**99.6**	60.2	61.2	98.4	60.7	63.8	99.5	59.1	62.3	99.2	**70.2**	**68.7**
cashew	98.4	68.8	63.9	**99.9**	98.0	94.5	**99.9**	**98.2**	**95.3**	**99.9**	96.4	92.3
chewinggum	98.7	78.3	72.4	**99.9**	**86.6**	**78.2**	99.5	84.3	77.6	99.7	85.5	78.1
fryum	91.0	28.2	31.8	97.9	76.7	**74.7**	97.4	74.5	69.5	**98.6**	**80.6**	70.8
macaroni1	98.2	4.2	10.6	99.8	28.7	37.6	**99.9**	**50.4**	**53.3**	**99.9**	47.1	49.7
macaroni2	90.6	0.1	0.0	97.2	19.7	30.3	97.3	17.8	26.7	**98.5**	**22.4**	**30.9**
pcb1	99.0	76.6	75.8	98.9	92.6	88.1	99.4	**94.2**	**90.3**	**99.7**	88.9	83.4
pcb2	98.2	23.1	37.1	96.1	35.8	46.6	**99.1**	49.7	49.3	94.8	**51.3**	**53.9**
pcb3	93.6	36.7	43.6	98.6	56.3	58.2	**98.8**	68.6	62.2	98.2	**69.0**	**64.6**
pcb4	98.2	44.3	53.3	98.8	52.3	53.3	98.6	61.9	60.5	**99.1**	**66.1**	**62.5**
pipe_fryum	99.0	69.9	66.7	99.4	79.1	69.6	**99.9**	**93.1**	**85.1**	99.8	88.6	78.2
Average	97.0	42.9	45.8	98.7	61.3	62.1	98.8	65.1	64.1	**98.9**	**67.2**	**64.6**

Table 16. Comparison on a trained U-Net segmentation model for **image-level** anomaly detection and localization on **MVTec-AD** dataset.

Category	**DFMGAN**			**AnomalyDiffusion**			**DualAnoDiff**			**SeaS**			**Ours**		
	AUC-I	AP-I	F1-I	AUC-I	AP-I	F1-I	AUC-I	AP-I	F1-I	AUC-I	AP-I	F1-I	AUC-I	AP-I	F1-I
bottle	99.3	99.8	97.7	99.8	99.9	98.9	**100**	**100**	**100**	99.9	99.9	98.9	**100**	**100**	**100**
cable	95.9	97.8	93.8	**100**	**100**	**100**	99.8	99.8	98.5	98.0	98.8	96.1	**100**	**100**	**100**
capsule	92.8	98.5	94.5	**99.7**	**99.9**	**98.7**	96.3	99.2	94.7	97.1	99.2	95.4	99.3	99.8	98.0
carpet	67.9	87.9	87.3	96.7	98.8	94.3	98.6	99.9	96.7	97.4	99.0	96.7	**100**	**100**	**100**
grid	73.0	90.4	85.4	98.4	99.5	98.7	**100**	99.7	**100**	99.9	99.9	98.8	**100**	**100**	**100**
hazelnut	**99.9**	**100**	**99.0**	99.8	99.9	98.9	**99.9**	**100**	**99.0**	99.8	99.8	**99.0**	**99.9**	**100**	**99.0**
leather	99.9	**100**	99.2	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.2
metal_nut	99.3	99.8	99.2	**100**	**100**	**100**	**100**	99.9	**100**	99.9	**100**	99.2	**100**	**100**	**100**
pill	68.7	91.7	91.4	98.0	99.6	97.0	98.2	99.0	95.9	98.4	99.6	**97.9**	**98.9**	**99.7**	97.5
screw	22.3	64.7	85.3	**96.8**	97.9	**95.5**	91.4	95.0	88.6	95.2	98.0	92.5	95.7	**98.2**	94.9
tile	**100**	**100**	**100**	**100**	**100**	**100**	99.5	100	98.3	**100**	**100**	**100**	**100**	**100**	**100**
toothbrush	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
transistor	90.8	92.5	88.9	**100**	**100**	**100**	**100**	99.7	**100**	99.8	99.5	96.4	**100**	**100.0**	**100**
wood	98.4	99.4	**98.8**	98.4	99.4	**98.8**	99.6	**99.9**	**98.8**	99.0	99.6	**98.8**	**99.7**	99.8	**98.8**
zipper	99.7	99.9	99.4	99.9	**100**	99.4	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Average	87.2	94.8	94.7	99.2	99.7	98.7	98.9	98.9	98.0	99.0	99.6	98.0	**99.6**	**99.8**	**99.2**

tween anomalies and object regions, and may either produce no visible defects or synthesize defects that deviate from the real anomaly distribution, thereby limiting downstream anomaly detection and classification. By comparison, our O2MAG does not require fine-tuning Stable Diffusion for each object category and under a single shared parameter setting, our O2MAG can synthesize realistic and diverse anomalies, substantially improving downstream anomaly detection accuracy.

H.3. Generalization to Real-World Scenes

Our method demonstrates robust generalization to real-world scenes (e.g., UAV imagery from UAV-RSOD dataset) beyond object-centric images by leveraging nano-banana generated masks (Fig.15).

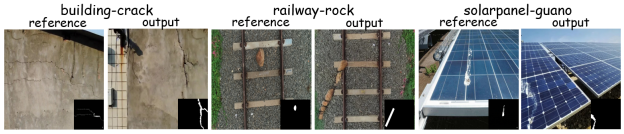

Figure 15. Generalization to real-world scenes.

Table 17. Comparison on a trained U-Net segmentation model for **pixel-level** anomaly detection and localization on **MVTec-AD** dataset.

Category	DFMGAN			AnomalyDiffusion			DualAnoDiff			SeaS			Ours		
	AUC-P	AP-P	F1-P	AUC-P	AP-P	F1-P	AUC-P	AP-P	F1-P	AUC-P	AP-P	F1-P	AUC-P	AP-P	F1-P
bottle	98.9	90.2	83.9	99.4	94.1	87.3	99.5	93.4	85.7	**99.7**	**95.9**	**88.8**	**99.7**	95.4	88.4
cable	97.2	81.0	75.4	99.2	90.8	83.5	98.5	82.6	76.9	96.0	83.1	77.7	**99.4**	**91.2**	**85.0**
capsule	79.2	26.0	35.0	98.8	57.2	59.8	**99.5**	**73.2**	**67.0**	93.7	41.9	47.3	97.0	60.6	59.0
carpet	90.6	33.4	38.1	98.6	81.2	74.6	99.4	**89.1**	**80.2**	99.3	86.4	78.1	**99.5**	88.5	80.0
grid	75.2	14.3	20.5	98.3	52.9	54.6	98.5	57.2	54.9	**99.7**	76.3	70.0	99.6	**78.6**	**71.6**
hazelnut	99.7	95.2	89.5	**99.8**	96.5	90.6	**99.8**	**97.7**	**92.8**	99.5	92.3	85.6	**99.8**	96.2	90.1
leather	98.5	68.7	66.7	99.8	79.6	71.0	**99.9**	**88.8**	78.8	99.8	85.2	77.0	99.7	88.0	**79.7**
metal_nut	99.3	98.1	94.5	**99.8**	98.7	94.0	99.6	98.0	93.0	**99.8**	**99.2**	**95.7**	**99.8**	**99.2**	**95.7**
pill	81.2	67.8	72.6	99.7	93.9	**90.8**	99.6	95.8	89.2	**99.9**	**97.1**	90.7	99.7	96.1	89.9
screw	58.8	2.2	5.3	97.0	51.8	50.9	98.1	57.1	56.1	98.5	58.5	57.2	**99.4**	**68.2**	**64.4**
tile	99.5	97.1	91.6	99.2	93.6	86.2	99.7	97.1	91.0	99.8	97.9	92.5	**99.9**	**98.2**	**92.7**
toothbrush	96.4	75.9	72.6	**99.2**	**76.5**	**73.4**	98.2	68.3	68.6	98.4	70.0	68.1	96.3	58.6	59.2
transistor	96.2	81.2	77.0	99.3	92.6	85.7	98.0	86.7	79.6	98.0	87.3	81.9	**99.9**	**98.2**	**93.2**
wood	95.3	70.7	65.8	**98.9**	84.6	74.5	99.4	**91.6**	**83.8**	99.0	87.0	79.6	99.4	89.4	81.1
zipper	92.9	65.6	64.9	99.6	86.0	79.2	99.6	**90.7**	**82.7**	**99.7**	88.2	81.6	99.5	88.5	82.0
Average	90.0	62.7	62.1	99.1	81.4	76.3	99.1	84.5	78.8	98.7	83.1	78.1	**99.2**	**86.3**	**80.8**

I. Limitations

Our method has two limitation: limit control on logical anomaly generation and self-attention grafting performs suboptimally when the reference anomaly is small.

Limit control on logical anomaly generation. Our method synthesizes anomalies by manipulating attention and optimizing text embedding (AGO). However, compared with training-based approaches, it is less effective at reasoning aboutlogical anomalies. On MVTec-AD, two representative cases: cable-cable_swap and transistor-misplaced, remain challenging. As shown in Fig. 16, Transistor-misplaced typically denotes that the transistor pins are not correctly connected to the solder pads, or the transistor is missing. The training-free AnomalyAny enlarges attention on anomaly tokens. By using the same prompt template "*A photo of a [cls] with a [anomaly_type]*" and LLM-generated detailed prompts, it still fails to match the dataset distribution and does not produce the intended anomalies. In our method, For misplacement case, the anomaly mask covers the transistor foreground. During self-attention grafting this can entangle features from the normal and reference branches, yielding inconsistent content as shown in Fig. 16. For the missing-transistor case, we do not manipulate attention in the initial five denoising steps, allowing the target branch to form an initial transistor silhouette that becomes difficult to remove later. For cable-cable_swap, among the expected green/blue/gray wires, two share the same color. Our approach further relies on precise mask–wire correspondence, so any misalignment of the mask often leads to bad generation.

We plan to integrate MLLMs to enrich prompt semantics for precise logical guidance in future work.

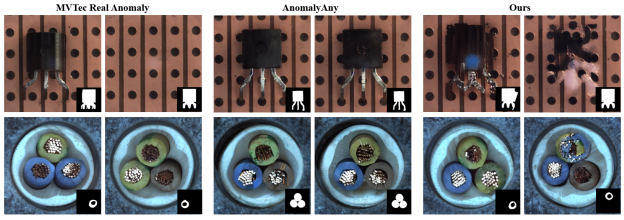

Figure 16. Logical anomaly generation.

Suboptimal on small anomaly generation. As discussed in Sec. 3.2, self-attention is applied to intermediate feature maps at 64×64, 32×32, 16×16, and 8×8. The reference anomaly image is accordingly downsampled to the corresponding spatial resolution. When the anomalous region is small, its representation in the self-attention space tends to exhibit unclear boundaries relative to the background. As illustrated in Fig. 14, several VisA cases with small reference anomalies yield low-contrast self-attention maps. Concretely, at the 30th denoising step we apply principal component analysis (PCA) to the self-attention maps and visualize the top three principal components. Because the variance contributed by the anomalous region is much smaller than that of the background, the leading components tend to "ignore" the anomaly, which hampers synthesis of tiny defects.

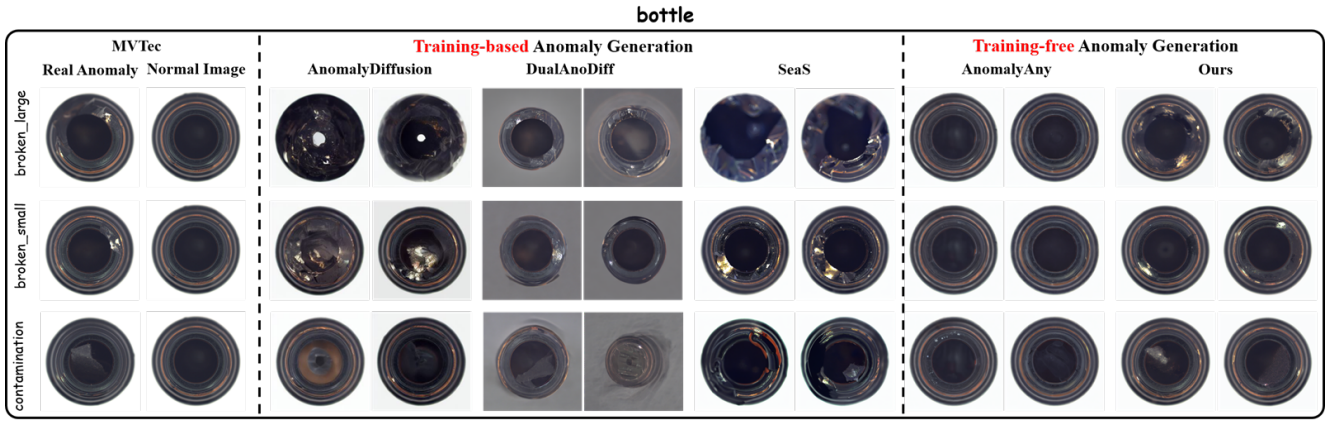

Figure 17. bottle qualitative results on MVTec-AD.

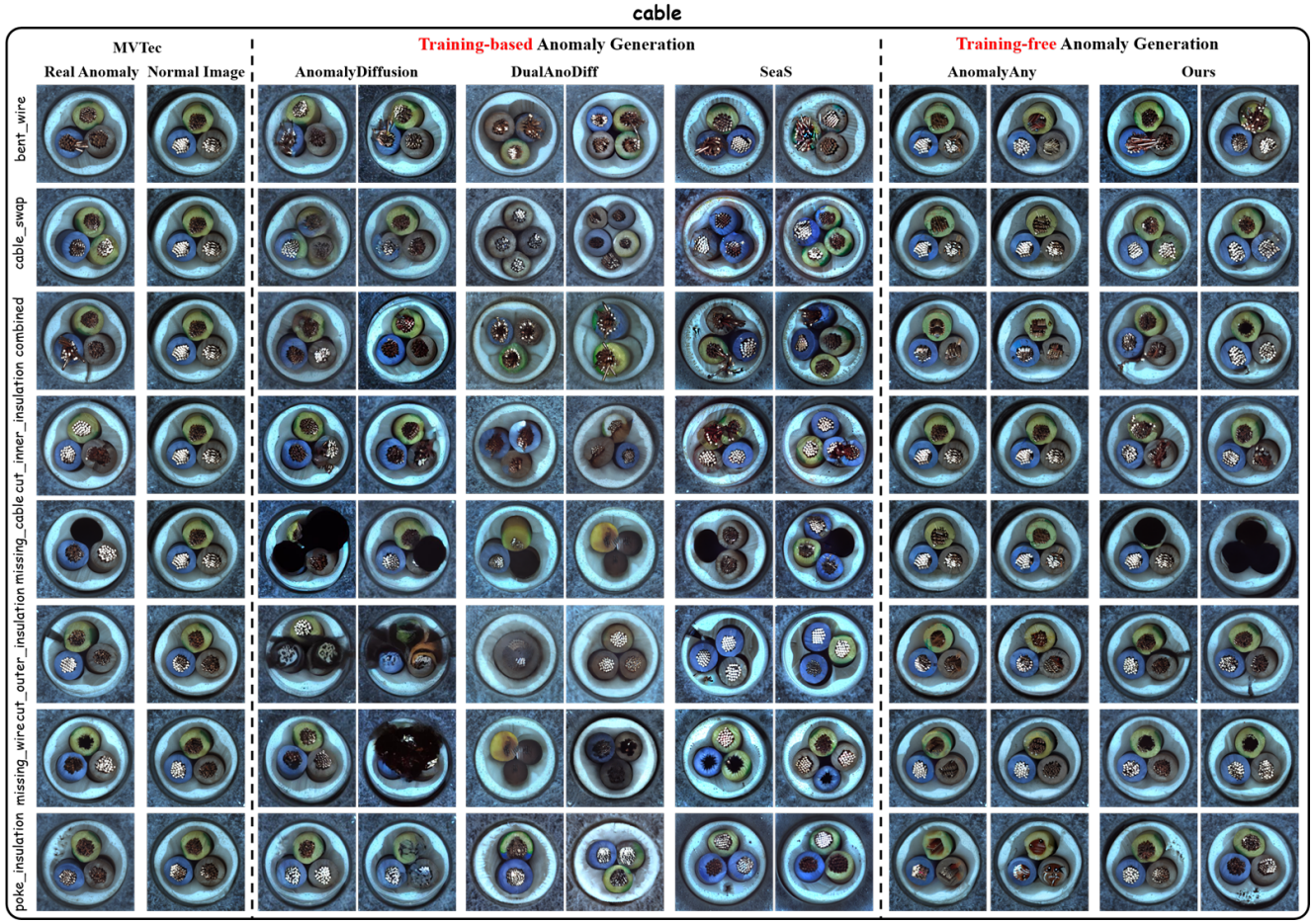

Figure 18. cable qualitative results on MVTec-AD.

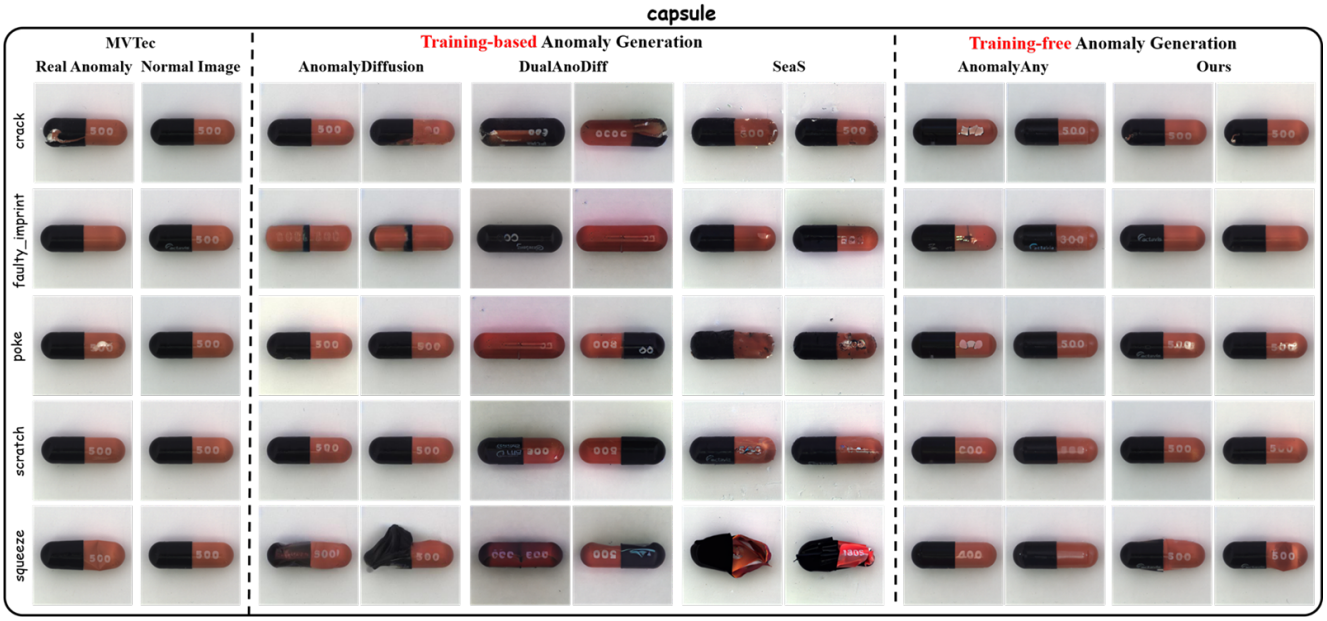

Figure 19. capsule qualitative results on MVTec-AD.

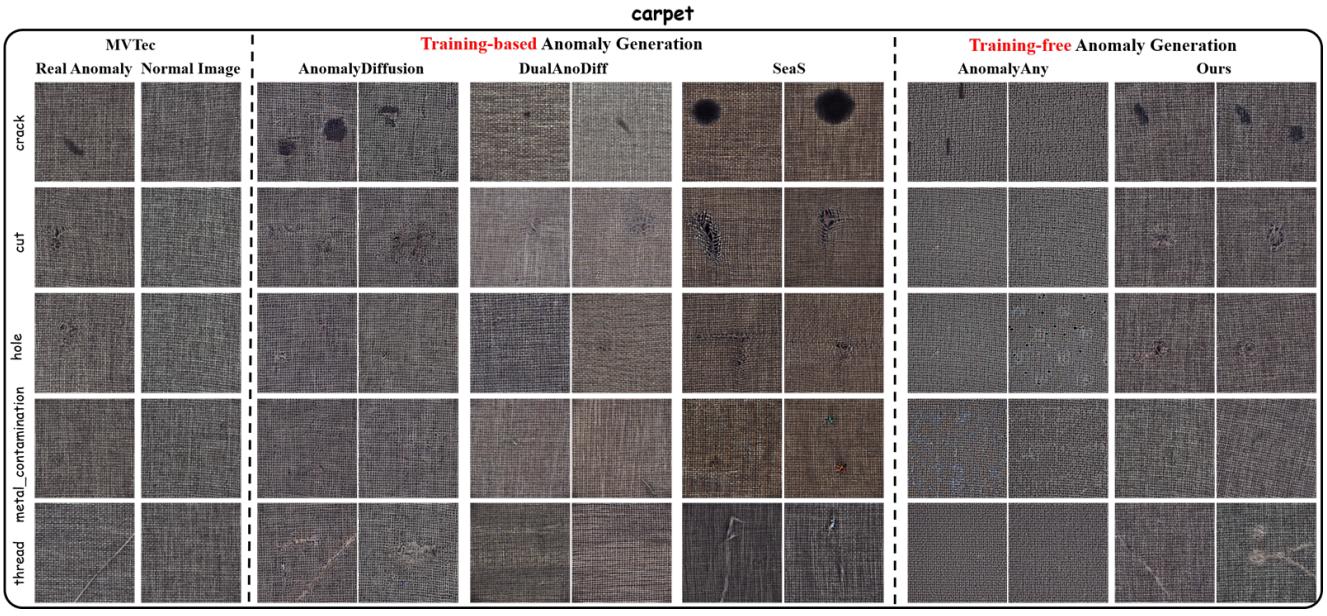

Figure 20. carpet qualitative results on MVTec-AD.

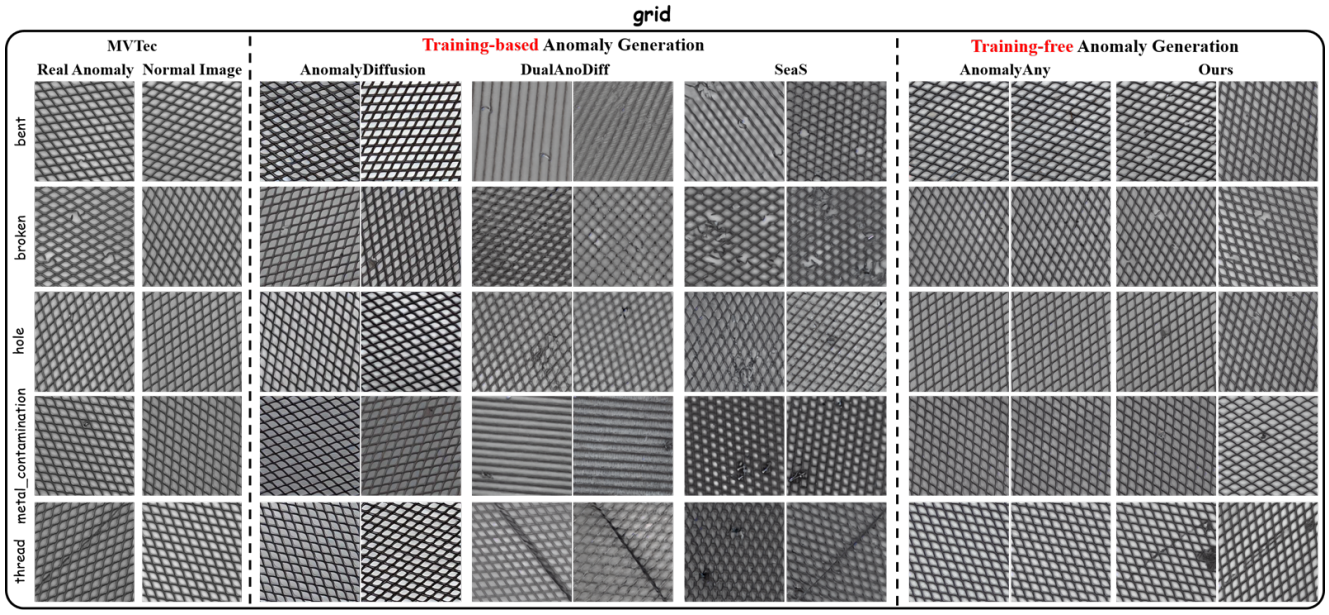

Figure 21. AnomalyDiffusion fails to synthesize the intended defects within the anomaly mask; DualAnoDiff and SeaS do not preserve background appearance and produce anomalies with distribution shift; AnomalyAny struggles to capture precise anomaly semantics. In contrast, our method preserves background fidelity and synthesizes diverse, realistic anomalies.

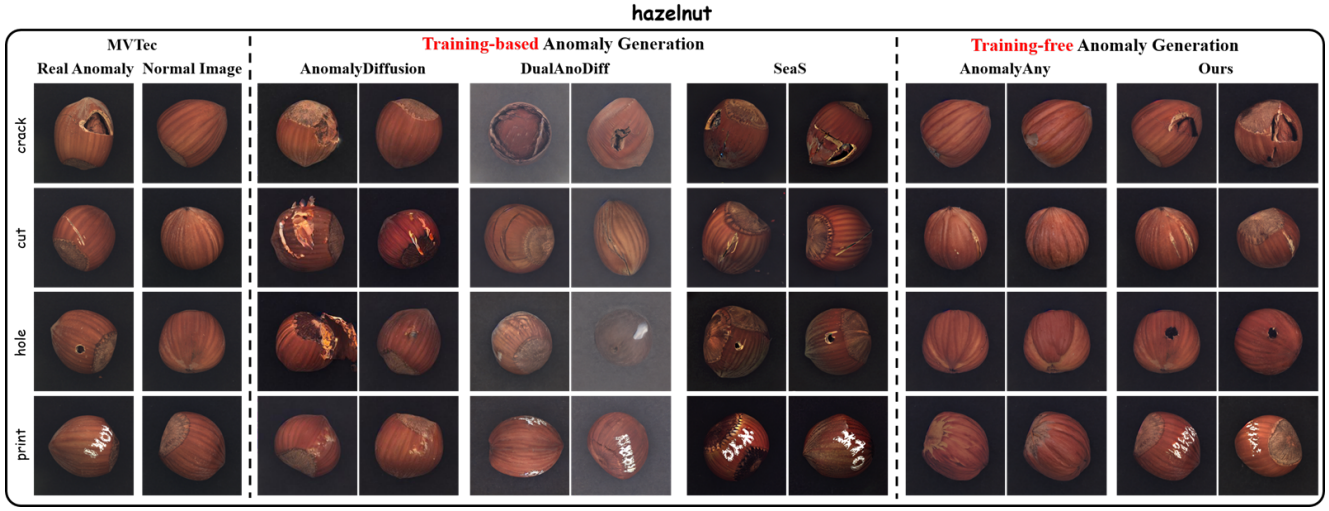

Figure 22. hazelnut qualitative results on MVTec-AD.

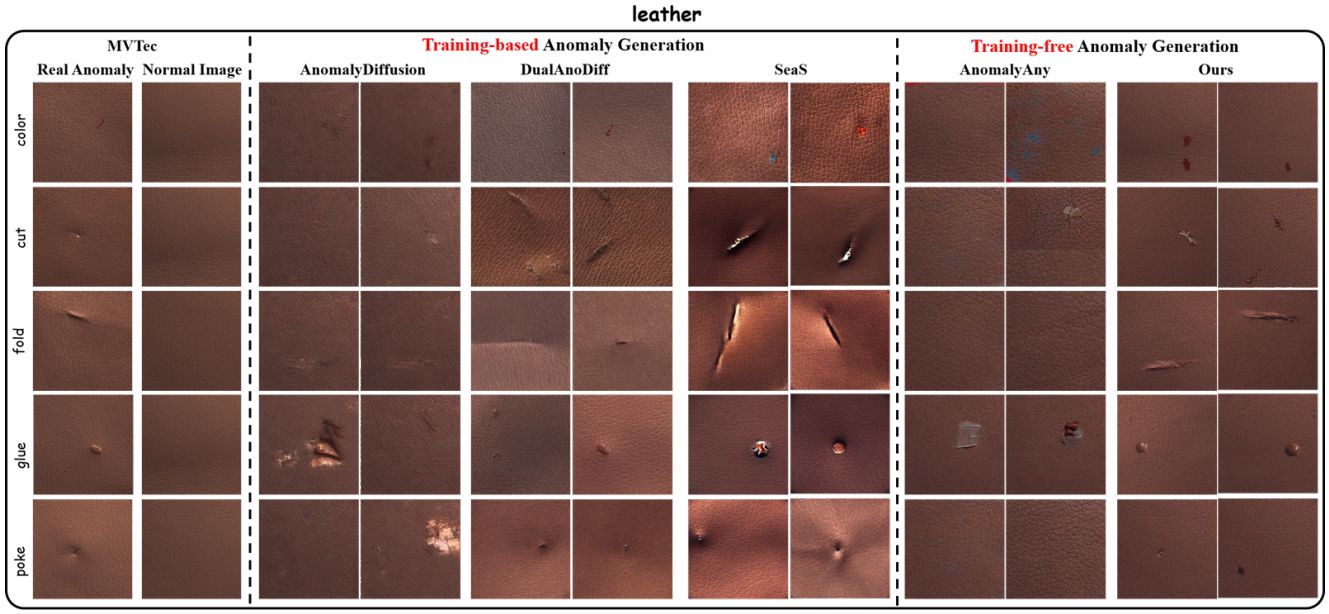

Figure 23. leather qualitative results on MVTec-AD.

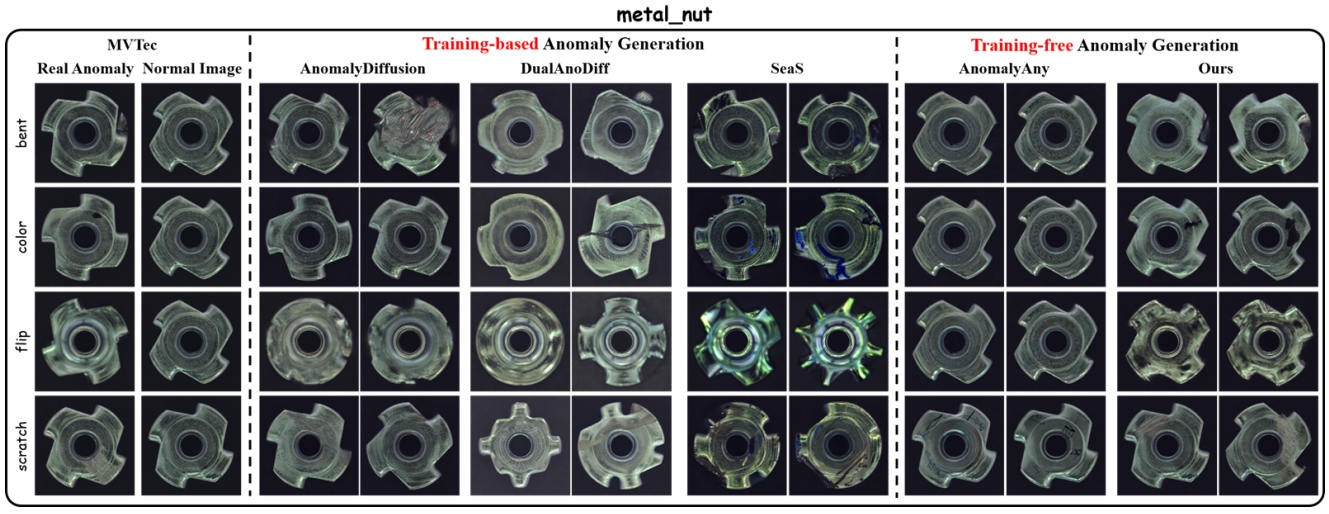

Figure 24. metal_nut qualitative results on MVTec-AD.

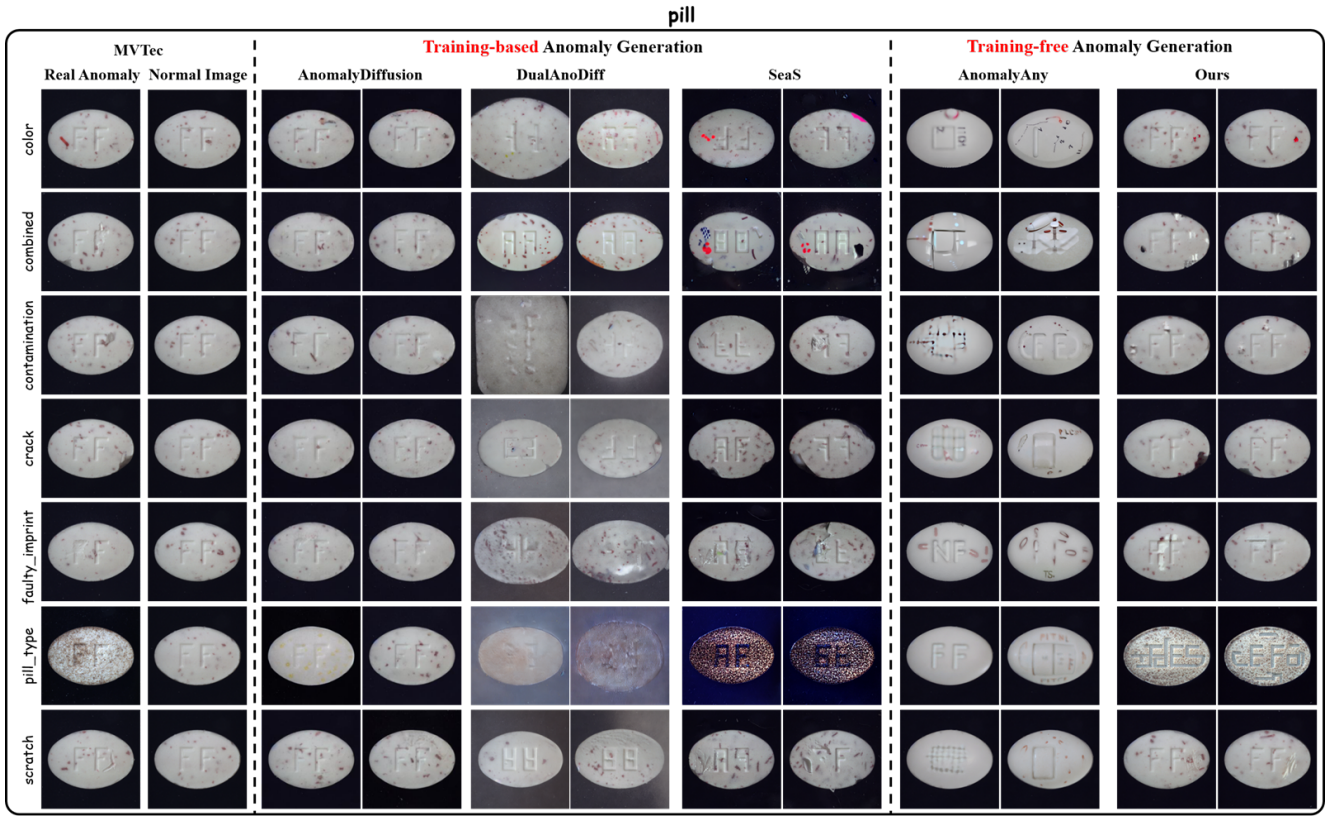

Figure 25. pill qualitative results on MVTec-AD.

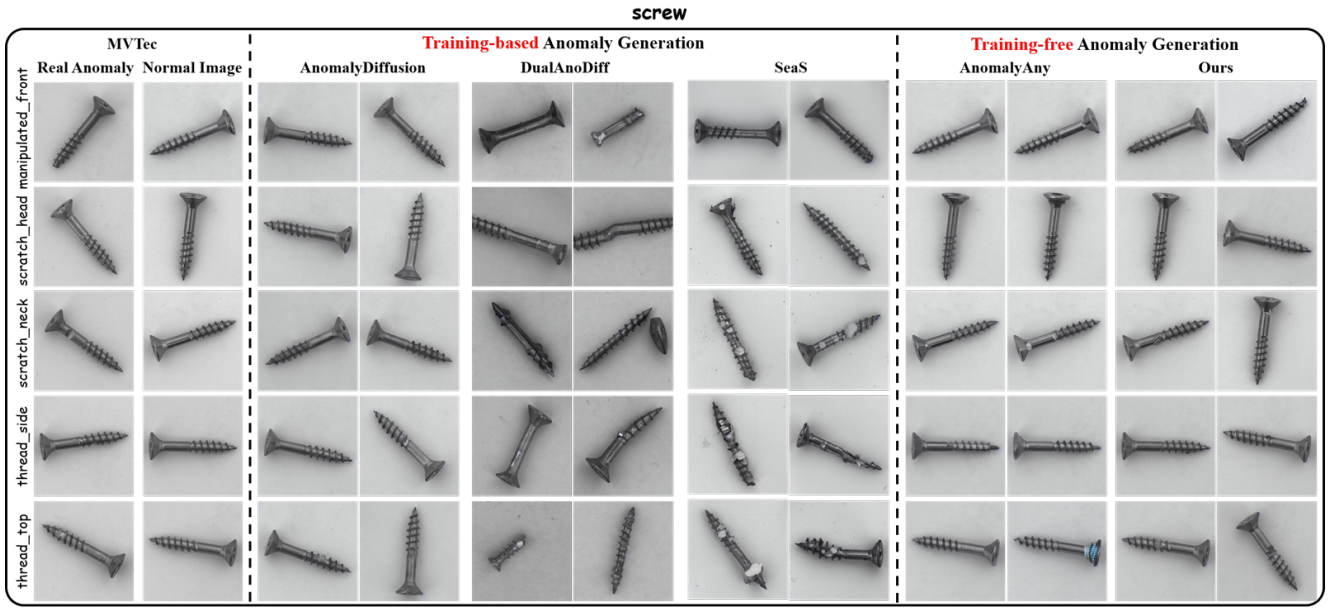

Figure 26. screw qualitative results on MVTec-AD.

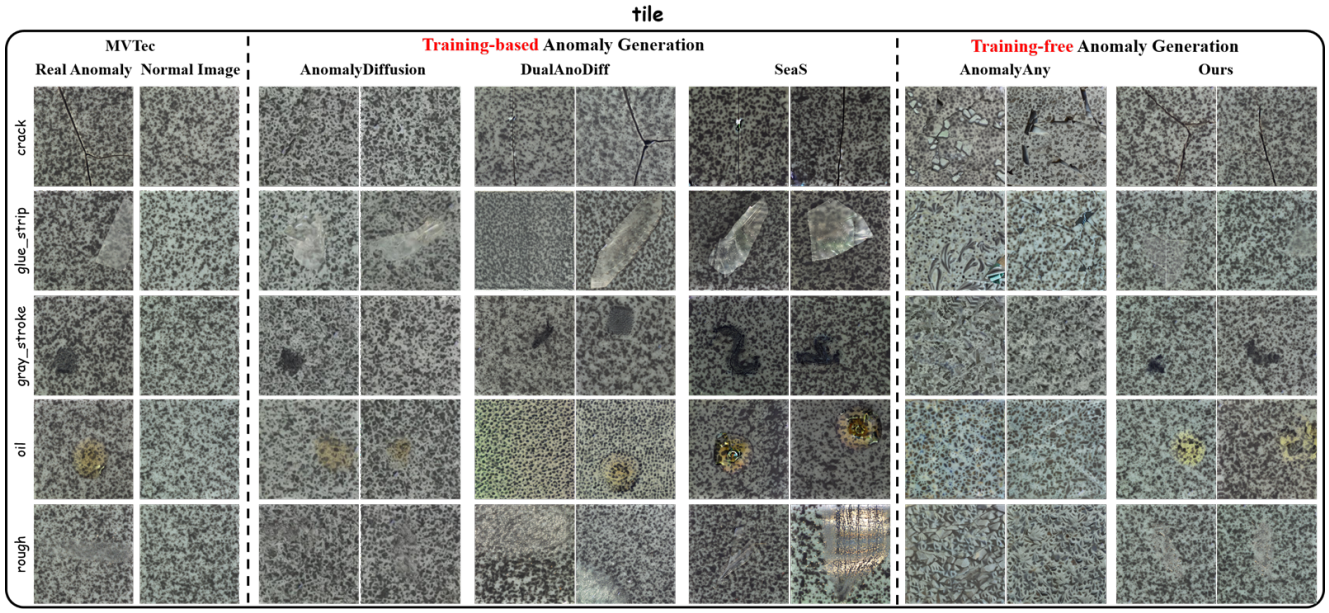

Figure 27. tile qualitative results on MVTec-AD.

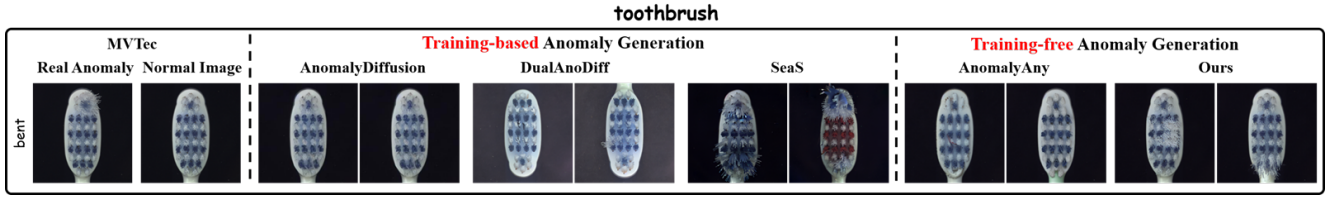

Figure 28. toothbrush qualitative results on MVTec-AD.

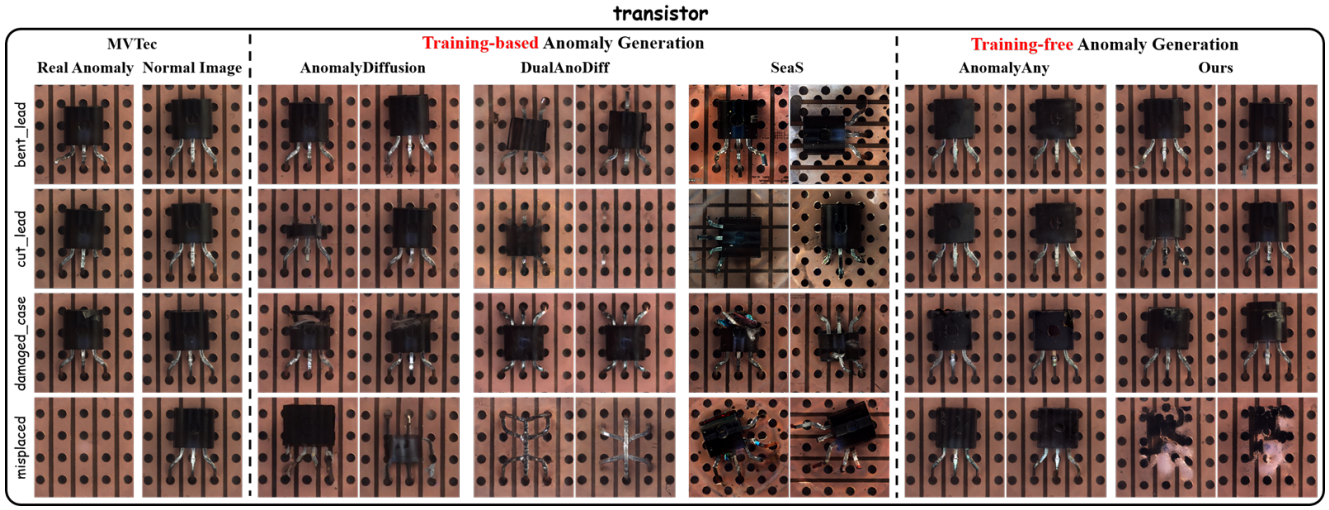

Figure 29. transistor qualitative results on MVTec-AD.

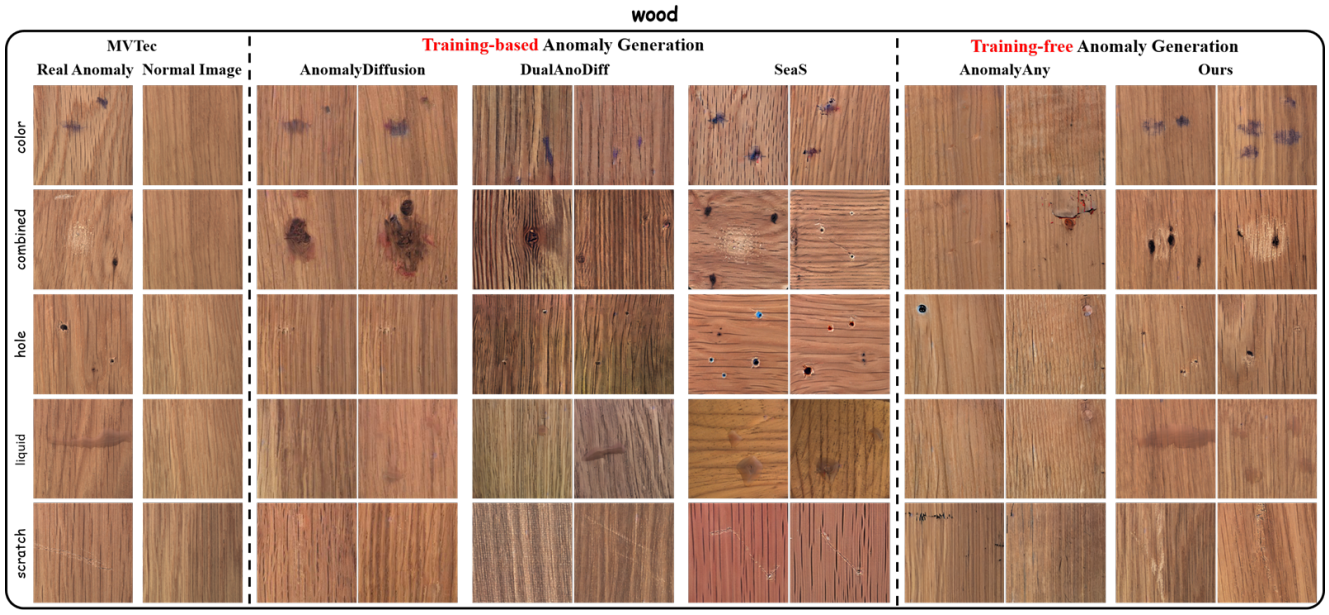

Figure 30. wood qualitative results on MVTec-AD.

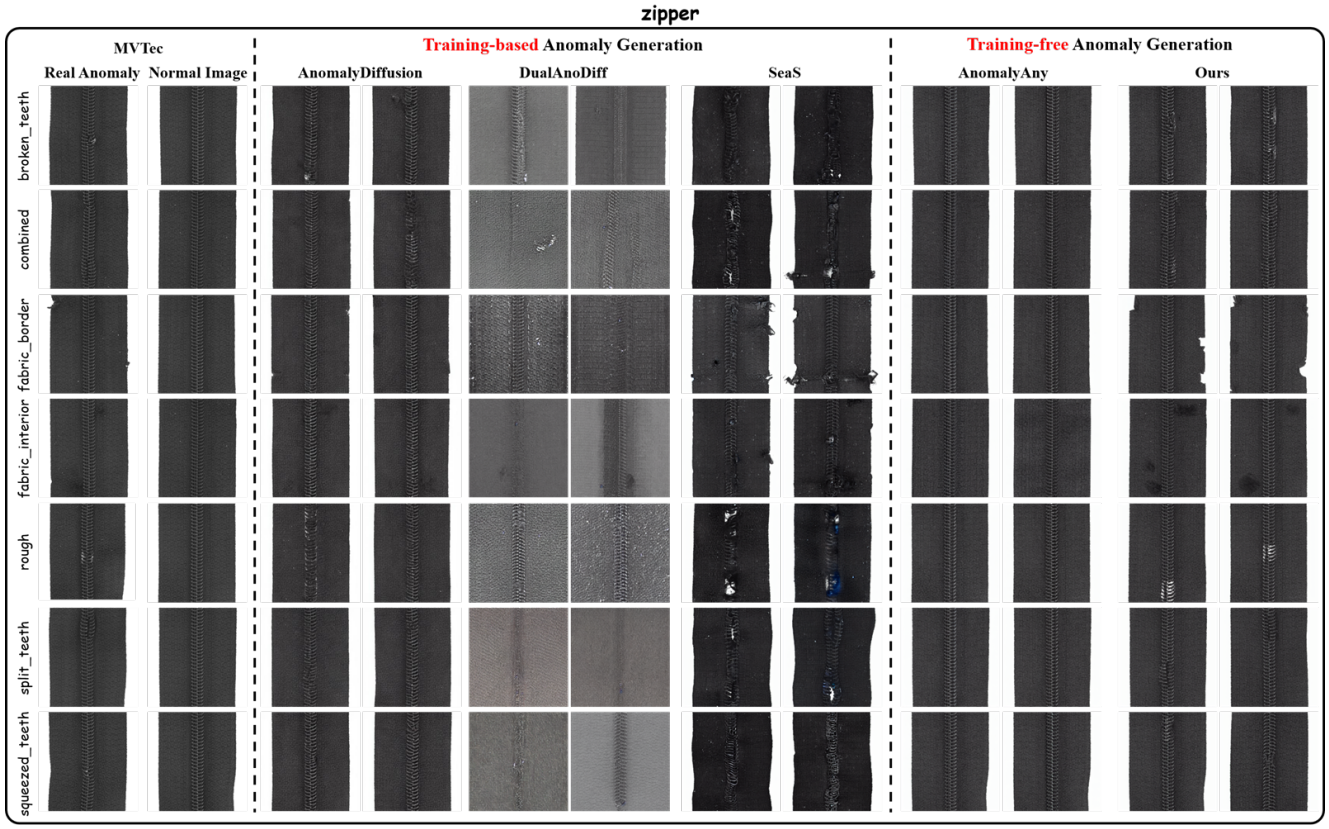

Figure 31. zipper qualitative results on MVTec-AD.