ANOMALYEDITOR: FEW-SHOT MASK-FREE IMAGE EDITING FOR HIGH-FIDELITY ANOMALY SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Realistic anomaly image synthesis can alleviate the scarcity of labeled anomaly data in industrial visual inspection. Existing reference-based methods require explicit anomaly masks, while global synthesis methods can compromise normal-region fidelity. Mask-free image editing offers a promising alternative, but its supervised adaptation requires spatially aligned normal–anomaly pairs, which are rarely available. We observe that such pairs can be constructed more readily in reverse. Adding an anomaly (forward: normal→anomaly) requires learning anomalous appearance from scarce examples, whereas removing an anomaly (reverse: anomaly→normal) can be formulated as inpainting guided by a normal prior learned from abundant normal images. Based on this observation, we propose AnomalyEditor, a few-shot reverse-to-forward framework for mask-free anomaly editing. Specifically, we first remove annotated anomalies from real images with an inpainter fine-tuned on normal images, then use the resulting aligned pairs to fine-tune a pretrained editor to add them back, guided by text instructions. As a result, AnomalyEditor learns from real anomalies how to add a specified anomaly type to an unseen normal image. During inference, AnomalyEditor generates a realistic anomaly without an input mask while leaving normal regions intact. Additionally, we introduce a spatial anomaly-attentive loss to strengthen supervision on small anomalies and severe-diverse anomaly generation to generate diverse anomalies of varying severity. Experiments on MVTec-AD, VisA, and MVTec-3D demonstrate improved Kernel Inception Distance and higher downstream anomaly classification accuracy than the evaluated synthesis baselines, alongside competitive detection and localization performance.

1 INTRODUCTION

Industrial visual inspection supports decisions on product rework, repair and disposal (Mokhtar et al., 2025), and informs root-cause analysis (Park et al., 2025). It involves anomaly detection, which identifies whether and where anomalies occur, and anomaly classification, which determines their types. While unsupervised detection methods exploit abundant normal samples (Batzner et al., 2024; Guo et al., 2025; Roth et al., 2022; Zhang et al., 2023a), classification remains challenging because few labeled examples exist for each anomaly type. Realistic anomaly synthesis potentially offers a direct way to alleviate this scarcity.

Nevertheless, existing diffusion-based synthesis methods face two major limitations. (i) Mask-conditioned synthesis (Hu et al., 2024; Gui et al., 2024; Song et al., 2025; Rao et al., 2026) requires an explicit target anomaly mask. When masks are generated independently of the input image, they can conflict with the object's structure and produce implausible anomalies (Fig. 1(a)). (ii) Global synthesis (Jin et al., 2025; Dai et al., 2025) generates entire images without a normal reference, often compromising normal-region fidelity and creating a real-synthetic domain gap (Fig. 1(b)).

Above limitations motivate a method that learns where and how to introduce an anomaly from a normal reference and a textual instruction without prescribing its region at inference. While mask-free editors provide this interface, their adaptation to industrial anomaly types requires spatially aligned normal–anomaly pairs. Without alignment, the desired edits are entangled with differences in object pose, illumination and background. Such pairs are rarely available, as collecting them requires imaging the same product instance in normal and anomalous states under identical conditions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

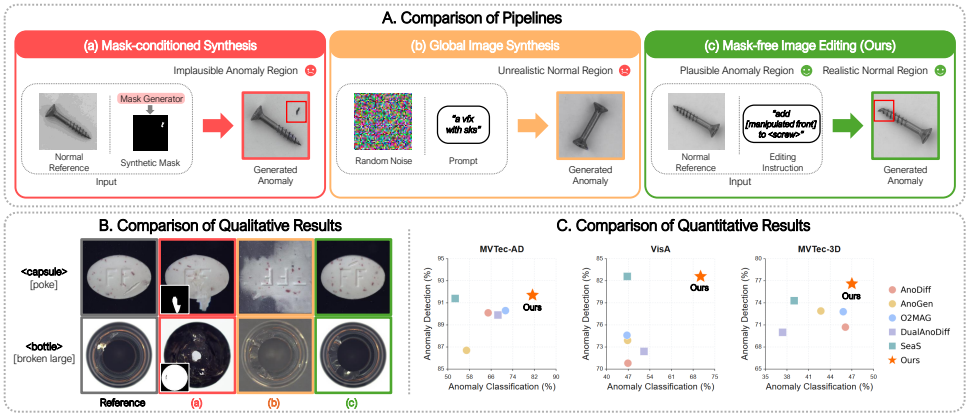


Figure 1: **Comparison of anomaly generation approaches. A.** (a) Mask-conditioned synthesis can produce implausible anomalies, while (b) Global synthesis compromises normal-region fidelity. (c) **Mask-free image editing (ours)** generates realistic anomalies while preserving normal regions. **B.** Qualitative comparisons. **C.** Our method achieves consistently strong downstream classification and detection performance across three datasets. Classification is measured by top-1 accuracy while detection scores average the AUC-I, AP-I, F1-I, AUC-P, AP-P, F1-P and PRO-P metrics.

To address this problem, we propose AnomalyEditor, a few-shot mask-free anomaly editing framework based on reverse-to-forward training. Our key insight is that, in the few-shot setting, reverse generation (anomaly→normal) is more tractable than forward generation (normal→anomaly) for constructing aligned pairs. While forward generation must learn anomalous patterns from scarce examples, reverse generation can be formulated as inpainting guided by surrounding context and a normal prior learned from abundant normal samples. AnomalyEditor therefore inpaints annotated anomaly regions to obtain spatially aligned pseudo-normal counterparts, then fine-tunes a pretrained editor on these pairs with real anomalous images as targets. At inference, the editor takes only a normal image and a textual instruction, without a supplied or separately generated anomaly mask.

Beyond pair construction, small anomalies can be overwhelmed by normal-region losses, and few-shot examples may offer limited severity variation. Hence, we introduce a spatial anomaly-attentive loss and anomaly intensity diversification, complemented by edit region-aware sampling to filter non-edit failures. Overall, our contributions are threefold: (i) we propose reverse-to-forward training to construct aligned pseudo-normal and anomaly pairs for mask-free editing, (ii) our loss and diversification strategy strengthen small-anomaly supervision and vary anomaly severity, and (iii) experiments on MVTec-AD (Bergmann et al., 2019), VisA (Zou et al., 2022) and MVTec-3D (Bergmann et al., 2021) demonstrate improved KID and downstream anomaly classification accuracy over the evaluated baselines without compromising detection and localization performance.

2 Related Work

2.1 Anomaly Image Generation

Mask-conditioned synthesis. Mask-conditioned methods synthesize anomalies on a normal reference using an explicit target mask for spatial control. AnomalyDiffusion (Hu et al., 2024) learns mask and anomaly generation separately, while AnoGen (Gui et al., 2024) uses masks derived from bounding boxes. DefectFill (Song et al., 2025) builds on an inpainting-specific prior to generate anomaly textures within supplied masks. These methods differ in how they obtain spatial conditions and model anomaly appearance. O2MAG (Rao et al., 2026) instead transfers features from a reference anomaly through attention grafting without model fine-tuning, using a target mask to control anomaly placement. Across these approaches, the anomaly region is supplied as a generation condition rather than inferred solely from the normal reference and instruction.

Global image synthesis. Global approaches model normal content and anomalous regions jointly, without conditioning on a particular normal reference. DualAnoDiff (Jin et al., 2025) coordinates

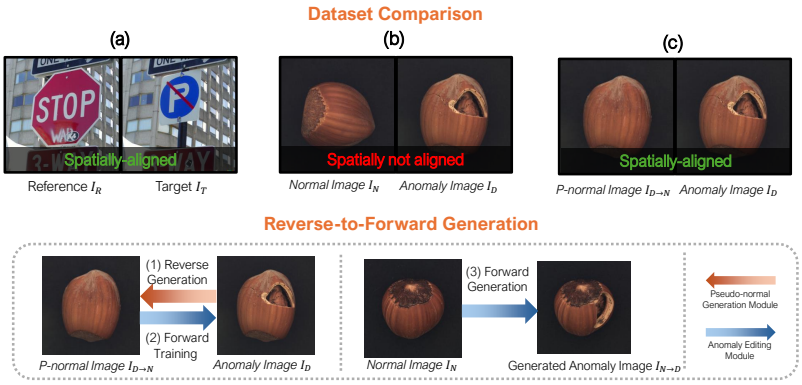


Figure 2: **Motivation and overview of our reverse-to-forward paradigm. (Top)** Image editing models require spatially (background)-aligned pairs (a), yet industrial settings often only provide unpaired normal and anomaly images (b). Our proposed reverse generation converts anomaly images into pseudo-normal counterparts, yielding spatially-aligned pairs (c). **(Bottom)** During *training*, (1) a reverse generation module produces pseudo-normal images from anomaly inputs, and (2) the resulting spatially aligned pairs are used to train the forward editing model. At *inference*, (3) the trained model generates realistic anomaly images from user-provided normal inputs.

coupled diffusion branches through attention-based feature sharing, whereas SeaS (Dai et al., 2025) introduces multiple anomaly tokens and a unified training approach for different anomaly types. These methods learn to synthesize complete anomalous images, in contrast to reference-based approaches that modify an existing normal image.

2.2 Image Editing

Image editing modifies a reference image according to specified conditions while preserving content unrelated to the requested change. Visually-guided approaches use masks or other spatial cues to control generation and editing (Lugmayr et al., 2022; Ju et al., 2024; Zhang et al., 2022; Pan et al., 2023; Zhang et al., 2023b). Instruction-guided editors instead infer the desired modifications from a reference image and textual instructions without requiring an explicit edit mask.

InstructPix2Pix (Brooks et al., 2023) demonstrated diffusion-based instruction following for image editing. More recent methods (Batifol et al., 2025; Wu et al., 2025a) adopt rectified flow (Liu et al., 2022b) and advance reference-preserving editing. Despite these model developments, obtaining spatially aligned source–target pairs remains an important challenge for supervised editor training (Chen et al., 2025; Winter et al., 2024).

Anomaly generation methods either need an anomaly mask or generate the entire image, which can lower normal-region fidelity. Instruction-guided editors can avoid both issues but require spatially aligned normal–anomaly pairs for training, which is rarely available for industrial anomalies. In this work, we note that anomalies can be removed without paired data by an inpainting model fine-tuned only on normal images. Thus, we remove annotated anomalies from a few real images to obtain aligned pairs, from which an editor learns anomaly appearance and location. This allows instruction-guided editors to be used for few-shot anomaly synthesis.

3 Preliminaries

3.1 Rectified Flow

Rectified flow (Liu et al., 2022b) is a type of generative model which, unlike DDPM (Ho et al., 2020), that performs probabilistic denoising, directly learns the vector field along a straight path from noise to data. In this framework, given a training data sample x (in our case, an anomaly image) and Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, the state at continuous time $t \in [0, 1]$ is defined by the linear interpolation $\mathrm{x}_t = t\mathrm{x} + (1-t)\epsilon$. Given conditioning information c, the model $V_\theta(\mathrm{x}_t, t, c)$

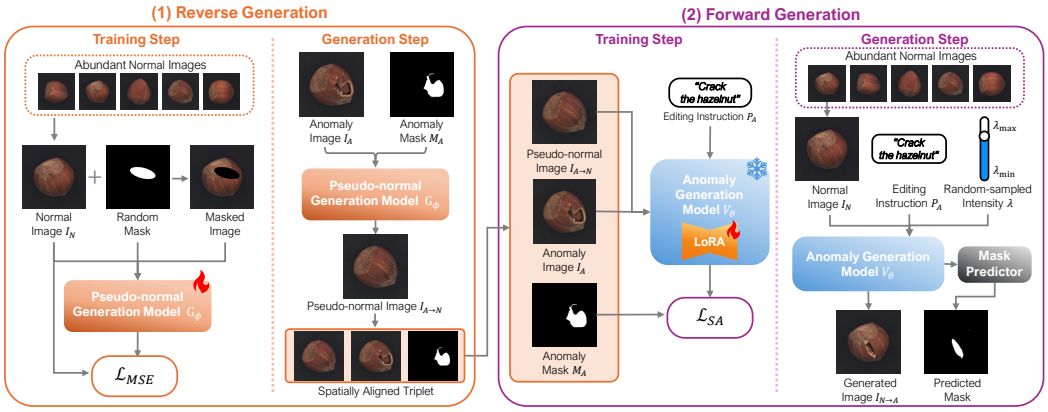


Figure 3: **Training and generation pipeline of AnomalyEditor. (1) Reverse generation:** An inpainting model trained on normal images with random masks converts real anomaly images into pseudo-normal images, yielding spatially aligned normal–anomaly pairs and anomaly masks. **(2) Forward generation:** These pairs and masks enable LoRA fine-tuning of an instruction-guided editing model with spatial anomaly-attentive loss $\mathcal{L}_{SA}$. At inference, randomly sampled guidance strengths diversify anomaly severity, generating anomaly images with corresponding masks.

learns to regress the target velocity $(\mathrm{x} - \epsilon)$ by minimizing $\mathcal{L} = \mathbb{E}_{\mathrm{x}_t,\epsilon}[|(\mathrm{x} - \epsilon) - V_\theta(\mathrm{x}_t, t, c)|^2]$, where $t \sim \mathcal{U}(0, 1)$. At inference, we map noise to data by numerically integrating V_θ from $t = 0$ to $t = 1$.

3.2 Negative Prompting

Negative prompting (Liu et al., 2022a; Ban et al., 2024; Sridhar & Vasconcelos, 2024; Gandikota et al., 2024) is a guidance method for conditional generation that promotes content specified by a positive prompt while suppressing content specified by a negative prompt. When applied to rectified flow, negative prompting combines a positive-conditioned velocity $V_+ = V_\theta(x_t, t, c_+)$, which guides generation toward samples consistent with the positive prompt, and a negative-conditioned velocity $V_- = V_\theta(x_t, t, c_-)$, which guides generation toward samples consistent with the negative prompt. Here, c_+ and c_- denote the positive and negative prompt conditions, respectively. The guided velocity is $V_G = V_- + \lambda \cdot (V_+ - V_-)$, where $V_+ - V_-$ defines the guidance direction and λ controls its strength.

3.3 Mask-free Image Editing Models

Mask-free image editing models (Brooks et al., 2023; Batifol et al., 2025; Wu et al., 2025a) perform local edits conditioned on text instruction and a reference image, while preserving the consistency of non-edit regions. For efficiency, image editing is carried out in a latent space. The target image I_T is mapped by an image encoder $\mathcal{E}$ to a latent representation $z = \mathcal{E}(I_T)$, and then reconstructed by a decoder $\mathcal{D}$ as $I_T \approx \mathcal{D}(z)$. The conditioning information consists of the text prompt P and the reference image I_R. These are embedded by a text encoder $\mathcal{T}$ and an image encoder $\mathcal{E}$, respectively, yielding $c = \mathcal{T}(P), \mathcal{E}(I_R)$. Training follows Sec. 3.1. For a randomly sampled continuous time t, we linearly interpolate z to obtain z_t and train the model $V_\theta(z_t, t, c)$ to regress the target vector from z_t toward z_1 under condition c. In practice, training pairs are constructed so that the reference image I_R and the target image I_T share background consistency, and the model is trained to implicitly infer and modify the edited regions while reconstructing non-edited regions to match I_R.

4 Proposed Method

As shown in Fig. 2, our mask-free anomaly image editing framework follows a *reverse-to-forward training paradigm*. The reverse step generates spatially aligned pseudo-normal images from real anomaly images (anomaly-to-normal). The forward step then uses the resulting pairs and placeholder-based instructions to fine-tune the anomaly editing model (normal-to-anomaly). We

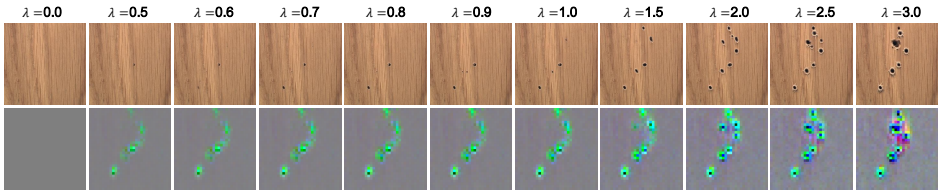


Figure 4: **Effect of anomaly severity-diverse anomaly generation.** Generated images and PCA visualizations of the scaled anomaly guidance direction show that increasing λ amplifies anomaly intensity and spatial extent, with limited changes to normal regions.

further introduce a *spatial anomaly-attentive loss* to strengthen learning signals from small anomalous regions during training and a *severity-diverse anomaly generation* to broaden the severity range of generated anomalies during inference.

4.1 Reverse (anomaly-to-normal) training and generation

We perform reverse generation (anomaly-to-normal) to convert anomaly images into pseudo-normal ones, obtain spatially-aligned pseudo-normal–anomaly pairs, and then use these pairs to train the model. Therefore, a key objective is to construct editing pairs $(I_{A\to N}, I_A)$ by generating pseudo-normal images $I_{A\to N}$ from real anomaly images I_A.

For this purpose, we first train a pseudo-normal generation module G_ϕ that captures knowledge of the texture and structure of normal objects. To implement this module, we build on a pre-trained Flux-Fill inpainting model (Black Forest Labs, 2024) that takes an image and a mask as input and inpaints the masked region. As shown on the left hand side of Fig. 3(1), we fine-tune this model using abundant normal samples and random masks. During this training process, G_ϕ is trained to restore the masked regions to their original normal textures, thereby acquiring the ability to generate realistic domain-specific surface. Once the reverse generation (normal context-aware) model G_ϕ is trained, we feed real anomaly images I_A and their corresponding masks M_A into this module as shown on the right side of Fig. 3(1). G_ϕ then inpaints the masked region M_A with normal textures and generates a pseudo-normal image $I_{A\to N} = G_\phi(I_A, M_A)$, yielding a set of spatially-aligned image editing pairs $(I_{A\to N}, I_A)$.

4.2 Forward (normal-to-anomaly) training

In the forward generation step, we train the anomaly image editor V_θ using the spatially-aligned pairs obtained from Sec. 4.1. This module is trained to generate anomaly images from a set of inputs comprising a normal image and instruction P_A. Specifically, we construct P_A using the template *"add [anomaly] to <object>"*, where the placeholders are filled with anomaly class and object category annotations (*e.g.*, *"add crack to hazelnut"*). For effective fine-tuning of the model, we propose a spatial anomaly-attentive loss to amplify learning signals from small anomaly regions during training which is later illustrated in this section. Using the spatially-aligned pairs and instructions, we obtain the complete training tuples $\{(I_{A\to N}, I_A), M_A, P_A\}$ needed for forward training.

Training with spatial anomaly-attentive loss. We fine-tune the editing model V_θ using the pseudo-normal image $I_{A\to N}$ generated from the reverse step as the reference and the real anomaly image I_A as the target. The training process follows the mechanism of Rectified flow described in Sec. 3.1. However, small anomaly regions contribute little to the overall loss and can be overlooked during fine-tuning, impairing the generation of subtle anomalies (*e.g.*, screw in Fig. 1).

To address this, we introduce the spatial anomaly-attentive loss $\mathcal{L}_{SA}$, which upweights velocity residuals in anomaly regions according to their relative area. Given a binary anomaly mask $M_A \in 0, 1^{H\times W}$, we compute the anomaly area ratio $\rho = \frac{1}{HW}\sum_{i,j} M_{A,ij}$ and define $M_w = (\mathbf{1} - M_A) + \rho^{-1} M_A$. This map scales anomaly-region squared velocity errors by $1/\rho$, counteracting their small

spatial contribution while retaining normal-region supervision. The resulting training objective is

$$\mathcal{L}_{SA} = \mathbb{E}_{z_t,t,\epsilon}\left[\sum_{i,j} M_{w,ij}\left\|\big((z-\epsilon) - V_\theta(z_t,t,c)\big)_{ij}\right\|_2^2\right] \tag{1}$$

We emphasize that the anomaly mask M_A is used only during training and that no anomaly mask is required at inference time. An ablation study on $\mathcal{L}_{SA}$ is provided in Sec. 5.3.

4.3 Forward (normal-to-anomaly) generation

Given a normal image I_N and an anomaly-inducing instruction P_A (*e.g.*, *"add crack to hazelnut"*), the trained editing model generates an anomaly image. However, the limited few-shot training samples may not fully capture the range of anomaly severity encountered in practice.

Severity-diverse anomaly generation. To generate anomalies of varying severity, we control anomaly intensity through guidance and randomly sample the guidance scale for each image. Inspired by negative prompting described in Sec. 3.2, we set the negative prompt to an identity-preserving instruction P_N, *"keep the image unchanged"*. Specifically, the model V_θ predicts two velocities conditioned on the same normal image I_N: one under the anomaly instruction P_A and the other under the identity-preserving instruction P_N. The corresponding conditions are $c_A = \{\mathcal{T}(P_A), \mathcal{E}(I_N)\}$ and $c_N = \{\mathcal{T}(P_N), \mathcal{E}(I_N)\}$, corresponding to positive and negative prompts respectively. We combine these predictions with guidance scale λ as follows:

$$V_G = V_\theta(z_t,t,c_N) + \lambda\big(V_\theta(z_t,t,c_A) - V_\theta(z_t,t,c_N)\big). \tag{2}$$

The difference between the two predictions serves as the *anomaly guidance direction*, and λ controls how strongly this direction is applied. At $\lambda = 1$, V_G equals the anomaly-conditioned prediction. Fig. 4 shows that increasing λ gradually strengthens and enlarges anomalies while largely preserving normal regions. PCA visualizations of the scaled anomaly guidance direction also show spatial patterns corresponding to the defects. We sample $\lambda \sim \mathcal{U}(\lambda_{\min}, \lambda_{\max})$ once per image to generate synthetic training data covering a broader severity range without additional training. We evaluate this diversification in Sec. 5.3.

Edit region-aware sampling. Mask-free image editors may occasionally leave input images nearly unchanged despite editing instructions (Brack et al., 2024; Yu et al., 2025). In anomaly synthesis, such non-edit samples can introduce label noise in downstream training. To identify potential non-edit samples, we use the area of an anomaly mask predicted from the editor's internal features. If this area falls below a threshold defined relative to the average training mask area of the corresponding anomaly class, we re-edit the original normal image (see Sec. 5 for details).

5 Experiments

Datasets. We evaluate experiments on the MVTec-AD (Bergmann et al., 2019), VisA (Zou et al., 2022) and MVTec-3D (Bergmann et al., 2021) datasets. MVTec-AD comprises 15 categories with a total of 73 anomaly classes, VisA includes 12 categories and 75 anomaly classes, and MVTec-3D includes 10 categories with 41 anomaly classes. Following the standard evaluation protocol (Hu et al., 2024), we allocated one-third of the anomaly images in each anomaly class to the train set and the remainder to the test set such that no overlapping images exist between the two sets.

Evaluation metrics. We evaluate distributional fidelity using KID (Sutherland et al., 2018) and perceptual diversity using IC-LPIPS (Ojha et al., 2021). For anomaly classification, we train a ResNet-34 (He et al., 2016) following the protocol of AnomalyDiffusion (Hu et al., 2024) and report top-1 accuracy. Further details are provided in the Appendix.

Implementation details. We use FLUX.1-Fill-dev (Black Forest Labs, 2024) for the reverse model and FLUX.1-Kontext-dev (Batifol et al., 2025) for the forward model. Both models are fine-tuned using LoRA (Hu et al., 2022) applied to the attention and feed-forward projections of the

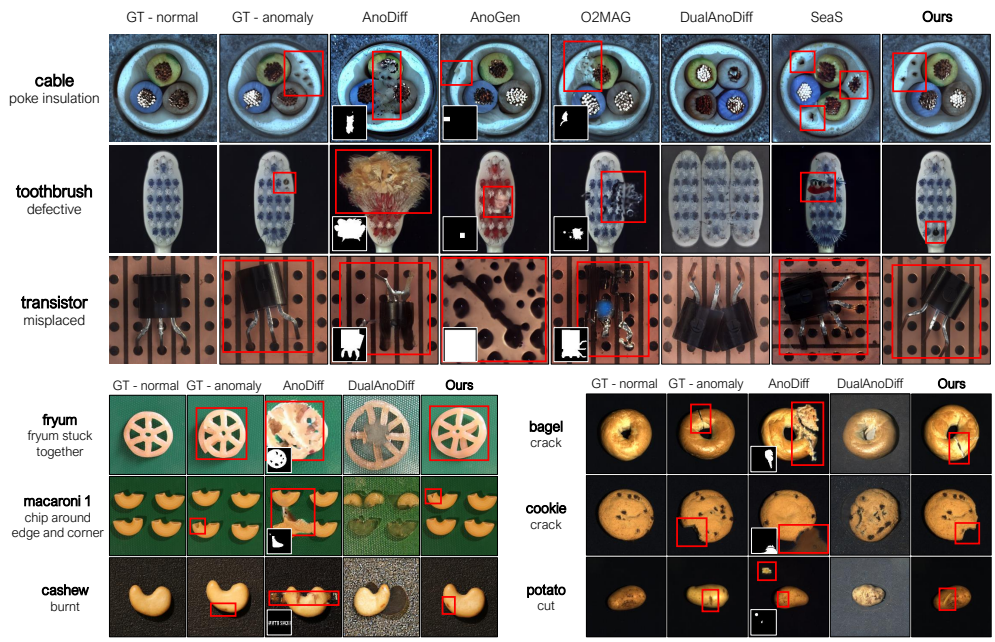


Figure 5: **Qualitative comparison of generated anomalies on MVTec-AD (Top), VisA (Bottom-left) and MVTec-3D (Bottom-right).** Red boxes indicate anomaly regions.

Table 1: Comparison of average KID and IC-LPIPS scores across three datasets. **Bold** and underlined indicate the best and second-best results, respectively. MCS: Mask-conditioned synthesis. GIS: Global image synthesis. MFIE: Mask-free image editing.

Type	Method	MVTec-AD		VisA		MVTec-3D		Average	
		KID ↓	IC-L ↑	KID ↓	IC-L ↑	KID ↓	IC-L ↑	KID ↓	IC-L ↑
MCS	AnoDiff	104.01	0.30	110.76	0.30	70.54	0.26	95.10	0.29
	AnoGen	105.39	0.31	106.41	0.30	73.10	0.26	94.97	0.29
	O2MAG	48.90	0.30	90.10	0.34	40.58	0.30	59.86	0.31
GIS	DualAnoDiff	96.82	**0.36**	152.64	**0.43**	143.20	**0.37**	130.89	**0.39**
	SeaS	126.59	0.35	97.00	0.23	101.42	0.33	108.34	0.30
MFIE	**Ours**	**30.55**	0.28	**70.78**	0.33	**40.13**	0.24	**47.15**	0.28

Transformer, with rank and alpha both set to 16. Both models use the Prodigy (Mishchenko & Defazio, 2023) optimizer with an initial learning rate of 1.0, a constant learning rate schedule without warmup, BF16 mixed precision, and an batch size of 4. The reverse model is trained per object class for 5,000 iterations, and the forward model per anomaly class for 2,000 iterations. Computational costs are reported in Appendix B.3. For the forward model, excluding the first 500 iterations, we sample 10 images every 100 iterations, compute KID against the training set, and use the best iteration to generate final images. At inference time, we generate images using 10 inference steps, with $\lambda_{\min} = 0.5$ and $\lambda_{\max} = 3.0$ for severity-diverse anomaly generation. Following prior work that derive anomaly masks from internal representations (Dai et al., 2025; Duan et al., 2023; Wu et al., 2025b), we predict initial anomaly masks from internal DiT features and refine them using the Mask Refinement Module (MRM) of SeaS (Dai et al., 2025) (See Appendix. A.4 for more details). To reduce non-edit samples, we re-edit the original normal image if the refined anomaly mask area is less than 20% of the average training mask area for the corresponding anomaly class. We allow up to two re-editing attempts after the initial generation. Using this procedure, we collect 500 anomaly images satisfying the area criterion per anomaly class for downstream tasks.

Baselines. We adopt AnomalyDiffusion (Hu et al., 2024), AnoGen (Gui et al., 2024) and O2MAG (Rao et al., 2026) as mask-conditioned baselines, and DualAnoDiff (Jin et al., 2025) and SeaS (Dai et al., 2025) as global image synthesis baselines, and compare all five in terms of generation quality and classification using their official implementations.

Table 2: Comparison of classification accuracies (%) across three datasets.

Type	Method	MVTec-AD	VisA	MVTec-3D	Average
MCS	AnoDiff	64.90	42.86	46.07	51.28
	AnoGen	56.92	46.75	42.68	48.78
	O2MAG	71.27	46.60	45.83	54.6
GIS	DualAnoDiff	68.50	52.14	37.41	52.68
	SeaS	52.73	36.55	39.52	42.76
MFIE	**Ours**	**81.34**	**70.44**	**46.99**	**66.26**

Table 3: Comparison of image-level anomaly detection results across three datasets.

Type	Method	MVTec-AD			VisA			MVTec-3D			Average		
		AUC-I	AP-I	F1-I	AUC-I	AP-I	F1-I	AUC-I	AP-I	F1-I	AUC-I	AP-I	F1-I
MCS	AnoDiff	98.5	**99.4**	**97.8**	85.0	81.2	78.1	86.1	94.2	91.0	89.9	91.6	88.9
	AnoGen	98.3	99.2	97.0	89.6	85.6	81.3	88.1	95.1	90.9	92.0	93.3	89.7
	O2MAG	97.9	99.1	97.2	92.4	90.4	84.4	91.0	96.6	92.1	93.8	95.4	91.2
GIS	DualAnoDiff	97.0	98.7	96.0	86.5	82.1	75.5	80.8	91.6	88.9	88.1	90.8	86.8
	SeaS	**98.7**	99.3	97.3	88.6	87.4	81.5	87.2	95.1	91.2	91.5	94.0	90.0
MFIE	**Ours**	98.4	99.3	97.5	**93.3**	**91.6**	**85.6**	**93.3**	**97.5**	**93.2**	**95.0**	**96.1**	**92.1**

Table 4: Comparison of pixel-level anomaly detection results across three datasets.

Type	Method	MVTec-AD				VisA				MVTec-3D				Average			
		AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P
MCS	AnoDiff	98.4	74.0	70.1	92.6	96.0	44.5	44.9	84.1	97.3	15.2	19.9	91.3	97.2	44.6	45.0	89.3
	AnoGen	97.2	68.7	65.0	90.2	97.3	45.4	46.9	87.4	98.3	20.0	26.5	91.3	97.6	44.7	46.1	89.7
	O2MAG	98.4	74.6	71.4	93.9	93.5	36.5	41.5	83.8	98.3	15.8	21.4	95.0	96.7	42.3	44.8	90.9
GIS	DualAnoDiff	97.4	76.8	72.9	90.8	97.0	49.5	51.0	86.7	97.3	16.8	24.6	89.7	97.2	47.7	49.5	89.1
	SeaS	98.3	78.3	73.7	94.4	**97.9**	**69.6**	**67.6**	85.9	97.7	**25.9**	30.7	92.6	98.0	**57.9**	**57.3**	91.0
MFIE	**Ours**	**98.8**	**78.6**	**74.7**	**94.7**	97.0	60.2	60.7	**90.0**	**99.2**	25.8	**32.0**	**95.4**	**98.3**	54.9	55.8	**93.4**

5.1 Comparison on anomaly generation results

Quantitative results. Tab. 1 presents a quantitative comparison of generated anomaly images. AnomalyEditor achieves the lowest KID scores across all datasets compared to over all baselines. We attribute this to the core property: unlike synthesis-based methods that generate images from scratch, our editing approach preserves realistic backgrounds, producing outputs that more closely match the true data distribution. Some baselines achieve higher IC-LPIPS scores, but perceptual diversity can increase this metric even when the generated images are not realistic. These scores should therefore be considered alongside distributional fidelity and downstream task performance when assessing generation quality (see Appendix A.7 for further discussion).

Qualitative results. Fig. 5 reveals specific failure modes of existing methods. mask-conditioned synthesis methods (AnoDiff, AnoGen, O2MAG) produce mask misalignment artifacts. Notably, in toothbrush (Fig. 5, top), AnoDiff transforms brush bristles into golden fur, because inaccurate synthetic masks lead to unrealistic filling of anomaly regions. Global image synthesis methods (DualAnoDiff, SeaS) generate physically implausible results: objects are replicated in transistor and toothbrush (Fig. 5, top), or backgrounds are completely altered in macaroni (Fig. 5, bottom-left) and bagel (Fig. 5, bottom-right). In contrast, our mask-free editing approach avoids these failure modes: by editing a reference image rather than synthesizing from scratch, the normal region is preserved, and by implicitly inferring the edit region, mask misalignment is mitigated. More results are in the Appendix.

5.2 Comparison on anomaly downstream tasks

Anomaly classification. To evaluate the utility of generated images, we trained ResNet-34 classifiers and compared against AnomalyDiffusion, AnoGen, O2MGA DualAnoDiff, and SeaS as in prior study (Hu et al., 2024). Toothbrush (MVTec-AD) and capsules (VisA) are excluded as they contain only a single anomaly class. Tab. 2 show classification accuracy across three datasets. Our method achieves 81.34% average accuracy on MVTec-AD, outperforming the previous state-of-the-art (SOTA) approach by 10.1%p. On VisA, we achieve 70.44%, showing a margin of 18.3%p. Overall, these results demonstrate that our approach delivers robust high performance across all three datasets at classification.

Table 5: Ablation studies on MVTec-AD dataset. SAG denotes severity-diverse anomaly generation. Without SAG, $\lambda = 1.0$; the full model samples $\lambda \sim \mathcal{U}(0.5, 3.0)$.

(a) Reverse-to-forward ablation.

Setting	Rev.	For.	Clf. Acc. (%)
Base editor	✗	✗	35.40
Forward only	✗	✓	<u>58.34</u>
Reverse + Forward	✓	✓	**81.34**

(b) Backbone ablation.

Backbone	# Params	Clf. Acc. (%)
Flux.2 Klein	4B	<u>79.62</u>
Flux.1 Kontext-dev	12B	**81.34**

(c) Component ablation.

Method	Clf. Acc. (%)
Ours *w/o* $\mathcal{L}_{SA}$	<u>78.69</u>
Ours *w/o* SAG	76.80
Ours	**81.34**

Anomaly detection. We evaluated anomaly detection performance using the same baseline methods (AnoDiff, AnoGen, O2MAG, DualAnoDiff, and SeaS). Tab. 3 and Tab. 4 present the quantitative results for image-level anomaly detection and pixel-level anomaly localization, respectively. For image-level detection, our method outperforms the best-performing baselines on average across the three datasets, increasing AUC-I by 1.2%p, AP-I by 0.7%p, and F1-I by 0.9%p. For pixel-level localization, our approach achieves state-of-the-art (SOTA) performance on average, improving AUC-P by 0.3%p and PRO-P by 2.4%p, while obtaining competitive second-best results in AP-P (54.9%) and F1-P (55.8%). Overall, these results demonstrate that our approach delivers robust high performance across all three datasets at both image and pixel levels.

5.3 Ablation study

Ablation on reverse-to-forward training. Without reverse generation, the training for forward generation can only use unaligned normal/anomalous images, where anomaly changes are entangled with pose, illumination, and background differences, making localized counterfactual editing difficult. Tab. 5 (c) isolates this effect. Reverse+forward training improves accuracy by 23.0%p over forward-only and 45.94%p over the base editor (Flux.1-Kontext) alone, while greatly reducing KID. Thus, the main gain does not come from base editor.

Ablation on editing backbone. We also evaluate the robustness of our approach to the choice of the base editing backbone. As shown in Tab. 5 (b), replacing the editor with a smaller backbone results in only 1.72%p drop, and still yields higher classification accuracy than the previous state-of-the-art (DualAnoDiff (Jin et al., 2025)). This suggests our gains are not tied to a particular editor.

Ablation on other components. Tab. 5(c) shows that removing the spatial anomaly-attentive loss or severity-diverse anomaly generation reduces classification accuracy from 81.34% to 78.69% and 76.80%, respectively. We additionally tested fixed severity scales of $\lambda = 0.5$ and 3.0, obtaining accuracies of 66.83% and 70.27%, both below the 76.80% obtained at $\lambda = 1.0$. This suggests that the benefit of severity-diverse generation comes from sampling across severity scales rather than simply setting a fixed scale.

6 Conclusion

In this work, we presented AnomalyEditor, a few-shot framework for learning mask-free anomaly editing through reverse-to-forward training. By restoring annotated anomaly regions using a normal-domain inpainter, we constructed spatially aligned pseudo-normal–anomaly pairs while retaining real anomalous images as targets. This provided the supervision needed to learn localized anomaly transformations from scarce examples. Spatial anomaly-attentive supervision emphasized small defects, complemented by anomaly intensity diversification for varying synthesis severity and edit region-aware sampling for filtering non-edit failures. Experiments on MVTec-AD, VisA, and MVTec-3D demonstrated improved Kernel Inception Distance and higher downstream anomaly classification accuracy than the evaluated synthesis baselines while also showing competitive detection and localization performance. Importantly, ablations showed substantial improvements over both the base editor and forward-only fine-tuning, highlighting the value of reverse-constructed supervision beyond the pretrained editing capability. Together, these findings support using abundant normal data to construct aligned supervision from scarce real anomalies, enabling the editor to learn anomaly appearance and placement without prescribing a target region at inference.

7 AI USAGE STATEMENT

In this work, we used generative AI tools for research ideation, initial literature retrieval and discovery, drafting initial sections of the manuscript, and refining and polishing the written text. We have not used generative AI tools for generating synthetic datasets or proving mathematical claims. Additionally, we used generative AI tools for formatting LaTeX tables and code snippets for experiments. We have reviewed all AI-assisted work. Specifically, all research ideas and literature suggested by the AI were manually verified through literature search to prevent hallucination, initial drafted texts were substantially edited and revised by the authors, and all claims, analyses, and code were rigorously checked for technical correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Yuanhao Ban, Ruochen Wang, Tianyi Zhou, Minhao Cheng, Boqing Gong, and Cho-Jui Hsieh. Understanding the impact of negative prompts: When and how do they take effect? In *European Conference on Computer Vision*, pp. 190–206. Springer, 2024.

Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv e-prints*, pp. arXiv–2506, 2025.

Kilian Batzner, Lars Heckler, and Rebecca König. Efficientad: Accurate visual anomaly detection at millisecond-level latencies. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 128–138, 2024.

Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9592–9600, 2019.

Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization. *arXiv preprint arXiv:2112.09045*, 2021.

Black Forest Labs. Flux.1-fill-dev. `https://huggingface.co/black-forest-labs/FLUX.1-Fill-dev`, 2024.

Manuel Brack, Felix Friedrich, Katharia Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8861–8870, 2024.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

Xi Chen, Zhifei Zhang, He Zhang, Yuqian Zhou, Soo Ye Kim, Qing Liu, Yijun Li, Jianming Zhang, Nanxuan Zhao, Yilin Wang, et al. Unireal: Universal image generation and editing via learning real-world dynamics. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12501–12511, 2025.

Zhewei Dai, Shilei Zeng, Haotian Liu, Xurui Li, Feng Xue, and Yu Zhou. Seas: few-shot industrial anomaly image generation with separation and sharing fine-tuning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23135–23144. IEEE, 2025.

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

Rohit Gandikota, Joanna Materzyńska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept sliders: Lora adaptors for precise control in diffusion models. In *European Conference on Computer Vision*, pp. 172–188. Springer, 2024.

Guan Gui, Bin-Bin Gao, Jun Liu, Chengjie Wang, and Yunsheng Wu. Few-shot anomaly-driven generation for anomaly classification and segmentation. In *European Conference on Computer Vision*, pp. 210–226. Springer, 2024.

Jia Guo, Shuai Lu, Weihang Zhang, Fang Chen, Huiqi Li, and Hongen Liao. Dinomaly: The less is more philosophy in multi-class unsupervised anomaly detection. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20405–20415. IEEE, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Teng Hu, Jiangning Zhang, Ran Yi, Yuzhen Du, Xu Chen, Liang Liu, Yabiao Wang, and Chengjie Wang. Anomalydiffusion: Few-shot anomaly image generation with diffusion model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 8526–8534, 2024.

Ying Jin, Jinlong Peng, Qingdong He, Teng Hu, Jiafu Wu, Hao Chen, Haoxuan Wang, Wenbing Zhu, Mingmin Chi, Jun Liu, et al. Dual-interrelated diffusion model for few-shot anomaly image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 30420–30429, 2025.

Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *European Conference on Computer Vision*, pp. 150–168. Springer, 2024.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022b.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

Konstantin Mishchenko and Aaron Defazio. Prodigy: An expeditiously adaptive parameter-free learner. *arXiv preprint arXiv:2306.06101*, 2023.

Sassan Mokhtar, Arian Mousakhan, Silvio Galesso, Jawad Tayyub, and Thomas Brox. Detect, classify, act: Categorizing industrial anomalies with multi-modal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4058–4067, 2025.

Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10743–10752, 2021.

Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 conference proceedings*, pp. 1–11, 2023.

Taekyeong Park, Yongho Son, Sanghyuk Moon, Seungju Han, and Je Hyeong Hong. Long-tailed detection and classification of wafer defects from scanning electron microscope images robust to diverse image backgrounds and defect scales. *Engineering Applications of Artificial Intelligence*, 162:112342, 2025.

Haoxiang Rao, Zhao Wang, Chenyang Si, Yan Lyu, Yuanyi Duan, Fang Zhao, and Caifeng Shan. One-to-more: High-fidelity training-free anomaly generation with attention control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28492–28501, 2026.

Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14318–14328, 2022.

Jaewoo Song, Daemin Park, Kanghyun Baek, Sangyub Lee, Jooyoung Choi, Eunji Kim, and Sungroh Yoon. Defectfill: Realistic defect generation with inpainting diffusion model for visual inspection. *ArXiv*, abs/2503.13985, 2025.

Deepak Sridhar and Nuno Vasconcelos. Prompt sliders for fine-grained control, editing and erasing of concepts in diffusion models. In *European Conference on Computer Vision*, pp. 17–31. Springer, 2024.

JD Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International conference for learning representations*, volume 6, 2018.

Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. Objectdrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. In *European Conference on Computer Vision*, pp. 112–129. Springer, 2024.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Linchun Wu, Qin Zou, Xianbiao Qi, Bo Du, Zhongyuan Wang, and Qingquan Li. Double helix diffusion for cross-domain anomaly image generation. *arXiv preprint arXiv:2509.12787*, 2025b.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26125–26135, 2025.

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

Hui Zhang, Zuxuan Wu, Zheng Wang, Zhineng Chen, and Yu-Gang Jiang. Prototypical residual networks for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16281–16291, 2023a.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023b.

Pengze Zhang, Lingxiao Yang, Jian-Huang Lai, and Xiaohua Xie. Exploring dual-task correlation for pose guided person image generation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 7713–7722, 2022.

Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European conference on computer vision*, pp. 392–408. Springer, 2022.

APPENDIX

A ADDITIONAL DETAILS

We provide additional explanations of our training and generation procedures in this section. Complete implementation details, including configuration files and scripts, are provided in the *code* directory of the zip package.

A.1 DATA PRE-PROCESSING

For the VisA (Zou et al., 2022) dataset, which contains non-square images, we applied a center crop based on the shorter edge to obtain square inputs, following baseline models. Class labels were assigned using the metadata provided with VisA, and categories containing multi-label samples were excluded. For MVTec-3D (Bergmann et al., 2021), we discard the depth channel and convert each sample to a 3-channel RGB image, and all experiments are conducted on these 3-channel representations.

A.2 REVERSE GENERATION

To construct the reverse (anomaly-to-normal) generation module, we fine-tune a pseudo-normal generation module independently for each object category. Specifically, we used 60 normal images per object and a simple category-specific text prompt of the form *"a photo of [object]"*. This step enables the model to reliably reconstruct normal textures across diverse industrial surfaces, forming the basis for generating spatially-aligned pseudo-normal images used in the forward generation.

A.3 FORWARD GENERATION

The forward (normal-to-anomaly) editing model was fine-tuned separately for each anomaly class. For parameter-efficient adaptation, we applied LoRA (Hu et al., 2022) only to the key, query, value, and output projection layers of the Flux.1 kontext-dev (Batifol et al., 2025), leaving the text encoder frozen. All experiments were conducted using bf16 precision to ensure stable and memory-efficient training.

A.4 MASK PREDICTOR

Downstream anomaly detection requires pixel-accurate masks. We therefore obtain a coarse mask from the internal features of the editor and refine it with the Mask Refinement Module (MRM) of SeaS (Dai et al., 2025). Among the 57 transformer blocks of the DiT (19 double-stream and 38 single-stream), we extract features from the 46th and 51st blocks, both of which are single-stream blocks. FLUX.1 Kontext processes the text tokens, the target-image tokens being denoised, and the reference-image tokens as a single joint sequence. From this sequence, we keep only the target-image tokens, which occupy the first L_x positions of the image sequence and form a 32×32 grid, and discard all other tokens. The reference tokens are excluded on purpose, so that the predictor localizes the anomaly rather than the reference image. The target-token features from the two blocks are concatenated along the channel dimension, compressed to 192 channels by a single 1×1 convolution, and bilinearly upsampled to 64×64. All other components follow SeaS: the 4-layer self-attention refinement, the three cascaded MRMs, the prediction head, and the focal-loss supervision with normal image alignment. As in SeaS (Dai et al., 2025), the high-resolution features are taken from the first three up-blocks of the VAE decoder. At inference, we use the mask predicted at the final step of the 10-step rectified-flow sampling.

A.5 ANOMALY CLASSIFICATION

For evaluating the utility of generated images, we trained a ResNet-34 (He et al., 2016) classifier under a setting identical to AnomalyDiffusion (Hu et al., 2024). The classifier was optimized using CrossEntropy loss, Adam, batch size of 8, and a learning rate of 1e-4 for 30 epochs. Importantly, the classifier was trained using only the 500 generated images per anomaly class, without incorporating any real samples, in order to isolate the contribution of synthetic data to classification performance.

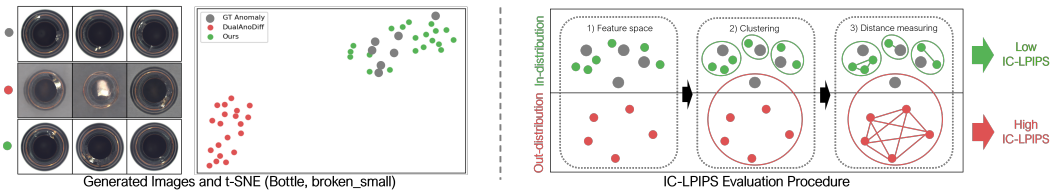


Figure 6: Relationship between IC-LPIPS and the real anomaly distribution.

A.6 Anomaly Detection

For evaluating the utility of generated images for anomaly localization, we trained a U-Net-based segmentation network, following the discriminative sub-network of DRAEM (Zavrtanik et al., 2021), under a setting identical to AnomalyDiffusion (Hu et al., 2024). The network takes an RGB image as input and outputs a two-channel (normal/anomaly) segmentation map. It was optimized using Focal loss (Lin et al., 2017), Adam, a batch size of 8, and an initial learning rate of 1e-4 for 200 epochs, with the learning rate decayed by a factor of 0.2 at 80% and 90% of the total epochs. Importantly, the detector was trained using only the 500 generated image–mask pairs per anomaly class, without incorporating any real samples, in order to isolate the contribution of synthetic data to detection performance.

A.7 Discussion on image diversity.

On MVTec-AD, DualAnoDiff achieves a higher IC-LPIPS than AnomalyEditor (0.36 vs. 0.28), while its KID is also higher (96.82 vs. 30.55). This illustrates that higher perceptual diversity does not necessarily imply closer agreement with the real image distribution. Fig. 6 provides an example for the bottle broken-small class: in the t-SNE visualization, DualAnoDiff samples form a group separated from real anomaly samples, whereas AnomalyEditor samples appear closer to them. The accompanying schematic illustrates how IC-LPIPS can be high when generated samples exhibit substantial perceptual variation, even if they are not realistic. Such variation may arise from changes in backgrounds or object structures as well as anomaly regions. Therefore, generation quality should be assessed by jointly considering IC-LPIPS, KID, qualitative comparisons, and downstream task performance.

A.8 Reproduction details of DualAnoDiff.

We reproduced DualAnoDiff (Jin et al., 2025) based on its official repository. The Background Compensation Module (BCM) was selectively activated for *bottle*, *toothbrush*, and *pill* on MVTec-AD as specified in the paper, and enabled for every category on both VisA and MVTec-3D. Foreground masks were obtained using U^2-Net with a binarization threshold of 0.5, as configured in the official codebase. Under this protocol, we conducted reproduction through two approaches: retraining from scratch using the released codebase, and directly evaluating the official pre-trained checkpoint. For fair evaluation, we adopt the data selection protocol of AnomalyDiffusion (Hu et al., 2024), which has been used as a standard protocol (Duan et al., 2023; Dai et al., 2025). This protocol provides a deterministic split by sorting anomaly images per anomaly class and selecting the first third without rounding. Although DualAnoDiff also reserves one-third of anomaly samples for training, details such as how fractional sample counts are rounded remain undocumented.

Our retrained model achieved 68.50%, and the official pre-trained model yielded 67.15%, both of which fall short of the performance reported in the original paper. In this regard, we note that the official repository also contains discussions reflecting similar experiences from other users. For all comparisons, we report the higher of the two reproduced scores. The detailed per-category comparison is provided in Tab. 6.

A.9 Reproduction details of O2MAG.

We reproduced O2MAG using its official implementation. O2MAG evaluates downstream performance by training task models on both synthetic images and real anomaly images from the training

Table 6: Reproduction results of DualAnoDiff and comparison with AnomalyEditor on MVTec-AD, evaluated by classification accuracy (%) using ResNet-34 (He et al., 2016). DualAnoDiff denotes our retrained results using the official codebase; DualAnoDiff* indicates results obtained from the official pre-trained checkpoint; DualAnoDiff† shows the values reported in the original paper. **Note that we have not been able to reproduce the reported values to this date.** Bold = best, underline = second-best.

Category	DualAnoDiff	DualAnoDiff*	DualAnoDiff†	Ours
bottle	72.09	58.14	<u>79.07</u>	**83.72**
cable	56.25	56.25	78.12	**84.38**
capsule	48.00	53.33	**70.67**	<u>56.00</u>
carpet	<u>70.97</u>	67.74	**79.03**	69.35
grid	60.00	70.00	<u>80.00</u>	**87.50**
hazelnut	85.42	83.33	<u>89.58</u>	**93.75**
leather	84.13	**95.24**	<u>90.48</u>	<u>90.48</u>
metal_nut	76.56	70.31	89.06	**92.19**
pill	33.33	50.00	<u>56.25</u>	**65.62**
screw	58.02	48.15	<u>70.37</u>	**81.48**
tile	98.25	**100.00**	**100.00**	<u>98.25</u>
transistor	71.43	67.86	71.43	**89.29**
wood	71.43	**92.86**	<u>85.71</u>	80.95
zipper	<u>73.17</u>	26.83	**75.61**	65.85
Average	68.50	67.15	<u>79.67</u>	**81.34**

Table 7: Reproduction results of O2MAG and comparison with AnomalyEditor on MVTec-AD, evaluated by classification accuracy (%) using ResNet-34 (He et al., 2016). Methods without a mark are trained on synthetic data only; * denotes training on both real and synthetic data; O2MAG† shows the values reported in the original paper. Bold = best, underline = second-best.

Category	O2MAG	O2MAG*	O2MAG†	Ours	Ours*
bottle	79.07	81.40	**86.05**	<u>83.72</u>	<u>83.72</u>
cable	73.44	75.00	<u>76.56</u>	**84.38**	61.33
capsule	49.33	65.33	**77.33**	56.00	<u>67.74</u>
carpet	62.90	<u>74.19</u>	<u>74.19</u>	69.35	**96.83**
grid	65.00	67.50	<u>80.00</u>	**87.50**	68.75
hazelnut	83.33	<u>93.75</u>	**95.83**	<u>93.75</u>	92.86
leather	61.90	84.13	<u>92.06</u>	90.48	**100.00**
metal_nut	81.25	<u>90.62</u>	84.38	**92.19**	82.81
pill	59.38	66.67	<u>72.92</u>	65.62	**79.27**
screw	51.85	59.26	56.79	<u>81.48</u>	**92.19**
tile	**100.00**	**100.00**	**100.00**	<u>98.25</u>	93.75
transistor	89.29	**96.43**	<u>92.86</u>	89.29	83.95
wood	66.67	73.81	78.57	<u>80.95</u>	**90.00**
zipper	74.39	76.83	**85.37**	65.85	<u>80.95</u>
Average	71.27	78.92	<u>82.35</u>	81.34	**83.87**

split. In contrast, all main experiments in our study exclude real anomaly images from downstream training to isolate the contribution of synthetic data. Under this synthetic-only setting, O2MAG achieved a classification accuracy of 71.27% on MVTec-AD, compared with 81.34% for AnomalyEditor. We additionally evaluated both methods under O2MAG's setting, including real anomaly training images alongside synthetic images. AnomalyEditor achieved 83.87%, exceeding both our reproduced O2MAG result of 78.92% and its originally reported accuracy of 82.35%. These results show that AnomalyEditor's advantage is maintained when real anomaly images are included in downstream training. The detailed per-category comparison under both settings is provided in Tab. 7.

B ADDITIONAL QUANTITATIVE RESULTS

B.1 CLASSIFICATION ACCURACY

We provide category-wise classification results for all benchmark datasets, including MVTec-AD (Bergmann et al., 2019), VisA (Zou et al., 2022), and MVTec-3D (Bergmann et al., 2021). Detailed per-category performance is summarized in Tab. 9, Tab. 10, Tab. 11, offering a comprehensive view of the model's classification capability across diverse anomaly types and object categories.

B.2 DETECTION RESULTS

We provide category-wise detection results for all benchmark datasets, including MVTec-AD (Bergmann et al., 2019), VisA (Zou et al., 2022), and MVTec-3D (Bergmann et al., 2021). Detailed per-category performance for image-level is summarized in Tab. 12, Tab. 13, Tab. 14, while detailed per-category performance for pixel-level is summarized in Tab. 15, Tab. 16, Tab. 17. offering a comprehensive view of the model's detection capability across diverse anomaly types and object categories.

Phase	AnoDiff	AnoGen	O2MAG	DualAno	SeaS	Ours
Train. (Time / Memory)	10.5h / 24.0GB	1.4h / 11.7GB	Training-free	12.8h / 26.5GB	4.6h / 24.0GB	33.8h / 39.5GB
Infer. (Time per img-mask pair)	5.76s	1.01s	35.7s	10.18s	1.74s	6.52s

Table 8: Comparison of the computational cost for training and inference on Tile object class of MVTec-AD

B.3 COMPUTATIONAL COST.

All timings are measured on an NVIDIA A100 GPU. Tab. 8 compares computational costs on the tile category of MVTec-AD. AnomalyEditor requires 33.8 hours of training and 39.5 GB of GPU memory, exceeding the training costs of the evaluated baselines. Generating an image–mask pair takes 6.52 seconds, including mask prediction, compared with 10.18 seconds for DualAnoDiff and 35.7 seconds for O2MAG. Although AnomalyEditor incurs higher training costs, training and synthetic data generation are performed offline and introduce no additional inference overhead for the downstream classifier. (See Tab. 8)

B.4 ABLATION STUDY ON EDIT-REGION SAMPLING.

We evaluate the effect of edit region-aware sampling on MVTec-AD. Removing this procedure reduces classification accuracy from 81.34% to 76.66%, a decrease of 4.68%p. This result suggests that re-editing potential non-edit samples helps improve the utility of generated anomaly images for downstream classification.

C ADDITIONAL QUALITATIVE RESULTS

We provide additional qualitative examples to further demonstrate the capability of the model in various datasets (MVTec-AD, VisA, and MVTec-3D).

C.1 REVERSE GENERATION RESULTS

Fig. 7-23 shows the outputs of the reverse generation stage, where anomaly images are converted into pseudo-normal versions. Across all categories, the generated pseudo-normal images closely resemble real normal samples, indicating that the model effectively removes various anomaly patterns while preserving the underlying textures and structures of each object. This ability is especially clear in large and severe anomalous cases, such as the *missing cable* example in Fig. 8, where the model reconstructs a coherent and realistic cable despite the extensive damaged region. In more challenging cases such as the *pill type* example in Fig. 15, where the appearance of the entire object

must be replaced, the method still produces a plausible and natural-looking normal pill that fits well with the surrounding context.

C.2 FORWARD GENERATION RESULTS

Fig. 24-40 presents additional forward generation results, in which anomalies are introduced into normal reference images. The edited anomalies align naturally with the geometry of the object and follow the expected anomaly locations with high spatial accuracy. Even in cases where the object appears in different orientations, such as *screws* whose alignment varies from image to image, the model generates anomaly patterns that remain properly positioned and visually consistent with the rotated object's appearance as observed in Fig. 33. For categories such as *grid*, where the structure is thin and delicate and the anomalies tend to be fine, the model is able to generate these subtle anomalies clearly without breaking the overall structure (Fig. 28). These results demonstrate that AnomalyEditor introduces changes only where necessary while preserving the normal regions and unedited regions almost identically to the original normal images.

Table 9: Comparison of classification accuracies (%) on the MVTec-AD (Bergmann et al., 2019) dataset.

Type	MCS			GIS		MFIE
Category	AnoDiff	AnoGen	O2MAG	DualAno	SeaS	Ours
bottle	81.40	62.79	79.07	72.09	65.12	**83.72**
cable	65.62	64.06	73.44	56.25	60.94	**84.38**
capsule	37.33	30.67	49.33	48.00	28.00	**56.00**
carpet	64.52	48.39	62.90	**70.97**	45.16	69.35
grid	40.00	42.50	65.00	60.00	32.50	**87.50**
hazelnut	87.50	68.75	83.33	85.42	47.92	**93.75**
leather	68.25	58.73	61.90	84.13	65.08	**90.48**
metal nut	82.81	58.33	81.25	76.56	68.75	**92.19**
pill	56.25	37.50	59.38	33.33	39.58	**65.62**
screw	24.69	48.15	51.85	58.02	44.44	**81.48**
tile	87.72	98.25	**100.00**	98.25	91.23	98.25
trans	75.00	46.43	**89.29**	71.43	60.71	**89.29**
wood	64.29	73.81	66.67	71.43	57.14	**80.95**
zipper	73.17	58.54	**74.39**	73.17	31.71	65.85
Average	64.90	56.92	71.27	68.50	52.73	**81.34**

Table 10: Comparison of classification accuracies (%) on the VisA (Zou et al., 2022) dataset.

Type	MCS			GIS		MFIE
Category	AnoDiff	AnoGen	O2MAG	DualAno	SeaS	Ours
candle	31.03	27.59	27.59	44.83	27.59	**72.41**
cashew	27.42	45.16	30.65	33.87	17.74	**64.52**
gum	64.10	84.62	53.83	76.92	46.15	**97.44**
fryum	31.67	35.00	31.67	38.33	15.00	**43.33**
mac1	32.73	29.09	50.91	40.00	25.45	**80.00**
mac2	30.19	24.53	52.83	37.74	35.85	**73.58**
pcb1	53.97	61.90	53.97	**68.25**	55.56	60.32
pcb2	59.68	**69.35**	51.61	59.68	62.90	67.74
pcb3	37.50	35.94	46.88	29.69	45.31	**75.00**
pcb4	61.54	57.69	57.69	**69.23**	53.85	53.85
pipe	41.67	43.33	55.00	75.00	16.67	**86.67**
Average	42.86	46.75	46.60	52.14	36.55	**70.44**

Table 11: Comparison of classification accuracies (%) on the MVTec-3D (Bergmann et al., 2021) dataset.

Type	MCS			GIS		MFIE
Category	AnoDiff	AnoGen	O2MAG	DualAno	SeaS	Ours
bagel	43.33	**60.87**	55.00	36.67	33.33	45.00
cable	33.90	45.76	**52.54**	45.76	37.29	45.76
carrot	**54.44**	42.22	53.33	42.22	23.33	**54.44**
cookie	44.29	38.57	32.86	37.14	41.43	**45.71**
dowel	45.07	35.21	38.03	28.17	42.25	**46.48**
foam	42.86	35.71	28.57	32.14	37.50	**50.00**
peach	36.11	**49.09**	43.06	27.78	30.56	48.61
potato	32.81	34.38	**40.62**	26.56	29.69	23.44
rope	78.72	51.06	70.21	55.32	**80.85**	68.09
tire	**49.15**	33.90	44.07	42.37	39.00	42.37
Average	46.07	42.68	45.83	37.41	39.52	**46.99**

Table 12: Comparison of image-level detection results (%) on the MVTec-AD (Bergmann et al., 2019) dataset.

Type	MCS									GIS						MFIE		
Category	AnoDiff			AnoGen			O2MAG			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	AUC-I	AP-I	F1-I
bottle	99.7	99.8	98.9	99.3	99.7	97.6	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
cable	99.8	99.9	98.4	97.9	98.6	94.4	**100.0**	**100.0**	**100.0**	98.2	99.0	96.8	97.2	97.9	94.4	99.8	99.9	98.5
capsule	96.5	98.9	95.2	**97.5**	**99.3**	95.5	95.8	98.9	95.2	96.5	99.0	94.6	96.9	99.2	**95.9**	94.4	98.4	93.4
carpet	92.9	97.5	93.2	95.2	98.1	93.4	93.0	96.7	91.9	99.0	99.6	**98.4**	**99.4**	**99.7**	96.9	94.5	97.9	93.2
grid	98.1	99.3	**98.7**	97.6	99.2	**98.7**	99.2	99.6	**98.7**	98.3	99.3	97.4	99.3	**99.7**	97.5	**99.4**	**99.7**	**98.7**
hazelnut	99.9	**100.0**	99.0	99.6	99.7	97.9	99.4	99.5	97.9	99.9	99.9	98.9	99.8	99.9	98.0	**100.0**	**100.0**	**100.0**
leather	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	98.9	99.3	98.4	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	99.2	**100.0**	**100.0**	**100.0**
metal_nut	**100.0**	**100.0**	**100.0**	99.7	99.9	98.4	**100.0**	**100.0**	**100.0**	99.3	**100.0**	97.7	99.9	**100.0**	99.2	**100.0**	**100.0**	**100.0**
pill	97.9	99.5	96.8	96.4	99.1	95.3	95.6	98.9	94.0	89.7	97.4	90.2	98.7	99.7	96.9	**99.0**	**99.8**	**98.9**
screw	93.5	96.9	**91.7**	92.4	96.1	90.7	88.1	94.2	87.3	81.9	91.3	82.9	**94.6**	**97.6**	90.8	90.3	95.6	87.0
tile	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	99.8	99.9	99.1	99.3	99.6	97.4
toothbrush	98.8	99.3	97.4	**100.0**	**100.0**	**100.0**	99.2	99.5	97.4	92.9	96.3	88.9	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
transistor	**100.0**	**100.0**	**100.0**	99.4	98.8	94.7	99.9	99.7	98.2	99.1	98.1	94.9	97.2	96.4	94.3	99.0	98.5	96.4
wood	99.9	99.9	98.8	99.0	99.5	98.8	**100.0**	**100.0**	**100.0**	99.9	99.9	98.8	98.1	99.3	97.6	**100.0**	**100.0**	**100.0**
zipper	**100.0**	**100.0**	99.4	**100.0**	**100.0**	**100.0**	99.9	**100.0**	99.4	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	99.8	99.9	99.4
Average	98.5	**99.4**	**97.8**	98.3	99.2	97.0	97.9	99.1	97.2	97.0	98.7	96.0	**98.7**	99.3	97.3	98.4	99.3	97.5

Table 13: Comparison of image-level detection results (%) on the VisA (Zou et al., 2022) dataset.

Type	MCS									GIS						MFIE		
Category	AnoDiff			AnoGen			O2MAG			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	AUC-I	AP-I	F1-I
candle	**93.6**	**93.1**	**87.0**	87.1	84.4	76.0	89.8	89.5	82.5	86.6	80.9	74.5	78.4	76.6	66.0	85.1	83.1	72.9
capsules	63.9	44.8	46.3	63.6	37.5	52.2	95.0	90.3	83.3	89.8	75.8	71.1	97.3	96.3	92.7	**99.0**	**96.9**	**93.0**
cashew	96.4	97.1	93.0	**98.7**	**99.0**	**95.3**	96.7	97.5	92.2	94.4	95.7	87.9	85.6	88.6	82.2	96.6	96.6	**95.3**
chewinggum	94.6	95.3	89.7	95.1	95.6	90.1	**99.0**	**98.8**	**95.0**	93.3	91.5	84.6	96.1	96.5	93.2	95.8	96.5	90.9
fryum	91.5	93.7	86.8	88.0	92.4	83.3	**97.0**	**97.7**	**91.4**	87.6	90.7	82.7	86.5	89.5	84.8	91.4	94.1	84.7
macaroni1	82.6	72.2	69.4	**98.9**	**98.1**	**93.6**	97.7	96.4	90.3	87.4	82.8	71.1	94.5	89.2	86.0	97.9	96.3	89.9
macaroni2	45.8	30.9	52.8	78.4	71.5	**66.7**	74.3	70.7	61.4	74.3	55.8	63.0	75.1	70.7	62.3	**81.0**	**75.8**	66.0
pcb1	**95.7**	**94.3**	84.2	94.9	93.7	**85.7**	87.7	82.4	78.7	78.6	80.3	71.3	92.7	90.5	79.7	93.9	88.3	85.3
pcb2	94.6	94.1	88.3	**97.0**	**96.2**	**89.8**	86.9	83.3	75.2	88.3	88.6	81.1	93.1	92.8	84.4	95.5	94.9	87.9
pcb3	85.3	85.7	78.9	91.7	88.3	82.7	**91.8**	**91.5**	**85.2**	77.5	74.9	65.4	85.4	84.8	76.5	90.5	89.6	80.0
pcb4	97.6	89.2	84.7	94.7	79.8	77.6	95.8	89.3	83.3	88.6	73.8	66.7	95.9	88.8	87.5	**98.2**	**91.4**	**90.9**
pipe_fryum	78.7	84.5	75.5	87.6	91.2	82.3	**96.6**	**97.6**	**94.1**	91.0	93.8	86.2	82.7	85.0	82.8	94.8	96.2	90.9
Average	85.0	81.2	78.1	89.6	85.6	81.3	92.4	90.4	84.4	86.5	82.1	75.5	88.6	87.4	81.5	**93.3**	**91.6**	**85.6**

Table 14: Comparison of image-level detection results (%) on the MVTec-3D (Bergmann et al., 2021) dataset.

Type	MCS									GIS						MFIE		
Category	AnoDiff			AnoGen			O2MAG			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	AUC-I	AP-I	F1-I
bagel	94.5	98.1	93.4	93.5	97.9	94.1	**99.3**	**99.8**	**98.3**	98.7	99.6	97.5	97.9	99.3	96.7	98.9	99.6	**98.3**
cable_gland	**98.9**	**99.6**	**97.5**	88.5	94.7	91.8	96.1	98.7	95.8	61.8	84.8	84.9	91.2	96.8	92.1	96.9	98.9	95.1
carrot	97.2	99.2	**96.7**	94.1	98.0	94.1	**97.8**	**99.4**	96.6	90.7	97.1	91.3	92.6	97.2	94.4	93.7	97.8	94.2
cookie	88.9	94.7	91.8	89.9	95.6	91.9	84.6	94.2	86.5	84.3	93.7	85.7	92.0	97.2	90.2	**95.1**	**98.1**	**93.2**
dowel	64.4	86.4	84.8	83.7	94.3	86.3	90.9	96.9	89.9	82.7	93.5	88.0	79.8	92.8	85.2	**95.0**	**98.5**	**94.3**
foam	**93.5**	**97.9**	91.6	92.9	97.5	**92.7**	71.3	88.6	85.3	82.1	93.7	86.6	71.9	90.1	85.9	88.9	96.2	89.5
peach	79.7	90.3	85.7	90.2	96.7	88.7	**97.4**	**99.0**	**96.0**	88.0	95.3	89.7	95.7	98.4	93.3	88.7	95.5	91.5
potato	71.4	87.8	87.0	74.1	88.3	87.1	82.7	93.7	88.4	62.9	83.1	85.3	61.6	83.5	85.9	**91.2**	**96.4**	**93.2**
rope	99.5	99.7	97.9	98.5	99.2	96.8	96.0	97.6	91.1	98.1	99.0	96.7	**100.0**	**100.0**	**100.0**	99.7	99.8	97.9
tire	72.7	88.3	83.2	75.6	88.6	85.5	**94.3**	**97.6**	**93.3**	58.3	75.9	83.1	89.7	96.0	88.7	84.8	94.1	85.1
Average	86.1	94.2	91.0	88.1	95.1	90.9	91.0	96.6	92.1	80.8	91.6	88.9	87.2	95.1	91.2	**93.3**	**97.5**	**93.2**

Table 15: Comparison of pixel-level detection accuracies (%) on the MVTec-AD (Bergmann et al., 2019) dataset.

Type	MCS												GIS								MFIE			
Category	AnoDiff				AnoGen				O2MAG				DualAnoDiff				SeaS				Ours			
	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P
bottle	99.3	92.6	84.6	95.8	98.0	83.0	74.5	89.9	99.2	92.4	85.3	94.1	98.7	88.4	81.4	91.8	99.5	92.6	86.8	**97.0**	**99.7**	**94.8**	**87.3**	95.3
cable	98.4	84.3	76.3	93.6	94.8	68.3	64.7	88.6	**99.1**	**88.4**	**81.1**	**96.9**	91.3	72.9	69.9	80.9	96.0	76.7	73.0	90.1	99.0	86.1	78.5	96.2
capsule	97.9	41.6	43.7	89.7	96.3	34.0	37.5	92.4	97.3	**48.9**	**53.1**	94.0	**98.8**	47.8	52.8	**95.5**	96.8	44.0	46.0	94.7	97.2	45.8	52.5	93.5
carpet	96.4	74.8	71.8	83.6	98.8	81.4	74.0	89.6	96.4	66.6	63.1	90.4	99.1	**87.1**	**78.5**	94.5	**99.3**	85.3	76.4	**95.1**	98.8	85.3	78.2	90.8
grid	97.3	43.6	46.0	93.6	96.1	33.1	41.8	93.9	98.7	55.6	57.4	96.8	95.4	54.9	55.0	93.2	98.9	63.9	61.0	**97.2**	**99.4**	**64.3**	**61.7**	97.0
hazelnut	99.3	89.4	81.9	94.5	97.2	58.3	56.7	93.1	99.7	92.5	85.3	96.9	99.5	88.7	81.0	95.4	99.6	90.1	82.7	**98.1**	**99.9**	**96.4**	**91.3**	98.0
leather	**99.8**	78.4	71.0	98.3	99.4	77.7	70.4	98.0	99.2	69.2	66.9	97.5	**99.8**	**82.8**	**73.6**	**98.4**	99.4	77.1	69.0	98.1	99.6	74.9	67.4	98.1
metal_nut	99.6	97.8	92.5	95.2	95.2	77.0	72.5	89.5	99.7	97.8	93.6	94.5	99.5	96.7	91.8	92.5	99.7	**98.5**	**94.1**	**95.6**	**99.8**	**98.5**	94.0	94.8
pill	**99.6**	**95.0**	**88.5**	94.4	99.2	89.1	80.7	93.2	99.3	92.2	85.3	95.6	97.1	83.0	77.9	78.4	99.5	90.5	82.9	**97.1**	**99.6**	94.3	87.4	96.9
screw	95.6	11.8	21.6	83.8	92.7	21.9	27.2	77.5	98.6	42.1	45.0	93.7	98.2	50.4	51.2	91.5	96.3	48.3	49.8	90.5	**98.9**	**51.4**	**52.3**	**94.8**
tile	99.5	96.0	88.9	96.9	99.2	93.1	85.0	97.0	**99.7**	96.7	90.2	**97.9**	99.6	95.4	89.2	97.5	**99.7**	**96.8**	**91.4**	97.5	99.4	95.3	89.5	97.3
toothbrush	97.9	56.6	58.5	84.9	**99.2**	**93.1**	**85.0**	72.2	95.5	44.3	50.8	82.1	93.8	56.2	60.9	71.1	97.1	66.0	65.8	86.0	99.1	58.1	62.3	**93.7**
transistor	**99.2**	**89.1**	**81.2**	**96.4**	95.0	63.2	61.2	88.8	97.2	77.8	72.1	93.7	92.5	73.9	71.1	89.8	97.8	83.7	77.0	92.6	96.1	73.4	70.9	92.1
wood	96.7	74.6	67.4	91.3	**98.5**	78.2	71.5	94.0	98.1	75.2	67.8	90.4	98.3	**84.6**	**77.2**	**94.5**	96.7	80.4	74.8	90.8	97.6	77.7	70.3	88.4
zipper	99.4	84.7	78.2	**96.6**	99.1	78.9	71.8	95.6	98.2	79.7	74.5	93.3	**99.5**	**89.0**	**82.0**	**96.6**	98.8	81.1	74.9	95.9	98.2	82.2	76.7	92.9
Average	98.4	74.0	70.1	92.6	97.2	68.7	65.0	90.2	98.4	74.6	71.4	93.9	97.4	76.8	72.9	90.8	98.3	78.3	73.7	94.4	**98.8**	**78.6**	**74.7**	**94.7**

Table 16: Comparison of pixel-level detection accuracies (%) on the VisA (Zou et al., 2022) dataset.

Type	MCS												GIS								MFIE			
Category	AnoDiff				AnoGen				O2MAG				DualAnoDiff				SeaS				Ours			
	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P
candle	98.4	33.8	39.4	90.3	96.8	40.2	43.6	88.1	90.9	32.2	38.7	89.4	92.1	18.3	28.3	85.0	93.7	**55.8**	**59.9**	82.9	**98.6**	40.1	42.7	**94.7**
capsules	87.6	8.0	12.6	56.4	92.3	3.8	11.4	68.8	97.1	55.4	54.4	89.3	97.5	51.2	50.6	80.7	**99.9**	72.9	65.1	93.3	98.9	**85.1**	**79.2**	**95.6**
cashew	99.0	83.0	78.1	81.5	97.8	68.4	66.0	**94.9**	89.1	12.9	18.5	87.8	**99.8**	**95.0**	**92.9**	92.4	98.1	85.0	81.5	83.9	99.6	92.3	87.2	92.3
chewinggum	99.4	69.2	64.8	92.0	99.4	80.8	73.9	90.1	99.0	83.8	77.9	93.2	99.4	65.5	59.5	90.4	99.8	**88.8**	**80.0**	**96.7**	**99.9**	81.4	72.2	94.9
fryum	96.8	53.3	51.8	**94.0**	97.1	55.5	53.5	89.6	93.8	39.6	43.1	89.4	97.6	55.3	55.9	91.1	**98.8**	**83.9**	**77.7**	77.7	97.2	50.0	52.2	87.2
macaroni1	92.6	3.4	9.6	83.8	97.8	35.6	41.1	**95.7**	97.7	25.2	34.4	93.4	94.5	5.2	11.1	90.2	**99.0**	**54.8**	**56.3**	89.3	94.7	44.2	51.1	94.2
macaroni2	93.0	0.2	0.6	78.0	97.8	8.5	13.3	**92.4**	93.3	15.8	26.1	80.3	96.1	9.5	12.2	87.7	**98.2**	14.2	23.6	90.3	96.5	**32.9**	**40.8**	83.0
pcb1	94.9	71.4	70.0	**88.9**	97.0	69.9	66.8	87.3	94.1	56.9	61.7	69.0	96.1	70.9	68.5	76.8	**98.4**	**86.0**	**84.5**	84.2	95.9	65.1	65.1	83.3
pcb2	94.9	25.6	33.5	82.5	**97.3**	31.3	39.4	**90.0**	87.7	13.1	22.6	66.5	96.4	34.4	43.3	85.1	93.3	**54.7**	**55.9**	79.8	91.9	47.2	52.3	82.7
pcb3	97.5	28.9	31.9	81.5	**97.9**	33.1	34.9	**86.3**	90.0	34.6	39.8	77.8	96.5	25.9	35.0	80.6	96.8	**68.2**	**65.8**	74.1	93.8	34.9	42.9	85.0
pcb4	98.3	66.4	62.1	**92.3**	96.4	37.9	46.2	79.5	93.4	26.8	34.3	75.6	98.0	68.5	68.9	85.4	**99.4**	**82.9**	**78.4**	91.9	98.3	72.2	70.6	91.9
pipe_fryum	**99.7**	90.6	84.4	88.3	99.4	79.9	72.1	86.6	96.0	42.1	47.0	93.4	**99.7**	**93.8**	**86.2**	95.4	99.2	88.0	82.0	86.7	99.0	77.5	72.6	**95.6**
Average	96.0	44.5	44.9	84.1	97.3	45.4	46.9	87.4	93.5	36.5	41.5	83.8	97.0	49.5	51.0	86.7	**97.9**	**69.6**	**67.6**	85.9	97.0	60.2	60.7	**90.0**

Table 17: Comparison of pixel-level detection accuracies (%) on the MVTec-3D (Bergmann et al., 2021) dataset.

Type	MCS												GIS								MFIE			
Category	AnoDiff				AnoGen				O2MAG				DualAnoDiff				SeaS				Ours			
	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P	AUC-P	AP-P	F1-P	PRO-P
bagel	98.4	3.4	8.8	94.7	98.5	4.7	11.7	95.6	99.0	3.8	9.1	97.9	99.3	**7.8**	**15.2**	95.5	99.3	5.3	12.2	97.1	**99.5**	6.0	14.0	**98.1**
cable_gland	95.7	1.5	4.4	86.2	**99.3**	**9.5**	**18.9**	**97.0**	97.2	3.1	8.7	96.0	95.3	1.6	5.7	83.6	93.4	4.3	12.2	89.4	98.9	8.4	17.6	96.1
carrot	98.9	7.2	17.1	95.2	99.4	6.3	14.0	97.8	99.6	5.7	12.1	**98.2**	99.2	14.2	23.7	97.1	**99.8**	**24.1**	**31.6**	98.1	99.3	9.8	22.3	98.0
cookie	88.4	2.3	6.2	87.8	91.2	6.8	17.4	85.1	93.6	3.0	7.9	**93.4**	94.9	18.9	33.5	93.3	94.7	8.3	19.2	92.2	**99.0**	**35.3**	**45.3**	**93.4**
dowel	99.3	33.8	38.4	87.2	99.0	25.7	36.8	89.4	99.0	23.7	31.5	92.3	98.4	30.5	39.7	89.1	99.8	**57.3**	**57.8**	86.9	**99.9**	52.9	54.4	**95.7**
foam	99.9	74.6	68.7	**90.2**	99.9	65.3	61.1	82.2	99.8	58.7	60.2	87.3	96.4	58.0	63.0	73.0	99.8	70.0	66.5	83.4	**100.0**	**88.6**	**80.3**	86.9
peach	97.2	3.4	9.3	91.8	99.3	15.5	24.3	96.1	**99.6**	21.5	32.2	**97.7**	99.2	21.3	29.6	96.1	99.3	**37.0**	**40.3**	96.8	99.1	15.0	23.8	96.9
potato	97.5	1.9	5.1	90.7	**99.5**	**14.8**	**25.4**	90.4	97.3	1.7	5.5	94.3	96.9	7.5	14.0	90.8	96.4	10.1	14.1	95.1	98.3	2.9	9.6	**97.1**
rope	**99.2**	7.3	12.9	**97.1**	98.5	11.0	11.5	96.5	99.1	**27.3**	**27.1**	96.8	98.8	5.2	11.7	96.6	94.7	10.3	17.0	93.5	98.9	6.3	15.1	96.2
tire	98.6	16.2	27.6	92.3	98.4	**40.0**	**43.5**	83.2	98.8	9.0	19.3	95.6	95.0	2.9	10.3	81.7	**99.5**	31.9	35.8	93.1	99.0	32.6	37.2	**96.0**
Average	97.3	15.2	19.9	91.3	98.3	20.0	26.5	91.3	98.3	15.8	21.4	95.0	97.3	16.8	24.6	89.7	97.7	**25.9**	30.7	92.6	**99.2**	25.8	**32.0**	**95.4**

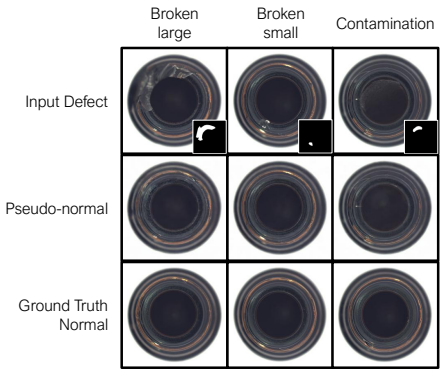


Figure 7: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: bottle).**

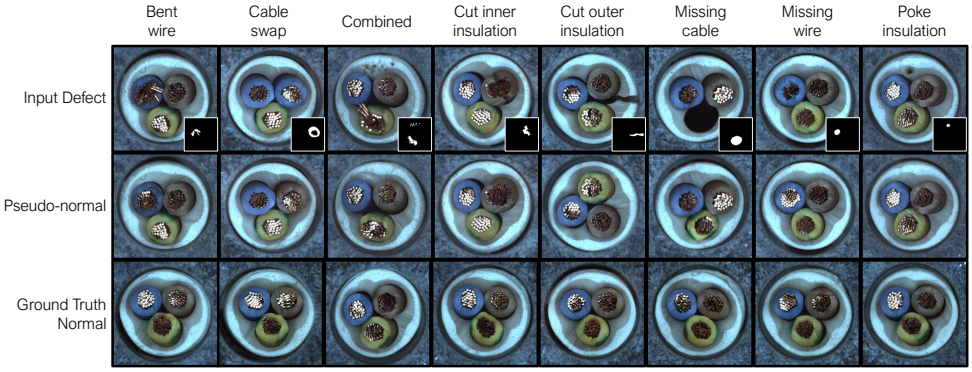


Figure 8: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: cable).**

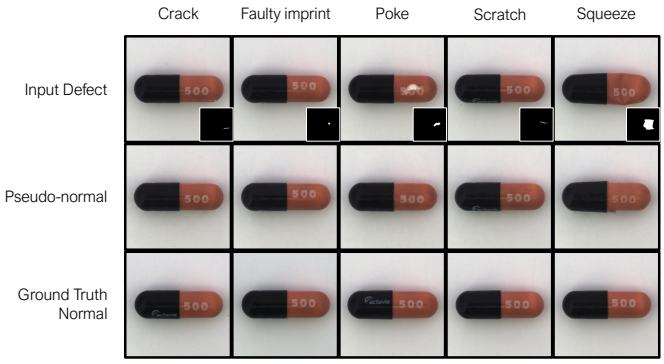


Figure 9: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: capsule).**

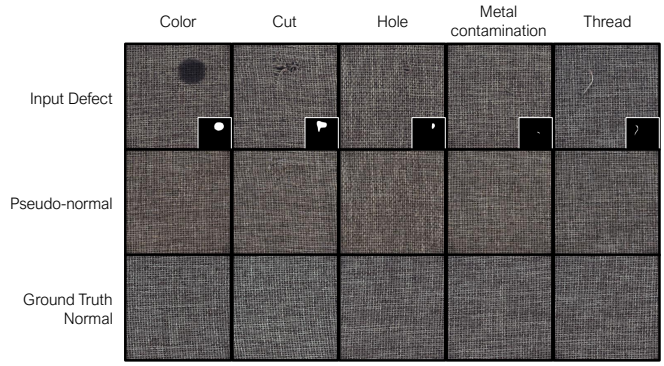


Figure 10: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: carpet).**

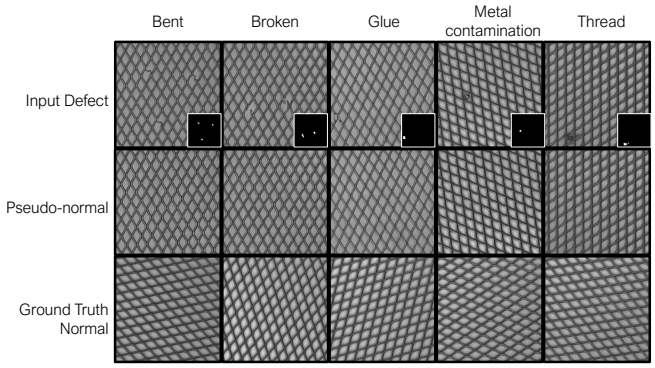


Figure 11: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: grid).**

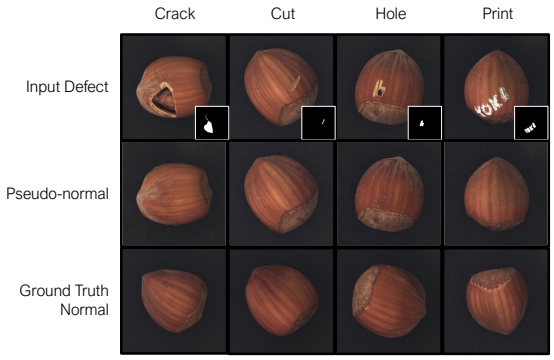


Figure 12: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: hazelnut).**

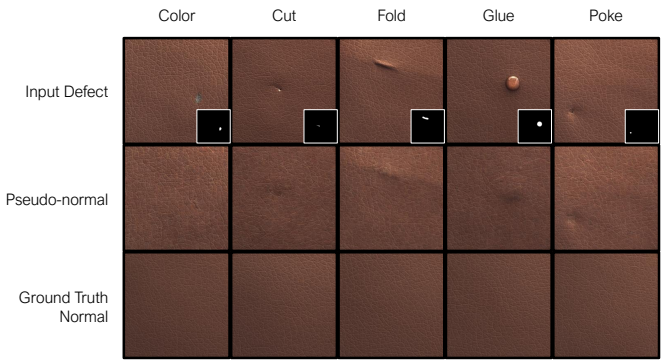


Figure 13: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: leather).**

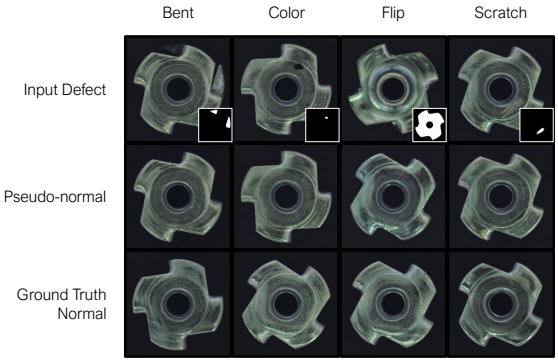


Figure 14: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: metal nut).**

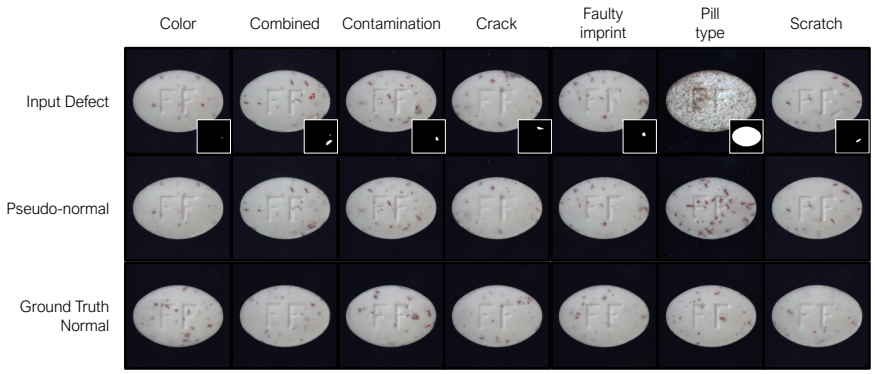


Figure 15: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: pill).**

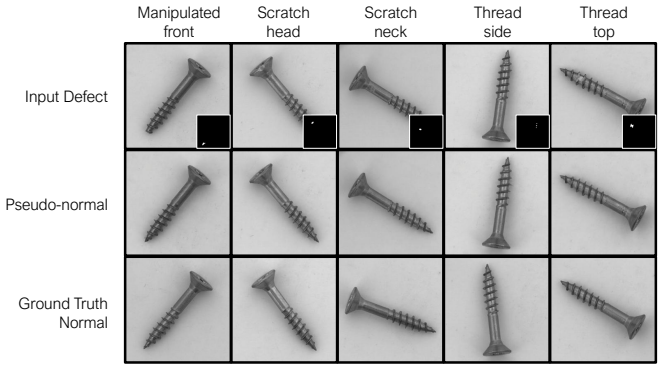


Figure 16: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: screw).**

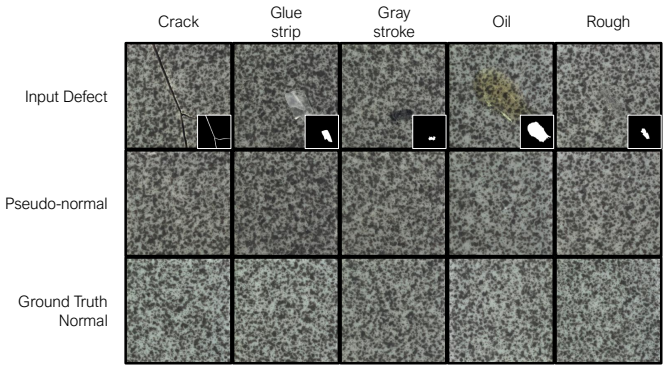


Figure 17: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: tile).**

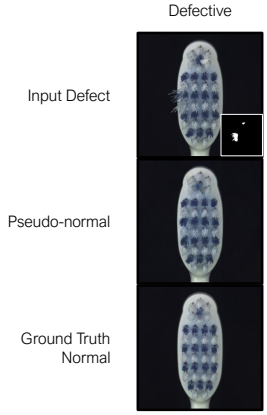


Figure 18: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: toothbrush).**

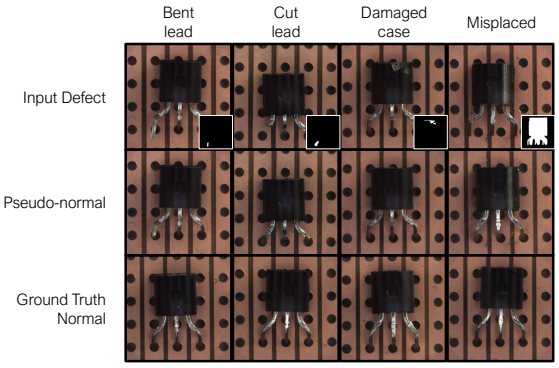


Figure 19: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: transistor).**

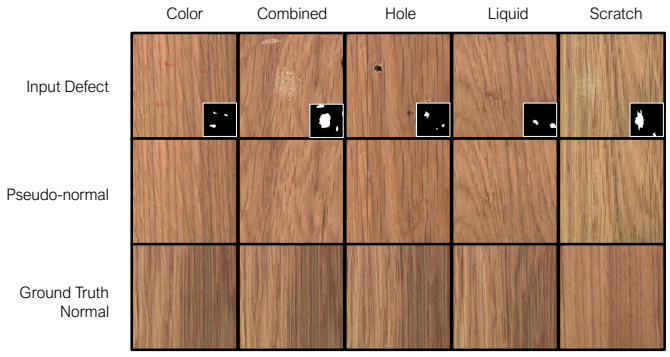


Figure 20: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: wood).**

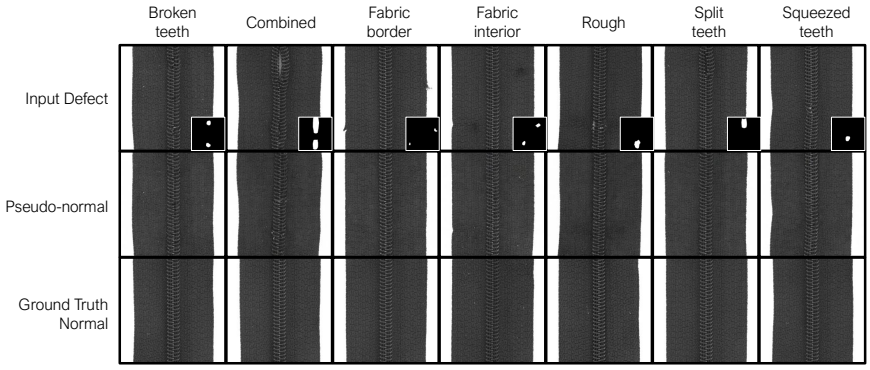


Figure 21: **Reverse (anomaly-to-normal) generation results on MVTec-AD dataset (object: zipper).**

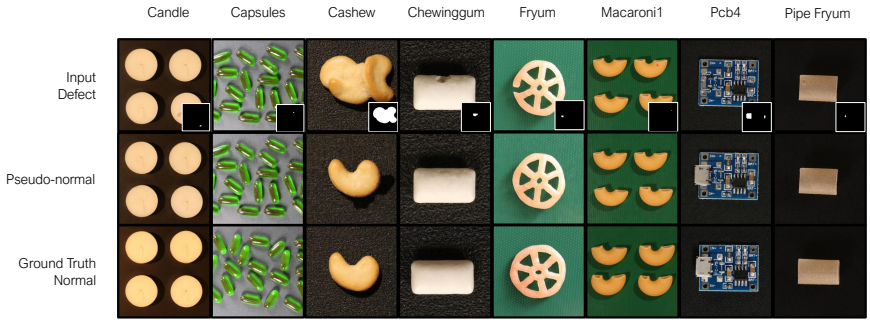


Figure 22: **Reverse (anomaly-to-normal) generation results on VisA dataset.**

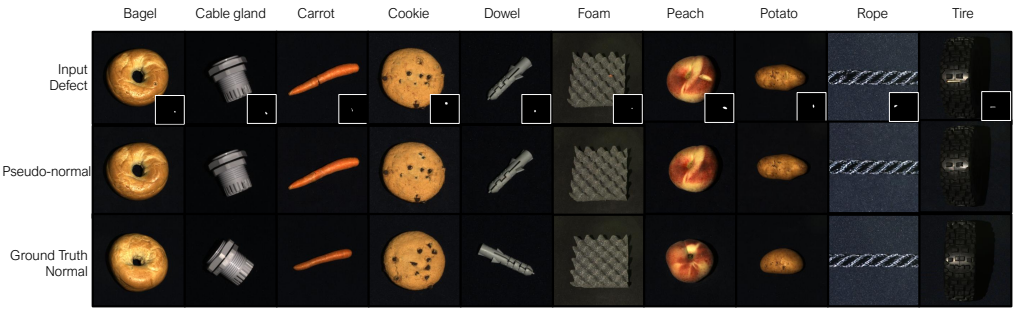


Figure 23: **Reverse (anomaly-to-normal) generation results on MVTec-3D dataset.**

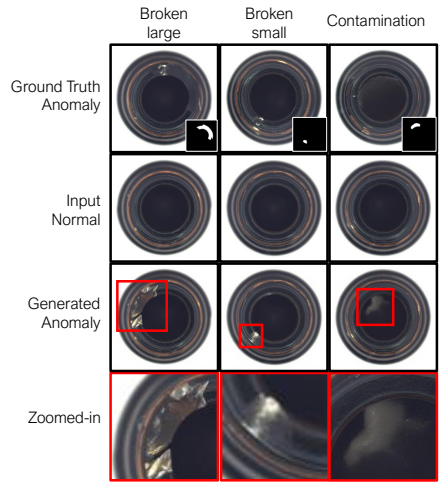


Figure 24: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: bottle).**

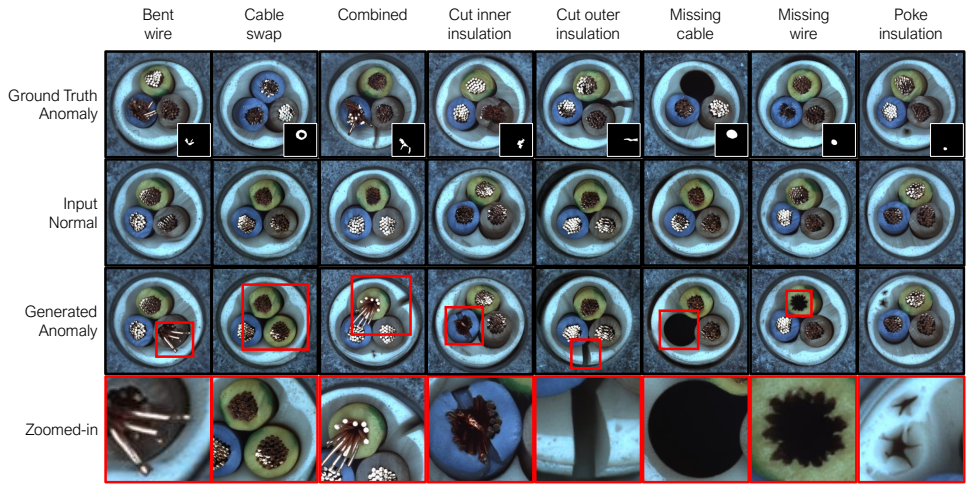


Figure 25: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: cable).**

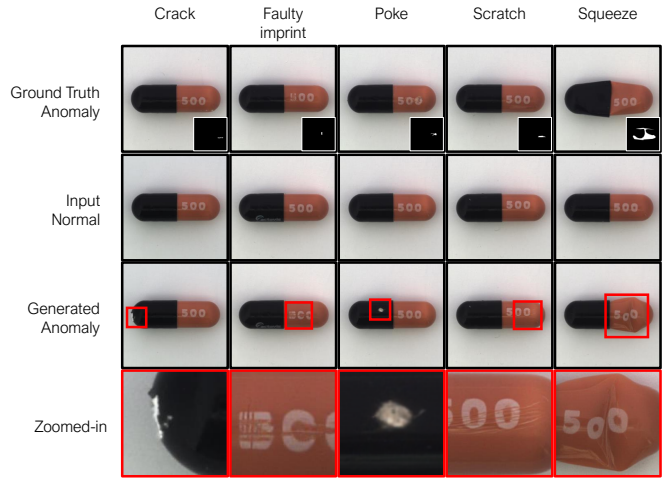


Figure 26: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: capsule).**

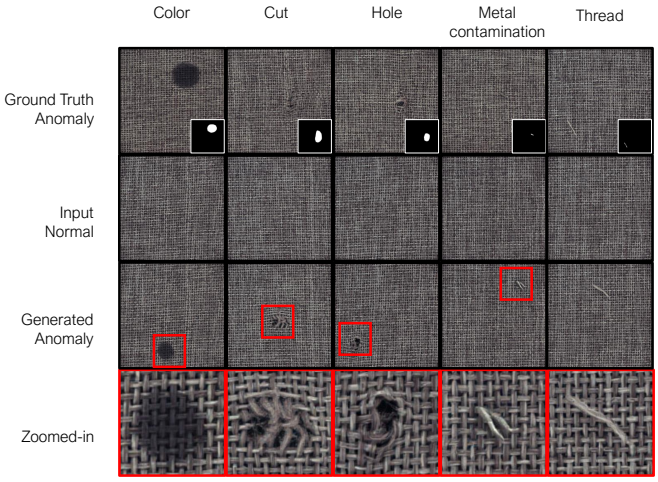


Figure 27: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: carpet).**

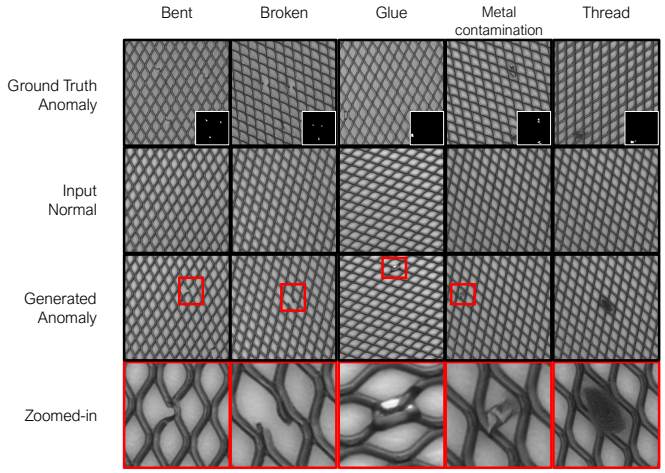


Figure 28: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: grid).**

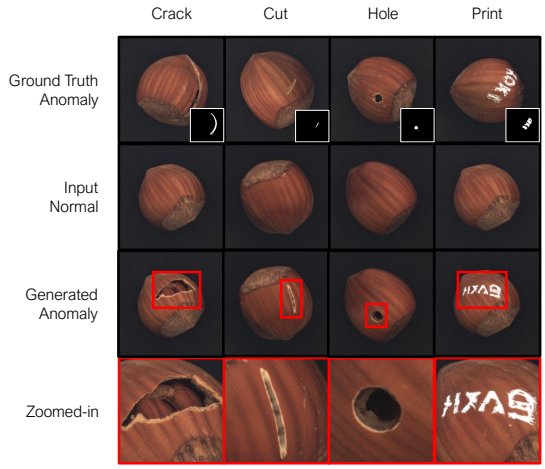


Figure 29: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: hazelnut).**

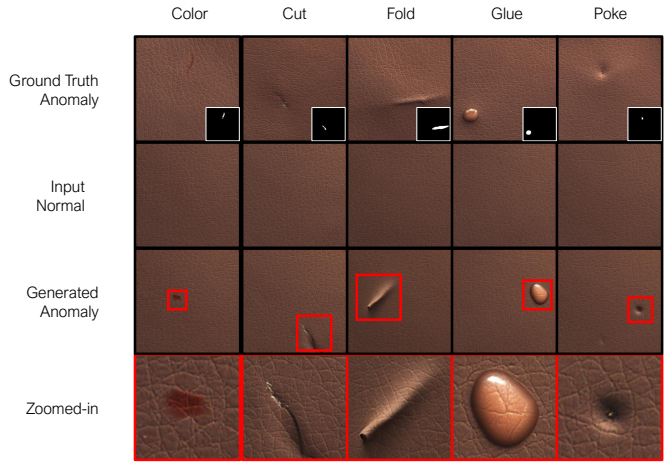


Figure 30: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: leather).**

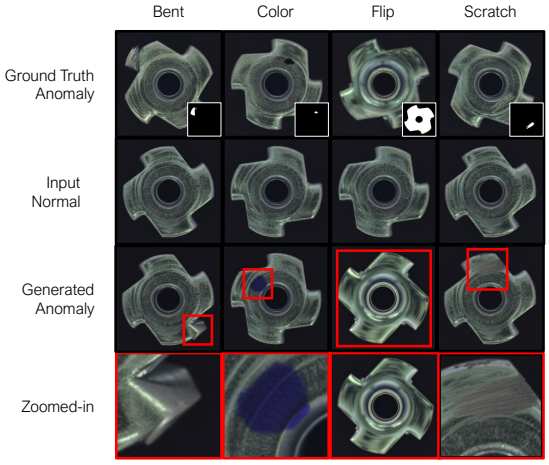


Figure 31: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: metal nut).**



Figure 32: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: pill).**

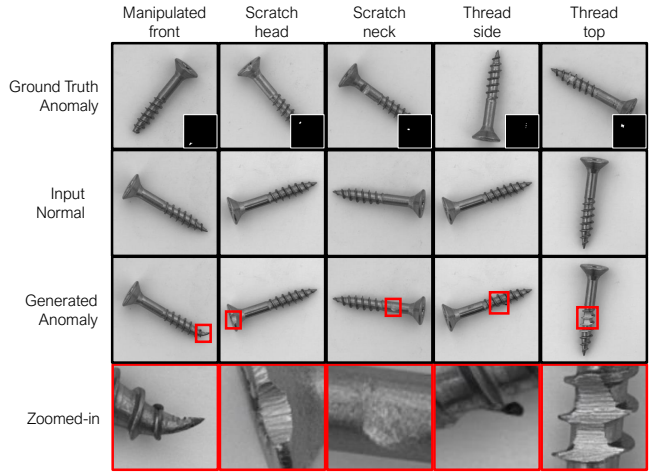


Figure 33: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: screw).**

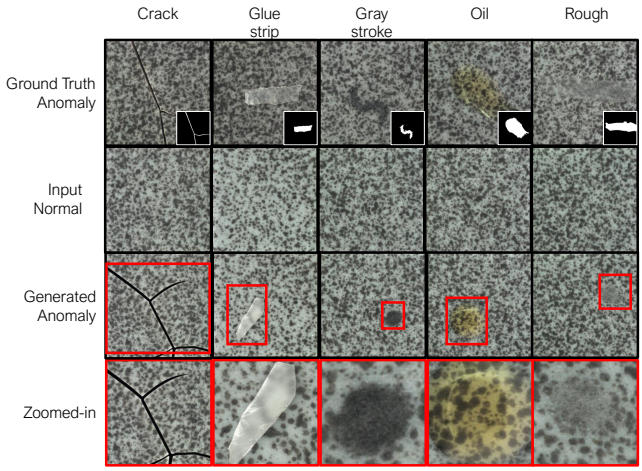


Figure 34: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: tile).**

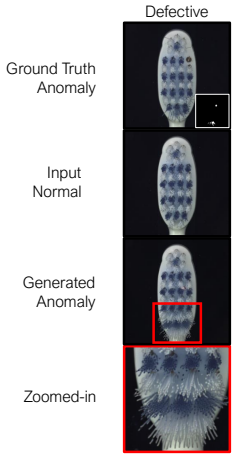


Figure 35: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: toothbrush).**

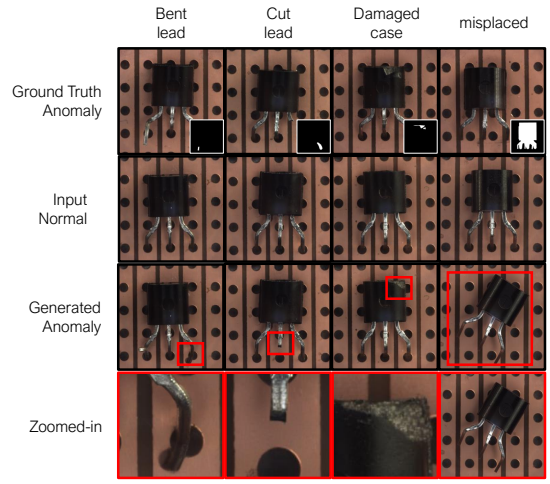


Figure 36: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: transistor).**

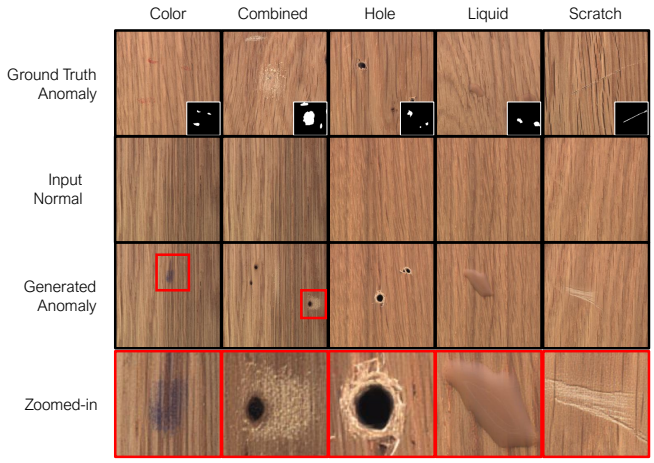


Figure 37: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: wood).**

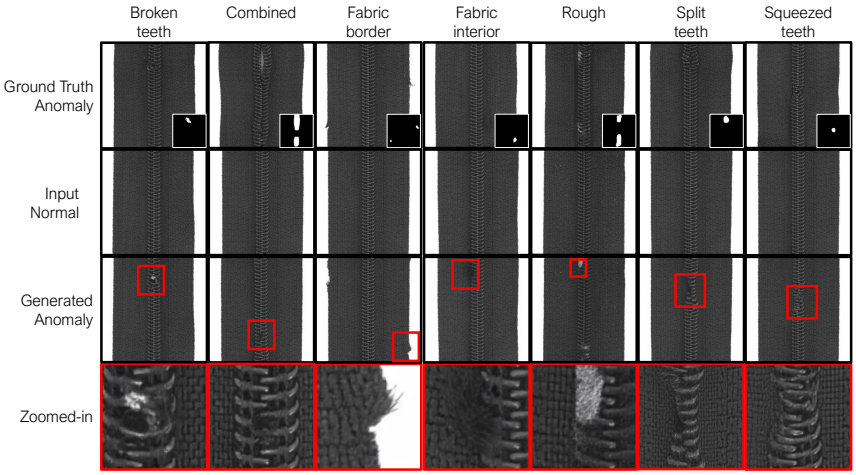


Figure 38: **Forward (normal-to-anomaly) generation results on MVTec-AD dataset (object: zipper).**

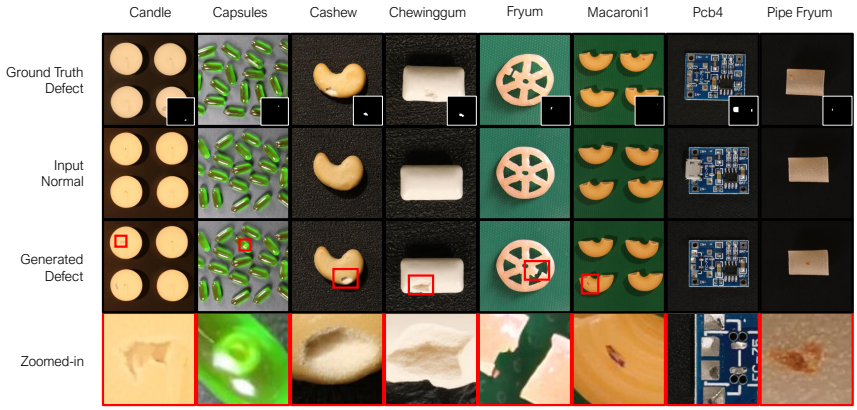


Figure 39: **Forward (normal-to-anomaly) generation results on VisA dataset.**

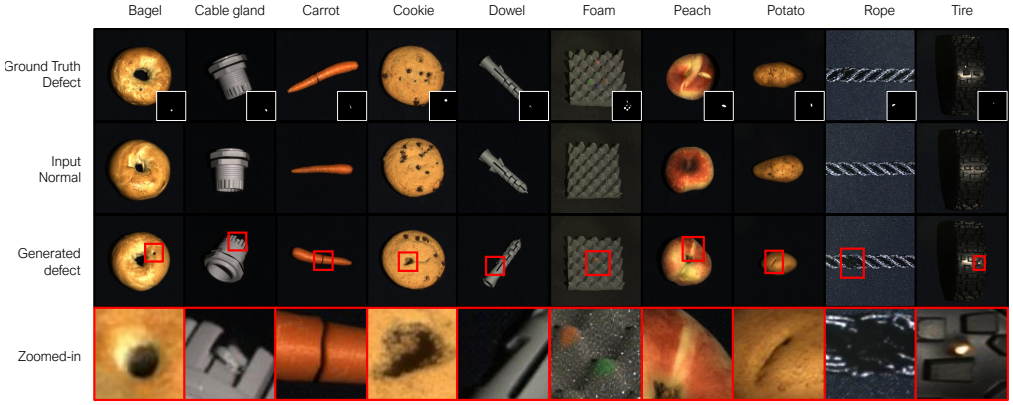


Figure 40: **Forward (normal-to-anomaly) generation results on MVTec-3D dataset.**