SWIFTAD: VISIBILITY-GUIDED HIERARCHICAL TOKEN PRUNING FOR EFFICIENT ZERO-SHOT 3D ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

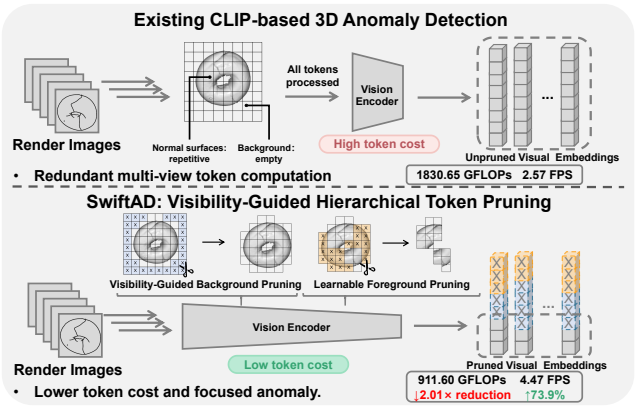
Zero-shot 3D anomaly detection must localize subtle defects without target-domain training data while remaining efficient enough for practical inspection. Existing CLIP-based methods render point clouds into multiple high-resolution views and process all visual tokens, even though many tokens correspond to empty backgrounds, regions without valid 3D support, or redundant normal surfaces. We present SwiftAD, an efficient framework centered on visibility-guided hierarchical token pruning. SwiftAD first converts rendering-derived 3D-to-2D correspondences into a geometric visibility mask and removes unsupported background tokens without additional learnable parameters. It then applies a lightweight anomaly-aware selector to retain informative tokens from the remaining object regions. We find that token pruning introduces a distributional shift in the visual features, weakening their alignment with the original CLIP text prototypes. Textual prototype adaptation therefore learns separate global and local prototypes for object-level recognition and point-level localization, respectively, and applies residual calibration to their anomaly prototypes. Experiments on MVTec3D-AD, Eyecandies, and Real3D-AD show a favorable accuracy–efficiency trade-off. SwiftAD preserves strong zero-shot performance while halving computational cost and improving throughput by 73.9% over the unpruned baseline.

1 INTRODUCTION

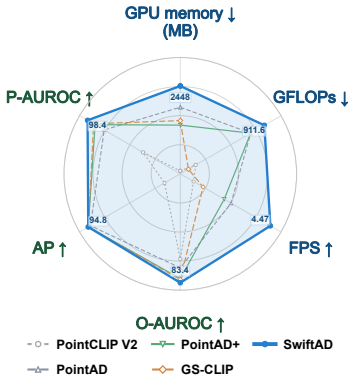
The rapid development of intelligent manufacturing and industrial automation has intensified the demand for industrial anomaly detection (Bergmann et al., 2020; Tao et al., 2022a; Gu et al., 2024). In practice, anomalous samples are rare and costly to annotate (Tao et al., 2022b; Bergmann et al., 2019; Dai et al., 2025). The types of defects also vary widely (Huang et al., 2025b). Privacy, safety, and operating conditions can limit data collection (Tao et al., 2022a; Wang et al., 2024). These limits motivate methods that need only a few normal samples (Gao, 2024). In 3D inspection, defects can be subtle changes in local surface geometry. Collecting point clouds also adds to the data acquisition burden (Bergmann et al., 2021; Bonfiglioli et al., 2022). Vision–language models such as CLIP (Radford et al., 2021) support zero-shot anomaly detection. These approaches do not require target-domain-specific training (Zhou et al., 2023).

A common way to transfer CLIP to 3D inspection is to render a point cloud into multiple high-resolution images and aggregate predictions across views. As shown in Fig. 1a, this strategy introduces a substantial computational burden because the visual encoder repeatedly processes all patch tokens in every view. Many of these tokens correspond to empty backgrounds or image regions without valid 3D-point support; even within the visible object region, large portions describe repetitive normal surfaces. In our experimental setting, the unpruned pipeline requires 1,830.65 GFLOPs for a nine-view object and processes only 2.57 objects per second, making token reduction imperative. Directly discarding tokens, however, is risky: dense anomaly localization depends on small, spatially sparse defects, and indiscriminate pruning can easily remove the very evidence required for point-level prediction. Moreover, pruning alters the distribution of the retained visual features, weakening their alignment with the original CLIP text prototypes. The central challenge is thus not merely to prune more tokens, but to identify and remove 3D-specific redundancy while preserving anomaly-sensitive foreground information and accommodating the resulting representation shift.

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.



(a) Motivation and framework comparison.



(b) Efficiency–accuracy comparison on MVTec3D-AD.

Figure 1: **Overview and performance of SwiftAD. (a) Motivation and framework comparison:** Existing CLIP-based 3D anomaly detectors process all tokens from multi-view renderings, causing redundant computation on background and normal regions. SwiftAD removes geometrically unsupported background tokens and then retains informative foreground tokens. **(b) Efficiency–accuracy comparison:** On MVTec3D-AD, SwiftAD achieves a favorable accuracy–efficiency trade-off, with lower peak GPU memory and GFLOPs and higher throughput.

To address this challenge, we propose **SwiftAD**, an efficient zero-shot 3D anomaly detection framework centered on visibility-aware dual-stage token pruning (VA-DSTP). The first stage, visibility-guided background pruning (VGBP), uses rendering-derived 3D-to-2D correspondences as a geometric prior to deterministically remove tokens without valid visible-point support while preserving a narrow boundary context. The second stage, learnable foreground pruning (LFP), applies a lightweight selector to retain informative tokens from the remaining visible object region. The experimental results substantiate both the efficiency and accuracy of this design. SwiftAD achieves a $2.01\times$ reduction in computation, from 1,830.65 to 911.60 GFLOPs. Throughput rises from 2.57 to 4.47 FPS, a 73.9% increase over the unpruned baseline. Figure 1b shows this accuracy–efficiency trade-off for zero-shot 3D anomaly detection. Our key contributions are summarized as follows:

- Motivated by the observation that rendering-derived geometric visibility is a free prior, we propose **SwiftAD**, an efficient zero-shot framework that transfers CLIP to 3D anomaly detection through visibility-guided hierarchical token reduction, removing unsupported tokens while preserving anomaly-sensitive foreground features.
- We introduce visibility-aware dual-stage token pruning (VA-DSTP), which removes unsupported background tokens via rendering-derived 3D-to-2D correspondences and then retains anomaly-relevant foreground tokens through a lightweight selector in the frozen CLIP vision encoder. Textual prototype adaptation via decoupling and residual calibration is applied to accommodate the resulting distributional shift.
- Experiments on MVTec3D-AD, Eyecandies, and Real3D-AD show a favorable accuracy–efficiency trade-off for zero-shot 3D anomaly detection. SwiftAD maintains strong performance while halving computation relative to the unpruned baseline. It also increases throughput by 73.9% over the unpruned baseline.

2 Related Work

2.1 3D Industrial Anomaly Detection

Industrial 3D anomaly detection aims to identify anomalous objects and localize surface defects. Existing methods mainly follow reconstruction- or memory-based paradigms. Reconstruction methods detect discrepancies from normal reconstructions, such as masking-based IMRNet (Li et al., 2024) and diffusion-based R3D-AD (Zhou et al., 2024b). Memory-based methods instead compare test features with stored normal representations, including Reg3D-AD (Liu et al., 2023), M3DM (Wang et al., 2023), CPMF (Cao et al., 2024a), and ISMP (Liang et al., 2025). Despite strong performance,

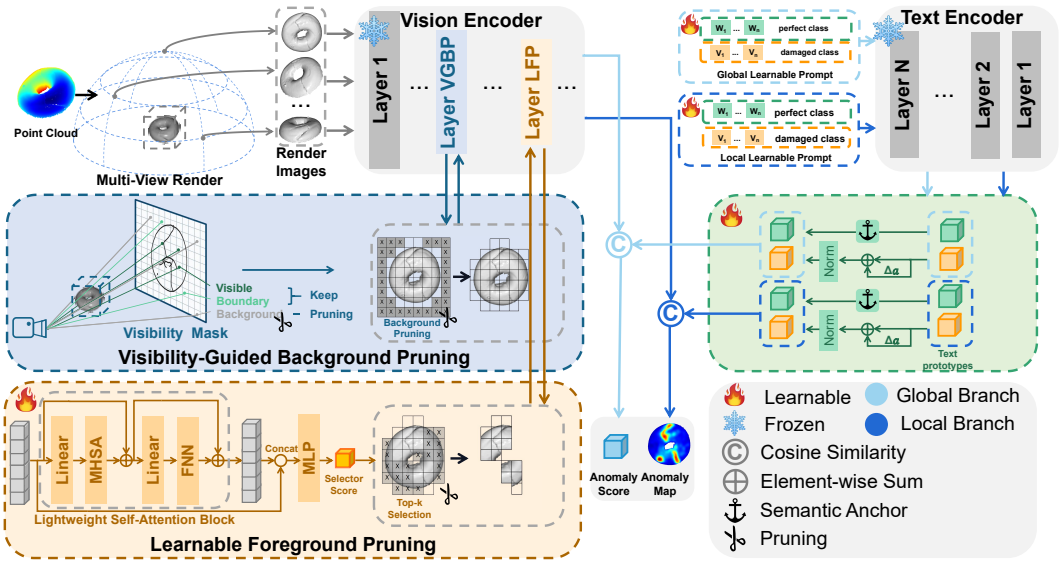


Figure 2: Overall framework of SwiftAD. Multi-view images are processed by the frozen CLIP vision encoder, while VGBP removes background and geometrically invalid tokens and LFP retains anomaly-relevant foreground tokens. Textual prototype adaptation via decoupling and residual calibration supports object-level recognition and point-level localization, producing the anomaly score and anomaly map through cosine similarity and multi-view aggregation.

these methods require target-domain normal samples and category-specific adaptation, motivating zero-shot approaches that generalize without target-domain-specific training.

2.2 CLIP-based Zero-shot Anomaly Detection

CLIP (Radford et al., 2021) enables zero-shot anomaly detection by aligning visual observations with textual descriptions of normal and abnormal states. In 2D inspection, WinCLIP (Jeong et al., 2023) uses handcrafted prompt ensembles, while APRIL-GAN, AnomalyCLIP, AdaCLIP, and VCP-CLIP (Chen et al., 2023; Zhou et al., 2023; Cao et al., 2024b; Qu et al., 2024) improve semantic alignment and localization through feature adaptation or prompt learning. This paradigm has been extended to 3D data through multi-view rendering. PointCLIP (Zhang et al., 2022; Zhu et al., 2023) demonstrates projected-view recognition, whereas MVP-PCLIP, PointAD, and PointAD+ (Cheng et al., 2025; Zhou et al., 2024a; 2025) introduce specialized rendering and prompt learning for 3D anomaly detection. GS-CLIP (Deng et al., 2026) further incorporates geometric priors and depth views. However, these methods process all tokens from high-resolution multi-view inputs, introducing substantial redundancy from background, invisible, and anomaly-irrelevant regions. SwiftAD instead removes such redundancy within the CLIP visual encoder.

2.3 Efficient Vision Transformers and Visual Token Pruning

Visual token pruning accelerates Vision Transformers and multimodal models by removing uninformative or highly redundant visual tokens. Training-free methods select tokens using attention, similarity, or spatial diversity (Chen et al., 2024; Shang et al., 2025; Chen et al., 2026; Alvar et al., 2025; Fang et al., 2026), whereas learning-based methods predict token importance under downstream supervision (Huang et al., 2025a; Zhu et al., 2025). However, most existing methods target object-level recognition or reasoning and may discard subtle defects required for dense anomaly localization. They also rarely exploit the geometric visibility inherent in rendered point clouds. Unlike these methods, which rely on attention or similarity heuristics, SwiftAD exploits the geometric visibility inherent in 3D rendering which is a prior unavailable in 2D tasks.

3 METHODOLOGY

3.1 OVERVIEW

SwiftAD targets efficient zero-shot 3D anomaly detection by building upon a frozen CLIP backbone. Given a 3D object, SwiftAD renders multiple 2D views while preserving point-to-pixel correspondences, which guide token pruning within the frozen CLIP image encoder and project 2D anomaly responses back to the 3D point domain. As shown in Fig. 2, visibility-aware dual-stage token pruning (VA-DSTP) first removes geometrically unsupported background tokens through visibility-guided background pruning (VGBP) and then selects anomaly-relevant foreground tokens through learnable foreground pruning (LFP). To compensate for the pruning-induced distribution shift, textual prototype adaptation learns decoupled global and local prototypes for object-level recognition and point-level localization, followed by residual calibration.

3.2 MULTI-VIEW POINT CLOUD RENDERING

To leverage the visual knowledge of CLIP for 3D anomaly detection, we adopt a rendering-based strategy to transfer 3D anomaly detection into the 2D vision-language space. For each point-cloud object, we generate a set of multi-view images by rotating the object around its geometric center. Let Θ_x and Θ_y denote the predefined rotation angle sets around the X-axis and Y-axis, respectively. Each rendered view is obtained by the composite rotation:

$$\mathbf{R}_{ij} = \mathbf{R}_y(\theta_j)\mathbf{R}_x(\theta_i), \quad \theta_i \in \Theta_x, \theta_j \in \Theta_y. \tag{1}$$

The rotated point cloud is projected onto the 2D image plane with a virtual camera. For a 3D point $\mathbf{p}'_k = (x'_k, y'_k, z'_k)^\top$, its image coordinate $\mathbf{u}_k = (u_k, v_k)^\top$ is computed by:

$$s \begin{bmatrix} u_k \\ v_k \\ 1 \end{bmatrix} = \mathbf{K} \left[\mathbf{R}_{cam} \mid \mathbf{t}_{cam}\right] \begin{bmatrix} x'_k \\ y'_k \\ z'_k \\ 1 \end{bmatrix}, \tag{2}$$

where $\mathbf{K}$ is the camera intrinsic matrix, $\mathbf{R}_{cam}$ and $\mathbf{t}_{cam}$ are the camera extrinsic parameters, and s is the depth scale.

To handle self-occlusion, we maintain a visibility indicator for each projected point. If multiple 3D points are projected to the same 2D pixel, only the point with the nearest camera-space depth is regarded as visible:

$$m_k = \mathbb{1}\left(z'_k = \min\{z'_q \mid \pi(\mathbf{p}'_q) = \pi(\mathbf{p}'_k)\}\right). \tag{3}$$

Therefore, each rendered view is associated with a visibility-aware 3D-to-2D correspondence map, which records the projected coordinate (u_k, v_k) and visibility indicator m_k for each 3D point. This correspondence map provides geometric support for visibility-aware token pruning and enables the back-projection of 2D anomaly responses to the 3D point cloud.

3.3 VISIBILITY-GUIDED BACKGROUND PRUNING

Multi-view rendering for 3D AD inherently produces per-pixel 3D-to-2D correspondences that encode geometric visibility, yet all existing methods discard this information and process every token uniformly. We observe that this free geometric prior can directly identify which tokens have valid 3D support, enabling parameter-free pruning that no 2D token pruning method can replicate. This first pruning stage, termed Visibility-Guided Background Pruning (VGBP), preserves a narrow spatial context around the visible object boundary.

For each rendered view, the rendering process provides the projected coordinate (u_k, v_k) and visibility indicator m_k for every 3D point $\mathbf{p}_k$. A pixel is identified as foreground if at least one visible 3D point projects onto it:

$$V(u, v) = \mathbb{I}\left[\exists k,\ (u_k, v_k) = (u, v) \wedge m_k = 1\right], \tag{4}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function and $\mathbf{V} \in \{0, 1\}^{H \times W}$ provides an explicit foreground–background partition. Given the CLIP image encoder patch size P, we aggregate $\mathbf{V}$ into a patch-level *Visible* mask:

$$M_{\text{vis}}^{(i)} = \max_{(u,v) \in \Omega_i} V(u, v), \qquad i = 1, \ldots, N_p, \tag{5}$$

where $N_p = \frac{H}{P} \times \frac{W}{P}$ and Ω_i is the pixel support of the i-th patch.

Directly removing all non-visible patches may discard useful contextual cues around the object contour. We therefore distinguish the immediate spatial neighbors of Visible patches as *Boundary*, while all remaining patches are labeled *Background*. Let $\overline{\mathcal{N}}_8(i) = \mathcal{N}_8(i) \cup \{i\}$ denote the closed 8-connected neighborhood. The Boundary mask is the one-patch dilation ring around the Visible region:

$$M_{\mathrm{bd}}^{(i)} = \left(1 - M_{\mathrm{vis}}^{(i)}\right) \max_{j \in \overline{\mathcal{N}}_8(i)} M_{\mathrm{vis}}^{(j)}, \qquad i = 1, \ldots, N_p. \tag{6}$$

We retain Visible and Boundary tokens and prune only Background tokens:

$$M_{\mathrm{keep}}^{(i)} = M_{\mathrm{vis}}^{(i)} + M_{\mathrm{bd}}^{(i)} = 1 - M_{\mathrm{bg}}^{(i)}. \tag{7}$$

We apply $\mathbf{M}_{\mathrm{keep}}$ at an intermediate ViT layer ℓ_b rather than at the input. Shallow self-attention layers rely on full token interaction to build contextual representations; pruning before this context has formed would irreversibly discard tokens whose significance only emerges through inter-token communication. The CLS token is always retained; Background tokens are removed, whereas Visible and Boundary tokens are compacted into a shorter sequence for the subsequent layers.

3.4 Learnable Foreground Pruning

Although VGBP removes background tokens without visible 3D support, the retained tokens may still contain redundant normal-surface patterns. Since anomalies are typically sparse and localized, we further introduce Learnable Foreground Pruning (LFP) to adaptively retain the most informative tokens among the visible object regions.

Let $\mathbf{X} \in \mathbb{R}^{N_f \times D}$ denote the compact visible-token sequence after VGBP, where N_f is the number of retained patch tokens and D is the feature dimension. The selector first enhances token representations through a lightweight self-attention block and a feed-forward network:

$$\widetilde{\mathbf{X}} = \mathbf{X} + \mathrm{MHSA}\left(\mathrm{LN}(\mathbf{X})\right), \tag{8}$$

$$\widehat{\mathbf{X}} = \widetilde{\mathbf{X}} + \mathrm{FFN}\left(\mathrm{LN}(\widetilde{\mathbf{X}})\right). \tag{9}$$

To improve generalization to unseen categories, we evaluate each token using both its local feature and sample-adaptive global context. Specifically, we compute the channel-wise mean and maximum over the context-enhanced token sequence:

$$\bar{\mathbf{x}} = \frac{1}{N_f} \sum_{j=1}^{N_f} \widehat{\mathbf{x}}_j, \qquad \mathbf{x}_{\max} = \max_{j=1,\ldots,N_f} \widehat{\mathbf{x}}_j. \tag{10}$$

For each token, we then construct the context-relative representation

$$\mathbf{z}_i = \left[\widehat{\mathbf{x}}_i; \bar{\mathbf{x}}; \mathbf{x}_{\max}; |\widehat{\mathbf{x}}_i - \bar{\mathbf{x}}|\right]. \tag{11}$$

The mean and maximum features summarize the current sample, while the element-wise absolute difference measures the deviation of each token from the sample-level context. This context-relative representation reduces reliance on category-specific appearance patterns. An MLP head then predicts the importance logit:

$$s_i = \mathrm{MLP}\left(\mathrm{LN}(\mathbf{z}_i)\right). \tag{12}$$

Given a target keep ratio r, LFP performs sample-wise top-k selection over the valid tokens retained by VGBP, where $k = \max\left(1, \mathrm{round}(rN_f)\right)$. The forward pass uses an exact binary top-k mask $\mathbf{m}^{\mathrm{hard}}$, ensuring genuine token sparsity in the subsequent Transformer layers. During optimization, we use a straight-through estimator:

$$m_i = m_i^{\mathrm{hard}} + \sigma\left(\frac{s_i - \tau_k}{T}\right) - \mathrm{sg}\left[\sigma\left(\frac{s_i - \tau_k}{T}\right)\right], \tag{13}$$

where τ_k is the sample-wise top-k threshold, T is the temperature, and $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator.

Token compaction preserves the original spatial indices, and only tokens selected by LFP are processed by the subsequent Transformer layers. After the final visual layer, the retained features are scattered back to their original positions in the patch grid, while pruned positions are filled with zeros before output normalization and projection. The restored grid is compared with the local text prototypes to produce patch-level anomaly maps, which are upsampled to the image resolution, back-projected to visible 3D points using the rendering correspondences, and aggregated across views.

VGBP and LFP are inserted at layers ℓ_b and ℓ_f, respectively, where $\ell_b < \ell_f$. This coarse-to-fine procedure first removes geometrically unsupported background tokens and then selects anomaly-relevant tokens from the foreground and boundary regions retained by VGBP.

3.5 Textual Prototype Adaptation

Token pruning affects global and local visual representations differently. The CLS representation aggregates information from the retained sequence, whereas patch-level representations are directly altered by spatial token removal. Consequently, sharing a single text prototype pair between object-level recognition and point-level localization may produce conflicting alignment objectives.

We therefore employ two independent prompt learners in Stage 1. A global prompt learner $\mathcal{P}_g$ is optimized for object-level recognition, while a local prompt learner $\mathcal{P}_l$ is optimized for point-level localization. The CLIP text encoder produces the corresponding normalized prototype pairs:

$$\mathbf{T}_b = \mathrm{Norm}\left(E_t(\mathcal{P}_b)\right), \qquad b \in \{g, l\}, \tag{14}$$

where each $\mathbf{T}_b \in \mathbb{R}^{2\times D}$ contains a normal prototype $\mathbf{t}_n^b$ and an anomaly prototype $\mathbf{t}_a^b$.

In Stage 2, the prompt-derived prototypes are fixed as semantic anchors. Because foreground pruning concentrates the retained sequence toward anomaly-relevant regions, we adapt only the anomaly prototype through a lightweight residual:

$$\widetilde{\mathbf{t}}_n^b = \mathbf{t}_n^b, \qquad \widetilde{\mathbf{t}}_a^b = \mathrm{Norm}\left(\mathbf{t}_a^b + \alpha_b \Delta_a^b\right), \qquad b \in \{g, l\}, \tag{15}$$

where Δ_a^b and α_b are the branch-specific anomaly residual and its learnable scale, respectively. Keeping the normal prototypes fixed preserves the source-domain semantic anchors, while the independent anomaly residuals accommodate the different recognition and localization requirements induced by token pruning.

3.6 Loss Design

SwiftAD is optimized in two stages while keeping the CLIP backbone frozen. We first define a task loss that jointly supervises object-level recognition and dense anomaly localization:

$$\mathcal{L}_{\mathrm{task}} = \underbrace{\mathcal{L}_{\mathrm{CE}}(\mathbf{p}_{\mathrm{view}}, y_{\mathrm{view}}) + \mathcal{L}_{\mathrm{CE}}(\mathbf{p}_{\mathrm{obj}}, y_{\mathrm{obj}})}_{\mathcal{L}_{\mathrm{global}}} + \underbrace{\mathcal{L}_{\mathrm{Focal}}(\mathbf{S}_{\mathrm{2D}}, \mathbf{Y}_{\mathrm{2D}}) + \mathcal{L}_{\mathrm{Dice}}(\mathbf{S}_{\mathrm{2D}}, \mathbf{Y}_{\mathrm{2D}}) + \mathcal{L}_{\mathrm{Dice}}(\mathbf{S}_{\mathrm{3D}}, \mathbf{Y}_{\mathrm{3D}})}_{\mathcal{L}_{\mathrm{local}}}. \tag{16}$$

Here, $\mathbf{p}_{\mathrm{view}}$ denotes the prediction for each rendered view, and $\mathbf{p}_{\mathrm{obj}}$ is obtained by aggregating the predictions across views. The 2D anomaly map $\mathbf{S}_{\mathrm{2D}}$ is projected to the point-cloud domain using the rendering-derived correspondences:

$$\mathbf{S}_{\mathrm{3D}}(i) = \frac{\sum_{v=1}^{N_v} m_i^v \mathbf{S}_{\mathrm{2D}}^v(u_i^v, v_i^v)}{\sum_{v=1}^{N_v} m_i^v + \epsilon}. \tag{17}$$

In Stage 1, we optimize the decoupled global and local prompt learners using

$$\mathcal{L}_{\mathrm{stage1}} = \mathcal{L}_{\mathrm{task}}, \tag{18}$$

while keeping the foreground selector frozen.

In Stage 2, the prompt-derived prototypes are fixed as semantic anchors, and only the branch-specific anomaly residuals and foreground selector are optimized. The selector loss combines a sparsity constraint with anomaly-aware token supervision:

$$\mathcal{L}_{\mathrm{sel}} = \lambda_{\mathrm{sp}} \mathcal{L}_{\mathrm{sp}} + \lambda_{\mathrm{focus}} \mathcal{L}_{\mathrm{focus}}, \tag{19}$$

Table 1: Zero-shot comparison on MVTec3D-AD, Eyecandies, and Real3D-AD, together with computational efficiency measurements. The best and second-best results in each column are highlighted in bold and underlined, respectively. Lower values are better for memory and GFLOPs, whereas higher values are better for all other metrics.

Method	Efficiency			MVTec3D-AD				Eyecandies				Real3D-AD		
				Source: Eyecandies				*Source: MVTec3D-AD*				*Source: MVTec3D-AD*		
	Memory	GFLOPs	FPS	O-AUROC	AP	P-AUROC	AUPRO	O-AUROC	AP	P-AUROC	AUPRO	O-AUROC	AP	P-AUROC
CPMF (Unsupervised)	–	–	–	95.2	98.2	93.5	92.9	–	–	–	–	–	–	–
PointCLIP V2	9747 MB	4660.49	0.66	78.3	49.4	87.4	–	48.5	50.9	46.3	–	57.4	58.8	51.9
AnomalyCLIP	3348 MB	1656.34	5.26	65.3	87.6	92.5	74.7	62.4	62.4	89.8	69.7	52.7	55.7	14.8
MVP-PCLIP	5694 MB	**467.35**	**8.03**	71.9	89.6	87.5	–	54.8	56.7	72.2	–	74.9	75.6	73.9
PointAD	4275 MB	1656.34	2.52	78.9	93.2	94.3	80.6	77.2	79.9	94.5	83.6	75.9	78.9	71.6
PointAD+	5811 MB	1658.34	2.18	82.5	**94.8**	96.9	88.7	80.3	83.1	94.8	85.2	74.3	78.2	72.5
GS-CLIP	5424 MB	5055.34	1.13	82.7	94.7	97.2	90.6	81.5	83.5	95.4	83.3	76.2	77.1	74.7
SwiftAD (Ours)	**2448 MB**	911.60	4.47	**83.4**	**94.8**	**98.4**	**93.7**	**83.2**	**84.7**	**96.9**	**87.0**	**79.3**	**82.4**	**81.8**

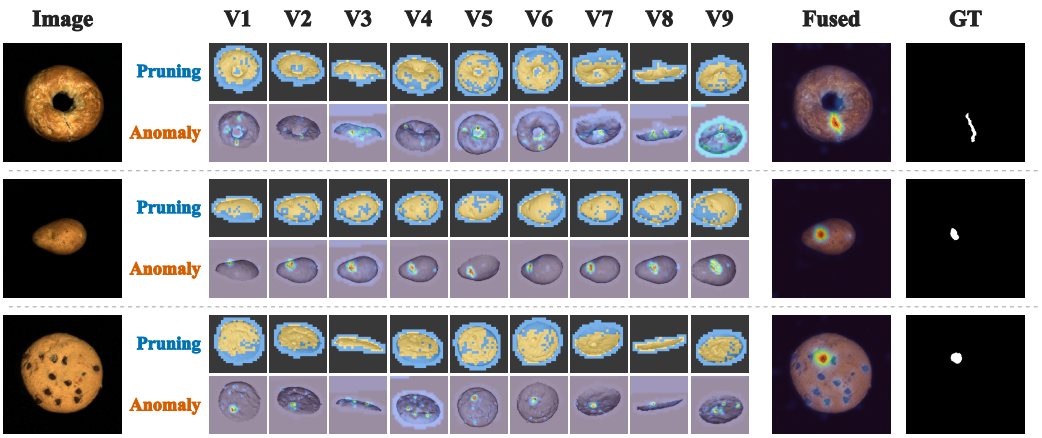


Figure 3: **Multi-view pruning and anomaly localization on MVTec3D-AD.** V1–V9 show pruning masks in the upper rows and anomaly heatmaps in the lower rows. Fused shows the final anomaly heatmap after multi-view aggregation, and GT denotes the ground-truth defect mask.

where $\mathcal{L}_{\text{sp}}$ combines soft keep-ratio alignment with score-spread regularization, and $\mathcal{L}_{\text{focus}}$ combines weighted patch-level binary classification with an anomaly-coverage penalty. To prevent the learned anomaly prototypes from deviating excessively from their Stage-1 semantic anchors, we regularize the anomaly residuals as

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{anchor}} \sum_{b\in\{g,l\}} \left\|\Delta_a^b\right\|_2^2. \tag{20}$$

The complete Stage-2 objective is therefore

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{sel}} + \mathcal{L}_{\text{reg}}. \tag{21}$$

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate SwiftAD under the **cross-dataset zero-shot** setting on MVTec3D-AD (Bergmann et al., 2021), Eyecandies (Bonfiglioli et al., 2022), and Real3D-AD (Liu et al., 2023), without target-category training samples. Specifically, when evaluating on MVTec3D-AD, the model is trained on Eyecandies; when evaluating on Eyecandies and Real3D-AD, the model is trained on MVTec3D-AD and transferred directly without any further target-domain adaptation. The first two datasets contain ten categories with paired RGB and geometric data, whereas Real3D-AD provides high-resolution point clouds of real industrial objects. We report object-level AUROC (O-AUROC) and AP for object-level recognition and P-AUROC and AUPRO for point-level localization.

Table 2: Component ablation on MVTec3D-AD. VGBP removes background tokens, LFP selects informative tokens from the regions retained by VGBP, and PA adapts the pruning-induced representation shifts. The best and second-best results are shown in **bold** and underline.

VGBP	LFP	PA	O-AUROC	AP	P-AUROC	AUPRO	GFLOPs	FPS
×	×	×	75.1	92.1	94.3	82.2	1830.65	2.57
✓	×	×	58.7	83.9	85.2	51.3	1165.12	3.71
✓	✓	×	66.8	87.2	86.7	67.6	**911.60**	**4.47**
✓	×	✓	**83.9**	**95.2**	98.1	93.1	1165.12	3.70
✓	✓	✓	83.4	94.8	**98.4**	**93.7**	**911.60**	**4.47**

Table 3: Ablation study of pruning layers and foreground keep ratio on MVTec3D-AD. **Bold**: best, underline: second-best.

VGBP Layer	LFP Layer	Keep Ratio	O-AUROC	AP	P-AUROC	AUPRO	FPS
1	6	0.25	52.2	81.5	89.2	57.1	**7.07**
1	6	0.50	51.9	81.2	90.0	59.4	5.20
1	6	0.75	53.9	82.5	89.4	55.4	4.12
1	12	0.25	79.9	93.4	97.9	93.2	5.14
1	12	0.50	83.0	94.6	98.1	93.3	4.36
1	12	0.75	82.3	**95.2**	98.1	93.2	3.81
6	12	0.25	79.0	93.0	97.9	93.5	5.23
6	12	0.50	**83.4**	94.8	**98.4**	**93.7**	4.47
6	12	0.75	53.5	81.8	89.6	62.1	3.88
6	18	0.25	55.9	83.7	96.7	91.5	4.09
6	18	0.50	79.7	93.2	98.2	**93.7**	3.81
6	18	0.75	52.5	81.2	90.6	63.5	3.58
12	18	0.25	56.7	83.4	90.9	65.9	4.18
12	18	0.50	53.3	81.7	87.1	53.1	3.90
12	18	0.75	52.5	81.2	87.2	53.5	3.65

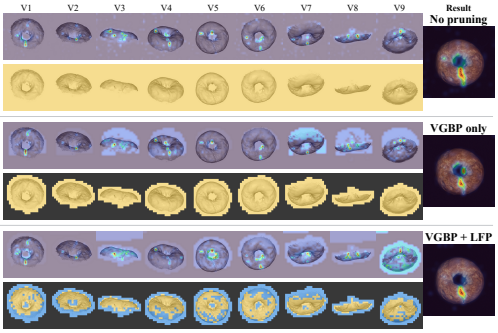


Figure 4: **Qualitative comparison of pruning settings.** Each group shows anomaly heatmaps above pruning masks across nine views, with the fused anomaly heatmap on the right.

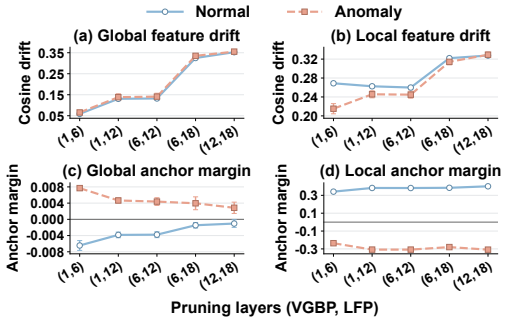


Figure 5: **Pruning-induced representation shift.** Global/local cosine drift and signed Stage-1 anchor margins across VGBP/LFP layer pairs; error bars denote 95% confidence intervals.

4.2 Implementation Details

We use pre-trained CLIP ViT-L/14@336px as the frozen vision and text backbone. By default, each point cloud is rendered into nine 336×336 views. The parameter-free VGBP and learnable LFP are inserted at Layers 6 and 12, respectively, with LFP retaining 50% of the tokens preserved by VGBP. Experiments use PyTorch 2.0.0 on a single NVIDIA RTX 3090 with a batch size of 2. We use the Adam optimizer ($\beta = (0.5, 0.999)$) for both stages. In Stage 1, the decoupled prompt learners are trained with a learning rate of 1×10^{-3}. In Stage 2, the residual prototypes and foreground selector are jointly trained with a learning rate of 1×10^{-4}, with the vision backbone fixed.

4.3 Main Results

We compare SwiftAD with representative zero-shot methods, including PointCLIP V2 (Zhu et al., 2023), AnomalyCLIP (Zhou et al., 2023), MVP-PCLIP (Cheng et al., 2025), PointAD (Zhou et al., 2024a), PointAD+ (Zhou et al., 2025), and GS-CLIP (Deng et al., 2026). We additionally include CPMF (Cao et al., 2024a), an unsupervised method with access to target-domain normal samples. SwiftAD offers a favorable accuracy–efficiency trade-off across all three benchmarks. It maintains strong zero-shot performance with about half the computation of the unpruned baseline (911.60 vs. 1,830.65 GFLOPs). Its throughput is 73.9% higher than that of the baseline. On MVTec3D-AD, it obtains 83.4% O-AUROC, 98.4% P-AUROC, and 93.7% AUPRO, improving over the strongest zero-shot competitor by 0.7, 1.2, and 3.1 percentage points, respectively. On Real3D-AD, it exceeds the best competing results by up to 7.1 percentage points on P-AUROC. Although MVP-PCLIP has lower computational cost, its accuracy is considerably lower. As visualized in Fig. 3, retained tokens and anomaly responses are complementary across views, explaining why SwiftAD can substantially reduce the token sequence while preserving fine-grained point-level localization.

Table 4: Impact of image resolution and view number on MVTec3D-AD. **Bold**: best, underline: second-best.

Resolution	Views	O-AUROC	AP	P-AUROC	AUPRO	FPS	GFLOPs
336×336	2	61.2	85.8	90.7	63.5	12.91	273.48
	4	81.0	94.1	98.1	93.2	8.32	455.80
	6	81.5	94.3	98.1	93.3	6.19	638.13
	8	82.5	94.6	98.1	93.5	5.02	820.45
	9	**83.4**	**94.8**	**98.4**	**93.7**	4.47	911.60
	10	81.3	94.4	98.3	93.5	4.12	1002.77
	12	82.2	94.5	98.2	93.2	3.47	1185.09
224×224	6	55.6	82.4	88.8	61.5	12.88	284.39
	9	57.7	84.0	88.3	59.3	9.71	406.28
	12	55.9	82.7	87.3	57.6	7.66	528.16

Table 5: Robustness to Gaussian noise and calibration shifts on MVTec3D-AD. **Bold**: best, underline: second-best.

Perturbation	Gaussian Noise	Calibration Shift	O-AUROC	AP	P-AUROC	AUPRO
Ideal Condition	std=0	$\pm$0 px	**83.4**	**94.8**	**98.4**	**93.7**
Gaussian Noise	std=0.05	$\pm$0 px	80.4	93.7	98.0	92.6
	std=0.10	$\pm$0 px	78.6	92.5	97.9	92.6
	std=0.20	$\pm$0 px	74.9	91.4	97.8	91.7
Calibration Shift	std=0	$\pm$5 px	83.2	94.7	98.1	93.4
	std=0	$\pm$10 px	82.7	94.6	98.0	93.1
Mixed	std=0.05	$\pm$5 px	79.9	93.6	98.0	92.5
	std=0.10	$\pm$10 px	77.1	92.4	97.9	92.6
	std=0.20	$\pm$10 px	73.7	91.3	97.6	90.9
	std=0.50	$\pm$20 px	65.8	88.0	96.5	87.8

4.4 Ablation Studies

Effectiveness of the Core Components. Table 2 evaluates VGBP, LFP, and PA. The unpruned baseline achieves 75.1% O-AUROC at 1,830.65 GFLOPs. Notably, applying VGBP without PA drops O-AUROC to 58.7%, confirming that token pruning induces a distributional shift the original CLIP prototypes cannot accommodate. The complete model achieves the best P-AUROC (98.4%) and AUPRO (93.7%) at 911.60 GFLOPs, halving computation relative to the baseline. While LFP slightly reduces O-AUROC compared to VGBP+PA (83.4% vs. 83.9%), it improves localization (P-AUROC 98.1%→98.4%, AUPRO 93.1%→93.7%) while further reducing computation by 21.8%. As visualized in Fig. 4, the fused anomaly response remains concentrated on the defect as the token set becomes progressively more compact, corroborating this efficiency–localization trade-off.

Pruning Position, Ratio, and Representation Shift. Table 3 studies VGBP/LFP insertion layers and keep ratio. Applying LFP too early severely harms accuracy due to insufficiently discriminative shallow representations, while delaying both stages increases feature drift and reduces throughput. VGBP at Layer 6 and LFP at Layer 12 with a keep ratio of 0.5 provides the best trade-off (83.4% O-AUROC, 98.4% P-AUROC, 93.7% AUPRO, 4.47 FPS).

Figure 5 further quantifies the representation shift induced by different pruning-layer configurations. Global drift increases sharply when pruning is delayed to the later encoder layers, whereas local patch features exhibit a substantial shift even under earlier pruning and rise further for late-layer configurations. Using the signed anchor margin $\cos(\mathbf{f}, \mathbf{t}_a) - \cos(\mathbf{f}, \mathbf{t}_n)$, the Stage-1 global anchors preserve only a small normal–anomaly separation, while the local branch exhibits a pronounced mismatch across all tested layer pairs. These results support the use of separate global and local prototypes and motivate residual prototype calibration after hierarchical token pruning.

Impact of View Number, Image Resolution, and Input Perturbations. Table 4 shows that nine 336×336 views achieve the best overall result; additional views introduce redundancy, while lower resolution substantially degrades accuracy. Table 5 demonstrates robustness to moderate noise and calibration shifts, with notable degradation occurring mainly under severe mixed perturbations.

5 Conclusion

In this paper, we presented **SwiftAD**, an efficient zero-shot 3D anomaly detection framework combining multi-view rendering with vision-language models. SwiftAD removes redundant tokens through VA-DSTP: VGBP deterministically removes background tokens without valid 3D support using rendering-derived visibility correspondences, while LFP employs a lightweight selector to retain anomaly-relevant foreground tokens. Textual prototype adaptation via decoupling and residual calibration is applied to accommodate the resulting distributional shift. Experiments on three benchmarks show a favorable accuracy–efficiency trade-off. SwiftAD maintains strong detection and localization performance while reducing memory use and computation.

Despite these advantages, aggressive token pruning increases sensitivity to point-cloud noise and calibration errors, potentially reducing robustness under imperfect geometric priors. Future work will therefore focus on noise-resilient pruning and topology-robust feature learning, while extending the proposed mechanism to broader 3D multimodal understanding and real-time robotic inspection.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and presentation. All AI-assisted text was carefully reviewed, revised, and verified by the authors. The authors take full responsibility for the final content of this work, including all technical claims, experimental results, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work uses publicly available industrial anomaly-detection benchmarks and does not involve human participants or personally identifiable information. We do not identify any direct ethical concerns arising from the methodology or experiments. Nevertheless, practical deployment in safety-critical industrial settings should include appropriate human oversight and validation under the target operating conditions.

REPRODUCIBILITY STATEMENT

We provide the model architecture, training objectives, implementation settings, datasets, and evaluation metrics in the main paper. Additional implementation details and ablation results are included in the appendix and supplementary material. Source code and configuration files will be released to facilitate reproduction of the reported results.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401. IEEE, 2025.

Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9592–9600, 2019.

Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4183–4192, 2020.

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.

Luca Bonfiglioli, Marco Toschi, Davide Silvestri, Nicola Fioraio, and Daniele De Gregorio. The eyecandies dataset for unsupervised multimodal anomaly detection and localization. In *Proceedings of the Asian Conference on Computer Vision*, pp. 3586–3602, 2022.

Yunkang Cao, Xiaohao Xu, and Weiming Shen. Complementary pseudo multimodal feature for point cloud anomaly detection. *Pattern Recognition*, 156:110761, 2024a.

Yunkang Cao, Jiangning Zhang, Luca Frittoli, Yuqi Cheng, Weiming Shen, and Giacomo Boracchi. Adaclip: Adapting clip with hybrid learnable prompts for zero-shot anomaly detection. In *European Conference on Computer Vision*, pp. 55–72. Springer, 2024b.

Junjie Chen, Xuyang Liu, Zichen Wen, Yiyu Wang, Siteng Huang, and Honggang Chen. Variation-aware vision token dropping for faster large vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3489–3499, 2026.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

Xuhai Chen, Yue Han, and Jiangning Zhang. April-gan: A zero-/few-shot anomaly classification and segmentation method for cvpr 2023 vand workshop challenge tracks 1&2: 1st place on zero-shot ad and 4th place on few-shot ad. *arXiv preprint arXiv:2305.17382*, 2023.

Yuqi Cheng, Yunkang Cao, Guoyang Xie, Zhichao Lu, and Weiming Shen. Toward zero-shot point cloud anomaly detection: A multiview projection framework. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–14, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23135–23144, 2025.

Zehao Deng, An Liu, and Yan Wang. Gs-clip: Zero-shot 3d anomaly detection by geometry-aware prompt and synergistic view representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35587–35596, 2026.

Zhengyao Fang, Pengyuan Lyu, Chengquan Zhang, Guangming Lu, Jun Yu, and Wenjie Pei. Prune redundancy, preserve essence: Vision token compression in vlms via synergistic importance-diversity. *arXiv preprint arXiv:2603.09480*, 2026.

Bin-Bin Gao. Learning to detect multi-class anomalies with just one normal image prompt. In *European Conference on Computer Vision*, pp. 454–470. Springer, 2024.

Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. Anomalygpt: Detecting industrial anomalies using large vision-language models. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 1932–1940, 2024.

Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, and Shaohui Lin. Dynamic-llava: Efficient multimodal large language models via dynamic vision-language context sparsification. In *International Conference on Learning Representations*, volume 2025, pp. 69927–69955, 2025a.

Ziming Huang, Xurui Li, Haotian Liu, Feng Xue, Yuzhe Wang, and Yu Zhou. Anomalyncd: Towards novel anomaly class discovery in industrial scenarios. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4755–4765, 2025b.

Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19606–19616, 2023.

Wenqiao Li, Xiaohao Xu, Yao Gu, Bozhong Zheng, Shenghua Gao, and Yingna Wu. Towards scalable 3d anomaly detection and localization: A benchmark via 3d anomaly synthesis and a self-supervised learning network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22207–22216, 2024.

Hanzhe Liang, Guoyang Xie, Chengbin Hou, Bingshu Wang, Can Gao, and Jinbao Wang. Look inside for more: Internal spatial modality perception for 3d anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 5146–5154, 2025.

Jiaqi Liu, Guoyang Xie, Ruitao Chen, Xinpeng Li, Jinbao Wang, Yong Liu, Chengjie Wang, and Feng Zheng. Real3d-ad: A dataset of point cloud anomaly detection. *Advances in Neural Information Processing Systems*, 36:30402–30415, 2023.

Zhen Qu, Xian Tao, Mukesh Prasad, Fei Shen, Zhengtao Zhang, Xinyi Gong, and Guiguang Ding. Vcp-clip: A visual context prompting model for zero-shot anomaly segmentation. In *European Conference on Computer Vision*, pp. 301–317. Springer, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, pp. 8748–8763, 2021.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22857–22867. IEEE, 2025.

Xian Tao, Xinyi Gong, Xin Zhang, Shaohua Yan, and Chandranath Adak. Deep learning for unsupervised anomaly localization in industrial images: A survey. *IEEE Transactions on Instrumentation and Measurement*, 71:1–21, 2022a.

Xian Tao, Dapeng Zhang, Wenzhi Ma, Zhanxin Hou, ZhenFeng Lu, and Chandranath Adak. Unsupervised anomaly detection for surface defects with dual-siamese network. *IEEE Transactions on Industrial Informatics*, 18(11):7707–7717, 2022b.

Chengjie Wang, Wenbing Zhu, Bin-Bin Gao, Zhenye Gan, Jiangning Zhang, Zhihao Gu, Shuguang Qian, Mingang Chen, and Lizhuang Ma. Real-iad: A real-world multi-view dataset for benchmarking versatile industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22883–22892, 2024.

Yue Wang, Jinlong Peng, Jiangning Zhang, Ran Yi, Yabiao Wang, and Chengjie Wang. Multimodal industrial anomaly detection via hybrid fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8032–8041, 2023.

Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8552–8562, 2022.

Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. *arXiv preprint arXiv:2310.18961*, 2023.

Qihang Zhou, Jiangtao Yan, Shibo He, Wenchao Meng, and Jiming Chen. Pointad: Comprehending 3d anomalies from points and pixels for zero-shot 3d anomaly detection. *Advances in Neural Information Processing Systems*, 37:84866–84896, 2024a.

Qihang Zhou, Shibo He, Jiangtao Yan, Wenchao Meng, and Jiming Chen. Pointad+: Learning hierarchical representations for zero-shot 3d anomaly detection. *arXiv preprint arXiv:2509.03277*, 2025.

Zheyuan Zhou, Le Wang, Naiyu Fang, Zili Wang, Lemiao Qiu, and Shuyou Zhang. R3d-ad: Reconstruction via diffusion for 3d anomaly detection. In *European conference on computer vision*, pp. 91–107. Springer, 2024b.

Jiaying Zhu, Yurui Zhu, Xin Lu, Wenrui Yan, Dong Li, Kunlin Liu, Xueyang Fu, and Zheng-Jun Zha. Visionselector: End-to-end learnable visual token compression for efficient multimodal llms. *arXiv preprint arXiv:2510.16598*, 2025.

Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2639–2650, 2023.

A Cross-Dataset Zero-Shot Protocol

We follow the cross-dataset zero-shot protocol commonly used in zero-shot 3D anomaly detection. Here, *zero-shot* refers to direct transfer to unseen target-domain categories without using any target-domain sample for training or adaptation, rather than training-free inference. The model is trained with labeled source-domain data, including object-level anomaly labels and dense anomaly annotations, and is then directly evaluated on the target dataset.

Specifically, for evaluation on MVTec3D-AD, SwiftAD is trained on Eyecandies. For evaluation on Eyecandies and Real3D-AD, it is trained on MVTec3D-AD. No sample from the target dataset is used to optimize the prompt learners, foreground selector, or residual prototypes. Stage 1 learns the decoupled global and local prompts from the source dataset while keeping the foreground selector frozen. Stage 2 fixes the prompt-derived prototypes as semantic anchors and optimizes only the branch-specific anomaly residuals and foreground selector, again using source-domain data. The resulting model is applied to the target dataset without category-specific fine-tuning or prototype adaptation.

B Detailed Experimental Configuration

All experiments are implemented in PyTorch 2.0.0 on a single NVIDIA RTX 3090 GPU with a batch size of 2 and a random seed of 3407. We use CLIP ViT-L/14@336px as the frozen vision–language backbone. Each point cloud is rendered into nine 336×336 views, producing a 24×24 patch grid with a patch size of 14. Unless otherwise specified, VGBP and LFP are inserted at Layers 6 and 12, respectively. The visibility mask is dilated by one patch to preserve boundary context, and LFP retains 50% of the tokens preserved by VGBP. The foreground selector uses four attention heads, a hidden dimension of 1024, a dropout rate of 0.1, and a straight-through temperature $T = 0.7$.

The prompt learner uses a context length of 12. Both training stages use Adam with $\beta = (0.5, 0.999)$, zero weight decay, and no learning-rate scheduler. Stage 1 trains the decoupled prompt learners with a learning rate of 1×10^{-3} while freezing the CLIP backbone and foreground selector. Stage 2 trains the branch-specific anomaly residuals and foreground selector with a learning rate of 1×10^{-4} while keeping the CLIP backbone and Stage-1 prompt-derived prototypes fixed. The anomaly residuals are initialized to zero, and their learnable scales are initialized to one. The classification temperature is set to 0.07.

For the Stage-2 objective, we set the keep-ratio loss weight λ_{sp} to 0.1 and the score-spread coefficient to 0.05. The focus-loss weight λ_{focus} is 0.5, with positive-token and anomaly-coverage weights of 5.0 and 1.0, respectively. The anchor regularization weight λ_{anchor} is set to 0.5.

C Additional Ablation on Textual Prototype Adaptation

Table 6 provides a detailed comparison of prompt-learning and prototype-adaptation strategies. Fixed and shared learnable prompts achieve only 60.5% and 66.8% O-AUROC, respectively, whereas decoupling the global and local prompts raises O-AUROC to 81.9% and P-AUROC to 98.3%. Direct prototype optimization improves object-level recognition but slightly reduces localization performance. In contrast, decoupled residual adaptation achieves the strongest overall result, with 83.4% O-AUROC, 98.4% P-AUROC, and 93.7% AUPRO. These results support branch-specific prompts for the asymmetric global and local shifts in Fig. 5. Residual adaptation calibrates the anomaly prototypes while preserving the Stage-1 semantic anchors.

Table 6: Comparison of prompt-learning and prototype-adaptation strategies on MVTec3D-AD. **Bold**: best, underline: second-best.

Method	Stage-1	Stage-2	O-AUROC	AP	P-AUROC	AUPRO
Fixed Prompt	Fixed	None	60.5	85.0	86.4	59.7
Shared Learnable Prompt	Shared	None	66.8	87.2	86.7	67.6
Decoupled Learnable Prompts	Decoupled	None	81.9	94.3	<u>98.3</u>	<u>93.4</u>
Shared + Direct Prototype	Shared	Direct	57.5	83.5	85.1	51.6
Decoupled + Direct Prototype	Decoupled	Direct	<u>83.3</u>	**94.8**	98.1	93.2
Shared + Residual Adaptation	Shared	Residual	56.5	83.0	85.3	52.0
Decoupled + Residual Adaptation	Decoupled	Residual	**83.4**	**94.8**	**98.4**	**93.7**

D Complete Qualitative Results Across Datasets

Figures 6–8 present complete qualitative results on MVTec3D-AD, Eyecandies, and Real3D-AD. For each object, we show an RGB or rendered reference, pruning masks and anomaly responses from nine rendered views, the fused point-level anomaly score map, and the ground truth. Across all three datasets, different views retain complementary object regions, and their anomaly responses are aggregated to produce the final point-level prediction.

Reading the MVTec3D-AD results. Each example shows an RGB reference, followed by pruning masks and anomaly responses from nine rendered views. The final two columns compare the fused point-level anomaly score map with the ground truth, allowing the localized response to be inspected against the annotated defect.

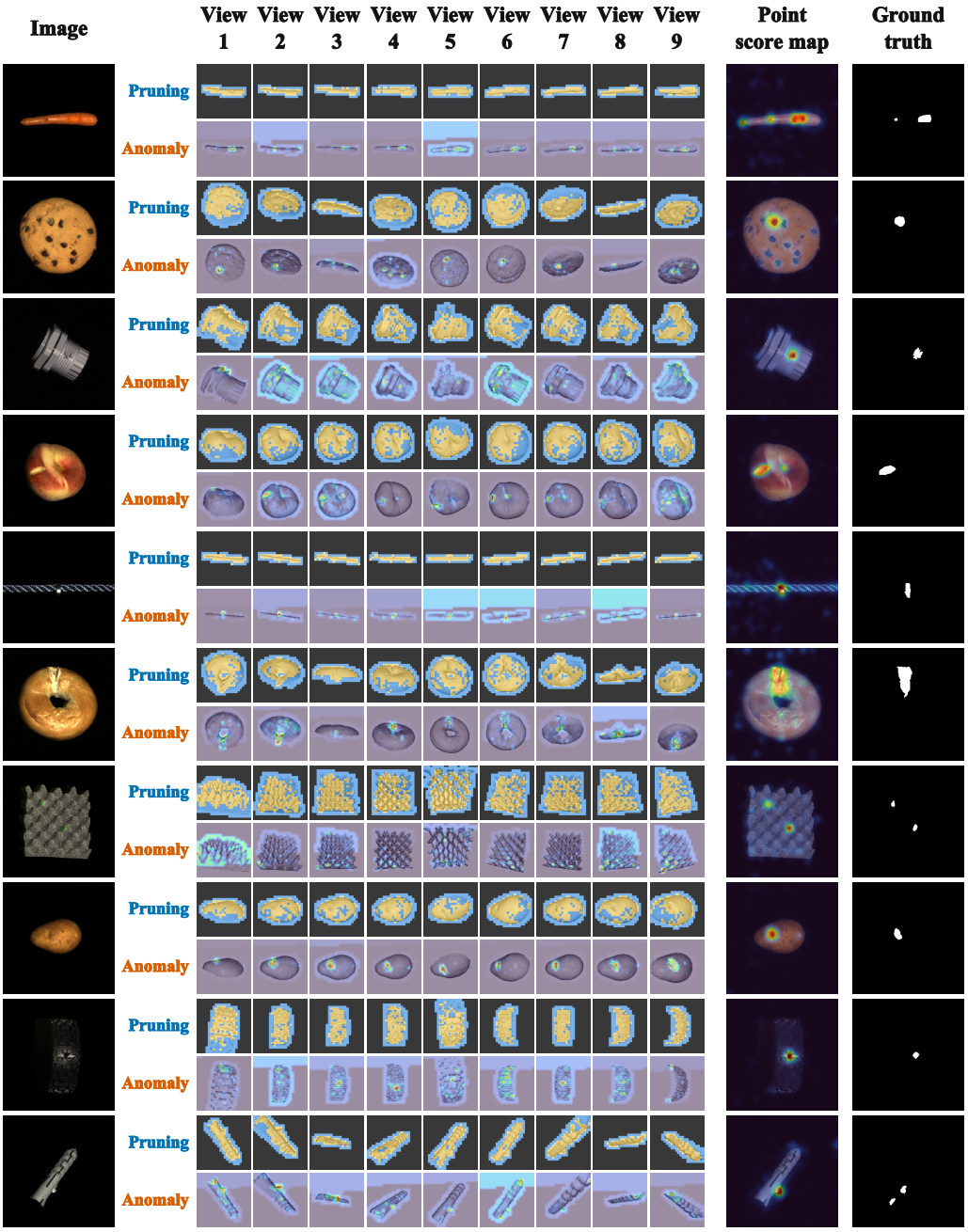


Figure 6: **Complete qualitative results on MVTec3D-AD.** From left to right: RGB reference, pruning masks and anomaly responses from nine rendered views, fused point-level anomaly score map, and ground truth. Within each example, the upper view-wise row shows the pruning mask and the lower row shows the anomaly response.

Reading the Eyecandies results. Each example shows an RGB reference, followed by pruning masks and anomaly responses from nine rendered views. The final two columns compare the fused point-level anomaly score map with the ground truth, allowing the localized response to be inspected against the annotated defect.

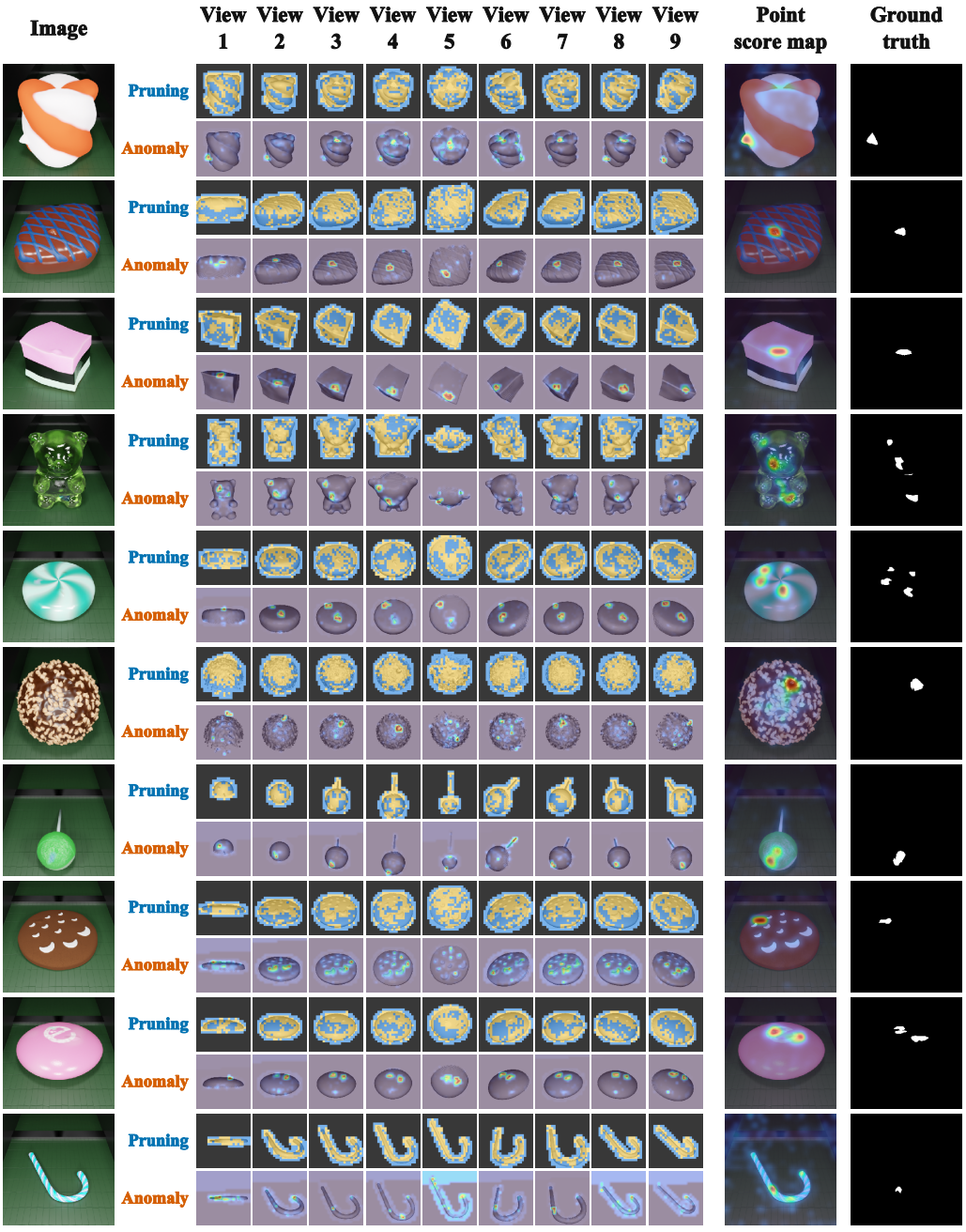


Figure 7: **Complete qualitative results on Eyecandies.** From left to right: RGB reference, pruning masks and anomaly responses from nine rendered views, fused point-level anomaly score map, and ground truth. Within each example, the upper view-wise row shows the pruning mask and the lower row shows the anomaly response.

Reading the Real3D-AD results. Each example shows a rendered reference, followed by pruning masks and anomaly responses from nine rendered views. The final two columns compare the fused point-level anomaly score map with the ground truth, allowing the localized response to be inspected against the annotated defect.

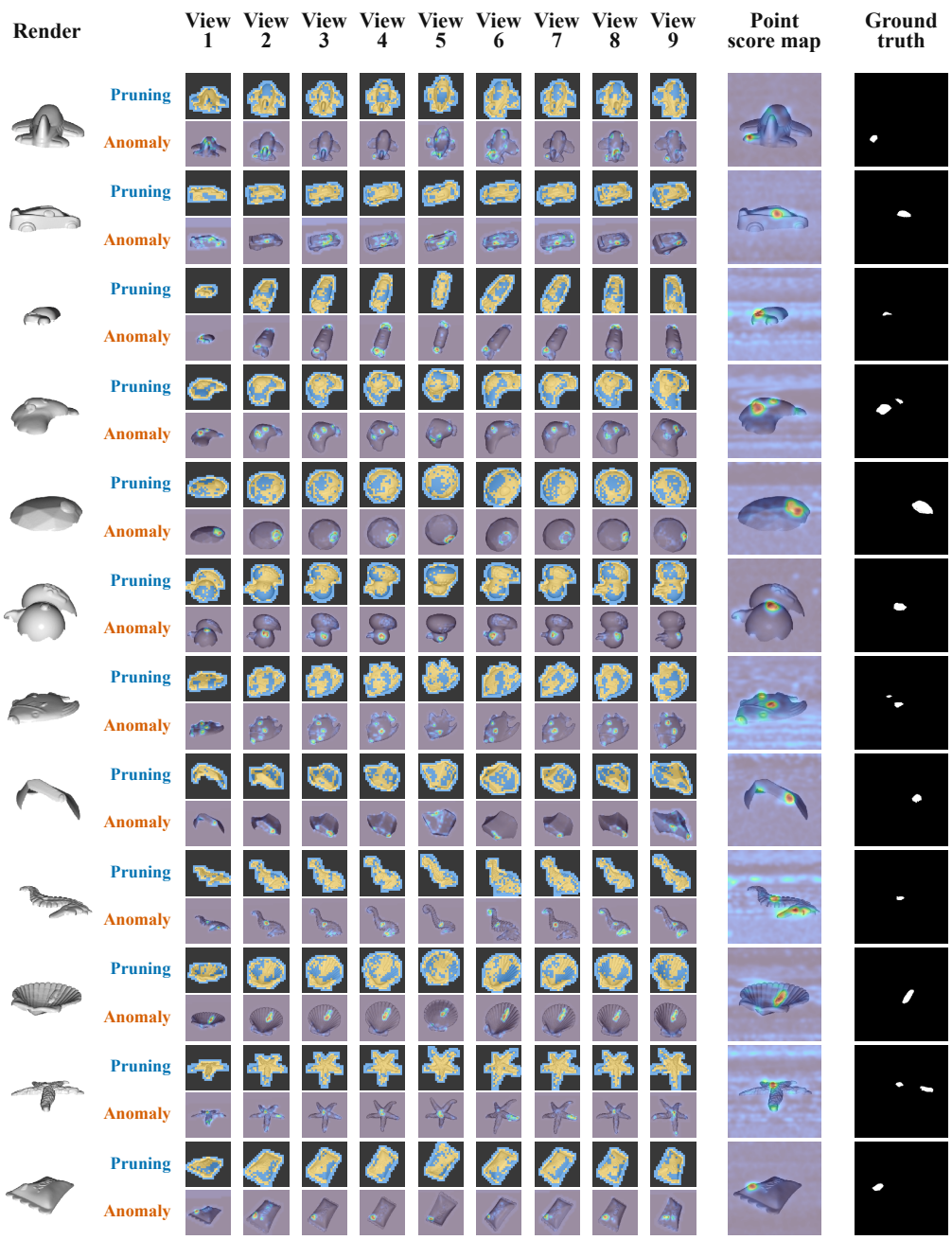


Figure 8: **Complete qualitative results on Real3D-AD.** From left to right: rendered reference, pruning masks and anomaly responses from nine rendered views, fused point-level anomaly score map, and ground truth. Within each example, the upper view-wise row shows the pruning mask and the lower row shows the anomaly response.