GPS: GlobalCLIP-PatchCore-SAM Based Zero-Shot Anomaly Detection and Localization in Smart Manufacturing

Junha Park and Jongpil Jeong(✉)

Department of Smart Factory Convergence, Sungkyunkwan University, Suwon-si, Gyeonggi-do 16419, Republic of Korea
{junha010102,jpjeong}@skku.edu

Abstract. Detecting defects on a factory floor remains challenging when production runs are short and defective parts seldom appear. Supervised and unsupervised inspection models both struggle under such data-scarce conditions, motivating zero-shot approaches that operate without target-domain defect samples. We introduce GlobalCLIP-PatchCore-SAM (GPS), a three-stage pipeline that first decides whether an image is anomalous at the global level and then drills down to the exact defect pixels, using no anomaly samples or defect labels from the target domain at any stage. Stage 1 (GlobalCLIP) aggregates 1280-dimensional features from two Contrastive Language-Image Pre-Training (CLIP) encoders (ViT-B/32 and ViT-L/14@336) and trains a self-cyclic ensemble of six autoencoders, guided by Adaptive Rejection Sampling for outlier removal and Adaptive Augmented sample Weighting for pseudo-label refinement, to gate images at the global level. Stage 2 (PatchCore) extracts WideResNet-50 multi-scale patch features and produces a 32×32 anomaly map via (K-Nearest Neighbor) k-NN (k=3) over a 25% random coreset. Stage 3 Segment Anything Model (SAM) refines anomaly boundaries through segment-wise scoring so that they snap to real object edges. With a single fixed hyperparameter set across all 15 MVTec AD categories, GPS reaches 92.49% image-level and 84.62% pixel-level Area Under Receiver Operating Characteristic (AUROC). Ablations expose clear trade-offs: dropping the GlobalCLIP gate collapses classification to chance level, k-NN runs 1.6× faster than Mahalanobis scoring with comparable accuracy, and SAM sharpens boundaries visually yet can hurt pixel AUROC when defects are tiny.

Keywords: Anomaly detection · CLIP · PatchCore · Segment Anything Model · Vision-language model · Vision Transformer · Zero-Shot · Anomaly localization

1 Introduction

Anomaly detection is a key task to ensure product quality and production stability in the manufacturing process [1]. Thanks to the miniaturization and cost

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16768, pp. 447–463, 2027.
https://doi.org/10.1007/978-3-032-30488-9_28

reduction of optical sensors, the proportion of image-based analysis systems is increasing. However, since the modern manufacturing process is stabilized, defective products rarely occur, which causes a serious data imbalance problem [2]. In other words, as product types become diversified in the modern variety of small-scale production, data-based inspection systems become more complex.

How much labeled data a method needs largely determines where it can be deployed. Fully supervised detectors demand pixel-level defect annotations, which are expensive and hard to scale [3]. Unsupervised approaches drop the need for defect labels but still assume access to a reasonably large pool of defect-free images [4,5]. Few-shot variants cut that pool down to a handful of reference pictures [2,6], yet they still expect at least some target-domain samples. Zero-shot methods [7,8] go one step further and operate with no target data at allan attractive property for factories that roll out new product lines on short notice. The term "Zero-shot" as used in this paper follows the general practice of the Zero-Shot-Anomaly-Detection (ZSAD) field, meaning that no anomaly samples or pixel-level defect labels of the target domain are used in the training process. Like existing ZSAD studies such as WinCLIP [7], AnomalyCLIP [10], and GlobalCLIP [8], our framework operates without any prior information about defect labels or defective images, while normal training images are used for training the ensemble model and constructing the memory bank. In other words, the zero-shot setting in this study is interpreted as "annotation-free" or "defect-label-free," which carries the practical value of eliminating the cost of collecting defective data when introducing new product lines. Recently, the GlobalCLIP method [8] demonstrated that zero-shot anomaly detection can achieve competitive performance using only global image features extracted by Contrastive Language-Image Pre-Training (CLIP) [9], without requiring text prompts or local feature tensors. However, GlobalCLIP is limited to image-level detection and cannot localize the spatial extent of anomalies a critical requirement for root cause analysis in manufacturing.

In this paper, we propose a multi-stage statistical framework that extends GlobalCLIP's zero-shot image-level detection capability to pixel-level anomaly localization. Existing zero-shot anomaly detection methods have the following limitations. Text prompt-based techniques such as WinCLIP [7], AnomalyCLIP [10], and AdaCLIP [11] require defect types to be defined in natural language in advance or learnable prompts to be trained separately, which is unsuitable for manufacturing environments where new defect types frequently appear. On the other hand, GlobalCLIP [8] eliminated the dependence on text prompts but is limited to image-level detection, making it impossible to localize defects, which constitutes a decisive limitation for root cause analysis on manufacturing sites.

The contributions of this study are summarized as follows.

– We present a unified framework that simultaneously addresses the problems of text-prompt dependence and the absence of localization, by integrating the text-prompt-free gating mechanism of GlobalCLIP with the memory-bank-

based localization of PatchCore and the boundary refinement of Segment Anything Model (SAM).
- We demonstrate generalizability without category-specific tuning by evaluating the framework with a single hyperparameter setting across all 15 MVTec AD categories.
- Through five ablation experiments, we quantitatively analyze the trade-offs among detection accuracy, localization quality, and computational efficiency, providing practical guidelines for industrial deployment.
- By analyzing performance variations and failure cases across categories, we present practical considerations for real-world deployment.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the proposed framework in detail. Section 4 presents experimental results and ablation studies. Section 5 discusses findings and limitations, and Sect. 6 concludes the paper.

2 Related Work

2.1 Zero-Shot Anomaly Detection

Zero-shot anomaly detection methods must operate without any task-specific training data, relying entirely on pre-trained representations and general priors. WinCLIP [7] utilizes CLIP's image-text alignment to compare local image features with manual text prompts describing normal and abnormal states. AnomalyCLIP [10] proposes object-agnostic learnable prompts that capture generic normal and abnormal patterns regardless of object semantics, demonstrating strong generalization performance across diverse domains. AdaCLIP [11] introduces hybrid learnable prompts that combine static prompts shared across all images with dynamic prompts generated per test image, improving ZSAD performance through auxiliary anomaly detection data.

Unlike these text-dependent methods, GlobalCLIP [8] performs detection using only global image features without text prompts. The technique aggregates features with a dual CLIP backbone (ViT-B/32 and ViT-L/14@336) and estimates anomaly scores through an adaptive self-cyclic reconstruction model ensemble. This study utilizes GlobalCLIP as the first stage and adds pixel-level localization capability.

2.2 Feature-Bank Anomaly Localization

PatchCore [2] builds a memory bank by collecting patch unit features from the normal learning image, and calculates an anomaly map using the k-nearest neighbor distance. High performance can be achieved with a small number of normal samples through multi-scale feature aggregation and greedy coreset sampling of the pre-trained network. In Stage 2 of this study, we adopt the memory-bank-based approach of PatchCore, while applying WideResNet-50 multi-scale features, random coreset sampling, and GPU-accelerated k-NN search.

Table 1. Comparison of feature characteristics with existing zero-shot anomaly detection methods.

Method	Text Prompt-Free	Pixel-Level Localization	Boundary-Aware Refinement
WinCLIP [7]	✗	✓	✗
AnomalyCLIP [10]	✗	✓	✗
AdaCLIP [11]	✗	✓	✗
GlobalCLIP [8]	✓	✗	✗
GPS (Ours)	✓	✓	✓

2.3 Foundation Models for Segmentation

SAM [19] is an image segmentation foundation model trained with more than 1 billion masks. Since SAM can generate class-independent segmentation masks through various prompts such as points, boxes, and automatic modes, in this study, SAM's automatic mask generation function is used as a boundary correction module in Stage 3 to assign anomaly scores to semantically consistent regions rather than individual pixels.

2.4 Comparison with Prior Work

As shown in Table 1, this framework is the only zero-shot anomaly detection method that satisfies all three characteristics: text-prompt independence, pixel-level localization, and boundary-aware refinement. WinCLIP [7], AnomalyCLIP [10], and AdaCLIP [11] support pixel-level localization but rely on either manually crafted or learnable text prompts, which incurs the cost of redesigning or retraining prompts whenever a new defect type is introduced. GlobalCLIP [8] eliminated the dependence on text prompts but cannot be used for visualizing defect regions or analyzing root causes because it lacks localization capability. In this study, the complementary strengths of the preceding techniques are integrated into a single pipeline: the text-prompt-independent gating mechanism of GlobalCLIP is adopted as Stage 1, while patch-level localization in Stage 2 and SAM-based boundary refinement in Stage 3 are applied only to images determined as anomalous. In addition, SAM-based boundary refinement suppresses pixel-level noise and produces an anomaly map aligned with object contours by computing semantically consistent region-level scores.

3 GPS Framework

The proposed GPS framework consists of three sequential step. Figure 1 is an overview of the entire pipeline showing how information flows from Stage 1 to Stage 3. In Stage 1 (GlobalCLIP) on the left, two CLIP encoders, ViT-B/32 and ViT-L/14@336, extract 1280-dimensional features from the input image, and a

self-cyclic reconstruction ensemble calculates image-level anomaly scores. In the Gate Decision step, the image is classified as normal or anomalous based on the 95th percentile threshold τ, and only the image that is judged as anomalous is transmitted to Stage 2. In Stage 2 (PatchCore k-NN) at the center, a rough 32×32 anomaly map is generated through the comparison of the multi-scale patch features of WideResNet-50 with the memory bank. In Stage 3 (SAM) on the right, SAM ViT-H refines the anomaly scores by semantically consistent area units through automatic mask generation and outputs final anomaly maps and bounding boxes aligned with object boundaries. This step-by-step separation structure greatly improves the computational efficiency by skipping Stage 2 and Stage 3 for images classified as normal by the gate, while allowing each step to be independently verified and replaced.

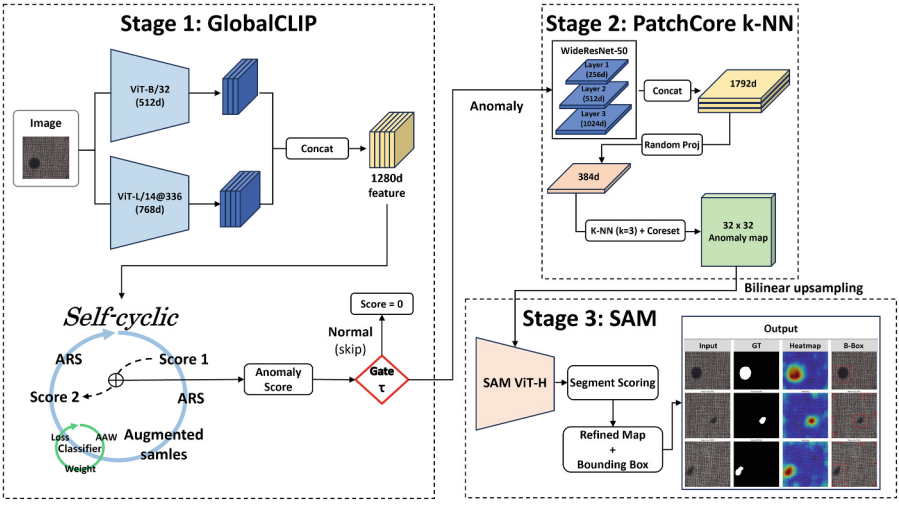


Fig. 1. GPS framework.

3.1 Stage 1: GlobalCLIP Zero-Shot Detection

In the first stage, when an image comes in, the framework determines whether the image contains an anomaly.

Multi-global Feature Aggregation. For each image $I_i \in \mathbb{R}^{W \times H \times 3}$, we extract global features using two CLIP image encoders:

$$v_i = \text{concat}\left[f_{\text{clip}}^{(1)}(I_i),\ f_{\text{clip}}^{(2)}(I_i)\right] \in \mathbb{R}^D \tag{1}$$

where $f_{\text{clip}}^{(1)}$ is ViT-B/32 ($D_1 = 512$) and $f_{\text{clip}}^{(2)}$ is ViT-L/14@336 ($D_2 = 768$), yielding $D = 1280$ dimensional concatenated features. Each feature vector is L2-normalized before concatenation. The dual-backbone aggregation captures both

coarse semantic information (ViT-B/32) and fine-grained visual details (ViT-L/14@336), providing a richer representation than either backbone alone.

Self-Cyclic Ensemble Learning. The core of Stage 1 is the self-cyclic ensemble learning mechanism, which progressively improves detection performance through iterative training cycles. The ensemble consists of $K = 6$ reconstruction models with varying bottleneck dimensions $\{32, 64, 96, 128, 160, 256\}$. Each reconstruction model $f(\cdot; \theta_k)$ is an autoencoder with architecture: Linear($D \rightarrow 512$)–ReLU–Linear($512 \rightarrow b_k$)–ReLU–Linear($b_k \rightarrow 512$)–ReLU–Linear($512 \rightarrow D$), where b_k is the bottleneck dimension.

During the cycle of $C = 3$, the model is trained with the following adaptive mechanism. Figure 2 is a visualization of the self-cyclic ensemble structure of Stage 1, which shows how the CLIP feature is converted into two complementary scores and integrated into the final anomaly score. The CLIP feature entering the left input is divided into a set of six independent reconstruction models with bottleneck dimensions of 32, 64, 96, 128, 160, and 256, respectively. Each model compresses and restores the input, and the average of the reconstruction error occurring in this process is calculated as *Score 1*. At the same time, the Adaptive Rejection Sampling (ARS) module applies a quartile range Interquartile Range (IQR)-based threshold to identify a pseudo-positive sample, and the Adaptive Augmented Sample Weight (AAW) module calculates *Score 2* by assigning a learning weight according to the prediction uncertainty of the classifier. The two scores are combined as a weighted sum and transferred to the Gate Decision stage, and at the same time, they are fed back to the next cycle through the self-cyclic loop. As the cycle is repeated in this way, the pseudo-label is gradually refined, which allows the model to learn the boundaries between normal and anomalous samples gradually without target-domain data.

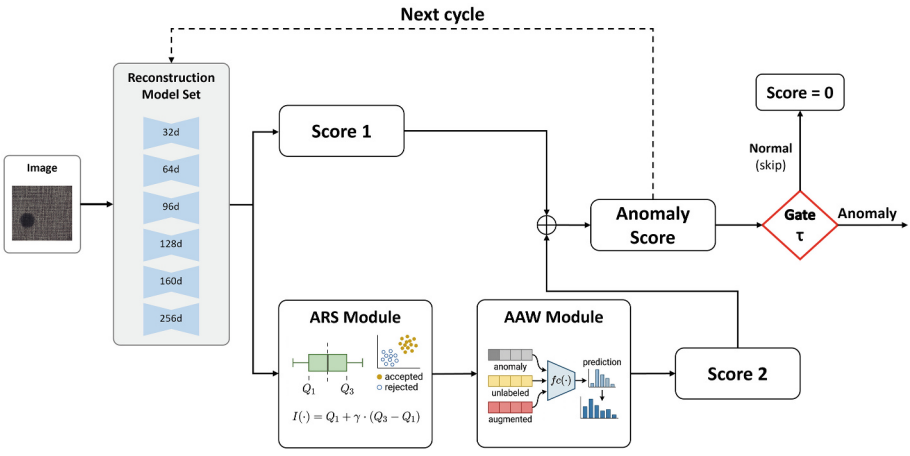


Fig. 2. Self-cyclic ensemble in Stage 1.

ARS. After each cycle c, the reconstruction error scores $s_c = \{s_{c,1}, \ldots, s_{c,N}\}$ are used to identify potential anomalies. The rejection threshold is computed using the interquartile range:

$$T_c = Q_1^c + \alpha \cdot (Q_3^c - Q_1^c) \tag{2}$$

where Q_1^c and Q_3^c are the first and third quartiles of s_c, and $\alpha = 1.5$ is the outlier sensitivity parameter (following the standard box-plot criterion). Samples with $s_{c,i} \geq T_c$ are flagged as potential anomalies and excluded from subsequent training rounds, progressively purifying the training set.

AAW. The AAW module assigns learning weights to samples based on the classifier's prediction uncertainty. For an unlabeled sample v with classifier output $f_p(v)$, the weight for the positive class is:

$$\omega(v) = \frac{(1-\gamma) \cdot f_p(v)}{1 - \gamma \cdot f_p(v)} \tag{3}$$

where $\gamma = 0.7$ is the label frequency parameter. This enables the classifier to implicitly learn the boundary between normal and anomalous samples from the unlabeled set.

Anomaly Score Computation. The final anomaly score for each test image combines the reconstruction error and classifier probability:

$$s_i = \lambda \cdot s_i^p + s_i^r \tag{4}$$

where $s_i^r = \Phi\left(\frac{1}{K}\sum_{k=1}^{K} s_i^k\right)$ is the normalized mean reconstruction error across all K models, $s_i^p = f_p(v_i^{\text{input}})$ is the classifier output, $\lambda = 0.5$, and $\Phi(\cdot)$ denotes min-max normalization.

Gating Decision. To determine which images enter Stage 2, we set the threshold τ as the 95th percentile of the training set's anomaly scores (mapped through the same sigmoid normalization). Images with $s_i \geq \tau$ are classified as anomalous and forwarded to Stage 2 for pixel-level analysis.

3.2 Stage 2: PatchCore *k*-NN Anomaly Localization

For images flagged as anomalous by Stage 1, we perform pixel-level anomaly localization using a PatchCore-inspired approach with several adaptations.

The memory bank in Stage 2 is constructed from normal training images of the target category. This is consistent with the zero-shot setting of our framework: as defined in 3.1, the zero-shot setting in this study refers to the absence of defect labels and anomaly samples, while the use of normal training images is permitted. This setting aligns with the standard practice in the unsupervised and zero-shot anomaly detection literature, including PatchCore [2] and

GlobalCLIP [8]. Normal training images are generally far easier to collect than defective images, and they can be readily secured even when introducing new product lines, which is consistent with practical deployment requirements.

Multi-scale Patch Feature Extraction. We employ a pre-trained Wide-ResNet-50 [20] as the feature extractor. For each image resized to 256×256, we extract intermediate features from three layers and aggregate them into a unified patch representation:

$$F = \text{concat}[\text{up}(F_1),\ \text{up}(F_2),\ \text{up}(F_3)] \in \mathbb{R}^{B\times 1792\times 32\times 32} \tag{5}$$

where $F_1 \in \mathbb{R}^{256\times h_1\times w_1}$, $F_2 \in \mathbb{R}^{512\times h_2\times w_2}$, and $F_3 \in \mathbb{R}^{1024\times h_3\times w_3}$ are features from Layer 1, 2, and 3, respectively, and $up(\cdot)$ denotes bilinear interpolation to the target grid size of 32×32. The concatenated 1792-dimensional features are reduced to 384 dimensions via random dimension selection (fixed random seed for reproducibility), yielding patch descriptors $p_j \in \mathbb{R}^{384}$ for each of the $32 \times 32 = 1024$ spatial positions.

Figure 3 shows the entire flow of the localization pipeline that combines Stage 2 and Stage 3. The anomaly image that has passed through the gate is first input into WideResNet-50 to extract multi-scale features (256, 512, 1024 dimensions, respectively) from Layers 1, 2, and 3, then concatenated into 1792 dimensions and reduced to 384 dimensions through random projection on a 32×32 grid. The patch descriptor obtained in this way is transmitted to the k-NN module, and patch-level anomaly scores are calculated based on the distance to the memory bank (25% coreset) composed of normal training features under the condition of $k = 3$. The rough 32×32 map presented on the right side of Fig. 3 has spatial alignment in patch units but lacks consistency with object boundaries. To compensate for this, SAM ViT-H generates an automatic mask for the anomaly map and allocates the 75th percentile within each segmented area as a representative score for the corresponding area. This area-wise scoring method has the effect of maintaining semantically consistent area boundaries while suppressing pixel-level noise, and finally, a refined pixel-level anomaly map with bounding boxes is output.

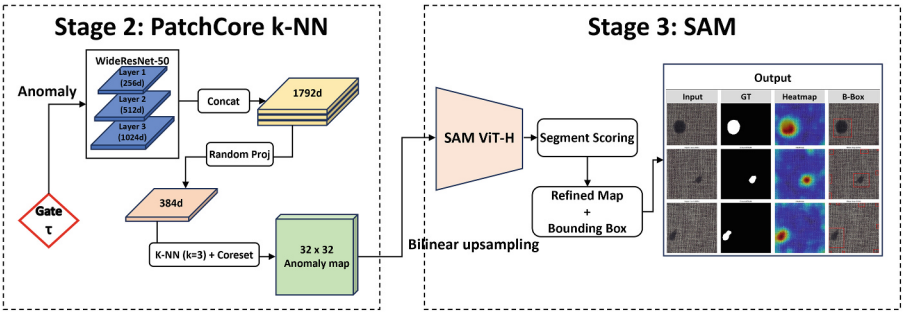


Fig. 3. Anomaly localization pipeline (Stages 2 and 3).

Memory Bank Construction and Coreset Sampling. A memory bank $\mathcal{M}$ is constructed from normal training images by collecting all patch features. Given N_{train} training images, the total number of patches is $N_{\text{train}} \times 1024$. To reduce memory and computation, we apply random coreset sampling at 25% ratio:

$$\mathcal{M}_{\text{core}} = \text{RandomSample}(\mathcal{M},\ |\mathcal{M}| \times 0.25) \tag{6}$$

***k*-NN Anomaly Scoring.** For each patch p_j of a test image, the anomaly score is the mean distance to the $k = 3$ nearest neighbors in the memory bank:

$$a_j = \frac{1}{k} \sum_{m=1}^{k} \|p_j - n_j^{(m)}\|_2 \tag{7}$$

where $n_j^{(1)}, \ldots, n_j^{(k)}$ are the k nearest neighbors of p_j in $\mathcal{M}_{\text{core}}$. The distances are computed via batch-wise GPU-accelerated Euclidean distance. The resulting 32×32 anomaly map is smoothed with a Gaussian filter ($\sigma = 1$) and min-max normalized to $[0, 1]$.

3.3 Stage 3: SAM Boundary Refinement

The coarse 32×32 anomaly map from Stage 2 is refined using the SAM [19] with ViT-H backbone for boundary-aware localization.

First, the anomaly map is upscaled to the original image resolution via bilinear interpolation. SAM's automatic mask generator then produces a set of class-agnostic segments $\{M_1, \ldots, M_S\}$ with parameters: 32 points per side, predicted Intersection over Union (IoU) threshold 0.86, stability score threshold 0.92, and minimum mask region area 100 pixels. For each segment M_s, a representative anomaly score is computed as the 75th percentile of the upscaled anomaly values within that segment:

$$a_s = Q_{75}\left(\{a(x) : x \in M_s\}\right) \tag{8}$$

The corrected anomaly map gives each pixel the highest score among the segments to which the pixel belongs. Pixels that are not included in any segment maintain the original value calculated and filled through the surrounding values. This segment-based scoring generates an anomaly map that respects object boundaries while smoothing noise within a consistent area.

Bounding Box Generation. For visualization and practical deployment, anomaly bounding boxes are generated by thresholding the refined map at the 90th percentile, applying connected component labeling, and computing axis-aligned bounding rectangles for each component exceeding 20 pixels in area.

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate on MVTec AD [1], a widely-used industrial anomaly detection benchmark containing 5,354 high-resolution images across 15 categories (5 texture types and 10 object types). Each category includes a training set of defect-free images and a test set with both normal and various defect types, along with pixel-level ground truth masks.

Implementation Details. All experiments were conducted in the NVIDIA A100 80GB GPU environment, and the entire deep learning pipeline was built on PyTorch [12]. In Stage 1, CLIP features were extracted from ViT-B/32 and ViT-L/14@336 models using the OpenCLIP library [13]. The reconstruction ensemble consists of $K = 6$ models, trained at 50 epochs per model with Adam optimizer (learning rate $= 10^{-3}$), batch size 64, and $C = 3$ cycles. In Stage 2, WideResNet-50 pre-trained with ImageNet was loaded from torchvision to extract multi-scale features from 256×256 resolution to 32×32 grid, which were reduced from 1792 dimensions to 384 dimensions. The memory bank uses a random coreset of 25% and a $k = 3$ nearest neighbor. Stage 3 utilizes the SAM ViT-H model through the official segment-anything library. All hyperparameters were equally fixed across all 15 categories.

Software Stack. Additional open-source software used for data processing and analysis in this study is as follows. Numerical operation and tensor processing were performed through NumPy [14]; Gaussian filtering ($\sigma = 1$), quartile IQR-based threshold calculation, and statistical distribution processing used SciPy [15]. The Area Under Receiver Operating Characteristic (AUROC) evaluation of anomaly scores, random coreset sampling, and verification metric calculation were performed through scikit-learn [16]. Bounding box generation through defect area visualization and connected component labeling utilized OpenCV [17], and heatmap visualization and result figure generation were performed through Matplotlib [18].

Evaluation Metrics. We use AUROC at both image level (detection quality) and pixel level (localization quality). Image-level AUROC (I-AUROC) measures the ability to distinguish normal from anomalous images, while Pixel-level AUROC (P-AUROC) evaluates the quality of spatial anomaly localization against ground truth masks.

4.2 Results

Table 2 presents the full experimental results across all 15 MVTec AD categories. The framework achieves a mean I-AUROC of 92.49% and a mean P-AUROC of 84.62%.

Table 2. MVTec AD (15 categories).

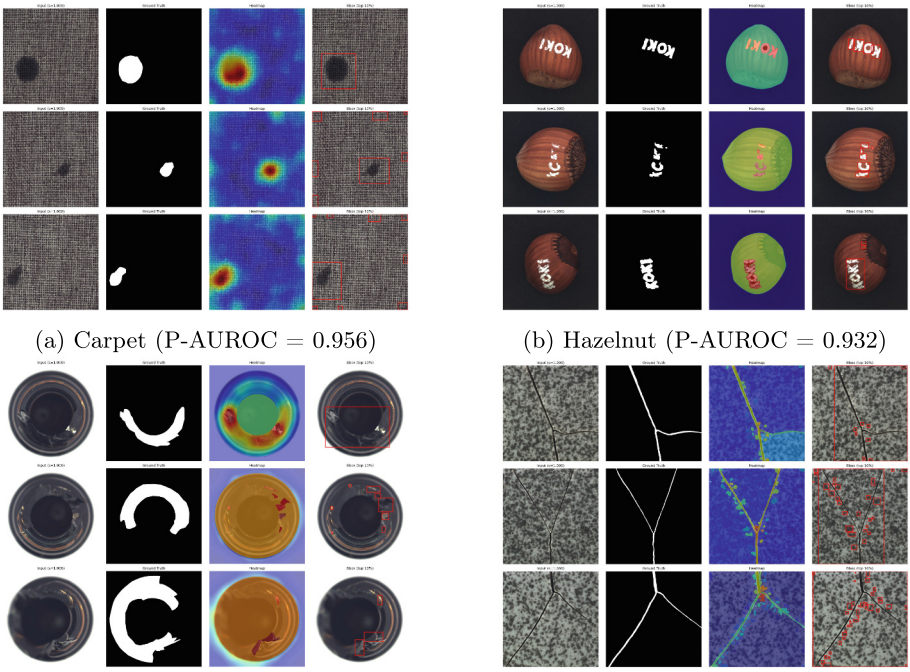
Category	I-AUROC	P-AUROC	τ	Anomaly	Time (s)
bottle	0.9921	0.9190	0.843	62/83	257
cable	0.9052	0.8468	0.863	64/150	349
capsule	0.8492	0.7881	0.881	51/132	305
carpet	*0.9663*	*0.9560*	*0.853*	*98/117*	*975*
grid	*0.9407*	*0.7850*	*0.879*	*58/78*	*284*
hazelnut	0.9689	0.9323	0.873	56/110	279
leather	*0.9993*	*0.8621*	*0.831*	*99/124*	*317*
metal_nut	0.9247	0.8453	0.839	71/115	239
pill	0.8140	0.8673	0.867	79/167	271
screw	0.7760	0.6193	0.852	44/160	228
tile	*0.9996*	*0.9285*	*0.791*	*87/117*	*405*
toothbrush	0.9556	0.9132	0.861	22/42	93
transistor	0.8571	0.8844	0.877	21/100	133
wood	*0.9974*	*0.7747*	*0.898*	*64/79*	*222*
zipper	0.9280	0.7711	0.872	83/151	276
Mean	0.9249	0.8462	–	–	78.4 m

Image-Level Detection. The Stage 1 zero-shot detector achieves excellent performance on texture categories (mean 0.9807) and strong performance on objects (mean 0.8941). The highest scores are observed for tile (0.9996), leather (0.9993), and wood (0.9974), all near-perfect detection. The most challenging categories are screw (0.7760) and pill (0.8140), where significant non-anomalous pose variations confound the global feature-based detection, consistent with findings in [8].

Pixel-Level Localization. P-AUROC varies more across categories, with carpet (0.9560), hazelnut (0.9323), and tile (0.9285) achieving strong localization, while screw (0.6193) and zipper (0.7711) show weaker performance. The latter cases involve fine-grained defects that are challenging for the 32×32 grid resolution and SAM's segment-level averaging.

Visualization. Figures. 4, 5 and 6 present visualization results, each showing four columns: input with anomaly score, ground truth mask, heatmap overlay, and bounding box predictions at the top 10% threshold. Figure 4 presents four texture and regular-shape categories. Figure 4(a) shows carpet, where the heatmap precisely covers defective regions and SAM aligns anomaly boundaries with the texture structure. Figure 4(b) shows hazelnut, where surface defects including printed markings and cracks are tightly enclosed by bounding boxes.

For bottle (Fig. 4(c)), the heatmap clearly highlights defective regions around the cap area. The SAM-refined boundaries align well with the circular geometry of the bottle, demonstrating the benefit of segment-aware scoring for objects with regular shapes. Tile (Fig. 4(d)) achieves near-perfect image-level detection (0.9996) and strong pixel localization; defects such as cracks and glue strips produce strong, well-localized heatmap responses.



(a) Carpet (P-AUROC = 0.956)

(b) Hazelnut (P-AUROC = 0.932)

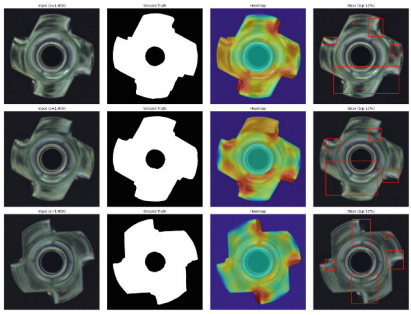
(c) Bottle (P-AUROC = 0.919)

(d) Tile (P-AUROC = 0.929)

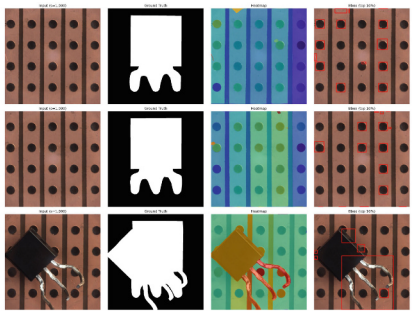
Fig. 4. Well-performing categories: Carpet, Hazelnut, Bottle, and Tile.

For object categories with complex geometry, Fig. 5(a) shows metal nut, where distributed anomalies across the object surface including scratches and bent regions are detected. The heatmap activates on edge regions, though some false positives appear on normal structural features. Transistor (Fig. 5(b)) demonstrates the framework's ability to localize misplaced and damaged components despite the complex multi-part geometry.

We also examine challenging cases where performance is weaker. Screw is the most difficult category, with significant non-anomalous pose variations that activate the entire object region, diluting the localization of actual defects (Fig. 6(a)). Wood shows partial defect coverage, where the heatmap highlights only portions of the ground truth regions. Long linear defects such as scratches are particularly challenging for the 32×32 grid resolution (Fig. 6(b)).

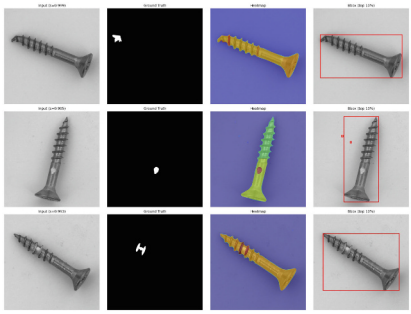


(a) Metal nut (P-AUROC = 0.845)

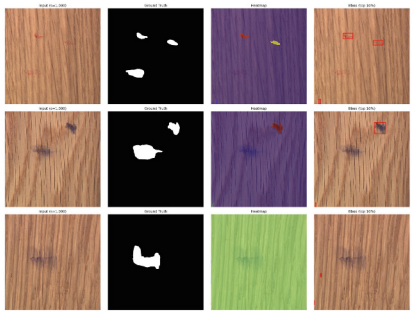


(b) Transistor (P-AUROC = 0.884)

Fig. 5. Categories with partial false positives: Metal nut and Transistor.



(a) Screw (P-AUROC = 0.619)



(b) Wood (P-AUROC = 0.775)

Fig. 6. Challenging categories with limited localization: Screw and Wood.

4.3 Ablation Studies

We conducted five ablation experiments on five representative categories (bottle, carpet, capsule, screw, leather) selected to cover both object and texture types with varying difficulty levels. Tables 3 and 4 present the results for image-level and pixel-level AUROC, respectively. The ablation studies reveal several important findings:

Finding 1: GlobalCLIP is Essential for Image-Level Detection. When GlobalCLIP is removed (A1), the I-AUROC drops to 0.500 (random level) in all categories. This is because PatchCore and SAM normalization generate similar maximum pixel scores for all images regardless of whether anomalies are present. That is, the GlobalCLIP gating mechanism is essential for determining which images contain anomalies.

Finding 2: The Detection Gate Creates a Localization Trade-off. Interestingly, the P-AUROC was higher when the GlobalCLIP gate was excluded (A1). This occurs because the gate misclassifies borderline anomalous images as normal

Table 3. I-AUROC.

Experiment	bottle	carpet	capsule	screw	leather	Mean
Full	0.992	0.965	0.844	0.741	0.999	**0.908**
A1: No GlobalCLIP	0.500	0.500	0.500	0.500	0.500	0.500
A2: No SAM	0.994	0.965	0.849	0.755	0.999	0.912
A3: Mahalanobis	0.992	0.966	0.848	0.760	0.999	0.913
A4: GlobalCLIP only	0.992	0.966	0.850	0.763	0.999	0.914

and assigns them a pixel score of 0, which prevents any localization from being recovered downstream. When all images pass through Stage 2 instead, more anomalous regions are correctly localized in aggregate. This reveals a technical limitation: improving operational efficiency through the gating mechanism inevitably sacrifices a portion of localization completeness.

Finding 3: SAM Refinement Has Mixed Effects. Removing the SAM (A2) rather improves the average P-AUROC from 0.827 to 0.871. This is because the segment-level score averaging of the SAM dilutes fine anomalous signals, especially for small defects (e.g., leather improves from 0.862 to 0.996 without SAM). However, SAM provides boundary-aware position estimation that fits the anomaly map to the object contour, which results in a clean analysis even if the pixel-level indicator falls.

Table 4. P-AUROC.

Experiment	bottle	carpet	capsule	screw	leather	Mean
Full	0.919	0.956	0.780	0.616	0.862	0.827
A1: No GlobalCLIP	0.827	0.941	0.929	0.933	0.854	0.897
A2: No SAM	0.975	0.983	0.795	0.605	0.996	0.871
A3: Mahalanobis	0.917	0.961	0.779	0.599	0.882	0.828
A4: GlobalCLIP only	N/A	N/A	N/A	N/A	N/A	N/A

Finding 4: k-NN Outperforms Mahalanobis Distance. The entire pipeline with k-NN is 1.6 times faster, achieving P-AUROC comparable to that of A3 (Mahalanobis). The Mahalanobis distance assumes that the feature follows a Gaussian distribution, and since this assumption does not really hold in multi-scale patch features, the nonparametric k-NN approach is more robust as it does not make assumptions about the distribution.

Finding 5: Speed–Accuracy Trade-offs. As shown in Table 5, the A2 variant (no SAM) runs 4.9× faster with better pixel metrics, making it the most practical for high-throughput scenarios. A4 (GlobalCLIP only) achieves 11.3× speedup with strong image-level detection (0.914 mean AUROC), suitable when localization is not required.

Table 5. Processing time.

Experiment	Total Time	Per Category	Speed vs Full	Note
Full	34.0 min	6.8 min	1.0×	Baseline
A1: No GlobalCLIP	48.3 min	9.7 min	1.4× slower	All images enter Stage 2
A2: No SAM	6.9 min	1.4 min	4.9× faster	No SAM overhead
A3: Mahalanobis	53.7 min	10.7 min	1.6× slower	Covariance computation
A4: GlobalCLIP only	3.0 min	0.6 min	11.3× faster	Stage 1 only

5 Discussion

The framework shows distinct performance differences between texture and object categories. Texture categories (carpet, grid, leather, tile, wood) achieve a higher I-AUROC (average of 0.981) because texture anomalies produce unique global feature changes that CLIP encoders effectively capture. On the other hand, object categories exhibit a larger variance (average of 0.894 I-AUROC), primarily because pose changes in categories such as screw and capsule produce feature differences that confuse global detection even when the samples are not anomalous. In pixel-level localization, texture categories show more diverse performance: carpet (0.956) and tile (0.929) achieve strong results, while wood (0.775) and grid (0.785) are weaker, suggesting that PatchCore grid resolution and SAM segment averaging may not capture long linear defects or grid alignment variations.

This study has several limitations. Due to the sequential dependence between modules, false negatives in Stage 1 are forcibly assigned a pixel score of 0, leaving no opportunity for recovery in later stages. SAM ViT-H is the primary computational bottleneck; the A2 ablation shows that removing SAM provides 4.9× speedup with better pixel metrics, suggesting bilinear upscaling may be preferable for high-throughput applications. The fixed 32×32 patch grid limits spatial precision for small defects, and uniform hyperparameters across all 15 categories, while demonstrating generalization, leave room for category-specific improvement. Finally, although our framework uses no defect samples or labels, it does utilize normal training images and therefore does not conform to the strictest zero-shot setting; future work may explore fully zero-shot localization that does not require normal training images, such as approaches exploiting pseudo-normal images within the test set (e.g., MuSc).

The self-cyclic learning process in Stage 1 exhibits statistical convergence characteristics. The ARS module's IQR-based threshold ensures robustness in non-Gaussian scoring distributions, and the ensemble of $K = 6$ reconstruction models with varying bottleneck dimensions acts as a form of multi-resolution analysis, capturing anomaly patterns at different compression rates. The AAW module connects to the Positive and Unlabeled (PU) learning literature [21] by providing principled learning from positive and unlabeled data. Across the 15

categories, the adaptive threshold τ ranges from 0.791 (Tile) to 0.898 (Wood), demonstrating that the framework automatically adjusts sensitivity based on each category's feature distribution.

6 Conclusion

We presented a multi-stage statistical framework for zero-shot industrial anomaly detection and localization that combines GlobalCLIP's adaptive ensemble learning with PatchCore k-NN localization and SAM boundary refinement. The framework achieves 92.49% mean I-AUROC under zero-shot conditions and 84.62% mean P-AUROC across all 15 MVTec AD categories, using fixed hyperparameters throughout.

Our ablation studies provide actionable insights for practitioners. The GlobalCLIP gate is indispensable for image-level detection. SAM refinement offers boundary-aware localization but may reduce pixel-level metrics for small defects. Removing SAM yields the best speed–accuracy trade-off for most applications. The framework's modular design allows practitioners to select the appropriate configuration based on their requirements for detection accuracy, localization precision, and computational budget.

Future research directions include replacing fixed gating thresholds with learning-based or adaptive mechanisms to reduce undetected anomalies, exploring higher resolution feature grids to improve small defect localization, and adding a heatmap calibration module to SAM to improve segment quality.

Acknowledgments. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP)-ICT Creative Consilience Program grant funded by the Korea government (MSIT)(IITP-2026-RS-2020-II201821), This work was supported by the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government under the Industrial Innovation Talent Growth Support Project.(KIAT-2026-RS-2026-25532917)

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bergmann, P., et al.: MVTec AD – a comprehensive real-world dataset for unsupervised anomaly detection. In: CVPR, pp. 9592–9600 (2019). https://doi.org/10.1109/CVPR.2019.00982
2. Roth, K., et al.: Towards total recall in industrial anomaly detection. In: CVPR, pp. 14298–14308 (2022). https://doi.org/10.1109/CVPR52688.2022.01392
3. Ding, C., Pang, G., Shen, C.: Catching both gray and black swans: open-set supervised anomaly detection. In: CVPR, pp. 7378–7388 (2022). https://doi.org/10.1109/CVPR52688.2022.00724
4. Deng, H., Li, X.: Anomaly detection via reverse distillation from one-class embedding. In: CVPR, pp. 9727–9736 (2022). https://doi.org/10.1109/CVPR52688.2022.00951

5. Meng, S., et al.: MoEAD: a parameter-efficient model for multi-class anomaly detection. In: ECCV, pp. 345–361 (2024). https://doi.org/10.1007/978-3-031-73013-9_20
6. Huang, C., et al.: Registration based few-shot anomaly detection. In: ECCV, pp. 303–319 (2022). https://doi.org/10.1007/978-3-031-20053-3_18
7. Jeong, J., et al.: WinCLIP: zero-/few-shot anomaly classification and segmentation. In: CVPR, pp. 19606–19616 (2023). https://doi.org/10.1109/CVPR52729.2023.01878
8. Shen, H., et al.: GlobalCLIP: zero-shot manufacturing anomaly detection with adaptive self-cyclic ensemble learning. Expert Syst. Appl. **298**, 129448 (2026). https://doi.org/10.1016/j.eswa.2025.129448
9. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning, PMLR, vol. 139, pp. 8748–8763 (2021)
10. Zhou, Q.: AnomalyCLIP: object-agnostic prompt learning for zero-shot anomaly detection. In: ICLR (2024)
11. Cao, Y., et al.: AdaCLIP: adapting CLIP with hybrid learnable prompts for zero-shot anomaly detection. In: ECCV, pp. 55–72 (2024). https://doi.org/10.1007/978-3-031-72761-0_4
12. Paszke, A.: PyTorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
13. Ilharco, G., et al.: OpenCLIP. Zenodo (2021)
14. Harris, C.R., et al.: Array programming with NumPy. Nature **585**(7825), 357–362 (2020). https://doi.org/10.1038/s41586-020-2649-2
15. Virtanen, P., et al.: SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods **17**(3), 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2
16. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
17. Bradski, G.: The OpenCV library. Dr. Dobb's J. Softw. Tools (2000)
18. Hunter, J.D.: Matplotlib: a 2D graphics environment. Comput. Sci. Eng. **9**(3), 90–95 (2007). https://doi.org/10.1109/MCSE.2007.55
19. Kirillov, A., et al.: Segment anything. In: ICCV, pp. 3992–4003 (2023). https://doi.org/10.1109/ICCV51070.2023.00371
20. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: BMVC (2016)
21. Bekker, J., Davis, J.: Learning from positive and unlabeled data: a survey. Mach. Learn. **109**(4), 719–760 (2020)