TRI-MEM: A TRIPLE-MEMORY FRAMEWORK FOR ONLINE INDUSTRIAL ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Industrial anomaly detection must often localize defects from few normal images while object appearance changes during testing. To address this setting, we propose Tri-Mem, an online few-shot anomaly detection method designed to adapt to those changes. It preserves support-derived descriptors in a frozen core and updates separate normal and anomaly memories from an unlabeled test stream. A frozen DINOv3 encoder and support-fitted PCA projection provide a shared space for causal, score-before-update retrieval. In the reported MVTec AD 2 comparison, Tri-Mem achieves the highest pixel AUROC and pixel AUPRO among the evaluated methods at 1, 2, 4, and 8 shots. At one shot it reaches 87.6% pixel AUROC, 64.2% pixel AUPRO, and 71.9% image AUROC. The complete configuration improves average pixel AUPRO by 4.6 points and image AUROC by 3.2 points over the frozen core across the four support budgets. Results on MVTec AD, MPDD, and BTAD further characterize where the visual-memory approach transfers.

1 INTRODUCTION

Industrial defect inspection often starts with few normal reference images, yet normal appearance changes after deployment. Illumination, reflections, and texture can make a defect-free patch look distant from stored support descriptors. Fixed-memory methods such as PatchCore (Roth et al., 2022) can then mistake these changes for defects. Yet updating the reference with unlabeled test patches can admit a wrongly selected patch and propagate its error to later images. The challenge is to absorb recurring normal appearances while preserving the few-shot support evidence.

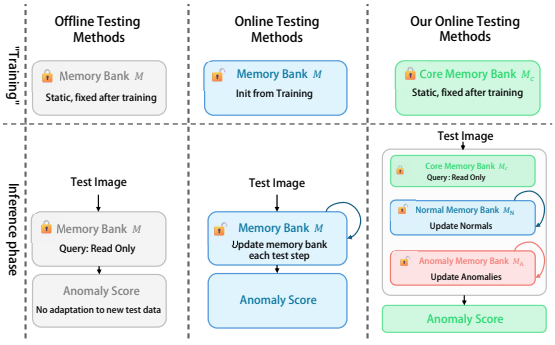


Figure 1: Comparison of memory bank update mechanisms in offline, conventional online, and proposed online testing scenarios. In Tri-Mem, each test image is scored before its selected descriptors update the adaptive memories for later images.

To address these problems, we propose Tri-Mem. A frozen core memory $\mathcal{M}_C$ retains normal support descriptors, while adaptive normal and anomaly memories collect low-score and high-score patches from the test stream. $\mathcal{M}_N$ expands coverage of benign appearance changes, and $\mathcal{M}_A$ supplies a bounded cue from earlier suspicious patches. Each image is scored before its selected patches enter update caches, so its prediction depends only on the support and earlier images. Figure 1 compares this causal order with fixed-memory and conventional online methods.

Tri-Mem controls both the feature space and timing of adaptation. Frozen DINOv3 features and support-fitted PCA place all memories in one retrieval space. The core stays fixed, the adaptive memories have separate budgets, and cached descriptors enter retrieval at later flushes. A bounded anomaly term limits the influence of $\mathcal{M}_A$ on patch scores. Figure 2 summarizes this causal pipeline without test-time gradient updates.

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

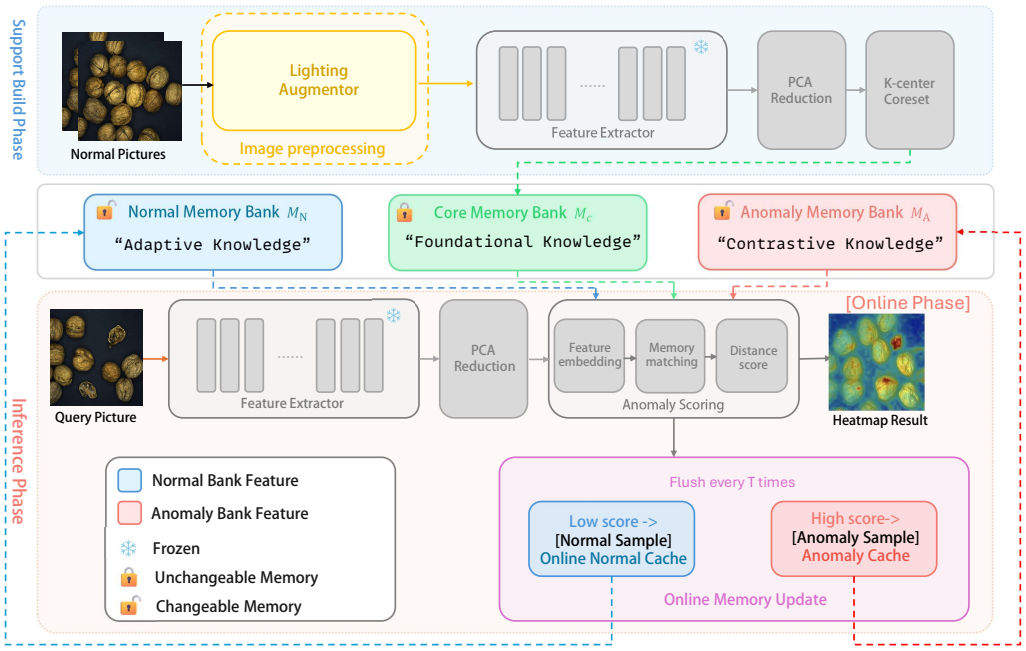


Figure 2: Tri-Mem has a frozen support-derived core and two adaptive memories. Test image t is scored before its selected patches can update the memories.

Our contributions are:

- We formulate online few-shot detection with a frozen support core and two independently bounded memories for rank-selected normal and high-score stream patches.
- We combine support-fitted frozen features, score-before-update retrieval, and a bounded anomaly-memory term so online evidence can change later scores without displacing the core.
- In the reported MVTec AD 2 comparison, Tri-Mem achieves the highest P-AUROC and P-AUPRO among the evaluated methods at all four few-shot budgets. The complete configuration improves average P-AUPRO by 4.6 points and I-AUROC by 3.2 points over the frozen core. Additional datasets and ablations define the scope of these gains.

2 Related Work

Few-Shot Anomaly Detection. Few-shot anomaly detection uses a small set of normal reference images to recognize defects in new images. Pretrained representations provide a starting point when these references cannot cover the full range of normal appearance. CLIP (Radford et al., 2021) supports this approach through visual and language representations. WinCLIP (Jeong et al., 2023) uses compositional prompts for zero-shot and few-shot inference, while APRIL-GAN (Chen et al., 2023) introduces lightweight adaptation of CLIP features. PromptAD (Li et al., 2024) learns prompts from normal samples. AnomalyGPT (Gu et al., 2023), CLIP-FSAC (Zuo et al., 2024), and FocusPatch AD (Ding et al., 2026) further explore vision-language modeling, synthetic anomalies, and patch prompts. RegAD (Huang et al., 2022) takes a visual registration approach to transferring a model across few-shot scenarios. These methods develop representations and matching rules for limited support data. A complementary question is how to incorporate the appearances observed after deployment.

Online Anomaly Detection. Online detection allows the model or its reference memory to evolve as new images arrive. PatchCore (Roth et al., 2022) stores representative normal patch descriptors for nearest-neighbor retrieval, but its original coreset construction does not specify how to maintain a memory through repeated online insertion and removal. Wei et al. (2024) study anomaly detection

and segmentation with few normal references and an incoming test stream. Ren et al. (2025) explore continual learning in dynamic manufacturing, combining a lightweight feature extractor with incremental k-center selection to incorporate informative features from new normal samples. SCAD (Wang et al., 2026) considers transductive zero-shot detection on an unlabeled test set, with automatic neighbor and sampling choices and filtering of suspicious anchors. These approaches highlight the roles of reference selection and adaptation, while also using different sources and timing of test-time information. Appendix B provides further context.

3 METHOD

Overview. Our method, Tri-Mem is a training-free few-shot anomaly detection framework for an ordered test stream. Let $\mathcal{S} = \{I_i^{\text{sup}}\}_{i=1}^{K}$ contain only normal support images, where $K \in \{1, 2, 4, 8\}$, and let $(I_1^{\text{test}}, \ldots, I_{T_{\text{test}}}^{\text{test}})$ denote the stream. For each image, the detector produces a pixel-level anomaly map and an image-level anomaly score without labels or gradient updates. As shown in Figure 2, Stage I converts the support images and their illumination-augmented views into multi-layer DINOv3 descriptors, fits a shared PCA projection, and stores deduplicated support descriptors in the frozen core memory $\mathcal{M}_C$. Stage II scores each image using $\mathcal{M}_C$, an adaptive normal memory $\mathcal{M}_N$, and a bounded cue from an adaptive anomaly memory $\mathcal{M}_A$. We denote the three-memory state after image t by $\Psi_t = (\mathcal{M}_C, \mathcal{M}_N^{(t)}, \mathcal{M}_A^{(t)})$. Low- and high-score patches selected after scoring enter separate caches and update the adaptive memories at periodic flushes. Consequently, image t uses support descriptors and observations from images $1, \ldots, t-1$, while its own descriptors become available only to later images.

3.1 STAGE I: SUPPORT BUILD PHASE

Support preparation and illumination views. First, each normal support image is resized to 720×720 and contributes its original view together with ten independently sampled illumination-augmented views. The original view bypasses all perturbations, which preserves an unaltered normal example alongside the simulated appearance changes. All resulting views are clipped and scaled to $[0, 1]$ before ImageNet normalization. Illumination augmentation is applied only to the support images, while test images retain their original appearance after resizing and normalization.

Each augmented view considers four operators in a fixed order, with highlight injection, diffuse blur, brightness and RGB shift, and specular reflection activated independently at probabilities of 0.5, 0.3, 0.7, and 0.4. Highlight injection adds three local Gaussian patches, while diffuse blur blends two locally blurred regions with the image. The third operator changes global intensity and shifts the red and blue channels asymmetrically, after which the final operator adds two anisotropic Gaussian reflections. If a Bernoulli switch is inactive, the corresponding operator leaves its input unchanged. Appendix A gives the exact masks and sampling ranges.

Frozen multi-layer feature extraction. Then, We use a frozen DINOv3 encoder (Siméoni et al., 2025). Let I_n denote an image after resizing, scaling, and ImageNet normalization. Following Dinomaly (Guo et al., 2025), we select spatial tokens from layers 2, 4, 6, and 8, normalize each token within its layer, and concatenate aligned tokens at corresponding spatial locations:

$$F_s = \text{Concat}\left[\text{Norm}_2(\mathcal{E}_2(I_n)), \text{Norm}_2(\mathcal{E}_4(I_n)), \text{Norm}_2(\mathcal{E}_6(I_n)), \text{Norm}_2(\mathcal{E}_8(I_n))\right] \in \mathbb{R}^{N \times C}. \quad (1)$$

The class token is discarded, leaving $N = H_f W_f$ spatial tokens from each selected layer. Each token has dimension D, and concatenating the four layers produces a feature dimension of $C = 4D$. Row-wise normalization within each layer places the selected tokens on a common scale before concatenation. A single frozen encoder and fusion rule process all support views and test images, keeping the feature space fixed throughout evaluation.

Constructing the frozen core memory $\mathcal{M}_C$. Finally, to construct $\mathcal{M}_C$, we flatten the feature maps of all support views in raster order and collect their descriptors as $\mathcal{Z} = \{\mathbf{z}_n\}_{n=1}^{N_f}$. We fit a PCA whitening transform to $\mathcal{Z}$, so the support descriptors determine its mean, eigenvectors, and

scaling factors. The corresponding sample mean and covariance are

$$\boldsymbol{\mu}_f = \frac{1}{N_f}\sum_{n=1}^{N_f}\mathbf{z}_n, \qquad \boldsymbol{\Sigma}_f = \frac{1}{N_f-1}\sum_{n=1}^{N_f}(\mathbf{z}_n-\boldsymbol{\mu}_f)(\mathbf{z}_n-\boldsymbol{\mu}_f)^\top. \tag{2}$$

Let $\mathbf{W}_p = [\mathbf{w}_1, \ldots, \mathbf{w}_{C_p}] \in \mathbb{R}^{C\times C_p}$ contain the retained eigenvectors of $\boldsymbol{\Sigma}_f$, and let $\boldsymbol{\Lambda}_p = \mathrm{diag}(\lambda_1, \ldots, \lambda_{C_p})$ contain their positive eigenvalues. They define the shared projection

$$\mathcal{P}(\mathbf{z}) = \boldsymbol{\Lambda}_p^{-1/2}\mathbf{W}_p^\top(\mathbf{z}-\boldsymbol{\mu}_f) \in \mathbb{R}^{C_p}. \tag{3}$$

We retain $C_p = 256$ components in the default configuration, subject to $C_p \le \mathrm{rank}(\boldsymbol{\Sigma}_f)$, and discard eigenvalues below the PCA solver's numerical tolerance. Whitening scales each retained direction by the inverse square root of its support variance. The fitted transform is reused unchanged for every query and cache flush, placing support, query, and adaptive-memory descriptors in the same space. The frozen core memory is initialized by order-preserving cosine deduplication,

$$\mathcal{M}_C = \mathrm{Dedup}_\tau(\mathcal{P}(\mathcal{Z}))\,, \qquad \tau = 0.99. \tag{4}$$

The operator scans projected support descriptors in their stored order, keeps the first, and retains a later descriptor only when its cosine similarity to every previously retained descriptor is at most τ. The core preserves a support-derived normal reference and remains unchanged throughout inference. It is stored independently of the capacity-limited adaptive memories, so online compression cannot displace support prototypes. We denote the resulting core size by $N_C = |\mathcal{M}_C|$.

3.2 Stage II: Inference Phase

With the frozen core memory $\mathcal{M}_C$ in place, Tri-Mem proceeds to the online test stage. Each arriving image is encoded and projected into the support-defined feature space, scored against the available memories, and then used to update the adaptive memories through selected patches.

Causal three-memory scoring. Each test image is resized and ImageNet-normalized without illumination augmentation. The frozen encoder produces $N_{\mathrm{qry}} = H_f W_f$ raw patch descriptors $\{\mathbf{z}_{t,j}\}_{j=1}^{N_{\mathrm{qry}}}$ in raster order, and the support-fitted transform gives $\mathbf{z}^p_{t,j} = \mathcal{P}(\mathbf{z}_{t,j})$. Before the stream begins, $\mathcal{M}_N^{(0)} = \mathcal{M}_A^{(0)} = \mathcal{C}_N^{(0)} = \mathcal{C}_A^{(0)} = \varnothing$. Image t is scored using the state after image $t-1$, with the logical normal retrieval set

$$\mathcal{M}_{\mathrm{nor}}^{(t-1)} = \mathcal{M}_C \cup \mathcal{M}_N^{(t-1)}. \tag{5}$$

The union is used only for retrieval: $\mathcal{M}_C$ remains an independent frozen memory, while $\mathcal{M}_N$ and $\mathcal{M}_A$ hold projected descriptors admitted from earlier test images. Selected raw descriptor copies reside in caches until a scheduled flush. Both memories therefore use the same support-fitted projection, but neither the test stream nor the caches refit PCA.

The normal retrieval set is nonempty because $\mathcal{M}_C$ contains support descriptors. The anomaly memory starts empty and is queried only after it has received projected high-score patches. These two cases use the same distance operator but different reference sets. Separating them avoids assigning an artificial distance to an empty anomaly memory and makes the first part of the stream a normal-reference-only scoring phase.

For a projected query $\mathbf{x}$ and memory $\mathcal{M}$, let $\kappa = \min(k_0, |\mathcal{M}|)$, and let $\mathrm{KNN}_\kappa^{\ell_2}(\mathbf{x}, \mathcal{M})$ denote its Euclidean nearest neighbors. Tri-Mem aggregates Euclidean and ℓ_1 distances:

$$d(\mathbf{x}, \mathcal{M}) = \frac{1}{\kappa}\sum_{\mathbf{u}\in\mathrm{KNN}_\kappa^{\ell_2}(\mathbf{x},\mathcal{M})}\left(\|\mathbf{x}-\mathbf{u}\|_2 + \|\mathbf{x}-\mathbf{u}\|_1\right), \qquad k_0 = 3. \tag{6}$$

Writing $d_N = d(\mathbf{z}^p_{t,j}, \mathcal{M}_{\mathrm{nor}}^{(t-1)})$, scoring begins with the normal-reference distance because the anomaly memory is initially empty:

$$s_{t,j} = d_N, \qquad \mathcal{M}_A^{(t-1)} = \varnothing. \tag{7}$$

Once $\mathcal{M}_A$ contains descriptors from earlier images, let $d_A = d(\mathbf{z}^p_{t,j}, \mathcal{M}_A^{(t-1)})$. The complete patch score is

$$s_{t,j} = d_N + w_A\frac{\gamma_A}{d_A+\gamma_A}, \qquad w_A = 0.3, \quad \gamma_A = 1.0. \tag{8}$$

The normal-reference term grows as a patch moves away from $\mathcal{M}_{\text{nor}}^{(t-1)}$. The anomaly-memory term assigns larger values to patches closer to stored high-score descriptors and is bounded above by w_A. Its scale parameter γ_A sets the distance at which this contribution reaches $w_A/2$. Scoring begins with the normal-reference term. After the first cache flush populates $\mathcal{M}_A$, the bounded proximity term adds evidence from previously observed high-score patterns.

Pixel and image outputs. The ordered patch scores are reshaped into an $H_f \times W_f$ grid, bilinearly upsampled to the resized image resolution, and smoothed by a 5×5 Gaussian kernel with $\sigma = 4$:

$$\mathbf{S}_{\text{map},t} = \mathcal{G}_{\text{map}}\left(\text{Upsample}_{H_r \times W_r}\left(\text{Reshape}_{\text{raster}}((s_{t,j})_{j=1}^{N_{\text{qry}}})\right)\right). \tag{9}$$

Image-level detection instead pools the highest patch scores before either upsampling or map smoothing. Let $\mathbf{s}_t = (s_{t,j})_{j=1}^{N_{\text{qry}}}$ collect the patch scores for image t. With $K_{\text{pool}} = \max(1, \lfloor \rho N_{\text{qry}} \rfloor)$ and $\rho = 0.02$, it uses

$$S_{\text{img},t} = \frac{1}{K_{\text{pool}}} \sum_{j \in \text{TopKIdx}(\mathbf{s}_t, K_{\text{pool}})} s_{t,j}. \tag{10}$$

Thus, image ranking uses the same memory-derived patch evidence as localization, while retaining the patch-level response before spatial interpolation.

Rank-selected candidates and cache flush. After all scores for image t are fixed, Tri-Mem selects low- and high-score descriptors as normal and anomaly candidates:

$$\begin{aligned} \mathcal{Q}_N^{(t)} &= \{\mathbf{z}_{t,j} : j \in \text{BottomKIdx}(\mathbf{s}_t, K_N)\}, \\ \mathcal{Q}_A^{(t)} &= \{\mathbf{z}_{t,j} : j \in \text{TopKIdx}(\mathbf{s}_t, K_A)\}, \\ K_X &= \min(K_{\max}, \lfloor r_X N_{\text{qry}} \rfloor), \qquad X \in \{N, A\}. \end{aligned} \tag{11}$$

We set $r_N = 0.05$, $r_A = 0.01$, and $K_{\max} = 256$. The BottomKIdx and TopKIdx operators select patches from opposite ends of a common deterministic ranking, which keeps their indices disjoint even when scores tie. Selection is performed directly on $\mathbf{s}_t$ before image-level pooling and pixel-map smoothing. The corresponding raw descriptors are then appended to the normal and anomaly caches. For $X \in \{N, A\}$, the provisional cache is

$$\widehat{\mathcal{C}}_X^{(t)} = \text{Trim}_{B_X}\left(\mathcal{C}_X^{(t-1)} \parallel \mathcal{Q}_X^{(t)}\right), \tag{12}$$

where Trim_{B_X} samples uniformly without replacement only if the appended collection exceeds its capacity. The raw caches have $B_N = 8192$ and $B_A = 2048$ descriptors. Caching raw features postpones PCA projection until insertion and prevents candidates from the current image from entering its own retrieval set.

Every $T = 10$ processed images, the provisional caches, including the current image's selected raw descriptors, are projected, deduplicated, and merged into their corresponding adaptive memories:

$$\begin{aligned} \mathcal{V}_N^{(t)} &= \text{Dedup}_\tau\left(\mathcal{M}_N^{(t-1)} \parallel \mathcal{P}(\widehat{\mathcal{C}}_N^{(t)}), \mathcal{M}_C\right), \\ \mathcal{V}_A^{(t)} &= \text{Dedup}_\tau\left(\mathcal{M}_A^{(t-1)} \parallel \mathcal{P}(\widehat{\mathcal{C}}_A^{(t)})\right), \\ \mathcal{M}_N^{(t)} &= \text{Compress}_{L_N}(\mathcal{V}_N^{(t)}), \qquad \mathcal{M}_A^{(t)} = \text{Compress}_{L_A}(\mathcal{V}_A^{(t)}). \end{aligned} \tag{13}$$

The second argument in the normal-memory deduplication is the fixed core, which is excluded from the candidates while serving as a redundancy reference. The adaptive capacities are $L_N = 80000$ and $L_A = 8000$. Appendix A gives the complete K-center compression definition. The core remains fixed while the other two memories adapt to earlier test images without gradient updates.

Update schedule and capacity control. Let $\widehat{\mathcal{C}}_X^{(t)}$ denote cache $X \in \{N, A\}$ after appending the candidates from image t and uniformly trimming it to capacity B_X. The update is explicitly separated into non-flush and flush cases:

$$\left(\mathcal{M}_X^{(t)}, \mathcal{C}_X^{(t)}\right) = \begin{cases} \left(\mathcal{M}_X^{(t-1)}, \widehat{\mathcal{C}}_X^{(t)}\right), & t \bmod T \neq 0, \\ \left(\text{Compress}_{L_X}(\mathcal{V}_X^{(t)}), \varnothing\right), & t \bmod T = 0. \end{cases} \tag{14}$$

Thus, an image can change only the raw cache immediately after it is scored. Projected candidates affect retrieval only at the next scheduled flush, and the flushed caches are cleared after their contents have been merged. This schedule makes the state used for every image unambiguous and prevents repeated insertion of the same cached descriptors.

With $T = 10$, images 1 through 10 are scored against the frozen core because the adaptive memories are still empty while their candidates accumulate in the caches. The tenth image is scored before its cache is flushed. The earliest image that can retrieve a projected adaptive descriptor is image 11. Subsequent ten-image blocks follow the same order: score using the preceding flush state, append selected raw features, and expose their projected versions only after the next flush.

Separate capacity control. Deduplication and compression serve different purposes. The cosine threshold removes near-duplicate projected descriptors before capacity enforcement. Normal candidates are additionally compared with $\mathcal{M}_C$, so a support-derived descriptor is not stored again in $\mathcal{M}_N$. If a deduplicated adaptive collection $\mathcal{V}_X^{(t)}$ exceeds L_X, K-center compression starts from one randomly chosen descriptor and, for $\ell = 2, \ldots, L_X$, adds the candidate farthest from its nearest selected center:

$$\mathbf{v}_\ell^* = \underset{\mathbf{v} \in \mathcal{V}_X^{(t)} \setminus \mathcal{K}_{\ell-1}}{\arg\max} \; \min_{\mathbf{a} \in \mathcal{K}_{\ell-1}} \|\mathbf{v} - \mathbf{a}\|_2. \tag{15}$$

The operator returns the unchanged collection when it fits within capacity and otherwise retains L_X centers. It runs independently for $\mathcal{M}_N$ and $\mathcal{M}_A$. The core is never included in adaptive compression, so its support prototypes cannot be displaced by later test samples. At any time, the normal retrieval set contains at most $N_C + L_N$ descriptors and the anomaly memory at most L_A, bounding stored references even when the stream length is not fixed.

Tri-Mem combines these two stages into a causal online detector. Stage I builds the frozen core $\mathcal{M}_C$ from normal support images. Stage II scores each test image before admitting its rank-selected patches to $\mathcal{M}_N$ and $\mathcal{M}_A$. The larger normal-memory budget covers recurring normal variation, while the smaller anomaly-memory budget retains high-score patterns for the bounded term in Eq. 8. Both adaptive memories and their raw caches have fixed capacities, while the support-derived core remains separate and unchanged. This design preserves the few-shot reference, incorporates observations from earlier test images, and keeps online storage controlled as the stream grows. Section 4 evaluates the resulting memory configurations.

4 Experiments

4.1 Experimental Setup

Datasets and metrics. We evaluate 1-, 2-, 4-, 8-, and full-shot normal-reference budgets on the officially released, publicly downloadable portion of MVTec AD 2 (Heckler-Kram et al., 2026). MVTec AD (Bergmann et al., 2019), MPDD (Jezek et al., 2021), and BTAD (Mishra et al., 2021) provide additional benchmark evaluations under the four few-shot budgets. The support set contains normal images only. We report pixel AUROC (P-AUROC), pixel AUPRO (P-AUPRO), and image AUROC (I-AUROC), which measure pixel ranking, region-level localization, and image-level discrimination, respectively.

Baselines and online protocol. We compare Tri-Mem with both CLIP-based and non-CLIP methods. The CLIP-based group includes WinCLIP (Jeong et al., 2023), IIPAD (Lv et al., 2025), and RareCLIP (He et al., 2025), while the non-CLIP group includes PatchCore (Roth et al., 2022), UniVAD (Gu et al., 2025), and AnomalyDINO (Damm et al., 2025). All online baseline variants are implemented by us on top of the corresponding released codebases, whereas the offline baselines use the official implementations. Table 1 identifies offline and online entries separately. Tri-Mem processes each test split in its fixed stream order with batch size one. Each image is scored before its candidates are admitted to the caches, and no test label is used for scoring, selection, or updating.

Implementation. All main experiments use a frozen DINOv3 ViT-B/16 encoder and 720×720 inputs. Tokens from layers 2, 4, 6, and 8 are concatenated and projected to 256 dimensions by support-fitted PCA whitening. Retrieval uses up to three Euclidean nearest neighbours and averages

Table 1: Few-shot anomaly detection results on MVTec AD 2. Each entry is reported as *P-AUROC / P-AUPRO / I-AUROC* (%). Offline and online settings are jointly reported. The best results are shown in **bold**, and the second-best results are underlined.

Category	Method	Setting	1-shot	2-shot	4-shot	8-shot
Non-CLIP	PatchCore	Offline	82.1/44.2/58.7	70.0/26.6/56.8	76.7/48.8/63.6	79.8/51.1/59.4
		Online	62.8/28.7/69.3	65.9/30.8/70.9	64.4/35.1/69.0	64.3/37.4/69.2
	AnomalyDINO	Offline	62.3/40.8/64.1	63.2/43.4/66.5	62.8/42.8/70.6	63.4/43.8/71.7
		Online	62.4/35.7/70.3	62.4/38.6/71.3	61.6/40.4/73.0	64.0/43.2/72.1
	UniVAD	Offline	86.5/54.3/64.3	85.1/55.6/66.9	85.4/54.5/67.8	86.3/55.8/69.6
		Online	65.3/38.5/54.0	67.6/37.9/58.1	72.4/37.9/56.7	69.6/38.1/56.9
CLIP-based	IIPAD	Offline	85.4/52.7/59.1	85.3/55.8/64.1	85.4/53.5/64.0	86.5/56.5/65.9
		Online	79.7/53.3/53.5	83.2/56.5/59.2	83.5/57.5/58.9	84.1/57.6/60.4
	WinCLIP	Offline	79.6/-/62.0	80.1/-/61.4	80.8/-/61.5	80.9/-/61.6
		Online	79.7/39.5/60.6	80.4/40.2/60.6	80.7/40.6/61.9	80.9/41.0/60.8
	RareCLIP	Offline	81.0/46.4/59.4	87.2/55.1/64.2	86.3/52.5/63.6	86.6/55.4/65.9
		Online	84.6/38.7/48.9	79.5/43.4/43.7	87.9/57.2/**80.1**	88.9/60.2/**78.2**
Ours	Tri-Mem	Online	**87.6/64.2/71.9**	**90.3/67.4/71.6**	**89.6/66.2**/73.1	**90.3/67.1**/73.8

Table 2: Online few-shot anomaly detection results on BTAD. Each entry is reported as *P-AUROC / P-AUPRO / I-AUROC* (%). The best results are shown in **bold**, and the second-best results are underlined.

Category	Method	1-shot	2-shot	4-shot	8-shot
Non-CLIP	PatchCore	81.1/45.6/50.3	81.3/43.4/50.6	85.7/53.9/56.7	90.4/58.9/70.6
	AnomalyDINO	92.9/69.6/95.3	74.4/55.6/93.5	74.3/55.6/93.3	74.3/55.5/92.9
	UniVAD	93.9/63.8/85.6	94.0/63.7/86.3	94.1/63.6/85.6	94.1/63.3/86.4
CLIP-based	IIPAD	96.6/**81.8**/88.2	97.0/80.6/89.1	97.2/80.7/88.7	97.2/80.2/88.1
	WinCLIP	81.5/38.8/71.9	81.5/38.9/71.2	81.5/39.0/71.1	81.4/39.0/70.8
	RareCLIP	97.3/79.0/**96.3**	**98.0**/84.4/**97.4**	98.0/85.0/**97.6**	**98.1**/82.4/97.0
Ours	Tri-Mem (Ours)	**97.8**/81.6/95.4	97.8/**84.5**/95.9	**98.1/87.2**/97.2	98.0/**87.5/97.3**

their combined $\ell_1 + \ell_2$ distances. The adaptive normal and anomaly memories are capped at 80,000 and 8,000 descriptors, and their raw caches are capped at 8,192 and 2,048 descriptors. The caches are flushed every ten processed images. Appendix A gives the augmentation and operator definitions. Appendix C reports the complete auxiliary benchmarks and sensitivity settings.

Main results. Table 1 presents the MVTec AD 2 few-shot comparison. Tri-Mem sets the highest reported P-AUROC and P-AUPRO among the evaluated methods at every support budget. At one shot, it improves P-AUPRO from the next highest 54.3% to 64.2%, a 9.9-point gain. It also leads I-AUROC at one and two shots and ranks second at four and eight shots. These rankings describe the methods and protocols shown in the table. Additional dataset results appear in Appendix C.

Full-shot comparison. Appendix Table 3 gives the full-shot MVTec AD 2 comparison. Tri-Mem achieves 91.2% P-AUROC, 64.6% P-AUPRO, and 72.6% I-AUROC, the highest values among the methods with numeric entries. AnomalyDINO, UniVAD, and IIPAD have OOM entries in the reported runs. The frozen core grows with the support set, while the two adaptive memories retain fixed capacities.

Additional benchmark results. Table 2 reports the complete 1-, 2-, 4-, and 8-shot comparison on BTAD. Tri-Mem has the highest P-AUROC at one and four shots, the highest P-AUPRO from two to eight shots, and the highest I-AUROC at eight shots. The complete MVTec AD and MPDD baseline-by-budget comparisons are retained in Appendix C.

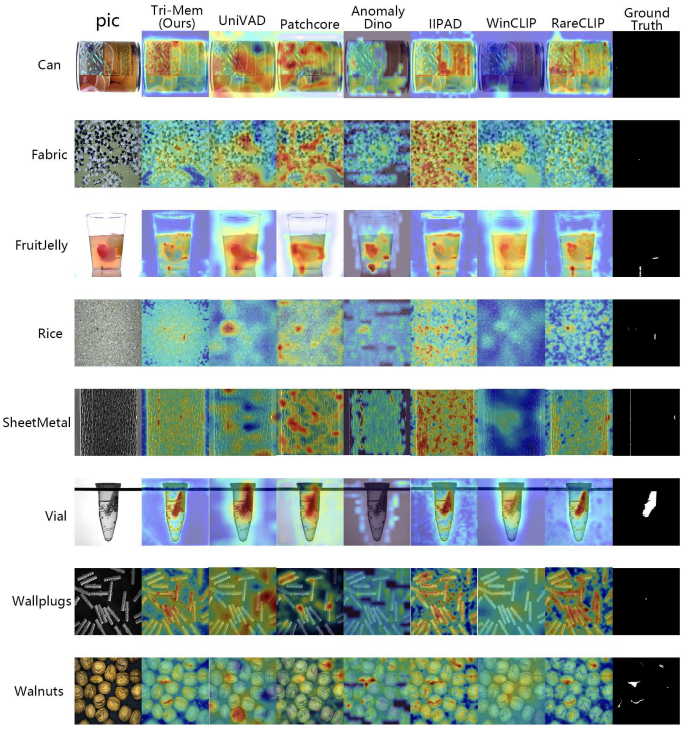


Figure 3: Qualitative anomaly-localization results on MVTec AD 2. Columns show the input image, Tri-Mem, competing methods, and the ground-truth mask.

4.2 Results on Additional Benchmarks

We additionally evaluate MVTec AD and MPDD with the same few-shot online protocol. Each benchmark uses its own normal support set and test stream. The comparisons cover CLIP-based and non-CLIP methods, showing how closely Tri-Mem's DINOv3 visual-memory pipeline matches vision-language baselines.

On MVTec AD, Tri-Mem's P-AUROC rises from 97.4% at one shot to 98.1% at eight shots, and its P-AUPRO rises from 90.5% to 93.4%. It ties the highest P-AUROC at two shots and leads at eight shots. RareCLIP has higher P-AUPRO and I-AUROC across the four budgets. The visual-memory approach therefore approaches the strongest pixel ranking on this benchmark, with a different localization and image-discrimination profile.

On MPDD, RareCLIP leads the reported table, while Tri-Mem is generally the second-ranked method for P-AUROC and remains close in image AUROC as the support budget grows. Its P-AUROC increases from 96.5% to 97.5%, and I-AUROC from 93.4% to 97.1%, between one and eight shots. The P-AUPRO values remain between 88.9% and 89.9%, so the additional support mainly changes pixel ranking and image discrimination rather than yielding a monotonic region-overlap gain.

Qualitative results. Figure 3 compares representative anomaly maps on MVTec AD 2. The examples expose the spatial concentration of each detector's response alongside the ground-truth mask, complementing the aggregate localization metrics in Table 1.

Ablations of the memories, resolution, illumination augmentation, PCA dimension, and retrieval components are reported in Appendix C.

5 CONCLUSION

Tri-Mem separates stable support knowledge from stream-adaptive descriptors and scores each image before updating its memories. In the reported MVTec AD 2 comparison, it leads P-AUROC and P-AUPRO at all four few-shot budgets and all three metrics among methods with numeric full-shot results. The complete configuration improves average P-AUPRO and I-AUROC over the frozen core. Results on MVTec AD, MPDD, and BTAD show how these gains vary by dataset and metric. The ablations support the joint configuration, while memory-activation measurements remain necessary to explain each adaptive branch's contribution.

Tri-Mem assumes clean normal support images and evaluates one fixed test-stream order. Support contamination, misranked adaptive candidates, and stream order can affect the learned projection and memory updates. Future work will quantify candidate purity, error accumulation, and order sensitivity across repeated stream permutations.

AI USE STATEMENT

In this work, we used generative AI tools to assist in writing proofs, implementing methods, translating text, reformatting datasets, and interpreting results. We did not use generative AI tools for other tasks requiring disclosure, and the remaining required disclosure tasks were not applicable to this work. We reviewed all AI-assisted work. The mathematical derivations were checked against the method definitions, the implementations were reviewed and tested for correctness, the translations were compared with the source text, the reformatted data were validated against the original records, and the interpretations were verified against the reported results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

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.

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 and 2: 1st place on zero-shot ad and 4th place on few-shot ad, 2023. URL https://arxiv.org/abs/2305.17382.

Simon Damm, Mike Laszkiewicz, Johannes Lederer, and Asja Fischer. Anomalydino: Boosting patch-based few-shot anomaly detection with dinov2, 2025. URL https://arxiv.org/abs/2405.14529.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

X. Ding, X. Li, M. Chen, J. Gong, and Y. Xie. FocusPatch AD: Few-shot multi-class anomaly detection with unified keywords patch prompts. *IEEE Transactions on Image Processing*, 35: 112–123, 2026. doi: 10.1109/TIP.2025.3646861.

Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. Anomalygpt: Detecting industrial anomalies using large vision-language models. *arXiv preprint arXiv:2308.15366*, 2023.

Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. Univad: A training-free unified model for few-shot visual anomaly detection, 2025. URL https://arxiv.org/abs/2412.03342.

Jia Guo, Shuai Lu, Weihang Zhang, Fang Chen, Huiqi Li, and Hongen Liao. Dinomaly: The less is more philosophy in multi-class unsupervised anomaly detection, 2025. URL https://arxiv.org/abs/2405.14325.

Jianfang He, Min Cao, Silong Peng, and Qiong Xie. Rareclip: Rarity-aware online zero-shot industrial anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24478–24487, October 2025.

Lars Heckler-Kram, Jan-Hendrik Neudeck, Ulla Scheler, Rebecca König, and Carsten Steger. The MVTec AD 2 dataset: Advanced scenarios for unsupervised anomaly detection. *International Journal of Computer Vision*, 134(4), March 2026. ISSN 1573-1405. doi: 10.1007/s11263-026-02743-0. URL http://dx.doi.org/10.1007/s11263-026-02743-0.

Teng Hu, Jiangning Zhang, Ran Yi, Yuzhen Du, Xu Chen, Liang Liu, Yabiao Wang, and Chengjie Wang. Anomalydiffusion: Few-shot anomaly image generation with diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. Registration based few-shot anomaly detection. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 303–319, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20053-3.

Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. Winclip: Zero-/few-shot anomaly classification and segmentation, 2023. URL https://arxiv.org/abs/2303.14814.

Stepan Jezek, Martin Jonak, Radim Burget, Pavel Dvorak, and Milos Skotak. Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions. In *2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 66–71, 2021. doi: 10.1109/ICUMT54235.2021.9631567.

Sangkyung Kwak, Jongheon Jeong, Hankook Lee, Woohyuck Kim, Dongho Seo, Woojin Yun, Wonjin Lee, and Jinwoo Shin. Few-shot anomaly detection via personalization. *IEEE Access*, 12: 11035–11051, 2024. ISSN 2169-3536. doi: 10.1109/ACCESS.2024.3355021. Publisher Copyright: © 2013 IEEE.

Xiaofan Li, Zhizhong Zhang, Xin Tan, Chengwei Chen, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Promptad: Learning prompts with only normal samples for few-shot anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16838–16848, June 2024.

Yuxuan Lin, Hanjing Yan, Xuan Tong, Yang Chang, Huanzhen Wang, Ziheng Zhou, Shuyong Gao, Yan Wang, and Wenqiang Zhang. Commonality in few: Few-shot multimodal anomaly detection via hypergraph-enhanced memory, 2025. URL https://arxiv.org/abs/2511.05966.

Wenxi Lv, Qinliang Su, and Wenchao Xu. One-for-all few-shot anomaly detection via instance-induced prompt learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Zzs3JwknAY.

Hongchi Ma, Guanglei Yang, Debin Zhao, Yanli Ji, and Wangmeng Zuo. Remp-ad: Retrieval-enhanced multi-modal prompt fusion for few-shot industrial visual anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20425–20434, October 2025.

Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. Vt-adl: A vision transformer network for image anomaly detection and localization. *arXiv preprint arXiv:2104.10036*, 2021. URL https://arxiv.org/abs/2104.10036.

Gyutae Oh and Jitae Shin. Patch core++: Augmenting coreset for superior industrial anomaly detection. In *2024 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pp. 1–4. IEEE, 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, 2021. URL https://arxiv.org/abs/2103.00020.

Haoyu Ren, Kay Koehle, Kirill Dorofeev, and Darko Anicic. On-device continual learning for unsupervised visual anomaly detection in dynamic manufacturing, 2025. URL https://arxiv.org/abs/2512.13497.

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

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

S. Wang, G. Wang, X. Liu, and et al. Scad: A self-constrained solution to automate context-guided zero-shot image anomaly detection. *Neural Networks*, pp. 108577, 2026.

Shenxing Wei, Xing Wei, Zhiheng Ma, Songlin Dong, Shaochen Zhang, and Yihong Gong. Few-shot online anomaly detection and segmentation. *Knowledge-Based Systems*, 300:112168, 2024.

Guoyang Xie, Jinbao Wang, Jiaqi Liu, Feng Zheng, and Yaochu Jin. Pushing the limits of fewshot anomaly detection in industry vision: Graphcore, 2023. URL https://arxiv.org/abs/2301.12082.

Zhenyu Yan, Qingqing Fang, Wenxi Lv, and Qinliang Su. Anomalysd: One-for-all few-shot anomaly detection via pre-trained diffusion models. *Computer Vision and Image Understanding*, 265: 104668, 2026. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2026.104668. URL https://www.sciencedirect.com/science/article/pii/S1077314226000354.

Jinsung Yoon, Kihyuk Sohn, Chun-Liang Li, Sercan O Arik, and Tomas Pfister. Spade: Semi-supervised anomaly detection under distribution mismatch. *arXiv preprint arXiv:2212.00173*, 2022.

Zuo Zuo, Yao Wu, Baoqiang Li, Jiahao Dong, You Zhou, Lei Zhou, Yanyun Qu, and Zongze Wu. Clip-fsac: Boosting clip for few-shot anomaly classification with synthetic anomalies. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 1834–1842. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/203. URL https://doi.org/10.24963/ijcai.2024/203. Main Track.

A SUPPLEMENTARY METHOD DETAILS

This appendix supplements the method in Section 3 with the illumination operators used in Stage I and the adaptive-memory compression operator used in Stage II.

A.1 STAGE I ILLUMINATION AUGMENTATION

Due to the issues of illumination variation, reflection artifacts, and texture inconsistency in anomaly detection, Tri-Mem adopts a unified image preprocessing pipeline for the support building stage to improve feature robustness under few-shot settings.

Support-Time Image Augmentation. To alleviate feature overfitting caused by limited visual diversity, support samples are further augmented using illumination-aware perturbations. Given a normal support image $I \in \mathbb{R}^{H\times W\times 3}$, we first resize it and represent its pixels in the 8-bit intensity domain:

$$I_r = \mathcal{R}(I) \in \{0,\dots,255\}^{H_r\times W_r\times 3}, \tag{16}$$

where $\mathcal{R}(\cdot)$ denotes resizing and H_r, W_r are the fixed resized dimensions.

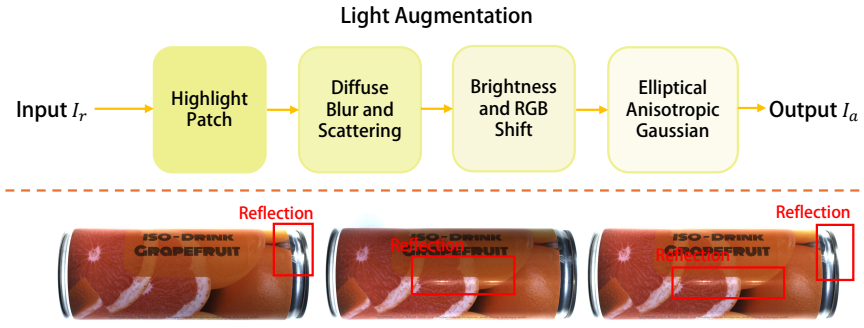


Figure 4: Support-time illumination augmentation. The top shows the fixed operator sequence from the resized image I_r to the augmented image I_a. The bottom highlights reflection, brightness, and color-temperature variations in the same normal MVTec AD 2 sample under different lighting conditions.

Support-time illumination augmentation is applied as shown in Fig. 4. Let $I^{(0)} = I_r$ and let $\delta_k \sim \mathrm{Bernoulli}(p_k)$ indicate whether the k-th operator is enabled. The four probabilities, in the order highlight injection, diffuse blur, brightness and RGB color shift, and specular reflection, are

$$(p_1, p_2, p_3, p_4) = (0.5, 0.3, 0.7, 0.4). \tag{17}$$

These fixed implementation probabilities are used throughout all experiments unless otherwise stated. For every support image, we retain the original view and generate ten independently sampled augmented views. Throughout this subsection, $\mathrm{Unif}(a,b)$ denotes the continuous uniform distribution on $[a,b]$, whereas $\mathrm{Unif}_{\mathbb{Z}}\{v_1,\dots,v_m\}$ denotes the discrete uniform distribution over the listed integer values. All primitive random draws are sampled independently unless otherwise stated. Pixel coordinates satisfy $x \in \{0,\dots,W_r-1\}$ and $y \in \{0,\dots,H_r-1\}$. The following definitions include δ_k explicitly, so that each stage reduces to the identity mapping when its switch is zero. The operator $\otimes \mathbf{1}_3$ broadcasts a two-dimensional mask across the three color channels, and $\odot$ denotes element-wise multiplication.

First, to mimic local overexposure caused by strong illumination or reflective surfaces, the implementation adds three Gaussian highlight patches. The resulting stage is

$$I^{(1)} = \mathrm{Clip}\left[I^{(0)} + 255\,\delta_1 \sum_{j=1}^{3} \alpha_j \left(M_{h,j} \otimes \mathbf{1}_3\right), 0, 255 \right], \tag{18}$$

where $M_{h,j}$ is centered at $(c^h_{x,j}, c^h_{y,j})$:

$$M_{h,j}(x,y) = \exp\left[-\frac{(x-c^h_{x,j})^2 + (y-c^h_{y,j})^2}{2\sigma^2_{h,j}}\right]. \tag{19}$$

For $j = 1, 2, 3$, the intensity, center, and integer patch size are sampled over the implemented ranges and the full resized-image domain:

$$\begin{aligned}
\alpha_j &\sim \mathrm{Unif}(0.3, 0.8),\\
c^h_{x,j} &\sim \mathrm{Unif}_{\mathbb{Z}}\{0,\dots,W_r-1\},\\
c^h_{y,j} &\sim \mathrm{Unif}_{\mathbb{Z}}\{0,\dots,H_r-1\},\\
\ell_{h,j} &\sim \mathrm{Unif}_{\mathbb{Z}}\{20,\dots,80\},\\
\sigma_{h,j} &= \frac{\ell_{h,j}}{2}.
\end{aligned}$$

Then, to approximate local surface scattering and optical blur, the implementation sequentially blends two locally blurred regions:

$$\begin{aligned}
I^{(0)}_d &= I^{(1)},\\
I^{(j)}_d &= (1-\delta_2 M_{d,j}) \odot I^{(j-1)}_d\\
&\quad + \delta_2 M_{d,j} \odot \mathcal{G}_{k_j}\left(I^{(j-1)}_d\right), \quad j = 1, 2,\\
I^{(2)} &= I^{(2)}_d.
\end{aligned} \tag{20}$$

where $\mathcal{G}_{k_j}(\cdot)$ denotes Gaussian filtering with a $k_j \times k_j$ kernel and automatically inferred standard deviation, as in the implementation. The regional blending mask is

$$M_{d,j}(x,y) = \max\left(1 - \frac{d_j(x,y)}{r_{d,j}}, 0\right), \tag{21}$$

$$d_j(x,y) = \sqrt{(x-c^d_{x,j})^2 + (y-c^d_{y,j})^2}. \tag{22}$$

The blur center covers the full resized image, the regional radius is sampled from 50 to 150 pixels, and the kernel size is sampled uniformly from the implemented odd sizes:

$$\begin{aligned}
c^d_{x,j} &\sim \mathrm{Unif}_{\mathbb{Z}}\{0,\dots,W_r-1\},\\
c^d_{y,j} &\sim \mathrm{Unif}_{\mathbb{Z}}\{0,\dots,H_r-1\},\\
r_{d,j} &\sim \mathrm{Unif}_{\mathbb{Z}}\{50,\dots,150\},\\
k_j &\sim \mathrm{Unif}_{\mathbb{Z}}\{5,7,9,11,13,15\}.
\end{aligned}$$

To simulate illumination intensity and RGB color shifts, let

$$\begin{aligned}
\beta_\delta &= 1 + \delta_3(\beta - 1),\\
\beta &\sim \mathrm{Unif}(0.7, 1.3),\\
\Delta &\sim \mathrm{Unif}_{\mathbb{Z}}\{-30,\dots,30\},\\
\Delta_+ &= \max(\Delta, 0), \qquad \Delta_- = \min(\Delta, 0).
\end{aligned} \tag{23}$$

The implementation stores support images in RGB order. Its actual channel operations, including the asymmetric positive and negative shifts, are therefore

$$\begin{aligned}
I^{(3)}_{\mathrm{R}} &= \beta_\delta I^{(2)}_{\mathrm{R}} + \delta_3\left(-\tfrac{1}{2}\Delta_+ - \Delta_-\right),\\
I^{(3)}_{\mathrm{G}} &= \beta_\delta I^{(2)}_{\mathrm{G}},\\
I^{(3)}_{\mathrm{B}} &= \beta_\delta I^{(2)}_{\mathrm{B}} + \delta_3\left(\Delta_+ + \tfrac{1}{2}\Delta_-\right),\\
I^{(3)} &= \mathrm{Clip}\Big[\mathrm{Concat}_{\mathrm{RGB}}\left(I^{(3)}_{\mathrm{R}}, I^{(3)}_{\mathrm{G}}, I^{(3)}_{\mathrm{B}}\right), 0, 255\Big].
\end{aligned} \tag{24}$$

Thus, $\Delta < 0$ increases the red channel by $|\Delta|$ and decreases the blue channel by $|\Delta|/2$, whereas $\Delta > 0$ increases the blue channel by Δ and decreases the red channel by $\Delta/2$. These two cases produce red- and blue-dominant illumination shifts, respectively, and $I^{(3)} = I^{(2)}$ when $\delta_3 = 0$.

Finally, to imitate mirror-like surface reflections, the implementation adds two anisotropic Gaussian masks. For $j = 1, 2$, define

$$\begin{aligned} x_{s,j} &= (x - c^s_{x,j}) \cos\theta_{s,j} + (y - c^s_{y,j}) \sin\theta_{s,j}, \\ y_{s,j} &= -(x - c^s_{x,j}) \sin\theta_{s,j} + (y - c^s_{y,j}) \cos\theta_{s,j}, \\ M_{s,j}(x, y) &= \exp\left[-\frac{1}{2} \left(\frac{x^2_{s,j}}{(q_j s_j/3)^2} + \frac{y^2_{s,j}}{(s_j/3)^2} \right) \right]. \end{aligned} \tag{25}$$

The centers cover the full resized-image domain. The integer size s_j, aspect ratio q_j, orientation $\theta_{s,j}$, and reflection intensity η_j are sampled from the implemented ranges

$$\begin{aligned} c^s_{x,j} &\sim \mathrm{Unif}_{\mathbb{Z}}\{0, \ldots, W_r - 1\}, \\ c^s_{y,j} &\sim \mathrm{Unif}_{\mathbb{Z}}\{0, \ldots, H_r - 1\}, \\ s_j &\sim \mathrm{Unif}_{\mathbb{Z}}\{30, \ldots, 100\}, \\ q_j &\sim \mathrm{Unif}(0.5, 2.0), \\ \theta_{s,j} &\sim \mathrm{Unif}(0, \pi), \\ \eta_j &\sim \mathrm{Unif}(0.5, 1.0). \end{aligned}$$

The final augmented image is

$$\begin{aligned} I_a &= I^{(4)} \\ &= \mathrm{Clip}\left[I^{(3)} + 255\, \delta_4 \sum_{j=1}^{2} \eta_j \left(M_{s,j} \otimes \mathbf{1}_3 \right), 0, 255 \right]. \end{aligned} \tag{26}$$

After augmentation, intensities are clipped, converted to $[0, 1]$, and normalized channel-wise using standard ImageNet statistics (Deng et al., 2009):

$$\bar{I}_a = \frac{\mathrm{Clip}(I_a, 0, 255)}{255}, \qquad I_n = \frac{\bar{I}_a - \boldsymbol{\mu}}{\boldsymbol{\sigma}}, \tag{27}$$

where

$$\begin{aligned} \boldsymbol{\mu} &= (0.485, 0.456, 0.406), \\ \boldsymbol{\sigma} &= (0.229, 0.224, 0.225). \end{aligned}$$

Subtraction and division are applied channel-wise.

To reduce GPU memory consumption and accelerate inference, illumination augmentation is disabled during inference and is applied only during the support build phase.

A.2 Stage II Adaptive Memory Compression

To bound each adaptive memory during inference, we define a generic capacity-triggered K-center operator. Let $\mathcal{V}$ be a deduplicated adaptive candidate collection and let $L \in \mathbb{N}_+$ be its capacity. When $|\mathcal{V}| > L$, the coreset is initialized with a randomly selected descriptor:

$$\mathcal{K}_1 = \{\mathbf{v}_r\}, \qquad \mathbf{v}_r \in \mathcal{V}. \tag{28}$$

For $\ell = 2, \ldots, L$, the next representative is selected by maximizing its minimum Euclidean distance to the current coreset:

$$\mathbf{v}^*_\ell = \operatorname*{arg\,max}_{\mathbf{v} \in \mathcal{V} \setminus \mathcal{K}_{\ell-1}} \; \min_{\mathbf{a} \in \mathcal{K}_{\ell-1}} \|\mathbf{v} - \mathbf{a}\|_2 . \tag{29}$$

The coreset is subsequently updated as

$$\mathcal{K}_\ell = \mathcal{K}_{\ell-1} \cup \{\mathbf{v}^*_\ell\} . \tag{30}$$

The resulting compression operator is

$$\mathrm{Compress}_L(\mathcal{V}) = \begin{cases} \mathcal{V}, & |\mathcal{V}| \le L, \\ \mathcal{K}_L, & |\mathcal{V}| > L. \end{cases} \tag{31}$$

Tri-Mem applies this operator separately to the adaptive normal and anomaly memories, which have capacities L_N and L_A, respectively. The core memory $\mathcal{M}_C$ is not compressed during online updates.

B Additional Related Work

B.1 Few-Shot Anomaly Detection

Few-shot anomaly detection builds a normal reference from a small support set. Pretrained features reduce the amount of target data needed to establish that reference, while support adaptation determines how those features are used. CLIP (Radford et al., 2021) provides visual and language representations for this setting. WinCLIP (Jeong et al., 2023) uses compositional prompts for zero-shot and few-shot prediction, APRIL-GAN (Chen et al., 2023) adapts CLIP features with lightweight layers, and PromptAD (Li et al., 2024) learns prompts from normal samples. AnomalyGPT (Gu et al., 2023) uses a large vision-language model for industrial anomaly detection. CLIP-FSAC (Zuo et al., 2024) incorporates synthetic anomalies, and FocusPatch AD (Ding et al., 2026) studies patch prompts for multiple categories. RegAD (Huang et al., 2022) uses image registration to transfer a model across few-shot scenarios.

Visual retrieval methods represent the support set through stored descriptors. PatchCore (Roth et al., 2022) selects a coreset of normal patch features for nearest-neighbor matching, and PatchCore++ (Oh & Shin, 2024) extends this coreset approach. GraphCore (Xie et al., 2023) introduces graph-based modeling for few-shot inspection, ReMP-AD (Ma et al., 2025) studies multimodal retrieval, and hypergraph-enhanced memory (Lin et al., 2025) models commonality across few examples. These approaches address how to represent limited support evidence. Tri-Mem preserves that evidence in a frozen core and uses separate memories to incorporate observations from the test stream.

Generative approaches provide another route to learning from few examples. Model personalization (Kwak et al., 2024), AnomalyDiffusion (Hu et al., 2024), and AnomalySD (Yan et al., 2026) use diffusion models for anomaly modeling or synthesis. The source of the support data matters here. Methods that synthesize defects from anomalous examples have access to information beyond the normal-only support set used by Tri-Mem. Our setting uses normal support images and unlabeled test observations to construct visual references.

B.2 Online Anomaly Detection

Online anomaly detection updates the model or reference state as observations arrive. A fixed support memory can miss normal appearances that recur during deployment, which motivates adapting the reference to new data. Wei et al. (2024) study few-shot online anomaly detection and segmentation. For dynamic manufacturing, Ren et al. (2025) combine a lightweight feature extractor with incremental k-center selection to incorporate informative features from new normal samples. This approach makes memory maintenance part of adaptation, since the stored descriptors must remain representative as new observations accumulate.

Test-time adaptation also depends on which observations are available before a prediction. SCAD (Wang et al., 2026) performs transductive zero-shot detection on an unlabeled test set, with automatic neighbor and sampling choices and filtering of suspicious anchors. SPADE (Yoon et al., 2022) studies semi-supervised anomaly detection under distribution mismatch. Tri-Mem uses an ordered unlabeled stream. An image is scored using the support core and memory state established by earlier images, and its selected descriptors become available through subsequent cache flushes. The fixed core and independently bounded normal and anomaly memories give support evidence and stream observations distinct roles in this update process.

C Detailed Experimental Results

C.1 Experimental Setup

Following the few-shot anomaly detection protocol, we report results under five reference-image budgets: 1-shot, 2-shot, 4-shot, 8-shot, and full-shot settings. Performance is measured by pixel-level AUROC (P-AUROC), pixel-level AUPRO (P-AUPRO), and image-level AUROC (I-AUROC). We compare Tri-Mem against six baselines: PatchCore (Roth et al., 2022), WinCLIP (Jeong et al., 2023), IIPAD (Lv et al., 2025), RareCLIP (He et al., 2025), UniVAD (Gu et al., 2025) and AnomalyDINO (Damm et al., 2025).

Table 3: Full-shot anomaly detection results on MVTec AD 2. Each entry is reported as *P-AUROC / P-AUPRO / I-AUROC* (%). OOM indicates out-of-memory during the reported full-shot run. The best results are shown in **bold**, and the second-best results are underlined.

Category	Method	Full-shot
Non-CLIP	PatchCore	85.1 / 52.0 / 68.0
	AnomalyDINO	*OOM*
	UniVAD	*OOM*
CLIP-based	IIPAD	*OOM*
	WinCLIP	81.3 / - / 60.5
	RareCLIP	88.3 / 56.4 / 67.8
Ours	Tri-Mem	**91.2 / 64.6 / 72.6**

Table 4: Few-shot online anomaly detection results on MVTec AD. Each entry is reported as *P-AUROC / P-AUPRO / I-AUROC* (%). The best results are shown in **bold**, and the second-best results are underlined.

Category	Method	1-shot	2-shot	4-shot	8-shot
Non-CLIP	PatchCore	44.6/18.8/46.0	77.6/56.6/62.1	89.7/66.8/60.8	92.3/75.1/72.3
	AnomalyDINO	79.7/48.2/49.3	81.8/53.2/56.0	83.0/57.5/60.3	85.7/64.3/72.3
	UniVAD	96.1/88.6/75.2	96.2/88.3/75.0	96.4/89.4/77.4	92.7/74.3/77.0
CLIP-based	IIPAD	91.6/73.2/60.9	91.6/73.1/60.9	91.2/72.5/63.3	92.6/76.1/68.3
	WinCLIP	92.1/74.6/77.0	92.6/78.3/63.9	92.5/78.3/63.8	92.5/79.3/64.3
	RareCLIP	**97.7/93.8/82.2**	**97.7/94.3/86.1**	**98.0/95.1/88.3**	98.0/**95.3/90.3**
Ours	Tri-Mem (Ours)	97.4/90.5/76.5	**97.7**/91.9/77.3	97.8/91.7/78.7	**98.1**/93.4/82.7

Full-shot MVTec AD 2 comparison.

C.2 PARAMETERS

Unless otherwise stated in the resolution ablation, all main experiments use a frozen DINOv3 ViT-B/16 backbone with a patch size of 16 and square RGB inputs resized to 720×720. We extract intermediate patch tokens from transformer layers 2, 4, 6, and 8. Features from each layer are independently ℓ_2-normalized and concatenated without learnable fusion parameters. ImageNet normalization is applied using mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$.

The concatenated patch descriptors are whitened and reduced to 256 dimensions by PCA fitted on the support features. For anomaly retrieval, up to three nearest neighbours are first selected according to Euclidean distance, after which their combined $\ell_1 + \ell_2$ distances are averaged. The frozen core memory contains N_C deduplicated support descriptors and is stored independently. The adaptive normal and anomaly memories are capped at 80,000 and 8,000 descriptors, respectively. The logical normal retrieval set therefore contains at most $N_C + 80{,}000$ descriptors. Capacity-triggered K-center compression is applied separately whenever either adaptive memory exceeds its own capacity. The bounded anomaly-memory proximity term uses a maximum weight of 0.3 and a half-similarity distance scale of 1.0.

The support-time illumination augmentation module generates ten independently sampled augmented views per support image, in addition to the retained original view. Highlight injection, diffuse blur, brightness and RGB color shift, and specular reflection are applied with probabilities 0.5, 0.3, 0.7, and 0.4, respectively. In the default 720×720 setting, the anomaly map is bilinearly upsampled to that resolution and smoothed with a 5×5 Gaussian kernel with $\sigma = 4$. The image-level patch score is computed from the top 2% patch responses. The reported inference path uses the three-memory patch-score formulation in Section 3: $\mathcal{M}_C$ and $\mathcal{M}_N$ jointly form the normal retrieval set, while $\mathcal{M}_A$ provides the anomaly reference.

Table 5: Few-shot online anomaly detection results on MPDD. Each entry is reported as *P-AUROC / P-AUPRO / I-AUROC* (%). The best results are shown in **bold**, and the second-best results are underlined.

Category	Method	1-shot	2-shot	4-shot	8-shot
Non-CLIP	PatchCore	75.6/43.3/89.5	83.0/56.1/89.8	84.7/64.2/88.9	87.9/70.4/90.5
	AnomalyDINO	89.1/75.9/92.5	89.5/76.8/92.8	90.3/77.6/94.3	90.7/78.9/94.7
	UniVAD	89.0/73.0/85.0	89.4/74.0/85.9	89.9/74.9/87.0	90.5/75.8/88.2
CLIP-based	IIPAD	92.4/77.7/87.3	92.8/78.3/88.4	96.7/91.3/96.5	97.0/91.5/96.7
	WinCLIP	85.8/69.1/75.7	86.3/70.9/77.9	86.9/71.6/79.0	87.3/72.3/79.9
	RareCLIP	**97.2/91.5/94.3**	**97.6/92.3/95.4**	**97.6/92.5/96.9**	**98.4/93.9/98.3**
Ours	Tri-Mem (Ours)	96.5/88.9/93.4	96.9/89.3/95.0	97.2/89.9/96.2	97.5/89.8/97.1

Online adaptation caches the lowest 5% of patch scores as normal candidates and the highest 1% as anomaly candidates. At most 256 patches of either type are collected from each image. The caches are flushed every 10 processed test images and are capped at 8,192 normal and 2,048 anomalous patches. We use batch size 1, enable online updating in all main experiments, and evaluate the support budgets $n_{\text{shot}} \in \{1, 2, 4, 8\}$.

C.3 Result Analysis

On MVTec AD 2, Tri-Mem leads the reported P-AUROC and P-AUPRO comparisons at every few-shot budget in Table 1. At one shot it reaches 87.6% P-AUROC and 64.2% P-AUPRO. The next highest values in the table are 86.5% and 54.3%, giving gains of 1.1 and 9.9 percentage points. At eight shots, Tri-Mem reaches 90.3% P-AUROC and 67.1% P-AUPRO. The localization advantage is largest in P-AUPRO.

Tri-Mem also leads I-AUROC at one and two shots. RareCLIP leads I-AUROC at four and eight shots, where Tri-Mem ranks second. The two methods use different representations, and the reported table locates Tri-Mem's strongest advantage in pixel and region-level localization.

In the full-shot MVTec AD 2 comparison, Tri-Mem attains the highest reported values among methods with numeric entries: 91.2% P-AUROC, 64.6% P-AUPRO, and 72.6% I-AUROC (Table 3). AnomalyDINO, UniVAD, and IIPAD have OOM entries for their reported runs. On MVTec AD, Tri-Mem ranks second in P-AUROC at one and four shots, ties for first at two shots, and leads at eight shots (Table 4). RareCLIP has higher P-AUPRO and I-AUROC across these budgets.

On MPDD, Tri-Mem's P-AUROC increases from 96.5% to 97.5% and its I-AUROC from 93.4% to 97.1% between one and eight shots (Table 5). RareCLIP leads the reported results across the main metrics. The MPDD comparison shows a smaller advantage for visual-memory adaptation than MVTec AD 2.

On BTAD, Tri-Mem leads P-AUROC at one and four shots and P-AUPRO at two, four, and eight shots (Table 2). At eight shots it reaches 87.5% P-AUPRO, 5.1 points above RareCLIP, and the highest I-AUROC of 97.3%. Among the additional benchmarks, BTAD shows Tri-Mem's clearest advantage in region-level localization.

Across the evaluated datasets, Tri-Mem's strongest reported gains lie in MVTec AD 2 pixel and region-level localization and BTAD region-level localization. Its frozen DINOv3 descriptors and causal memory updates provide these results without target-domain gradient updates. Overall, Tri-Mem performs at a similar level to the CLIP-based baselines across the evaluated datasets and support budgets. It uses DINOv3 visual features without language prompts, while matching or exceeding the CLIP-based methods on several localization metrics.

C.4 Ablation

Unless explicitly stated otherwise, all ablations use 720×720 inputs and the default parameters described above. The complete-model reference is always taken from the Tri-Mem row of Table 1:

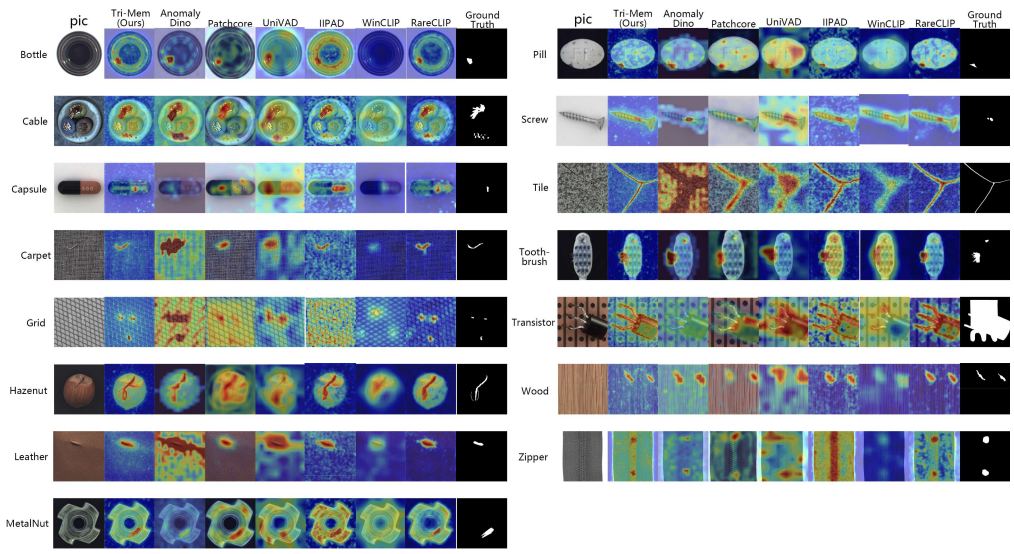


Figure 5: Qualitative visualization results on MVTec AD.

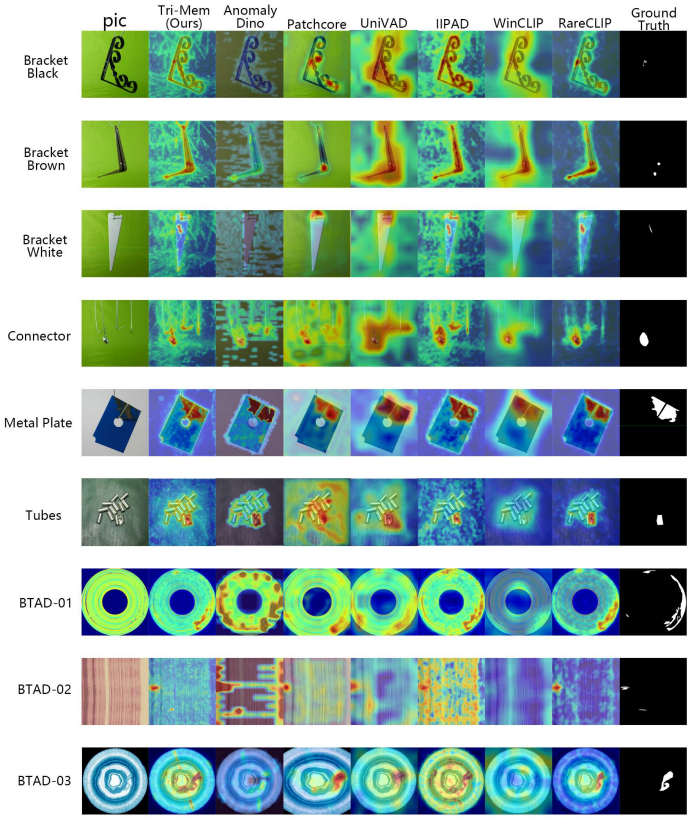


Figure 6: Qualitative visualization results on MPDD and BTAD.

Table 6: Selected one-shot MVTec AD 2 sensitivity results (%). Full illumination and PCA-dimension ablations follow below.

Variant	I-AUROC	P-AUROC	P-AUPRO
Default (PCA 256)	71.90	87.60	64.20
No illumination augmentation	67.65	87.87	63.34
PCA 64	65.52	87.66	67.06
PCA 512	68.87	89.39	64.54
No PCA	67.95	85.02	64.16

Table 7: Resolution and shot sensitivity of Tri-Mem on MVTec AD 2. Each entry is reported as *P-AUROC / P-AUPRO / I-AUROC* (%). The best results are shown in **bold**, and the second-best results are underlined.

Image size	1-shot	2-shot	4-shot	8-shot
512	**89.2**/61.2/68.1	88.5/61.5/69.3	88.5/62.8/70.0	89.4/63.2/69.5
600	88.2/61.8/66.3	89.6/62.3/67.1	**89.9**/62.8/67.8	90.0/63.4/71.4
720	87.6/**64.2/71.9**	**90.3/67.4/71.6**	89.6/**66.2/73.1**	**90.3/67.1/73.8**

$87.6/64.2/71.9$, $90.3/67.4/71.6$, $89.6/66.2/73.1$, and $90.3/67.1/73.8$ for the 1-, 2-, 4-, and 8-shot settings, respectively, where each entry denotes P-AUROC/P-AUPRO/I-AUROC (%). Ablated variants are compared with the corresponding complete-model result under the same shot budget.

Selected one-shot variants.

C.4.1 Resolution and Shot Sensitivity

We further evaluate the influence of input resolution and shot number on Tri-Mem. Since small industrial defects are often spatially subtle, the input resolution directly affects whether fine-grained anomaly structures can be preserved by the dense DINOv3 patch-token grid. Table 7 reports the detection performance under different image sizes and few-shot settings. Each entry is reported as *P-AUROC / P-AUPRO / I-AUROC*.

Table 7 shows that 720×720 gives the highest P-AUPRO among the tested resolutions at every support budget. Relative to 512 pixels, P-AUPRO rises from 61.2%, 61.5%, 62.8%, and 63.2% to 64.2%, 67.4%, 66.2%, and 67.1% at one, two, four, and eight shots. We use 720×720 as the default input size. Its row reproduces the Tri-Mem values in the main comparison table.

Table 8 reports the corresponding computational cost. More support tokens increase the stored core size as resolution and shot number grow. Peak GPU memory reaches 1.57 GB at 720 pixels and eight shots in the tested implementation.

High-resolution DINOv3 patch-token extraction and nearest-neighbour retrieval dominate inference cost. At the default 720-pixel resolution, Table 8 reports 283.51 ms per image and 0.65 GB peak GPU memory at one shot, rising to 329.67 ms and 1.57 GB at eight shots. These measurements characterize the tested implementation alongside its localization results.

C.4.2 Effect of Memory Banks

We compare four memory configurations across all four few-shot budgets. Table 9 reports the complete configuration-by-budget results.

Mechanism interpretation. The frozen core provides a strong reference. Averaged over the four budgets, $\mathcal{M}_C$ obtains 89.6% P-AUROC, 61.6% P-AUPRO, and 69.4% I-AUROC. Adding $\mathcal{M}_N$ changes these averages to 89.0%, 62.1%, and 69.7%. Its clearest gain occurs at one shot, where P-AUPRO rises from 57.7% to 60.3%. The adaptive normal memory expands normal coverage most visibly with the smallest support set, though its effect varies across metrics and budgets.

Table 8: Efficiency analysis under different image resolutions and few-shot settings on MVTec AD 2.

Size	Shot	Time (ms/img)	GPU Mem. (GB)
512	1-shot	218.76	0.43
	2-shot	253.66	0.45
	4-shot	257.84	0.48
	8-shot	387.41	0.71
600	1-shot	314.16	0.48
	2-shot	251.36	0.49
	4-shot	252.60	0.54
	8-shot	264.89	0.95
720	1-shot	283.51	0.65
	2-shot	236.47	0.69
	4-shot	369.57	0.78
	8-shot	329.67	1.57

Table 9: Memory-bank ablation on MVTec AD 2 across four few-shot budgets. Each entry is *P-AUROC / P-AUPRO / I-AUROC* (%). Bold marks the best result for each metric.

Memory configuration	1-shot	2-shot	4-shot	8-shot
$\mathcal{M}_C$	**88.4**/57.7/66.7	89.9/63.2/68.2	**90.0**/63.6/71.9	90.0/62.0/70.7
$\mathcal{M}_C + \mathcal{M}_N$	**88.4**/60.3/67.6	89.1/62.6/68.1	89.2/62.1/71.9	89.3/63.5/71.0
$\mathcal{M}_C + \mathcal{M}_A$	**88.4**/57.7/66.7	89.9/63.1/68.2	**90.0**/63.6/71.9	90.0/62.0/70.7
$\mathcal{M}_C + \mathcal{M}_N + \mathcal{M}_A$	87.6/**64.2/71.9**	**90.3/67.4/71.6**	89.6/**66.2/73.1**	**90.3/67.1/73.8**

Adding $\mathcal{M}_A$ to the frozen core produces identical P-AUROC and I-AUROC values at all four budgets and changes P-AUPRO only from 63.2% to 63.1% at two shots. The anomaly memory therefore provides little measurable benefit when used without the adaptive normal memory. In contrast, the complete configuration reaches average P-AUROC, P-AUPRO, and I-AUROC values of 89.5%, 66.2%, and 72.6%. Relative to $\mathcal{M}_C$, this corresponds to gains of 4.6 percentage points in P-AUPRO and 3.2 points in I-AUROC, while average P-AUROC remains within 0.1 point.

The budget-wise rows show where this average improvement arises. The complete model has the highest P-AUPRO and I-AUROC at every shot, with P-AUPRO gains over the frozen core of 6.5, 4.2, 2.6, and 5.1 points for 1, 2, 4, and 8 shots. Its P-AUROC is lower by 0.8 and 0.4 points at one and four shots, but higher by 0.4 and 0.3 points at two and eight shots. The joint design therefore primarily improves region-level localization and image-level discrimination rather than uniformly increasing pixel AUROC.

The complete configuration improves region-level localization and image-level discrimination, while $\mathcal{M}_C + \mathcal{M}_A$ closely matches the frozen core. The aggregate results therefore locate the gain in the joint configuration without identifying an independent anomaly-memory contribution. Activation time, occupancy, retrieved-neighbour identity, realized anomaly-term contribution, and candidate purity would distinguish interaction between the adaptive memories from infrequent or contaminated anomaly updates.

Table 9 contains the complete budget-specific results. Table 10 summarizes their arithmetic means. The summary separates the stable pixel-AUROC average from the larger P-AUPRO and I-AUROC gains of the complete configuration. Removing online updates is equivalent to the $\mathcal{M}_C$-only row because both adaptive memories remain empty.

Table 10: Average memory-ablation results over the 1-, 2-, 4-, and 8-shot MVTec AD 2 settings. Δ columns are changes from the frozen-core configuration in percentage points.

Memory configuration	P-AUROC	P-AUPRO	I-AUROC	ΔP-AUPRO	ΔI-AUROC
$\mathcal{M}_C$	89.6	61.6	69.4	0.0	0.0
$\mathcal{M}_C + \mathcal{M}_N$	89.0	62.1	69.7	+0.5	+0.3
$\mathcal{M}_C + \mathcal{M}_A$	89.6	61.6	69.4	0.0	0.0
$\mathcal{M}_C + \mathcal{M}_N + \mathcal{M}_A$	89.5	**66.2**	**72.6**	**+4.6**	**+3.2**

Table 11: Ablation of the illumination augmentation strategy on MVTec AD 2. Results are reported in percent. The last row reproduces the 1-shot Tri-Mem result from Table 1. I-AP and P-AP are not reported in the main table.

Variant	I-AUROC	P-AUROC	P-AUPRO	I-AP	P-AP
w/o highlight injection	**70.76**	87.58	63.98	**80.54**	**18.66**
w/o illumination augmentation	67.65	87.87	63.34	79.16	17.08
w/o specular reflection	67.61	86.58	62.87	80.20	17.85
w/o diffuse blur	67.28	87.72	63.14	79.48	17.60
w/o brightness and RGB color shift	65.95	**88.71**	**64.00**	78.19	16.50
Tri-Mem	71.90	87.60	64.20	–	–

C.4.3 Effect of Illumination Augmentation

Table 11 evaluates the contribution of the individual illumination augmentation operations. All variants use the default 1-shot setting and input resolution of 720×720. The best result among the ablated variants for each metric is highlighted in bold.

Compared with the main-table reference, removing the complete illumination augmentation pipeline reduces I-AUROC from 71.90% to 67.65%. Among the single-operation removals, omitting specular reflection causes the largest decreases in the two pixel-level metrics (1.02 points in P-AUROC and 1.33 points in P-AUPRO), whereas omitting the joint brightness-and-RGB-color-shift operation causes the largest decrease in I-AUROC (5.95 points). Because I-AP and P-AP are unavailable for the complete-model reference, those two columns are compared only among the ablated variants and are not used to infer an operation's contribution relative to the complete model. These metric-specific trends suggest that the illumination augmentation operations affect localization and image-level ranking differently.

C.4.4 Effect of PCA Dimension

Table 12 studies the dimensionality of the PCA-projected patch features under the 1-shot setting. The default implementation uses 256 dimensions. The results reveal a trade-off between compact representation, image-level detection, and pixel-level localization.

The default 256-dimensional main-table reference has the highest reported I-AUROC in this table. Among the non-default ablated variants, the 128-dimensional projection gives the highest I-AUROC and I-AP, 512 dimensions give the highest P-AUROC and P-AP, and 64 dimensions give the highest P-AUPRO. The no-PCA variant has the lowest reported P-AP among these ablations. Because I-AP and P-AP were not reported for the default reference, conclusions for those metrics are restricted to the non-default variants. The implemented $\ell_1 + \ell_2$ metric is not dimension-normalized, so changing C_p changes both representation dimension and relative distance scale. This is an end-to-end dimension-setting ablation. We retain 256 dimensions as the default and use the copied main-table row as the 1-shot reference for the three primary metrics.

C.4.5 Additional Component Ablations

To further examine the main design choices, we conduct additional component ablations on MVTec AD 2 under the default 1-shot setting with an input resolution of 720×720. The complete-model

Table 12: Ablation of the PCA output dimension on MVTec AD 2. Results are reported in percent. Bold denotes the best result among the non-default ablated variants. The separated last row reproduces the default 256-dimensional 1-shot result from Table 1. I-AP and P-AP are unavailable for this reference.

Variant	I-AUROC	P-AUROC	P-AUPRO	I-AP	P-AP
PCA dimension 64	65.52	87.66	**67.06**	79.56	15.69
PCA dimension 128	**68.95**	88.17	63.88	**79.63**	16.97
PCA dimension 512	68.87	**89.39**	64.54	78.52	**17.87**
w/o PCA	67.95	85.02	64.16	79.51	7.95
Tri-Mem(PCA dimension 256)	71.90	87.60	64.20	–	–

result is taken from the corresponding entry in the main comparison table to ensure that the reported reference performance is consistent throughout the paper. We select variants that directly correspond to components described in Section 3 and report the same three primary metrics used in the main experiments.

Table 13: Additional component ablations on MVTec AD 2 under the 1-shot setting at an input resolution of 720×720. P-AUROC, P-AUPRO, and I-AUROC are reported separately in percent. The complete-model row follows the main-table result.

Variant	P-AUROC	P-AUPRO	I-AUROC
Complete Tri-Mem	87.6	64.2	71.9
Feature layers 6 and 8 only	89.5	67.4	69.5
Joint normal-memory compression	87.5	63.9	69.1
Euclidean distance only	88.7	63.4	67.0

In the "Joint normal-memory compression" variant, support-derived and adaptive normal descriptors share the single capacity L_N. Support descriptors may be displaced, and the total normal-memory budget is smaller than the complete model's $N_C + L_N$ bound. Its 2.8-point I-AUROC gap reflects both changes, so this comparison cannot separate their effects. Using only feature layers 6 and 8 increases the two pixel-level metrics but lowers I-AUROC by 2.4 percentage points. Using Euclidean distance alone improves P-AUROC by 1.1 percentage points but reduces P-AUPRO and I-AUROC by 0.8 and 4.9 percentage points, respectively. The combined L1–L2 distance yields the stronger P-AUPRO and I-AUROC values in this ablation.