Bidirectional Multimodal Prompt Learning with Scale-Aware Training for Few-Shot Multi-Class Anomaly Detection

Supplementary Material

1. Comparison to Recent Approaches

To situate AnoPLe within the broader landscape of recent anomaly detection frameworks, we provide a comparative analysis of prompt usage, scalability, and architectural dependencies (Table 1). Earlier prompt-based approaches such as WinCLIP and PromptAD rely heavily on manually designed or class-specific anomaly descriptions, causing their prompt pool to grow proportionally with both the number of classes and the descriptive diversity. PromptAD further amplifies this issue by requiring VV-CLIP [6] forward passes for prompt evaluation, adding additional computational overhead beyond its prompt construction process. IIPAD improves class scalability via instance-induced prompts but still requires textual descriptions and an additional Q-Former module, introducing substantial inference overhead.

In contrast, training-free frameworks such as UniVAD eliminate textual dependencies but achieve this by stacking multiple large visual foundation models—including SAM [3], RAM [11], and Grounding DINO [7]—which significantly increases inference cost and system complexity. INP-Former is lightweight and avoids external modules, but as a vision-only, single-image reconstruction approach, it exhibits limited category-level generalization when scaling to unseen classes in multi-class industrial environments. Our method, AnoPLe, avoids both textual anomaly descriptions and heavy auxiliary models, while maintaining linear scalability with respect to the number of categories and significantly lower inference cost. This design makes AnoPLe particularly suitable for few-shot multi-class industrial anomaly detection, where scalability, efficiency, and robustness to unseen categories are all critical.

2. Multi-Class Confusion Analysis

The few-shot MCAD setting assumes that while the underlying notion of being "intact" is broadly shared across categories, the visual appearance of normal objects differs significantly by class, and abnormality manifests in category-dependent ways. As a result, a unified model must preserve shared normality concepts while maintaining class-aware separation in the abnormal space. To empirically analyze how different prompting frameworks cope with this asymmetry, we visualize class-wise normal and abnormal prototypes using two-dimensional PCA projections. Figure 1-(a) ˜(c) present the embedding structures for PromptAD (multi-class adaptation), IIPAD, and our AnoPLe, respectively.

Figure 1(a) shows that PromptAD produces embeddings in which normal and abnormal prototypes from different classes are intermixed, indicating that the method fails to preserve class-dependent structure in the multi-class setting. This phenomenon arises because PromptAD learns its abnormal prototype from a pooled set of anomaly descriptions; however, such descriptions are inherently category-dependent, and collapsing them into a shared textual pattern limits the model's ability to encode class-specific defect cues when extended to a multi-class scenario.

Figure 1(b) illustrates that IIPAD exhibits a coarse global separation between normal and abnormal prototypes. However, the separation forms two elongated bands, and class-wise structure is not preserved within either region. Although instance-induced prompts provide finer detail than PromptAD, class semantics remain weak: normal prototypes from different categories intermingle within a single band, and abnormal prototypes behave similarly. As a result, the normal–abnormal distance becomes inconsistent across categories, allowing a normal example from class A to lie closer to an abnormal example from class B than to its own class-specific abnormal direction. This entanglement indicates that IIPAD captures a generic abnormality axis but lacks the class-conditioned structure required for reliable multi-class discrimination.

In contrast, Figure 1(c) shows that AnoPLe produces well-organized, class-consistent clusters in which normal and abnormal prototypes form coherent pairs for each category. These pairs remain clearly separated across classes without exhibiting collapse or band-shaped entanglement. This demonstrates that bi-directional multimodal prompting effectively aligns class-level textual anchors with instance-level visual cues, enabling the model to learn category-aware abnormality representations without relying on defect-type descriptions. The resulting representation space preserves both shared normality concepts and stable class-conditioned normal–abnormal boundaries. Overall, these visualizations confirm the necessity of lightweight class semantics in the few-shot MCAD setting and highlight how AnoPLe successfully mitigates multi-class confusion while remaining description-free and scalable.

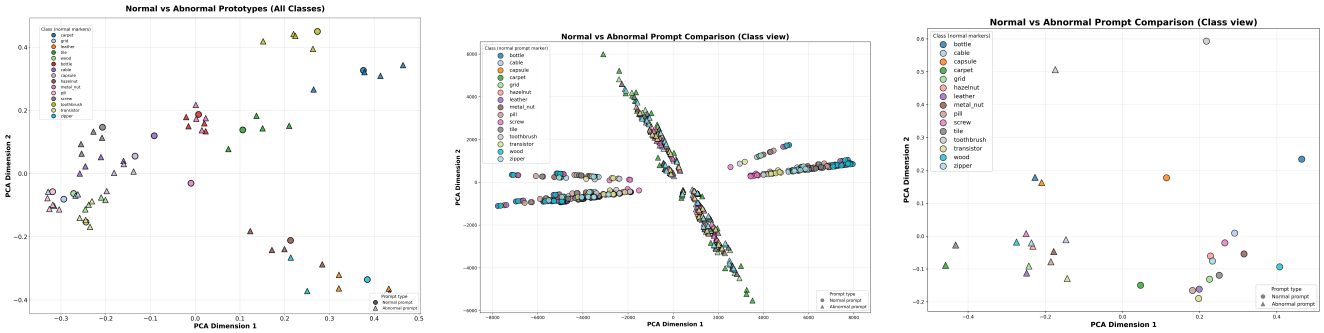
3. Further Implementation Details

3.1. Data Pre-processing

We normalize all images using CLIP's pre-computed mean values $[0.48145466, 0.4578275, 0.40821073]$ and standard

Table 1. **Comparison of prompt-related properties across recent anomaly detection frameworks.** Prompt-based methods increase their prompt pool size with class count and anomaly description diversity, while training-free approaches rely on multiple heavy visual modules. INP-Former provides a lightweight alternative but generalizes poorly to unseen categories due to its single-image, vision-only formulation. AnoPLe avoids textual prompts and external modules while maintaining high multi-class scalability and low inference cost. C, D, T denote the number of categories, the number of anomaly descriptions per category, and the number of textual prompt templates, respectively.

Method	Anomaly Descriptions	Multi-class Scalable	External Module	Inference Cost	Prompt Pool Growth
WinCLIP (CVPR 2023)	△	Moderate	✗	High	$C \times T$
PromptAD (CVPR 2024)	✓	Low	VV-CLIP	Moderate	$C \times (D + T)$
IIPAD (ICLR 2025)	✓	High	Q-Former	High	$C \times (D + T)$
UniVAD (CVPR 2025)	✗	High	SAM, RAM, Grounding DINO	High	N/A
INP-Former (CVPR 2025)	✗	Low	✗	Low	N/A
AnoPLe (Ours)	✗	High	✗	Low	C



(a) PromptAD (multi-class adaptation) (b) IIPAD (c) AnoPLe (ours)

Figure 1. **Comparison of prototype embeddings under the multi-class setting.**

deviations $[0.26862954, 0.26130258, 0.27577711]$. Unless otherwise stated, images are resized to 480×480 during training and are further divided into four non-overlapping crops, while the full image is additionally resized to 240×240 and used as a global view. At inference time, we use only the resized 240×240 images to reduce computational overhead. For Real-IAD, we follow the official setting and use the provided 256×256 resolution without additional cropping. The medical datasets follow the same 480-resolution pipeline, with the exception of Brain MRI where grey mask regions are treated as abnormal targets.

3.2. Training Details

The model is trained for 60 epochs using the SGD optimizer with a batch size of 1, momentum of 0.9, and weight decay of $1\mathrm{e}{-5}$. The learning rates for the prompt learner and decoder are set to 0.001 and 0.0002, respectively, with a linear warm-up from zero to the target learning rate. For the alignment loss (Eq. 10), we apply a temperature-scaled softmax with $\tau = 2$ to normalize $\hat{\mathbf{M}}$. Following OpenCLIP [1], the hidden dimensions of the textual and visual branches are set to 640 and 896. Across all benchmarks, we use a prompt depth of 9. The textual and visual context lengths $(n_{\text{ctx}}, n_{\text{vis}})$ are chosen within a narrow range to match the scale and variability of each benchmark while keeping the architecture identical. Specifically, we use $(3, 3)$ for MVTec-AD, $(5, 8)$ for VisA and Real-IAD, $(4, 5)$ for Brain MRI, $(3, 7)$ for Liver CT, and $(3, 4)$ for Retinal OCT. These values only control the expressive capacity of the learnable prompt and do not introduce any dataset-specific modifications to the model design or optimization.

3.3. Visual Memory Construction

To provide details for how to obtain visual memory assisted anomaly map, we acquire patch features from the i-th layer of E_V and store them in the visual memory $\mathbf{R}$, following previous works [2, 5]. At inference time, patch features $\mathbf{F} \in \mathbb{R}^{h \times w \times d_v}$ from the i-th layer of query images are compared with $\mathbf{R}$. Here, h and w are height and width of the image, respectively. The visual memory assisted anomaly score map $\mathbf{M}_{mem} \in [0, 1]^{h \times w}$ is obtained as $\mathbf{M}_{ij} = \min_{\mathbf{r} \in \mathbf{R}} \frac{1}{2}(1 - \langle \mathbf{F}_{ij}, \mathbf{r} \rangle)$. For implementation, we use l intermediate layers for feature extraction, obtaining $\mathbf{F} \in \mathbb{R}^{h \times w \times l \times d_v}$ and average them across layers, resulting in $\mathbf{F} \in \mathbb{R}^{h \times w \times d_v}$. Specifically, we extract features from the 7th to 10th layers for both datasets. Following WinCLIP and PromptAD , we aggregate prompt-guided and memory-guided anomaly maps with harmonic mean.

3.4. Computational Environments

Our model is trained and evaluated using an NVIDIA A100 80GB GPU and an Intel(R) Xeon(R) Gold 6338 CPU @ 2.00GHz.

4. Ablation Studies

We provide additional ablation results. The results are reported without visual memory to fairly evaluate the impact of each component on AnoPLe. Furthermore, unless otherwise specified, we conduct ablations in the 1-shot setting.

4.1. Prompt Depth

Figure 2 presents the results of an ablation study on prompt depth. To ensure a fair evaluation of the effect of prompt depth on performance, we measure both Image and Pixel AUROC without visual memory. In pixel-level detection, the results show greater robustness to changes in prompt depth compared to image-level detection. However, for image-level tasks, AUROC consistently improves as prompt depth increases. The highest performance is achieved when the prompt depth was set to 9, which is our default setting.

4.2. Analysis on Prompt Lengths

Figure 3 analyzes context length sensitivity.. In MVTec, the model remains robust across different lengths, with optimal performance at a textual and visual context length of 3. In VisA, longer contexts are more effective due to increased image complexity, likely due to the increased complexity of the images. Image-level anomaly detection improves as textual context length increases, while pixel-level performance remains stable. However, as textual context length grows, sensitivity to visual context length decreases, with the best configuration at 5 and 8 for textual and visual context lengths, respectively.

4.3. Textual Prompt Templates

The input textual prompts for AnoPLe are constructed as follows:

$$\mathbf{e}_0^+ = [class], \quad \mathbf{e}_0^- = [abnormal][class]. \tag{1}$$

We measure the impact of words placed in $[class]$ (i.e., class word) and $[abnormal]$ (i.e., state word) in Figure 4 and Table 2, respectively.

4.3.1. Ablation on a Class Word.

Figure 4 presents an ablation study comparing the use of a generic term ("Object") versus class names ("Class") in textual prompts. While one might expect the class-agnostic "Object" to perform better in a multi-class setting, the results indicate that using class names leads to higher Image-AUROC scores, particularly in MVTec-AD. This suggests that explicitly incorporating class names provides beneficial semantic information that enhances anomaly detection.

Our hypothesis is that textual and visual contexts in our framework interact bidirectionally, meaning that textual prompts contribute to refining the representation of visual features. When only the generic term "Object" is used, the text encoder lacks sufficient differentiation between normal and abnormal patterns across different categories, leading to a weaker anomaly representation. In contrast, when class names are included, the textual prompt provides more meaningful guidance, allowing the model to better distinguish normal patterns per category while still learning a unified concept of "abnormality" through the shared state word.

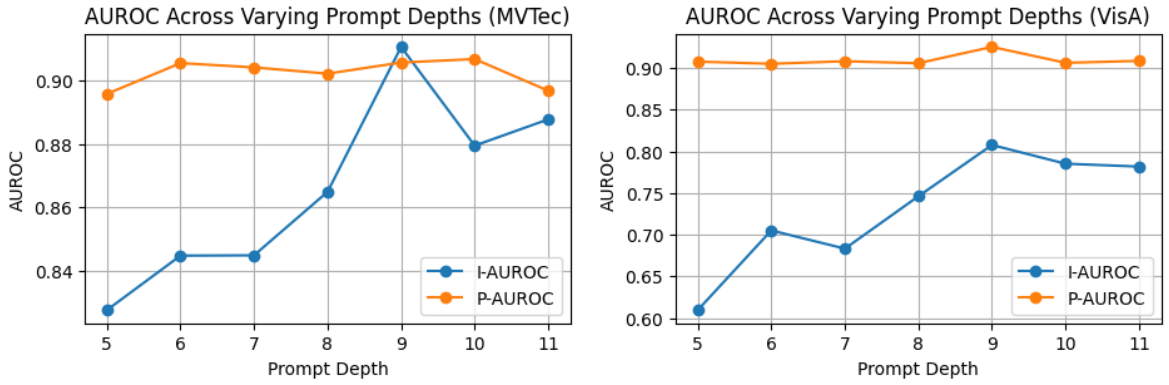
Importantly, this does not contradict our core argument that category-specific anomaly priors hinder multi-class generalization. The critical issue in prior works was the use of class-specific anomaly descriptions (e.g., "cable with bent wire," "screw with manipulated front"), which restricted generalization. In contrast, our approach does not encode category-specific anomaly knowledge but instead allows textual prompts to provide additional context for normal class characteristics, improving the effectiveness of the textual-visual interaction. This result supports our design choice to retain class names in prompts while removing category-dependent anomaly descriptions. By doing so, our model maintains multi-class generalization while still benefiting from the informative nature of class-specific textual context.

4.3.2. Ablation on a State Word.

We compare several state words (e.g., "damaged," "anomalous," "defective") and the Compositional Prompt Ensemble (CPE) proposed in [2] with our chosen state word, "abnormal." In the original paper, CPE combines multiple state-level templates with multiple text templates; however, in this ablation, we only use an ensemble of state-level templates. As shown in Table 2, our simple state word, "abnormal," outperforms or at least competes with other domain-specific options (e.g., "damaged" and "defective"). Although CPE achieves slightly higher AUROC scores at both the image and pixel levels in VisA, it employs a combination of seven domain-specific words for the normal state and four for the abnormal state. In contrast, AnoPLe leverages a single word—"abnormal"—and competes with CPE on VisA, even outperforming it on MVTec-AD.

4.4. Anomaly Simulation

Under constraint that we are not able to utilize any cues on true anomalies, whether textual or visual, we simulate anomalies in the pixel and latent space to achieve robust anomaly detection with only few normal images. Specifically, to generate pixel-level pseudo anomalies, we apply Perlin noise to the raw image following DRAEM [10]. For



(a) **AUROC across varying prompt depths (MVTec-AD)**

(b) **AUROC across varying prompt depths (VisA)**

Figure 2. **Comparison of image and pixel AUROC across different prompt depths for MVTec-AD and VisA datasets.**

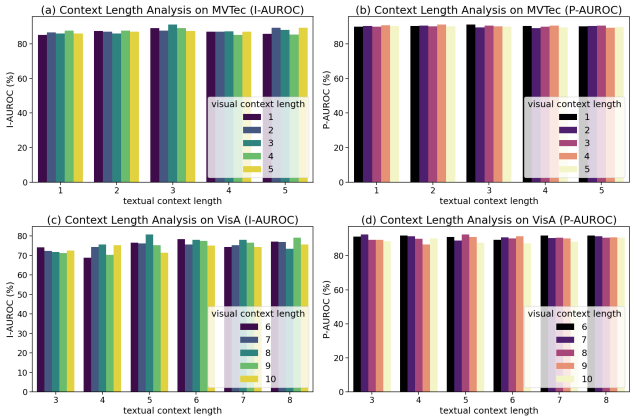


Figure 3. **Comparison Across Different Textual and Visual Context Lengths for MVTec and VisA.**

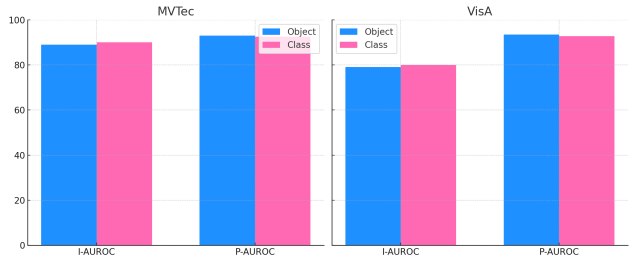


Figure 4. **Ablation on the word used in place of $[class]$ within the textual prompt.** "Object" (colored in blue) uses the word "object" for all classes, while "Class" (colored in orange) represents our default setting, where the specific name of each class is used.

anomaly simulation in the latent space, we follow SimpleNet [8] by adding Gaussian noise to the output of the visual encoder. Table 3 presents the ablation results on the impact of pseudo anomalies at each level on our method. Simulating anomalies in the latent space leads to notably poor performance, indicating that merely adding Gaussian noise to the features of a limited number of normal images fails to provide the model with sufficient signals for anomaly detection. In contrast, using only pixel-level anomalies performs quite well across both datasets. Notably, in MVTec-AD, it slightly outperforms AnoPLe, though the difference is marginal. However, for image-level detection (especially in VisA), the performance remains suboptimal. Overall, the best results are achieved when both levels of pseudo anomalies are used together. While anomaly simulation in the latent space alone is not highly informative, it complements pixel-level pseudo anomalies, resulting in further performance improvements when both are combined.

Table 2. **Ablation results on state-level templates.** CPE refers to the ensemble of multiple state tokens, using state words as described in [2]. The best scores are in bold, and the second-best are underlined.

State Tokens	MVTec-AD		VisA	
	I-AUROC	P-AUROC	I-AUROC	P-AUROC
[damaged]	90.0	<u>90.3</u>	79.7	**93.2**
[anomalous]	88.7	89.9	79.5	<u>92.9</u>
[defective]	<u>90.9</u>	90.0	79.6	92.7
CPE	89.4	89.3	**81.2**	92.8
[abnormal] (Ours)	**91.1**	**90.6**	<u>80.8</u>	92.5

We further examine whether the performance depends on the specific design of pixel-level pseudo anomalies. Replacing the default DRAEM-style synthesis with substantially different strategies (e.g., NSA and CutPaste) yields consistently similar performance (Table 4). This indicates that AnoPLe is not sensitive to the particular form of anomaly synthesis, as long as pseudo anomalies provide a sufficient contrastive signal. Taken together, these results suggest that pseudo anomalies in AnoPLe do not act as semantic teachers that define the appearance of defects. Instead, they serve as a weak contrastive scaffold that introduces the notion of "non-intactness" relative to class-conditioned normality.

Table 3. **Ablation results on pseudo anomaly generation in pixel and latent spaces.** The last row indicates our default configuration. The highest scores are indicated in bold, while the second-highest are underlined.

Pseudo Anomaly		MVTec-AD		VisA	
Pixel Space	Latent Space	I-AUROC	P-AUROC	I-AUROC	P-AUROC
✗	✓	56.7	73.3	59.6	66.4
✓	✗	<u>89.5</u>	**90.8**	<u>72.7</u>	<u>92.0</u>
✓	✓	**91.1**	<u>90.6</u>	**80.8**	**92.5**

Table 4. **Ablation results on pseudo-anomaly generation strategies.** We use DRAEM as the default pseudo-anomaly generation method in AnoPLe and compare it with alternative strategies.

Method	MVTec-AD		VisA	
	I-AUROC	P-AUROC	I-AUROC	P-AUROC
NSA [9]	94.4	95.3	84.6	95.8
CutPaste [4]	94.3	95.1	84.4	95.7
DRAEM (ours) [10]	**94.5**	**96.0**	**86.0**	**97.0**

Table 5. **Performance across different backbone variants.** AnoPLe maintains stable performance across CLIP ViT-B/16+, CLIP ViT-L/14+, and SigLIP-B/16, with only marginal variations across datasets.

Dataset	CLIP ViT-B/16+		CLIP ViT-L/14+		SigLIP-B/16	
	I-AUC	P-AUC	I-AUC	P-AUC	I-AUC	P-AUC
MVTec	94.5	**96.0**	93.2	95.9	**95.0**	94.7
VisA	**86.0**	**97.0**	**86.0**	95.5	85.0	96.4
RealIAD	**81.2**	**96.7**	80.1	94.9	81.1	**96.7**

4.5. Performance across Backbone Variants

Table 5 evaluates AnoPLe with different pretrained vision–language backbones. CLIP ViT-B/16+, our current setting, yields the best overall performance, while replacing it with CLIP ViT-L/14+ or SigLIP-B/16 results in only marginal differences across all datasets. This indicates that AnoPLe is not tightly coupled to a specific backbone, but instead leverages shared latent notions of normality and abnormality that are consistently captured across different pretrained models.

4.6. Decoder

To better understand the contribution of the proposed lightweight decoder, we compare performance with and without it across both MVTec and VisA benchmarks. Incorporating the decoder consistently improves both image-level and pixel-level anomaly detection: on MVTec, I-AUROC increases from 87.1 to 91.1 (+4.0) and P-AUROC from 82.8 to 90.6 (+7.8), while on VisA it yields improvements from 75.9 to 80.8 (+4.9) and from 79.4 to 92.5 (+13.1), respectively. These gains come with only a minor reduction in inference speed (–2.5 FPS on MVTec and –3.2 FPS on VisA), confirming that the decoder enhances spatial localization and overall detection reliability with negligible runtime overhead.

Table 6. **Effect of the decoder on anomaly detection and localization performance.** Numbers in parentheses denote gain/loss compared to the variant without the decoder (gain, loss).

Decoder	MVTec-AD			VisA		
	I-AUROC	P-AUROC	FPS	I-AUROC	P-AUROC	FPS
✗	87.1	82.8	30.4	75.9	79.4	30.1
✓	**91.1** (+4.0)	**90.6** (+7.8)	27.9 (–2.5)	**80.8** (+4.9)	**92.5** (+13.1)	26.9 (–3.2)

Table 7. **Effect of scaling factor s.** Performance is largely insensitive to the choice of scaling factor, suggesting that the coexistence of global and local views is more important than the exact scale.

s	MVTec-AD		VisA	
	I-AUROC	P-AUROC	I-AUROC	P-AUROC
3	**94.7**	95.2	85.6	95.8
4	94.1	95.4	85.3	96.0
2 (base)	94.5	**96.0**	**86.0**	**97.0**

4.7. Scale of local patches

Table 7 analyzes the effect of varying the granularity of local patches. We observe only marginal performance differences across scaling factors, indicating that AnoPLe is not sensitive to patch scale. Instead, the coexistence of global and local views plays a more critical role in capturing both holistic context and fine-grained details. Accordingly, we adopt $s = 2$ as a simple and efficient design choice.

5. Scalability Analysis

5.1. Per-class Evaluation for Unseen Categories

This section provides expanded per-class results corresponding to the unseen-category evaluation described in the main paper. Unlike the aggregate tables in the main text, the per-class tables presented here compare two different training conditions for each category:

- **Train**: performance on each class when the model is trained *with all classes included*, i.e., the held-out class is *not* removed during training. This reflects the fully supervised per-class performance for reference.
- **Held-out**: performance on the same class when it is *excluded from training* and evaluated only during testing, following the leave-one-class-out setting.

Thus, the gap between the two columns indicates how much the model's performance drops when encountering a category it has never seen during training. This differs from the main paper's reported "Train" values, which represent the average performance computed only over the categories included for training under each held-out condition. In this

section, we present the *per-class* supervised performance to allow direct class-wise comparison with the corresponding unseen-class results.

As illustrated in Table 10, AnoPLe generally shows consistently smaller gaps between supervised and unseen performance across most datasets and metrics. This demonstrates that class-name textual anchoring provides a stable semantic prior that helps the model preserve category-aware behavior even when the object type has never been observed during training. Across MVTec and VisA, this trend is pronounced: AnoPLe maintains high supervised accuracy while exhibiting only modest degradation in the held-out condition, clearly outperforming INP-Former both in image-level and pixel-level metrics.

In RealIAD, the pattern becomes slightly more nuanced for image-level AUROC. Owing to the dataset's many visually similar categories with limited semantic separability, the contribution of textual anchors does not manifest as strongly at the global image level, allowing INP-Former's purely visual representation to surpass AnoPLe on a subset of held-out classes. Even so, this difference remains largely confined to image-level metrics. At the pixel level, AnoPLe continues to show consistent advantages, suggesting that its textual grounding still provides reliable and transferable localization cues even when global semantics offer weaker guidance. Overall, the per-class results across datasets collectively support the conclusion that AnoPLe yields stronger unseen-category generalization.

5.2. Full Results on Medical Anomaly Detection

To further validate that AnoPLe is not only effective in its primary target domain of industrial anomaly detection but also serves as a competitive and generalizable methodology in the medical domain, we extend our evaluation on the BMAD benchmark to a broader few-shot setting. While the main paper reports performance under the 4-shot configuration, here we additionally assess 1-shot, 2-shot, and 4-shot settings—as shown in Table 19—to analyze how the number of available normal samples influences both anomaly detection and localization performance .

For each medical dataset—Brain MRI, Liver CT, and Retinal OCT—we report detailed image-level AUROC and pixel-level AUROC metrics. While AnoPLe demonstrates moderate performance in the highly constrained 1-shot and 2-shot cases, its capability improves substantially when more visual cues are provided, yielding strong performance under the 4-shot configuration. Indeed, under the 4-shot setting, AnoPLe consistently ranks first or second across datasets, underscoring its competitive capability even with minimal supervision. These observations collectively suggest that providing additional normal exemplars would further enhance its ability to model class-specific normality and fine-grained anomaly cues, ultimately enabling robust adaptation to visually diverse medical modalities.

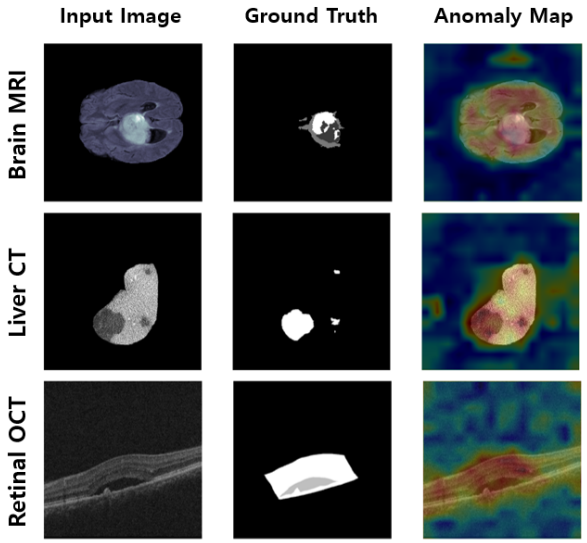


Figure 5. **Qualitative results on the BMAD benchmark across three medical modalities (Brain MRI, Liver CT, and Retinal OCT) under the 4-shot setting.** For each sample, we show the input image, ground-truth mask, and the anomaly map produced by our method.

In addition, we provide 4-shot qualitative results in Figure 5, which further demonstrate AnoPLe's ability to extend beyond industrial scenarios and effectively operate in the medical domain. Across Brain MRI, Liver CT, and Retinal OCT, AnoPLe generates anomaly maps that closely follow the ground-truth lesion regions, successfully capturing subtle and spatially localized pathological patterns. These results highlight that AnoPLe's defect-agnostic prompt formulation generalizes well to fine-grained medical anomaly localization, despite the significant domain shift.

6. Additional Quantitative Results

6.1. Inference Efficiency Analysis

We provide a detailed comparison of inference throughput on MVTec-AD, VisA, and Real-IAD. Table 8 reports the frames-per-second (FPS) achieved by WinCLIP, PromptAD, IIPAD, and AnoPLe under identical hardware and input conditions. As shown, AnoPLe consistently delivers the highest throughput across all three datasets, substantially outperforming existing VLM-based approaches.

This efficiency stems from AnoPLe's design, which avoids the computational overheads prevalent in prior methods. WinCLIP incurs significant latency due to its sliding-window inference strategy, while PromptAD requires additional VV-CLIP forward passes, and IIPAD relies on a Q-Former module that increases per-image processing cost. In contrast, AnoPLe employs a single-pass architecture with lightweight text–image interaction, resulting in markedly

higher throughput. Across datasets, AnoPLe attains 27.9 FPS on MVTec-AD, 26.9 FPS on VisA, and 29.6 FPS on Real-IAD.

The efficiency benefits of AnoPLe extend beyond speed to memory consumption during inference. Under the 1-shot MVTec-AD setting, PromptAD reaches a maximum inference memory of 15.5GB and IIPAD peaks at 24.1GB—both reflecting the heavy auxiliary modules they employ—whereas AnoPLe requires only 4.3GB. The combination of high throughput and low memory footprint makes AnoPLe well suited for real-time or resource-constrained industrial inspection systems.

Table 8. **Inference speed comparison (FPS) on MVTec-AD, VisA, and Real-IAD.** The results highlight AnoPLe's substantial efficiency gains over prior VLM-based anomaly detection methods.

Dataset	WinCLIP	PromptAD	IIPAD	AnoPLe
MVTec-AD	4.6	6.2	15.0	**27.9**
VisA	5.1	7.2	14.8	**26.9**
Real-IAD	10.6	15.5	14.5	**29.6**

Table 9. **Peak inference memory usage (1-shot, MVTec-AD).** AnoPLe exhibits significantly lower memory requirements than prior VLM-based anomaly detection methods.

Method	PromptAD	IIPAD	AnoPLe
Memory (GB)	15.5	24.1	**4.3**

6.2. Category-wise Few-shot Performance

We report category-wise I-AUROC and P-AUROC results for 1-shot, 2-shot, and 4-shot settings across MVTec-AD, VisA, and Real-IAD. These results, summarized in Tables 16, 17, and 18, extend the aggregate metrics presented in the main paper and reveal how shot count influences performance at the granularity of individual object categories.

Overall, increasing the number of normal reference images yields predictable yet non-uniform improvements across categories. Object types characterized by regular textures or low intra-class variation (e.g., *carpet*, *leather*) show rapid saturation, achieving near-optimal performance even in the 1-shot setting. In contrast, categories with more complex structures or greater structural variability (e.g., *cable*, *screw*) benefit more significantly from the additional support images, particularly in I-AUROC. VisA results indicate a similar trend, where texture-centric categories (*chewinggum*, *pipe_fryum*) exhibit strong one-shot performance, while categories involving finer geometric variation (*macaron2*, *pcb1*) improve steadily as shot count increases. Real-IAD, which contains a broader range of real manufacturing parts, displays more diverse behavior. Despite this complexity, P-AUROC remains consistently high across all shots, while I-AUROC gradually strengthens with more support samples, underscoring the benefit of category-specific visual context in few-shot industrial inspection. These tables provide an expanded view of the model's behavior across datasets, demonstrating how different object categories respond to limited supervision and highlighting the robustness of the proposed approach in low-data scenarios.

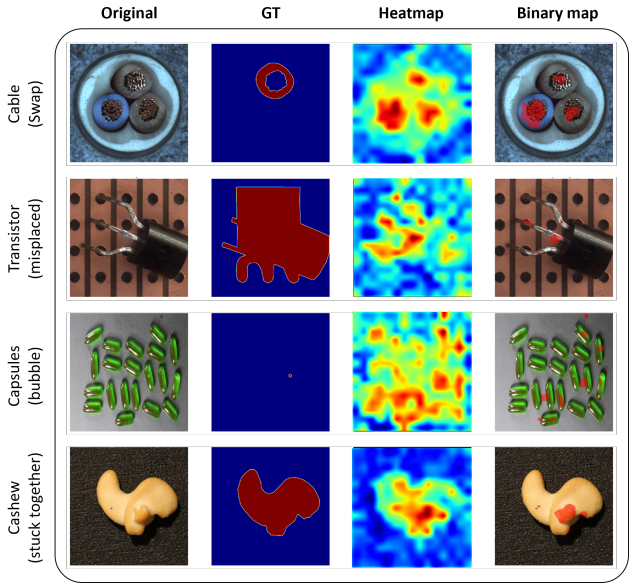


Figure 6. **Failure case on MVTec-AD and VisA.** The top two rows present examples from the MVTec-AD dataset, while the bottom two rows are from the VisA dataset.

7. Additional Qualitative Results

7.1. Visualization of Anomaly Localization

Figure 7 and 8 present segmentation maps under the 1-shot setting across all classes of the MVTec-AD and VisA datasets. The first, second, third, and fourth columns represent the original image, ground truth, heatmap, and the binary map generated by applying a mask to the binarized heatmap, respectively. Although AnoPLe is trained in a 1-shot setting, it demonstrates the ability to accurately segment anomalies in localized areas across all datasets and classes. Additionally, it shows a tendency to avoid falsely identifying anomalies, resulting in a map that closely resembles the ground truth mask.

7.2. Failure Cases

We presents some failure examples in Figure 6 from both MVTec-AD and VisA. The top two rows present examples from the MVTec-AD dataset, while the bottom two rows are

from the VisA dataset. The "Cable" case illustrates a scenario where components have been swapped, and the "Transistor" case represents an anomaly caused by the object being misplaced. In such logical anomaly cases, our method fails to accurately localize the anomaly. This can be attributed to the fact that our approach is designed to generate pseudo anomalies for training by creating anomalies at the image-level and pixel-level through noise. As a result, our method is more suited to structural anomalies, showing vulnerability when faced with logical anomalies, as seen in the provided examples. In particular, in the case of "misplaced" objects, the ground truth mask is configured to localize both the correct and incorrect locations, making the ground truth itself ambiguous and difficult to accurately align with the predictions. Additionally, the "Capsules" example from the VisA dataset illustrates a multi-object image. As seen in the heatmap and binary map, the model not only detects anomalies but also captures multiple objects, highlighting limitations when dealing with such multi-object images. Finally, in the case of "Cashew," the entire object exhibits a different shape. Our results show that the model focuses more on localizing small defects rather than the overall shape.

7.3. Learnable Contexts

AnoPLe encourages active interaction between the two modalities through a bi-directional projection of the learnable context vector, enabling the information in both modalities to complement each other and achieve robust anomaly detection. Our goal is to qualitatively confirm the interaction between the learnable contexts of each modality.

Figure 9 presents the attention matrix corresponding to the positions of the learnable context vectors extracted from AnoPLe for each query image. For MVTec-AD, the first 3 elements in the matrix represent the visual context, while the last 3 represent the textual context. Similarly, for VisA , the first 8 elements correspond to the visual context, and the last 5 correspond to the textual context. Each row of the matrix represents Q in the QKV attention mechanism, while each column represents K. In other words, the rows indicate which parts of the context are being referenced by a given context, and the columns reflect how much a particular context is being attended to by other contexts.

In the early layers (layers 3 and 4), both visual and textual contexts attend to each other in a balanced manner across the matrix. However, in the middle layers (layers 6-8), the columns associated with the textual context show increased activation, suggesting that both contexts are progressively referencing the textual information. This aligns with a common pattern in Transformer architectures, where earlier layers rely more on local, fine-grained information (e.g., visual). In contrast, later layers shift toward more abstract representations, such as textual context. In contrast to the similar pattern observed in the earlier layers across samples, the later layers of the encoder (layers 10 and 11) reveal two distinct trends: either inter-modal interaction is highly active, or intra-modal interaction dominates (as in the "pcb" and "candle"). However, inter-modal interaction is recovered by the final layer (layer 12).

7.4. Scale-Aware Prefix

AnoPLe undergoes multi-scale training by simultaneously learning from sub-images (local view) and full images (global view) during the training phase. However, using sub-images during testing can increase computational load, so only the global view is employed at test time. This difference between the training and testing conditions can introduce a distributional shift. To mitigate this, AnoPLe learns a scale-aware prefix c during training, conditioning the model to distinguish between local and global views, which facilitates multi-scale training. During inference, only the global view is used as the condition.

If training is successful, the global view signal is expected to attend to the entire image, similar to how the [CLS] token functions in image classification by focusing on the object as a whole. In contrast, the local view signal is likely to attend to only small portions of the image, potentially failing to capture meaningful semantics. To investigate this, we visualize the attention maps conditioned on both the global and local view signals (Figure 10). For comparison, the attention map of the [CLS] token is also visualized. We visualize attention maps from layer 3.

In Figure 10, the global view signal generally focuses on the object, or in cases where no distinct object is present, it attends to the entire image. Notably, the [CLS] token in AnoPLe, trained to detect abnormalities, not only focuses on the object but also emphasizes anomalous regions in abnormal images. In normal images, both the global view signal and the [CLS] token behave similarly, attending to the object. However, in abnormal images, the [CLS] token specifically highlights the anomalous areas, while the global view signal—without explicit training for abnormality detection—primarily focuses on the object itself. Furthermore, the local view signal often struggles to capture the object and, in the case of "tile", tends to focus on localized regions rather than the image as a whole.

References

[1] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. 2

[2] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19606–19616, 2023. 2, 3, 4

[3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 1

[4] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9664–9674, 2021. 5

[5] 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*, pages 16838–16848, 2024. 2

[6] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks. *arXiv preprint arXiv:2304.05653*, 2023. 1

[7] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024. 1

[8] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20402–20411, 2023. 4

[9] Hannah M Schlüter, Jeremy Tan, Benjamin Hou, and Bernhard Kainz. Natural synthetic anomalies for self-supervised anomaly detection and localization. In *European Conference on Computer Vision*, pages 474–489. Springer, 2022. 5

[10] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draema discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8330–8339, 2021. 3, 5

[11] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1724–1732, 2024. 1

Table 10. **Per-class I-AUROC results on MVTec-AD.** Columns denote performance when the class is included during training (*Train*), excluded under the leave-one-class-out setting (*Held-out*), and the resulting performance drop between the two (*Gap*).

Class	AnoPLe			INP-Former		
	Train	**Held-out**	**Gap**	**Train**	**Held-out**	**Gap**
bottle	99.3	87.5	11.8	100.0	72.5	27.5
cable	82.0	65.9	16.1	93.8	55.4	38.4
capsule	86.6	76.3	10.3	85.3	47.1	38.2
carpet	99.9	99.1	0.8	100.0	89.2	10.8
grid	99.8	98.0	1.8	99.9	68.2	31.7
hazelnut	97.1	97.9	-0.9	98.1	67.2	30.9
leather	100.0	99.9	0.1	100.0	99.3	0.7
metal_nut	97.1	84.0	13.1	100.0	52.5	47.5
pill	90.0	77.2	12.8	95.7	53.0	42.7
screw	63.9	69.3	-5.4	65.5	52.3	13.2
tile	99.6	97.5	2.1	100.0	84.4	15.6
toothbrush	90.0	90.0	0.0	97.2	68.6	28.6
wood	99.9	99.5	0.4	99.6	80.1	19.5
zipper	90.1	86.9	3.2	97.0	76.0	21.0
mean±std	92.5±9.8	**87.5**±11.4	**5.1**±6.6	**94.7**±9.3	67.7±16.0	27.0±13.3

Table 11. **Per-class P-AUROC results on MVTec-AD.**

Class	AnoPLe			INP-Former		
	Train	**Held-out**	**Gap**	**Train**	**Held-out**	**Gap**
bottle	97.7	80.9	16.8	98.8	84.1	14.8
cable	95.1	81.1	14.0	95.8	64.4	31.4
capsule	95.9	95.3	0.6	97.9	91.5	6.4
carpet	99.4	98.2	1.2	99.2	80.9	18.3
grid	98.5	96.7	1.8	98.4	72.6	25.8
hazelnut	98.2	85.6	12.6	99.1	87.1	12.0
leather	99.4	99.3	0.1	99.3	95.2	4.1
metal_nut	88.2	74.6	13.6	92.9	73.6	19.3
pill	96.1	84.3	11.8	97.5	81.0	16.4
screw	94.8	95.8	-1.0	97.3	91.7	5.6
tile	96.6	96.8	-0.2	96.9	72.7	24.2
toothbrush	98.6	95.5	3.1	98.8	93.9	4.8
wood	95.4	93.9	1.5	95.2	81.4	13.8
zipper	93.0	95.9	-2.9	96.8	84.9	11.9
mean±std	95.9±3.1	**89.5**±9.5	**6.4**±8.0	**96.7**±3.4	81.0±10.5	15.6±8.5

Table 12. **Per-class I-AUROC results on VisA.**

Class	AnoPLe			INP-Former		
	Train	**Held-out**	**Gap**	**Train**	**Held-out**	**Gap**
candle	92.7	74.2	18.5	93.9	63.6	30.3
capsules	77.6	64.7	12.8	87.5	69.5	18.0
cashew	90.2	86.4	3.8	91.6	75.9	15.7
chewinggum	96.8	93.2	3.5	96.7	86.6	10.1
fryum	90.7	81.6	9.1	89.9	56.6	33.3
macaroni1	68.5	65.4	3.1	80.3	54.8	25.5
macaroni2	65.7	48.9	16.8	55.2	53.0	2.3
pcb1	46.7	71.0	-24.3	87.5	26.2	61.3
pcb2	76.0	66.4	9.6	87.9	50.8	37.1
pcb3	79.2	52.7	26.5	81.8	53.0	28.8
pcb4	95.3	76.8	18.5	58.4	45.8	12.6
mean±std	81.4±15.5	**72.8**±14.2	**8.6**±12.7	84.0±13.7	58.7±15.6	25.3±15.4

Table 13. **Per-class P-AUROC results on VisA.**

Class	AnoPLe			INP-Former		
	Train	**Held-out**	**Gap**	**Train**	**Held-out**	**Gap**
candle	97.1	94.4	2.8	99.0	86.8	12.2
capsules	96.0	94.3	1.7	97.0	69.0	28.0
cashew	98.9	90.8	8.1	97.6	96.6	0.9
chewinggum	99.3	98.8	0.5	98.8	97.6	1.2
fryum	93.3	91.9	1.4	95.4	91.5	4.0
macaroni1	90.1	94.6	-4.5	95.1	93.0	2.1
macaroni2	85.4	91.7	-6.3	93.4	91.4	2.0
pcb1	84.7	84.9	-0.1	99.2	86.2	12.9
pcb2	94.2	82.2	12.0	96.7	90.9	5.8
pcb3	97.1	86.5	10.6	96.4	85.0	11.4
pcb4	93.8	93.0	0.8	93.2	90.5	2.7
mean±std	94.1±5.0	**91.7**±4.9	**2.4**±5.5	96.7±2.1	89.7±7.8	7.0±8.0

Table 14. **Per-class I-AUROC results on Real-IAD.**

Class	AnoPLe			INP-Former		
	Train	**Held-out**	**Gap**	**Train**	**Held-out**	**Gap**
audiojack	88.5	58.2	30.3	89.4	66.0	23.4
bottle_cap	88.9	59.0	29.9	93.4	79.7	13.7
button_battery	67.8	61.0	6.8	77.1	57.7	19.5
end_cap	74.2	47.4	26.8	69.4	44.4	25.0
eraser	84.9	57.7	27.2	85.8	87.6	-1.8
fire_hood	87.1	73.9	13.2	90.6	78.2	12.4
mint	55.9	50.5	5.4	65.4	52.3	13.1
mounts	86.9	61.1	25.8	91.1	80.4	10.7
pcb	71.2	59.4	11.7	75.6	64.2	11.4
phone_battery	87.6	39.8	47.9	90.1	54.5	35.7
plastic_nut	73.4	46.2	27.2	81.5	55.0	26.5
plastic_plug	89.0	69.6	19.4	90.9	47.8	43.1
porcelain_doll	74.4	69.6	4.8	85.4	80.3	5.0
regulator	49.2	43.9	5.3	66.7	52.2	14.5
rolled_strip_base	96.8	69.6	27.2	97.2	63.6	33.6
sim_card_set	97.8	53.8	44.1	97.8	61.4	36.4
switch	81.5	48.3	33.2	89.2	61.4	27.8
tape	95.0	68.6	26.5	99.0	63.3	35.8
terminalblock	91.9	61.0	31.0	97.9	50.7	47.2
toy	67.4	52.1	15.4	77.3	43.1	34.2
toy_brick	75.9	56.3	19.6	67.4	52.8	14.6
transistor1	62.4	52.9	9.5	68.3	69.1	-0.8
u_block	85.6	53.9	31.7	83.7	60.4	23.3
usb	89.8	58.2	31.6	95.5	43.8	51.7
usb_adaptor	84.4	57.6	26.8	84.4	64.1	20.4
vcpill	83.9	72.9	11.0	83.4	83.5	-0.1
wooden_beads	81.2	71.2	10.0	80.1	63.0	17.2
woodstick	83.4	57.8	25.7	76.8	62.4	14.4
zipper	96.0	59.2	36.8	98.7	81.7	17.0
mean±std	81.2±11.9	58.5±8.9	22.7±11.5	**84.4**±10.3	**63.1**±12.6	**21.3**±13.8

Table 15. **Per-class P-AUROC results on Real-IAD.**

Class	AnoPLe			INP-Former		
	Train	**Held-out**	**Gap**	**Train**	**Held-out**	**Gap**
audiojack	97.8	94.7	3.0	98.6	94.3	4.2
bottle_cap	97.4	95.1	2.3	99.3	98.1	1.2
button_battery	97.2	96.1	1.1	98.9	94.5	4.4
end_cap	95.1	84.9	10.2	96.3	86.5	9.8
eraser	99.4	99.1	0.4	99.5	99.1	0.4
fire_hood	99.5	97.8	1.7	99.2	97.4	1.8
mint	94.7	94.0	0.8	97.6	80.9	16.7
mounts	99.1	98.7	0.4	99.6	97.1	2.6
pcb	95.5	92.5	3.0	98.3	85.7	12.6
phone_battery	96.0	97.2	-1.1	98.7	88.4	10.2
plastic_nut	95.7	93.0	2.7	99.3	89.3	10.0
plastic_plug	98.6	94.7	3.9	99.3	96.6	2.7
porcelain_doll	93.6	96.6	-3.0	94.1	97.9	-3.8
regulator	87.8	88.2	-0.4	96.9	76.6	20.3
rolled_strip_base	99.1	95.4	3.7	99.7	86.9	12.8
sim_card_set	99.8	98.0	1.8	99.6	94.0	5.6
switch	98.1	73.2	24.9	98.1	80.3	17.9
tape	98.6	98.5	0.2	99.7	97.2	2.5
terminalblock	98.6	93.8	4.8	99.7	89.7	10.0
toy	82.2	72.5	9.7	85.1	75.5	9.6
toy_brick	97.3	95.8	1.5	95.9	90.8	5.1
transistor1	91.2	97.0	-5.8	96.4	94.1	2.2
u_block	99.0	99.0	-0.0	96.7	94.0	2.8
usb	98.9	95.9	3.0	99.5	91.7	7.8
usb_adaptor	97.3	91.4	5.9	97.4	88.7	8.6
vcpill	98.8	97.4	1.3	98.2	96.2	2.0
wooden_beads	98.6	98.0	0.6	97.8	95.7	2.1
woodstick	99.0	94.5	4.5	98.5	88.6	9.9
zipper	98.4	88.5	10.0	99.2	95.7	3.4
mean±std	96.7±3.8	**93.5**±6.5	**3.2**±5.4	97.9±2.8	91.3±6.5	6.6±5.7

Table 16. **Category-wise few-shot results on MVTec-AD.** These results compare image-level and pixel-level AUROC under 1-, 2-, and 4-shot settings, illustrating how additional normal exemplars enhance class-wise few-shot performance.

Class	1-shot		2-shot		4-shot	
	I-AUROC	**P-AUROC**	**I-AUROC**	**P-AUROC**	**I-AUROC**	**P-AUROC**
bottle	99.3	97.7	99.6	98.1	100.0	98.0
cable	82.0	95.1	94.5	96.4	95.3	96.5
capsule	86.6	95.9	69.7	93.3	89.1	97.0
carpet	99.9	99.4	100.0	99.3	100.0	99.3
grid	99.7	98.5	99.8	98.4	100.0	98.7
hazelnut	97.1	98.2	100.0	98.7	100.0	98.6
leather	100.0	99.4	100.0	99.3	100.0	99.2
metal_nut	97.1	88.2	100.0	91.3	100.0	92.2
pill	90.0	96.1	95.0	96.8	96.6	96.8
screw	63.9	94.8	60.9	88.0	65.2	93.9
tile	99.6	96.6	100.0	95.4	100.0	95.5
toothbrush	90.0	98.6	97.5	98.7	100.0	98.8
transistor	92.7	92.0	87.5	89.6	92.0	95.3
wood	99.9	95.5	99.9	96.5	99.7	93.8
zipper	90.1	93.0	94.0	93.6	92.5	92.9
mean±std	92.5±9.8	95.9±3.1	93.2±12.0	95.6±3.6	95.4±9.1	96.4±2.4

Table 17. **Category-wise few-shot results on VisA.**

Class	1-shot		2-shot		4-shot	
	I-AUROC	**P-AUROC**	**I-AUROC**	**P-AUROC**	**I-AUROC**	**P-AUROC**
candle	92.7	97.1	87.5	97.3	96.4	98.2
capsules	77.6	96.0	76.2	96.5	78.9	96.6
cashew	90.2	98.9	91.7	99.2	92.6	99.3
chewinggum	96.8	99.3	97.3	99.2	97.8	99.3
fryum	90.7	93.3	86.2	93.4	93.2	95.4
macaroni1	68.5	90.1	84.7	94.0	87.8	92.4
macaroni2	65.7	85.4	61.6	90.6	57.8	94.3
pcb1	46.7	84.7	91.8	98.1	90.4	98.6
pcb2	76.0	94.2	80.2	94.1	79.0	96.4
pcb3	79.2	97.1	69.8	96.9	82.3	97.6
pcb4	95.3	93.8	95.9	96.1	95.3	95.3
pipe_fryum	97.4	98.7	97.7	98.9	98.9	98.9
mean±std	81.4±15.5	94.1±5.0	85.0±11.3	96.2±2.7	87.5±11.7	96.9±2.2

Table 18. **Category-wise few-shot results on Real-IAD.**

Class	1-shot		2-shot		4-shot	
	I-AUROC	**P-AUROC**	**I-AUROC**	**P-AUROC**	**I-AUROC**	**P-AUROC**
audiojack	88.5	97.7	81.9	98.1	81.4	98.4
bottle_cap	88.9	97.4	86.8	98.0	88.2	97.5
button_battery	67.8	97.2	72.3	97.2	73.6	97.9
end_cap	74.2	95.1	73.6	93.5	71.8	93.1
eraser	84.9	99.4	85.5	99.5	90.7	99.6
fire_hood	87.1	99.5	87.8	99.5	90.0	99.6
mint	55.9	94.7	59.0	96.1	61.7	97.2
mounts	86.9	99.1	85.6	98.9	85.8	98.9
pcb	71.2	95.5	78.5	98.1	79.6	97.6
phone_battery	87.6	96.0	88.9	96.2	87.2	97.1
plastic_nut	73.4	95.7	73.1	97.2	75.1	96.6
plastic_plug	89.0	98.6	89.7	98.9	92.8	98.8
porcelain_doll	74.4	93.6	83.1	97.0	79.1	95.2
regulator	49.2	87.8	51.1	89.1	53.5	90.1
rolled_strip_base	96.8	99.1	97.4	99.3	98.6	99.4
sim_card_set	97.8	99.8	97.8	99.8	97.3	99.8
switch	81.5	98.1	83.0	98.8	83.7	98.7
tape	95.0	98.7	93.8	98.6	94.0	98.6
terminalblock	91.9	98.6	94.1	99.0	88.4	99.0
toothbrush	84.0	98.2	80.8	98.5	82.2	98.5
toy	67.4	82.2	66.8	85.0	69.7	85.1
toy_brick	75.9	97.3	76.6	97.4	69.8	97.8
transistor1	62.4	91.2	76.0	95.7	82.2	95.7
u_block	85.6	98.9	83.2	99.2	85.6	99.6
usb	89.8	98.9	91.5	99.4	92.1	99.5
usb_adaptor	84.4	97.3	86.8	97.7	82.4	96.1
vcpill	83.9	98.7	88.9	98.9	90.8	98.9
wooden_beads	81.2	98.6	83.4	99.0	86.4	99.1
woodstick	83.4	99.0	90.2	99.2	83.8	99.0
mean±std	81.2±11.9	96.7±3.8	82.8±10.9	97.4±3.2	83.2±10.6	97.4±3.1

Table 19. **Comparisons with state-of-the-art medical anomaly detection methods with K=1,2,4.** The best results in bold, and the second-best result is underlined.

Shot Number	Method	BrainMRI		LiverCT		RetinalOCT	
		I-AUROC	P-AUROC	I-AUROC	P-AUROC	I-AUROC	P-AUROC
1-shot	MedCLIP	69.7	91.7	40.5	93.8	66.9	91.5
	PatchCore	70.5	95.4	58.2	95.8	81.1	87.7
	WinCLIP	46.5	85.7	61.4	**97.6**	68.6	92.7
	PromptAD	80.1	96.1	60.4	95.9	**90.1**	96.2
	UniVAD	**80.2**	96.8	**70.0**	96.3	85.5	94.9
	AnoPLe	69.0	**97.0**	62.7	96.3	72.7	**96.9**
2-shot	PatchCore	70.1	95.5	56.2	96.6	65.5	82.6
	WinCLIP	47.8	84.4	63.0	**97.9**	71.8	93.4
	PromptAD	75.1	95.9	63.8	97.0	87.7	95.1
	UniVAD	**82.3**	**96.7**	**69.7**	97.7	86.1	94.9
	AnoPLe	70.4	95.9	67.5	96.4	**91.8**	**97.1**
4-shot	MedCLIP	76.9	90.9	60.7	94.4	66.6	89.6
	PatchCore	73.4	96.3	54.9	96.0	65.3	83.2
	WinCLIP	53.7	86.2	61.9	97.8	75.4	94.4
	PromptAD	73.9	94.3	62.1	96.4	91.2	96.6
	UniVAD	**82.6**	97.0	72.8	**97.9**	88.9	95.3
	AnoPLe	79.8	**97.1**	**74.8**	95.9	**91.4**	**97.0**

	1-shot						2-shot						4-shot					
	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe
bottle	92.86	98.7	49.9	**99.6**	98.6	99.2±0.5	91.43	99.2	45.0	99.7	98.5	**99.8±0.2**	94.37	99.7	37.1	99.6	85.0	**99.9±0.2**
cable	72.06	78.0	84.9	50.9	**89.9**	89.3±4.5	78.75	79.9	48.7	50.7	88.1	**92.1±2.2**	78.05	91.4	48.1	53.1	65.0	**95.3±1.7**
capsule	72.0	64.7	26.1	62.4	77.1	**83.0±10.2**	75.11	64.1	29.2	67.8	70.2	**87.9±3.3**	79.54	85.8	67.4	63.6	59.0	**90.4±2.3**
carpet	93.94	97.8	49.8	96.3	**100.0**	**100.0±0.1**	88.64	98.8	90.8	96.5	**100.0**	99.9±0.1	92.46	97.1	95.3	96.2	**99.0**	**99.9±0.1**
grid	32.58	66.0	68.3	63.2	90.8	**99.7±0.3**	33.92	68.0	69.8	67.6	95.0	**99.5±0.6**	37.01	79.3	66.5	66.6	93.0	**99.5±0.5**
hazelnut	66.0	80.1	45.0	86.5	99.0	**99.2±0.6**	71.82	95.1	57.6	86.9	97.9	**99.7±0.4**	80.79	92.7	61.9	86.5	88.0	**99.7±0.4**
leather	94.7	99.9	36.4	45.4	**100.0**	**100.0±0.0**	89.13	**100.0**	75.2	99.0	**100.0**	**100.0**±0.0	89.57	**100.0**	63.1	98.9	99.0	**100.0±0.0**
metal_nut	55.38	67.8	29.2	**98.7**	95.3	98.4±1.4	57.23	80.0	32.6	50.0	98.2	**99.4±0.4**	56.89	92.0	36.1	54.6	65.0	**99.4±1.2**
pill	63.07	74.2	55.5	84.0	90.5	**94.0±1.2**	73.1	86.0	61.2	83.4	88.3	**94.0±0.8**	75.2	87.5	49.7	84.4	59.0	**94.9±0.6**
screw	42.06	45.2	39.8	70.8	65.2	**72.2±4.0**	43.57	40.1	39.0	70.1	**76.1**	75.7±4.8	50.28	51.0	44.4	69.8	78.0	**78.3±2.6**
tile	90.73	99.2	64.4	65.0	**99.9**	99.4±0.3	94.84	99.7	**100.0**	95.7	99.8	99.7±0.5	93.25	99.5	77.3	96.2	99.0	**99.9±0.1**
toothbrush	72.78	85.3	46.1	**96.1**	91.9	92.8±3.8	66.39	88.3	45.3	70.0	**99.2**	98.1±1.8	63.06	85.6	31.1	66.7	76.0	**98.8±1.8**
transistor	81.83	**94.0**	61.8	57.4	79.0	92.5±2.1	80.17	**96.9**	35.7	58.5	83.8	92.8±2.7	82.46	**97.8**	38.8	56.8	58.0	94.8±1.2
wood	91.84	98.2	93.7	94.0	**99.4**	99.2±0.3	96.05	98.8	96.8	94.3	97.5	**99.1±0.6**	97.89	98.4	98.3	94.6	96.0	**99.0±0.4**
zipper	80.38	92.8	18.7	91.9	91.5	**93.5±1.3**	91.75	94.8	22.2	93.0	93.5	**95.6±0.7**	89.86	95.6	20.6	93.2	85.0	**95.8±1.4**
mean	73.5	82.8	51.3	77.5	91.2	**94.2±0.1**	75.5	86.0	56.6	78.9	92.4	**95.6±0.3**	77.4	90.2	55.7	78.7	80.3	**96.4±0.2**

Table 20. Comparison of category-wise Image-AUROC performance in multi-class setting on MVTec-AD under 1-shot, 2-shot, and 4-shot scenarios. The best performance is in bold, and the second-best performance is underlined.

	1-shot						2-shot						4-shot					
	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe
bottle	97.78	99.6	80.1	**99.9**	99.6	99.8±0.1	97.29	99.7	76.7	**99.9**	99.6	**99.9±0.0**	98.17	99.9	74.3	99.9	**100.0**	**100.0±0.0**
cable	79.36	84.6	**95.5**	64.4	94.6	94.1±2.5	85.59	88.5	67.6	63.5	93.2	**95.5±1.5**	84.58	94.7	68.0	65.0	93.0	**97.4±1.0**
capsule	92.67	85.3	75.5	87.8	93.8	**95.1±4.2**	94.1	84.9	77.9	90.6	88.4	**97.2±0.9**	94.95	96.6	89.2	89.1	96.0	**97.8±0.6**
carpet	98.08	99.4	68.8	98.8	**100.0**	**100.0±0.0**	96.32	99.7	97.3	98.9	**100.0**	**100.0±0.0**	97.67	99.2	98.7	98.9	**100.0**	**100.0±0.0**
grid	64.51	81.5	85.3	82.5	96.8	**99.9±0.1**	63.73	83.6	84.9	86.0	98.3	**99.8±0.2**	66.36	91.3	84.2	83.8	**100.0**	99.8±0.1
hazelnut	76.29	88.7	67.2	93.1	99.4	**99.6±0.3**	83.51	97.4	74.3	93.4	99.0	**99.8±0.2**	89.33	95.5	75.8	93.2	99.0	**99.8±0.2**
leather	98.36	**100.0**	70.4	80.5	**100.0**	**100.0**±0.0	96.41	**100.0**	91.2	99.7	**100.0**	**100.0**±0.0	96.68	**100.0**	86.1	99.6	**100.0**	**100.0**±0.0
metal_nut	86.61	89.7	70.6	**99.6**	98.9	**99.6±0.3**	87.36	92.8	74.0	84.0	99.6	**99.9±0.1**	87.52	98.1	75.7	85.5	99.0	**99.8±0.3**
pill	90.91	93.9	89.1	96.7	98.2	**98.8±0.3**	93.52	97.4	89.8	96.6	97.6	**98.9±0.1**	94.07	97.6	85.6	96.7	98.0	**99.1±0.1**
screw	69.66	69.3	69.0	87.8	83.7	**88.0±3.3**	70.83	67.7	68.5	87.0	89.7	**89.8±3.3**	73.82	73.5	71.2	87.8	87.0	**91.4±1.7**
tile	96.67	99.7	85.6	77.2	**100.0**	99.8±0.1	97.97	99.9	**100.0**	98.5	99.9	99.9±0.2	97.71	99.8	91.9	98.6	**100.0**	**100.0±0.0**
toothbrush	88.72	94.7	74.2	**98.6**	96.8	97.4±1.4	85.23	95.3	70.2	86.4	**99.7**	99.3±0.6	79.14	94.4	64.6	82.4	99.0	**99.6±0.6**
transistor	80.29	**92.0**	64.6	45.5	71.3	90.6±3.5	76.9	**96.8**	37.4	45.7	84.4	90.9±4.1	80.12	**97.2**	40.4	44.1	84.0	93.1±1.5
wood	97.55	99.5	98.2	98.2	**99.8**	**99.8±0.1**	98.88	99.6	99.2	98.2	99.2	**99.7±0.2**	99.4	99.5	99.5	98.3	**100.0**	99.7±0.1
zipper	94.55	97.7	64.4	97.3	97.8	**98.2±0.4**	97.67	98.4	65.9	97.6	98.2	**98.8±0.2**	97.32	98.6	65.4	97.7	**99.0**	98.9±0.4
mean	87.5	91.7	77.2	87.2	95.4	**97.4±0.2**	88.4	93.4	78.3	88.4	96.5	**98.0±0.4**	89.1	95.7	78.0	88.0	96.9	**98.4±0.2**

Table 21. Comparison of Image-AUPR performance for models trained in a multi-class setting on MVTec-AD under 1-shot, 2-shot, and 4-shot scenarios. The best performance is in bold, and the second-best performance is underlined.

	1-shot						2-shot						4-shot					
	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe
bottle	54.13	**97.7**	40.3	91.5	97.0	97.6±0.3	54.11	**98.3**	46.1	91.6	97.6	97.8±0.2	54.11	**98.3**	60.4	91.6	97.4	97.9±0.1
cable	63.55	83.6	90.9	54.0	79.6	**94.4±0.8**	63.55	86.0	70.0	53.7	81.2	**95.7±0.9**	63.55	90.8	59.9	53.8	84.2	**96.1±0.6**
capsule	62.79	**96.8**	89.1	89.5	95.2	96.0±0.9	62.81	**97.1**	89.5	90.1	92.3	96.5±0.6	62.81	**98.2**	97.8	90.1	95.0	96.4±1.0
carpet	63.99	99.0	51.8	98.3	99.3	**99.4±0.1**	63.99	98.9	91.4	98.3	99.3	**99.5±0.1**	63.99	98.9	95.5	98.4	**99.4**	99.3±0.2
grid	56.29	57.4	36.5	83.9	94.8	**98.6±0.0**	55.96	63.7	35.7	85.1	95.9	**98.4±0.2**	56.05	77.9	38.9	84.3	98.2	**98.3±0.2**
hazelnut	82.27	93.3	83.8	97.3	**97.5**	97.0±1.1	82.27	95.4	82.5	97.3	97.8	**97.9±0.4**	82.27	96.1	92.3	97.3	**98.5**	98.3±0.6
leather	57.91	99.2	73.5	54.6	99.3	**99.5±0.0**	57.95	99.3	90.3	98.3	99.2	**99.5±0.1**	57.94	99.3	84.4	98.3	**99.4**	99.4±0.1
metal_nut	50.36	88.8	74.3	**98.2**	88.9	86.2±1.0	50.35	**93.2**	74.5	55.3	90.2	88.7±1.8	50.34	**96.3**	75.7	55.9	90.8	90.5±2.1
pill	62.81	95.0	84.4	94.8	92.3	**95.9±0.3**	62.78	95.8	57.8	94.7	93.1	**96.3±0.2**	62.77	96.2	89.0	95.0	92.4	**96.9±0.2**
screw	58.67	88.0	89.3	93.6	94.1	**94.4±0.6**	58.7	88.9	89.0	93.8	93.7	**95.2±0.6**	58.7	92.1	91.5	93.9	92.5	**95.4±1.8**
tile	63.54	95.5	60.9	93.6	96.3	**97.2±0.3**	63.52	95.7	95.2	89.4	96.4	**97.2±0.4**	63.49	95.4	66.5	89.2	**96.8**	96.6±1.1
toothbrush	54.88	94.5	85.9	88.6	98.4	**98.7±0.2**	54.86	97.6	85.0	91.8	**98.9**	98.7±0.1	54.86	97.5	87.3	93.8	**99.0**	98.8±0.1
transistor	50.3	**91.1**	87.6	69.0	87.7	90.3±1.2	50.3	**93.6**	67.9	69.1	89.2	90.6±2.1	50.3	**93.1**	71.6	69.2	89.2	92.9±1.6
wood	63.14	92.0	79.1	90.3	94.4	**96.2±0.5**	63.44	94.0	79.9	90.3	95.1	**96.2±0.6**	63.44	94.0	82.2	90.4	95.5	**96.1±1.0**
zipper	43.75	**96.2**	79.7	93.3	88.4	94.6±0.9	43.81	**97.4**	76.9	94.1	90.0	95.8±0.8	43.74	**97.8**	80.8	94.7	91.4	95.1±1.3
mean	59.2	91.2	73.8	86.0	93.5	**95.7±0.2**	59.2	93.0	75.4	86.2	94.0	**96.3±0.2**	59.2	94.8	78.3	86.4	94.6	**96.5±0.2**

Table 22. Comparison of Pixel-AUROC performance for models trained in a multi-class setting on MVTec-AD under 1-shot, 2-shot, and 4-shot scenarios. The best performance is in bold, and the second-best performance is underlined.

	1-shot						2-shot						4-shot					
	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe
bottle	19.87	**95.1**	4.8	76.5	92.5	94.7±0.5	19.95	**95.8**	7.2	76.4	93.1	94.8±0.4	19.95	**95.6**	21.2	76.9	92.7	94.3±0.7
cable	23.43	74.7	70.9	36.6	76.4	**84.7±1.9**	23.4	79.0	30.0	35.9	80.2	**86.2±1.6**	23.4	85.8	36.6	37.5	83.0	**88.4±1.4**
capsule	19.38	85.4	70.5	72.3	83.6	**87.9±2.8**	19.39	86.5	72.2	74.1	78.8	**89.3±1.2**	19.39	**91.0**	87.4	73.1	85.7	89.2±2.5
carpet	23.32	95.6	18.7	92.3	97.2	**98.2±0.3**	23.48	96.1	73.3	92.6	97.0	**98.3±0.3**	23.42	94.9	83.0	93.1	97.2	**97.4±1.1**
grid	19.93	19.4	2.9	57.3	86.7	**96.2±0.3**	18.27	21.9	3.1	60.2	89.2	**95.8±0.4**	18.26	44.7	4.0	58.6	94.4	**95.6±0.6**
hazelnut	38.58	78.3	57.5	88.5	**93.0**	90.9±1.4	38.51	84.3	62.1	88.7	**92.9**	90.4±1.0	38.6	83.8	75.8	88.8	**94.6**	91.7±1.1
leather	25.73	97.0	42.1	53.7	97.7	**98.9±0.1**	25.87	97.3	74.9	95.2	96.6	**98.8±0.2**	25.91	97.3	70.8	95.4	97.5	**98.0±0.9**
metal_nut	14.69	69.8	11.3	**95.0**	83.2	86.6±1.2	14.67	79.0	15.1	56.4	86.8	**89.3±0.9**	14.67	88.8	14.7	57.0	86.1	**90.4±1.0**
pill	18.63	92.2	46.9	76.2	89.5	**95.6±0.1**	18.98	93.0	30.4	76.8	90.4	**95.7±0.4**	18.96	94.4	51.7	77.5	90.4	**95.5±0.7**
screw	19.99	62.8	69.1	76.9	77.4	**79.8±1.2**	20.0	64.1	66.9	77.5	78.6	**81.9±1.5**	20.0	72.3	74.1	78.2	73.9	**79.7±5.1**
tile	24.02	91.8	30.1	69.3	91.2	**95.5±0.1**	24.0	92.2	85.5	73.9	91.4	**95.5±0.4**	24.06	91.9	55.4	73.4	92.0	**94.6±1.8**
toothbrush	18.54	86.6	53.4	73.6	87.4	**92.0±2.1**	18.49	85.7	52.1	68.2	89.9	**93.3±1.6**	18.49	84.6	58.8	70.6	91.4	**93.3±1.1**
transistor	12.15	**84.5**	54.2	43.9	69.5	76.4±2.3	12.14	**88.3**	27.1	43.6	69.8	77.5±3.1	12.14	**86.3**	35.3	44.4	69.8	80.6±2.5
wood	18.99	83.5	58.4	71.3	90.0	**93.5±0.9**	18.84	87.2	61.0	71.3	86.9	**92.6±1.3**	18.83	85.8	67.2	71.7	**91.1**	90.7±3.4
zipper	13.12	**90.8**	41.4	78.1	76.2	89.9±1.6	13.12	**92.9**	36.1	79.7	78.8	91.8±1.3	13.05	**93.7**	40.9	82.9	81.1	90.1±2.7
mean	20.7	80.5	42.1	70.8	86.1	**90.7±0.2**	20.6	82.9	46.5	71.4	86.7	**91.4±0.2**	20.6	86.1	51.8	71.9	88.1	**91.3±1.1**

Table 23. Comparison of Pixel-PRO performance for models trained in a multi-class setting on MVTec-AD under 1-shot, 2-shot, and 4-shot scenarios. The best performance is in bold, and the second-best performance is underlined.

	1-shot						2-shot						4-shot					
	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe
candle	77.91	84.9	27.9	86.6	87.4	**91.4±2.3**	65.23	91.2	47.8	86.7	89.7	**93.7±3.1**	56.29	90.4	31.7	86.1	91.8	**94.5±0.8**
capsules	45.23	62.7	58.0	55.7	74.9	**78.4±2.1**	62.82	59.7	61.9	62.9	71.4	**76.7±5.9**	62.87	54.5	61.1	63.3	75.2	**79.5±3.3**
cashew	80.26	90.8	63.6	72.4	80.0	**93.3±1.7**	77.32	90.2	50.0	71.6	81.7	**93.1±1.3**	85.6	92.2	61.4	69.6	85.6	**93.9±0.8**
chewinggum	88.12	96.5	43.5	84.3	**97.0**	96.8±0.5	82.54	96.3	59.4	86.0	93.8	**96.6±0.7**	91.26	**98.4**	51.9	85.5	96.6	96.4±0.5
fryum	63.16	78.7	40.8	66.9	81.2	**87.3±3.5**	76.0	82.9	73.6	64.6	82.3	**89.9±0.7**	70.4	89.3	41.3	66.3	88.1	**90.1±1.9**
macaroni1	44.15	61.9	40.1	57.9	**83.0**	80.6±3.4	47.74	65.7	29.6	56.4	79.1	**83.7±1.3**	57.26	81.0	31.3	56.6	83.6	**84.1±1.4**
macaroni2	49.32	46.2	37.9	58.0	63.2	**64.2±3.1**	38.49	57.2	39.9	57.0	**73.3**	63.9±2.2	45.77	60.7	40.2	57.2	**76.4**	65.3±1.7
pcb1	90.13	80.8	79.4	72.4	88.3	**91.0±1.4**	42.42	79.1	22.9	73.1	62.4	**92.7±1.3**	79.19	83.7	55.3	75.9	87.2	**91.5±2.2**
pcb2	72.08	75.1	40.8	60.0	**80.7**	77.2±2.1	74.46	**80.9**	47.7	57.3	76.6	79.8±2.1	74.05	**84.1**	70.9	58.3	79.8	82.1±1.4
pcb3	69.84	**80.8**	31.2	56.9	74.5	77.6±2.4	67.89	**84.2**	37.5	57.0	80.6	78.6±2.3	64.07	**86.9**	51.6	57.0	79.0	80.6±2.9
pcb4	45.79	92.1	36.8	76.2	80.4	**93.4±1.0**	66.35	**98.2**	34.6	78.8	88.6	92.3±4.4	73.94	**98.7**	51.1	79.1	90.9	93.9±2.3
pipe_fryum	67.48	93.6	53.0	92.8	97.6	**98.1±0.7**	69.9	91.1	49.0	92.5	96.9	**98.1±0.9**	73.18	97.4	62.7	91.5	96.6	**98.6±0.3**
mean	66.1	78.7	46.1	70.0	82.4	**85.8±0.7**	64.3	81.4	46.2	70.3	81.4	**86.6±0.2**	69.5	84.8	50.9	70.5	85.9	**87.5±0.7**

Table 24. Comparison of Image-AUROC performance for models trained in a multi-class setting on VisA under 1-shot, 2-shot, and 4-shot scenarios. The best performance is in bold, and the second-best performance is underlined.

	1-shot						2-shot						4-shot					
	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe
candle	73.45	87.9	40.9	86.3	89.5	**92.9±1.4**	60.92	92.2	52.3	86.2	91.9	**94.7±2.3**	51.83	89.3	41.8	85.6	93.1	**95.1±0.9**
capsules	61.42	72.8	71.3	68.2	86.0	**87.8±1.0**	70.1	71.9	73.3	74.3	84.2	**87.0±3.1**	69.68	70.9	73.0	74.3	86.1	**88.3±1.7**
cashew	88.63	95.7	75.7	84.1	91.3	**97.0±0.7**	87.16	95.0	66.7	84.0	91.7	**96.8±0.9**	91.92	96.1	74.0	82.3	93.6	**97.3±0.4**
chewinggum	94.27	98.5	66.7	92.3	98.6	**98.7±0.2**	91.24	**98.6**	75.4	93.1	97.3	98.6±0.3	95.84	**99.3**	73.7	92.9	98.5	98.5±0.2
fryum	80.51	89.5	64.2	76.6	90.7	**94.1±1.4**	87.87	91.6	86.9	75.0	92.0	**95.4±0.3**	83.69	94.7	65.0	76.0	94.0	**95.4±0.8**
macaroni1	44.55	63.3	44.0	58.8	**86.3**	82.0±2.4	47.56	66.7	40.0	58.1	83.5	**85.8±0.3**	56.89	81.5	40.4	58.0	**86.9**	85.5±0.8
macaroni2	50.3	47.1	43.1	56.2	**67.5**	63.2±4.9	42.1	53.2	43.7	55.5	**78.7**	62.5±2.5	48.56	62.0	44.2	55.6	**79.2**	65.9±4.3
pcb1	87.42	75.8	75.3	69.3	87.7	**88.6±2.1**	53.87	73.0	35.9	69.5	67.8	**92.0±1.4**	79.83	78.7	50.6	71.4	85.4	**90.2±2.6**
pcb2	75.55	76.2	48.2	57.1	**80.3**	74.5±2.8	78.14	**81.4**	50.7	56.0	77.6	78.3±1.7	77.81	**86.4**	77.3	55.3	81.8	80.6±2.2
pcb3	67.99	**81.7**	38.8	54.8	77.4	78.0±4.1	68.42	**85.6**	41.5	54.5	82.3	79.6±2.4	69.37	**88.0**	51.4	54.7	81.6	80.9±3.0
pcb4	45.31	91.3	42.1	76.4	77.6	**91.7±1.2**	61.9	**98.0**	40.8	79.0	85.4	91.1±4.2	68.55	**98.8**	50.0	78.9	90.3	92.8±1.9
pipe_fryum	81.12	96.9	71.3	96.5	99.0	**99.2±0.4**	82.83	96.0	67.2	96.4	98.7	**99.1±0.5**	85.57	98.8	79.1	95.9	98.5	**99.4±0.1**
mean	70.9	81.4	56.8	73.0	86.0	**87.3±0.3**	69.3	83.6	56.2	73.5	85.9	**88.4±0.4**	73.3	87.0	60.0	73.4	89.1	**89.2±0.4**

Table 25. Comparison of Image-AUPR performance for models trained in a multi-class setting on VisA under 1-shot, 2-shot, and 4-shot scenarios. The best performance is in bold, and the second-best performance is underlined.

	1-shot						2-shot						4-shot					
	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe
candle	48.32	95.9	69.1	86.2	92.5	**97.7±0.3**	48.31	**97.7**	71.0	86.1	94.6	97.5±0.1	48.31	**98.6**	65.4	86.4	93.3	98.2±0.2
capsules	44.83	92.4	56.5	63.2	94.1	**97.2±0.7**	44.83	95.0	56.2	65.8	94.3	**98.0±0.4**	44.83	97.2	59.4	66.6	94.8	**97.7±0.4**
cashew	76.52	93.9	89.5	89.9	97.1	**99.2±0.0**	76.51	97.1	87.8	89.8	97.2	**99.3±0.1**	76.51	97.3	89.0	89.9	97.5	**99.2±0.2**
chewinggum	72.0	99.0	93.5	97.2	98.5	**99.4±0.0**	72.0	99.1	91.5	97.4	97.8	**99.3±0.1**	72.0	99.0	95.4	97.4	97.9	**99.2±0.1**
fryum	73.3	84.6	90.3	88.2	93.3	**95.6±0.8**	73.29	87.8	95.3	88.4	92.8	**96.1±0.7**	73.29	88.3	89.9	88.5	94.1	**96.3±0.4**
macaroni1	59.32	95.8	86.4	87.0	**96.7**	94.4±1.5	59.32	**98.0**	87.4	87.2	96.5	95.2±1.0	59.32	**99.3**	84.7	87.6	96.7	95.2±0.7
macaroni2	45.77	89.3	83.5	81.4	**93.7**	93.2±1.6	45.77	93.3	82.5	81.2	**93.9**	93.0±1.4	45.77	95.7	82.8	81.5	**96.0**	93.7±1.5
pcb1	49.65	96.3	**98.5**	71.3	95.1	97.6±0.7	49.65	96.8	79.7	72.2	92.3	**97.6±0.6**	49.64	96.9	90.8	73.4	94.8	**98.6±0.7**
pcb2	60.98	94.3	84.9	79.4	93.7	**96.0±0.5**	60.99	95.7	82.8	80.7	91.8	**96.9±0.5**	60.99	96.3	95.5	80.6	92.7	**96.4±0.5**
pcb3	52.5	**98.2**	83.8	82.4	94.9	96.6±0.7	52.57	**98.6**	83.4	83.1	96.3	96.5±0.5	52.57	**98.9**	82.1	83.6	96.8	97.7±0.2
pcb4	64.84	92.3	83.2	94.6	92.4	**96.1±0.6**	64.84	94.3	76.7	94.8	94.5	**96.7±0.6**	64.84	96.0	86.2	94.8	96.3	**97.3±0.2**
pipe_fryum	88.11	95.5	96.0	97.3	**98.8**	98.7±0.3	88.11	98.2	95.0	97.3	**98.9**	98.8±0.2	88.11	98.7	97.4	97.2	**98.9**	98.8±0.2
mean	61.3	94.0	84.6	84.8	95.1	**96.8±0.2**	61.3	96.0	82.4	85.3	95.1	**97.1±0.2**	61.3	96.9	84.9	85.6	95.8	**97.4±0.1**

Table 26. Comparison of Pixel-AUROC performance for models trained in a multi-class setting on VisA under 1-shot, 2-shot, and 4-shot scenarios. The best performance is in bold, and the second-best performance is underlined.

	1-shot						2-shot						4-shot					
	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe	SPADE	PatchCore	FastRecon	WinCLIP	PromptAD	AnoPLe
candle	10.76	**89.6**	43.2	87.0	84.3	89.5±0.6	10.75	**92.4**	47.7	87.3	86.71	89.9±1.0	10.73	**95.3**	36.2	87.6	86.2	91.6±1.1
capsules	19.54	61.3	19.0	22.7	60.4	**79.1±4.3**	19.52	68.6	17.8	25.8	57.91	**82.0±2.0**	19.52	72.2	19.0	24.4	61.8	**81.8±2.9**
cashew	43.98	77.8	74.6	77.9	77.7	**89.1±2.4**	43.96	**89.4**	69.4	78.0	79.77	87.3±2.8	43.95	**93.1**	73.4	78.0	79.2	88.4±1.1
chewinggum	26.94	85.4	70.9	72.5	79.5	**85.5±2.3**	26.88	83.3	58.4	73.8	76.06	**83.7±1.2**	26.88	**85.3**	77.4	73.6	73.5	82.0±1.9
fryum	39.45	81.4	67.8	77.7	77.6	**90.5±1.3**	39.42	83.2	83.1	78.1	73.78	**90.6±0.8**	39.4	90.2	65.2	78.2	73.5	**90.5±1.8**
macaroni1	26.88	**87.9**	68.0	50.2	84.5	87.0±2.0	26.88	**94.3**	70.3	50.6	85.5	87.3±2.1	26.88	**97.9**	65.4	50.8	88.7	87.2±0.8
macaroni2	13.9	66.6	58.8	36.8	79.8	**85.4±2.9**	13.88	78.5	56.3	36.4	75.96	**85.0±2.1**	13.88	86.1	58.7	36.6	83.5	**86.5±2.2**
pcb1	32.55	76.3	84.2	28.2	86.2	**95.3±0.7**	32.57	79.1	46.1	28.3	59.65	**95.4±0.5**	32.57	79.4	38.2	29.6	72.2	**94.9±0.8**
pcb2	33.66	83.6	53.2	45.4	69.5	**84.7±1.9**	33.67	85.8	53.4	47.9	62.04	**87.5±0.8**	33.67	**88.4**	80.8	47.0	65.7	87.3±1.7
pcb3	18.83	**84.2**	56.4	62.7	75.6	82.5±2.4	18.83	**87.8**	65.2	63.5	81.23	84.9±1.1	18.83	**89.2**	43.9	64.0	80.9	84.2±3.0
pcb4	29.3	65.6	49.5	79.4	64.6	**83.7±3.9**	29.27	69.6	39.3	80.0	71.32	**85.9±1.9**	29.27	77.0	61.4	80.0	79.3	**87.5±1.4**
pipe_fryum	51.61	94.4	84.7	94.3	93.4	**97.4±0.4**	51.62	95.2	80.4	94.5	94.73	**97.6±0.3**	51.62	96.6	86.7	94.3	93.4	**97.6±0.3**
mean	29.0	79.5	60.9	61.2	77.8	**87.5±1.6**	28.9	83.9	57.3	62.0	75.4	**88.1±0.6**	28.9	87.6	58.9	62.0	78.2	**88.3±0.7**

Table 27. Comparison of Pixel-PRO performance for models trained in a multi-class setting on VisA under 1-shot, 2-shot, and 4-shot scenarios. The best performance is in bold, and the second-best performance is underlined.

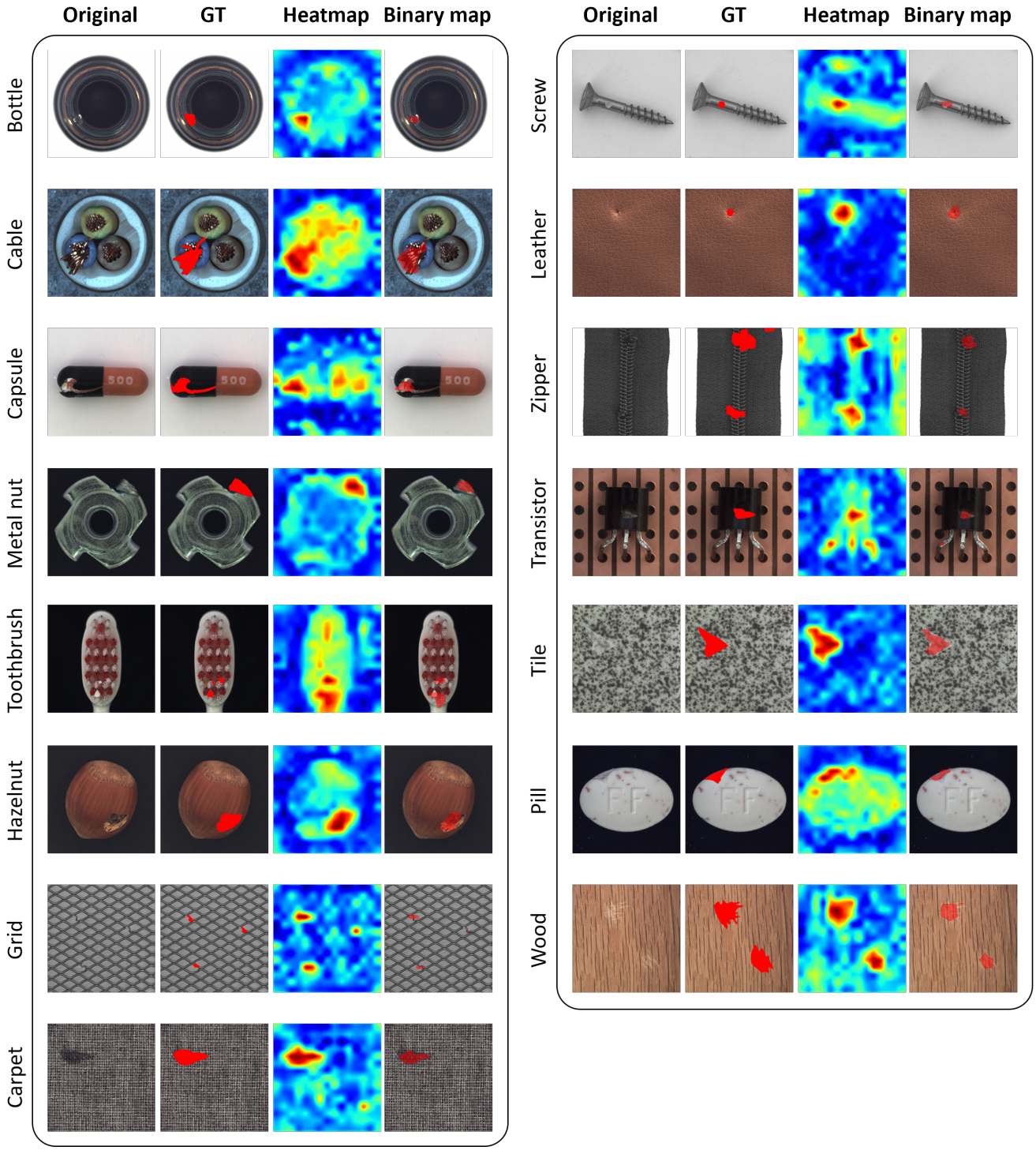


Figure 7. **Detailed qualitative results on MVTec-AD from 1-shot AnoPLe.**

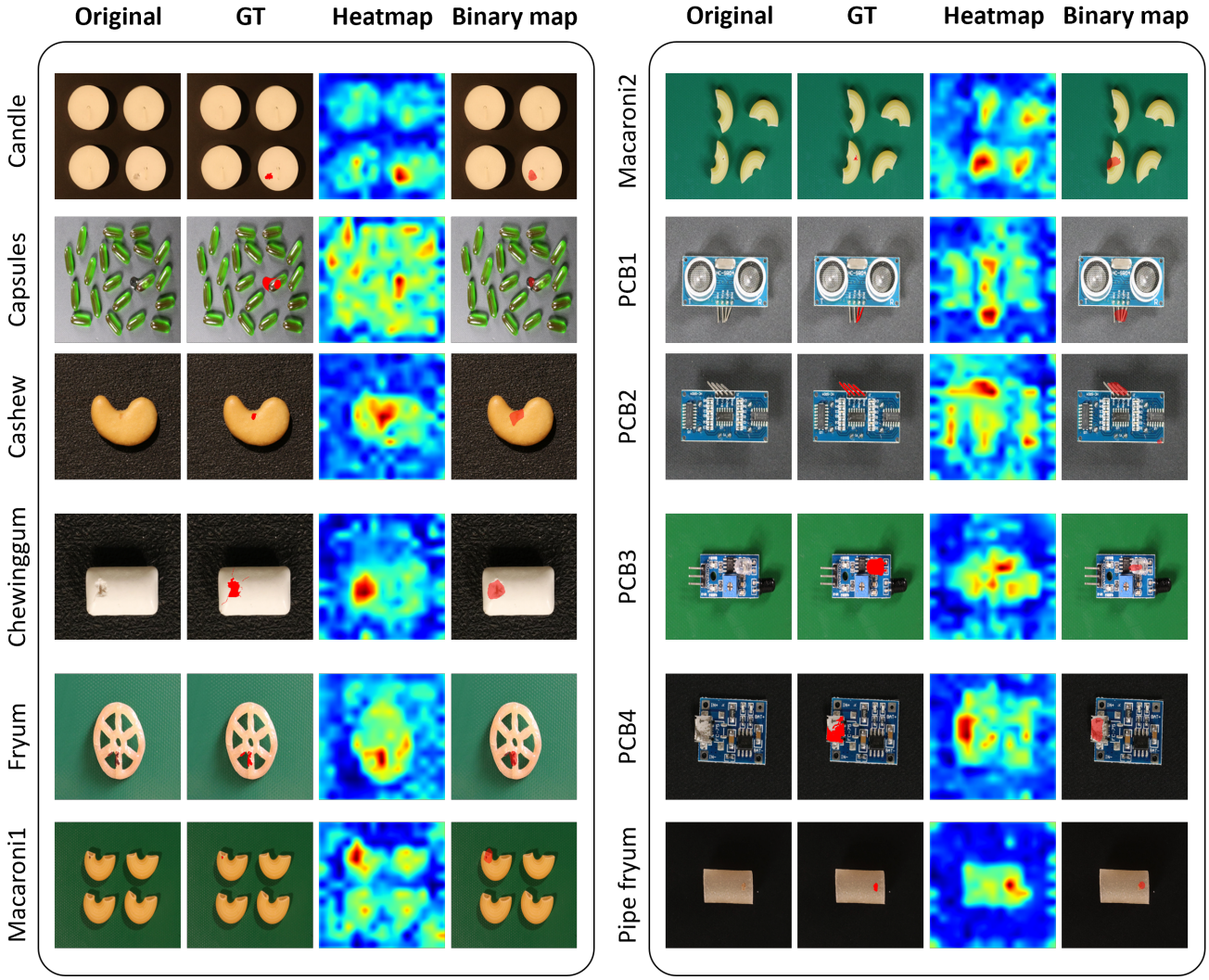


Figure 8. **Detailed qualitative results on VisA from 1-shot AnoPLe.**

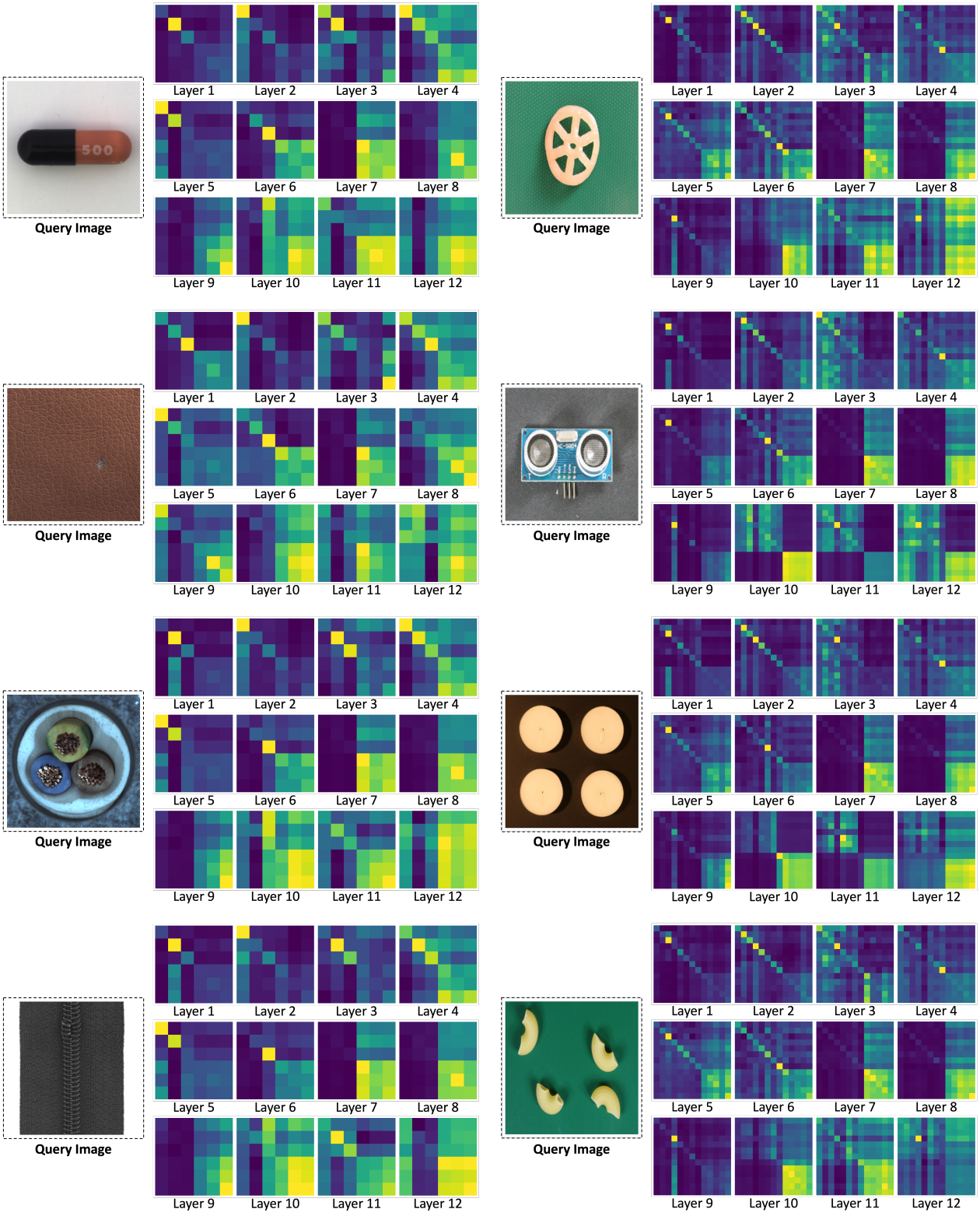


Figure 9. **Attention map visualization of learnable contexts.** The results of the MVTec-AD dataset are presented on the left side, while the results of the VisA dataset are displayed on the right side.

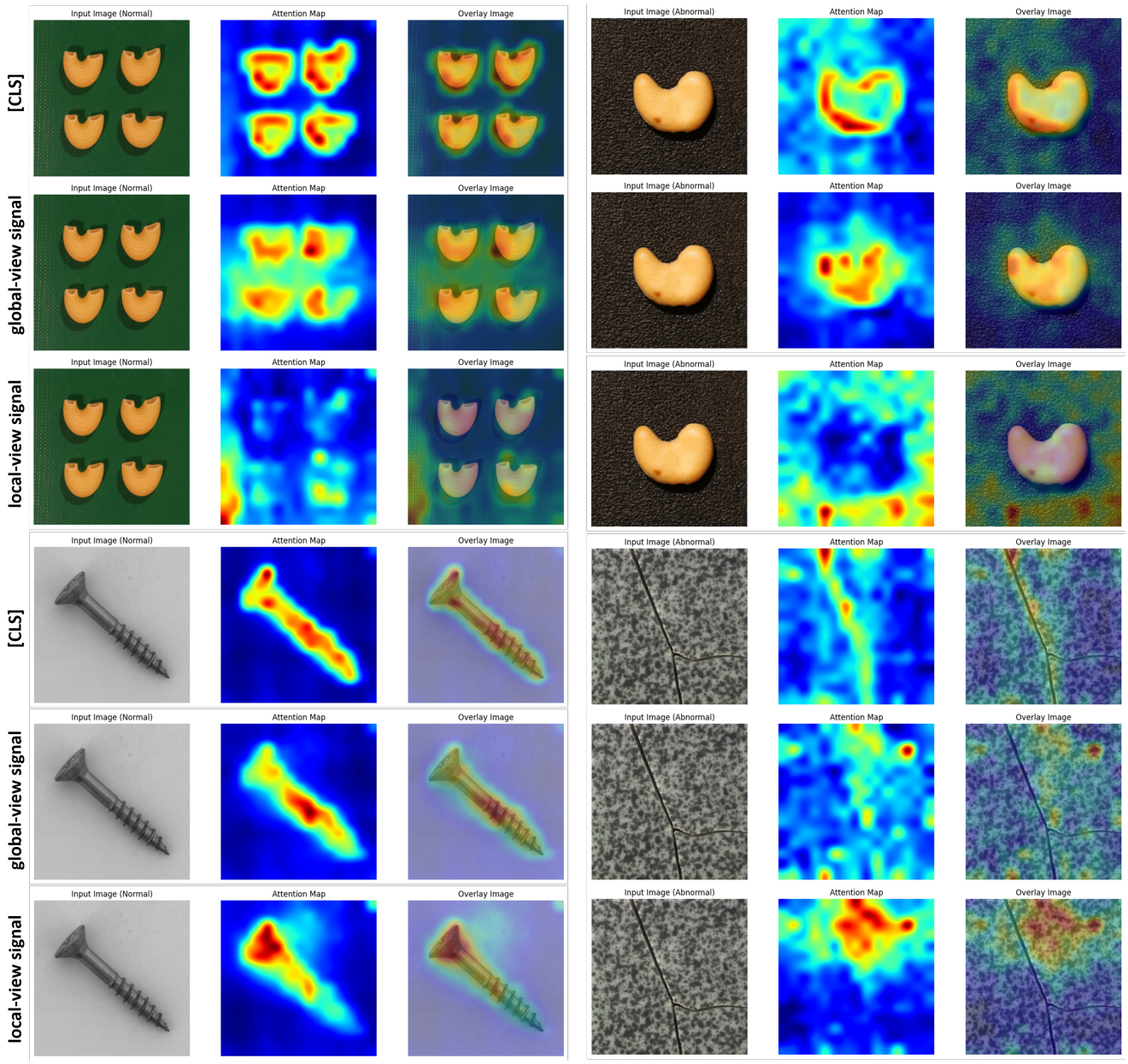


Figure 10. **Comparison of visualized attention maps for scale-aware prefix.** The first 3 rows present results from the VisA dataset, while the last 3 rows showcase results from the MVTec-AD dataset. The first 3 columns display normal images, and the last 3 columns present abnormal images. Columns 1 and 4 show the original images, columns 2 and 5 display the attention maps, and columns 3 and 6 present the attention overlay images.