VisualAD: Language-Free Zero-Shot Anomaly Detection via Vision Transformer

Supplementary Material

Appendix

This appendix includes the following five parts: 1) Implementation details of our VisualAD and the introduction of the state-of-the-art approaches in Section A; 2) Additional experimental results, including ablation studies, and further analysis in Section B; 3) Introduction to 13 industrial and medical datasets in Section C; 4) Presentation of more detailed quantitative and qualitative results in Section D; 5) Limitations of our method in Section E.

A. Implementation Details and SOTA Methods

A.1. Implementation Details

Details of the model architecture. By default, VisualAD employs the CLIP-pretrained [43] ViT-L/14@336 as the backbone, which contains 24 transformer layers. Following prior arts [58], patch tokens extracted at layers 6, 12, 18, 24 are used as equidistant multi-scale representations for fine-grained anomaly localization. Input images are uniformly resized to 518×518 before being fed into the encoder. To inject task-specific anomaly modeling, two learnable special tokens—anomaly token and normal token—are explicitly inserted into the patch sequence; both tokens together with their positional embeddings are updated during training, while the remaining pretrained parameters are kept frozen. At each selected layer, the frozen patch features first pass through the SCA module: a small set of learnable anchor queries dynamically aggregate localized spatial evidence via cross-attention, and a token-guided gating mechanism injects fine-grained positional cues into the anomaly/normal tokens, yielding joint semantic–spatial enhancement. Subsequently, the patch features enter the SAF, a single-layer MLP that nonlinearly recalibrates the feature distribution before token–patch matching, accentuating anomaly contrast. The enhanced tokens and recalibrated patches are then compared via cosine-similarity contrast to produce a per-layer anomaly map;multi-layer maps are summed, and the image-level score is obtained by averaging the top 1% (industrial) or 5% (medical) most anomalous pixels. Entirely eliminating the text branch, this pipeline achieves state-of-the-art cross-domain zero-shot detection solely through visual token learning, SCA spatial refinement, and SAF feature self-alignment.

Details of token enhancement and patch recalibration. Instead of directly comparing frozen patch tokens with the global anomaly/normal tokens, we perform two lightweight, learnable steps on-the-fly. First, every selected layer feeds its patch feature map $\mathbf{P}_\ell \in \mathbb{R}^{B\times N\times d}$ into the Spatial-Aware Cross-Attention (SCA) module. SCA generates $m \ll N$ learnable anchor queries, computes cross-attention between these anchors and the patches, and produces anchor-summary features $\mathbf{U}_\ell \in \mathbb{R}^{m\times d}$. A token-guided gating vector $\mathbf{g}(\mathbf{t})$ (shared for $\mathbf{t}_a$ and $\mathbf{t}_n$) re-weights the m anchors and injects the spatial mixture back into the global tokens, yielding enhanced representations $\tilde{\mathbf{t}}_a^{(\ell)}$ and $\tilde{\mathbf{t}}_n^{(\ell)}$ that are spatially grounded for the current image. Second, the identical patch map $\mathbf{P}_\ell$ is passed through the Self-Alignment Function (SAF), implemented as a MLP F_ℓ with residual connection, producing recalibrated patch features $\hat{\mathbf{P}}_\ell$ whose distribution is implicitly aligned with the evolving tokens.

Details of training and inference. We conduct zero-shot anomaly-detection experiments on six industrial and seven medical benchmarks. VisA [60] serves as the auxiliary training set; after convergence, its parameters are frozen and directly transferred to the remaining target sets. When VisA itself is the target, we swap the auxiliary role to MVTec [6]. During auxiliary training, we freeze the entire CLIP image encoder and update only the two inserted tokens, their positional embeddings, the SCA anchor queries / gating weights, and the per-layer SAF MLPs; consequently, a single forward pass is sufficient to compute the gradient and no memory-intensive ensemble is required.

At inference, the identical lightweight path is executed once per image: SCA first injects sample-specific spatial evidence into the anomaly/normal tokens, SAF recalibrates the patch features, and the cosine-difference anomaly map is produced in one shot.

Details of hyperparameters. Unless otherwise specified, all experiments adopt the hyperparameter configuration reported in Table 6. For both ViT-L/14@336px and DINOv2-ViT-L/14 backbones, input images are uniformly resized to 518×518, and features with dimensionality 1024 are extracted from four intermediate layers $\{6, 12, 18, 24\}$. The trainable parameters of VisualAD are optimized using AdamW with a learning rate of 1×10^{-3}, a batch size of 8, and 4 anchor queries per SCA module. The number of epochs is reported in the form "VisA / MVTec-AD" (for example, $1/2$ for ViT-L/14@336px and $4/2$ for DINOv2-ViT-L/14), reflecting the slightly longer training schedule required by DINOv2 under the same setting. The number of patches differs across backbones due to their native tokenization strategies (24×24 for ViT-L/14@336px and 37×37 for DINOv2-ViT-L/14), and a dropout rate of 0.1 together with a residual scale initialization of 0.01 is used to stabilise optimization.

Details of the loss function. To enable the VisualAD

Table 6. The table summarizes the key hyperparameter configurations used in experiments with ViT-L/14@336px and DINOv2-ViT-L/14 backbones, including input image size, feature dimensionality, selected network layers, optimizer settings, training epochs (where the former corresponds to training on VisA and the latter on MVTec-AD), batch size, and the number of image patches.

Hyperparameters	ViT-L/14@336px	DINOv2_ViTL14
Image Size	518	518
Feature Dimension	1024	1024
Feature Layers	[6, 12, 18, 24]	[6, 12, 18, 24]
Learning rate	1e-3	1e-3
Optimizer	AdamW	AdamW
Number of Anchors	4	4
Epochs	1/2	4/2
Batch size	8	8
Number of Patches	576 (24×24)	1369 (37×37)
Dropout	0.1	0.1
Residual Scale Init	0.01	0.01

framework to learn highly discriminative visual prototypes and achieve precise anomaly detection and localization in a zero-shot setting, we propose a unified hybrid supervised training objective $\mathcal{L}$. This objective is optimized exclusively for the learnable tokens, the Spatial-Aware Cross-Attention (SCA) modules, and the Self-Alignment Function (SAF), while keeping the Vision Transformer backbone frozen. The total loss is a summation of three core components (with default weights of 1), designed to jointly optimize classification accuracy, segmentation precision, and feature separability:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{seg}} + \mathcal{L}_{\text{ctr}}. \tag{14}$$

We employ the Binary Cross-Entropy (BCE) loss $\mathcal{L}_{\text{cls}}$ to supervise the image-level anomaly decision. This loss is applied directly to the final image-level anomaly score S, guiding the model toward robust binary classification:

$$\mathcal{L}_{\text{cls}} = -\Big[y \log \sigma(S) + (1-y) \log\big(1-\sigma(S)\big)\Big]. \tag{15}$$

where $y \in \{0, 1\}$ is the ground-truth class label and $\sigma(\cdot)$ is the Sigmoid activation function. Notably, the score S is not derived from a global average, but rather from averaging the top-k highest-scoring pixels in the multi-layer fused anomaly map H. This is expressed as $S = \frac{1}{k}\sum_{i=1}^{k} H_{(i)}$, where $k = [0.01HW]$, emphasizing the anomaly decision's reliance on the most salient anomaly cues.

The pixel-level segmentation loss $\mathcal{L}_{\text{seg}}$ is essential for achieving accurate anomaly localization. Considering the complementary nature of ViT features across different layers (shallow layers capture local details, deep layers encode global semantics), $\mathcal{L}_{\text{seg}}$ is designed to be computed and aggregated across all selected intermediate layers $l \in \mathcal{L}$. It is composed of Focal Loss ($\mathcal{L}_{\texttt{focal}}$) and Dice Loss ($\mathcal{L}_{\text{dice}}$), which jointly operate on the predicted mask $\hat{\mathbf{M}}^{(\ell)} = \sigma(\mathbf{H}_\ell)$:

$$\mathcal{L}_{\text{seg}} = \sum_{\ell \in \mathcal{L}} \left(\mathcal{L}_{\text{focal}}^{(\ell)} + \mathcal{L}_{\text{dice}}^{(\ell)} \right). \tag{16}$$

$\mathcal{L}_{\texttt{focal}}$ is specifically designed to address the severe foreground-background class imbalance prevalent in industrial and medical datasets. By down-weighting the contribution of easily classified samples (most of the background), the loss focuses the training process on hard-to-classify, sparse anomaly regions and ambiguous boundaries. Its calculation is defined as:

$$\mathcal{L}_{\texttt{focal}} = -\frac{1}{N}\sum_{i=1}^{N}(1-p_i)^{\gamma}\log(p_i) \tag{17}$$

where N is the total number of pixels, and p_i is the predicted probability for the correct class. The focusing parameter γ controls the modulation intensity and is set to 2 in this study. As an overlap-based metric, the Dice Loss effectively quantifies the spatial agreement between the predicted mask and the true anomaly region, directly optimizing the shape and boundary coherence of the segmentation output. It is calculated as follows:

$$\mathcal{L}_{dice} = 1 - \frac{2\sum_i y_i \hat{y}_i}{\sum_i y_i + \sum_i \hat{y}_i} \tag{18}$$

where y_i and $\hat{y}_i$ represent the true label and predicted probability, respectively, for the i-th pixel in the image.

To explicitly establish highly distinct Normal and Abnormal semantic prototypes within the visual feature space, we introduce the contrastive separation loss $\mathcal{L}_{\text{ctr}}$. This loss applies a cosine-margin penalty only to the l_2-normalized enhanced Normal Token ($\bar{t}_n^{(L)}$) and Abnormal Token ($\bar{t}_a^{(L)}$) at the deepest selected layer L (the maximum index in $\mathcal{L}$):

$$\mathcal{L}_{\text{ctr}} = \max\Big(0,\ \langle \bar{\mathbf{t}}_a^{(L)}, \bar{\mathbf{t}}_n^{(L)} \rangle + \tau\Big), \quad \tau = 0.5, \tag{19}$$

This formulation enforces that the cosine similarity between the two Tokens must be less than $-\tau$. The margin τ is set to 0.5, ensuring that the Normal and Abnormal Tokens maintain a sufficient angular separation (greater than $120°$) in the feature space, thereby providing strong and highly-separated feature anchors for subsequent contrastive scoring.

Trainable Parameter Analysis of VisualAD. To quantify the learnable capacity of VisualAD, we provide a consolidated analysis of all trainable components, including the Spatial-Aware Cross-Attention (SCA) modules, the Self-Alignment Function (SAF), and the global anomaly/normal tokens. The ViT backbone remains entirely frozen, ensuring that all adaptation stems from a compact set of auxiliary modules. As summarized in Table 7, the majority of

Table 7. Trainable Parameter Breakdown of VisualAD

Module	Params	Share	Description
SCA	23,121,940	73.4%	4 layers
q/k/v proj	3,145,728		
out proj	1,049,600		
2D pos enc	1,572,864		
anchors	4,096		
gating	4,100		
LN (q,k)	4,096		
res scale	1		
SAF	8,396,800	26.6%	MLPs (4 layers)
Linear1	1,048,576		
biases	2,048		
Linear2	1,048,576		
Global Tokens	4,096	0.01%	Tokens + LN
anomaly token	1,024		
normal token	1,024		
LN weight	1,024		
LN bias	1,024		
Total	**31,522,836**	**100%**	

parameters (23.12M, 73.4%) originate from the multi-layer SCA blocks, each of which integrates query/key/value projections, output projections, high-resolution positional encodings, anchor queries, gating parameters, and normalization terms. These components constitute the primary mechanism through which VisualAD injects spatial priors and cross-layer aggregation. The SAF contributes an additional 8.40M parameters (26.6%) through lightweight MLP-based transforms that recalibrate patch representations for stable multi-layer alignment. In contrast, the global anomaly and normal tokens, including their associated LayerNorm parameters, account for only 0.004M parameters (0.01%), highlighting the compactness of the semantic representations themselves. Overall, VisualAD contains 31.52M trainable parameters, offering a favorable balance between adaptability and parameter efficiency.

A.2. State-of-the-art Methods

- **WinCLIP** [24] is one of the earliest works based on CLIP for the ZSAD task. Since the vanilla CLIP does not align text with fine-grained image features during pretraining, it addresses this limitation by dividing the input image into multiple sub-images using windows of varying scales. The final anomaly segmentation results are derived by harmoniously aggregating the classification outcomes of sub-images corresponding to the same spatial locations. In addition, a two-class text prompt design method, named Compositional Prompt Ensemble, is proposed and has been widely adopted in subsequent works.
- **APRIL-GAN** [11] adopts the handcrafted textual prompt design strategy from WinCLIP. However, for aligning textual and visual features, it introduces a linear adapter layer to project fine-grained patch features into a joint embedding space. After training on an auxiliary dataset, it can directly generalize to novel categories.
- **CLIP-AD** [12] builds upon the modality alignment design of APRIL-GAN, continuing to use a linear adapter to project patch-level image features. The key difference is that it incorporates a feature surgery strategy to further address the issues of opposite predictions and irrelevant highlights.
- **AnomalyCLIP** [58] employs a text design strategy based on prompt optimization. By training on an auxiliary dataset, it learns object-agnostic text prompts that can be directly transferred to unseen object categories.
- **AdaCLIP** [9] introduces a hybrid prompt mechanism that integrates both dynamic and static prompts, embedding them into the text and image encoder layers. By incorporating visual prompts, the output text embeddings are able to dynamically adapt to the input image, thereby enhancing generalization performance.

Since the official code for WinCLIP has not been released, we use the reproduced code from AnomalyCLIP [58]. For APRILGAN, CLIP-AD, AnomalyCLIP, and AdaCLIP, we retrained the models using the official code, maintaining the same backbone, input image resolution, and experimental settings (training on the VisA dataset and testing on other datasets) as those used in our VisualAD. This ensures the fairness of the comparison between our VisualAD and other state-of-the-art (SOTA) methods.

B. Additional Experimental Results

B.1. Motivation

Existing mainstream Zero-Shot Anomaly Detection (ZSAD) methodologies predominantly leverage the transferability of Vision-Language Models (VLMs), such as CLIP. These methods rely on aligning visual representations with either heuristically-designed or learnable textual prompts to establish Normal and Abnormal semantic prototypes. This leads to a critical theoretical inquiry: **If the ultimate decision boundary for open-set discrimination is governed exclusively by the contrast between two learned reference feature vectors, is the inclusion of the computationally intensive text modality fundamentally necessary?**

To rigorously evaluate this premise, we performed an exploratory ablation study: we modified the AnomalyCLIP [58] architecture by decoupling the Text Encoder and substituting the trainable textual prompts with two learnable visual tokens—one for normality and one for abnormality—which were directly integrated into the frozen Vision Transformer (ViT) backbone.

The empirical results, visually encapsulated in Figure 1

of the main text, demonstrated that this purely visual variant achieved comparable, and occasionally slightly superior, ZSAD performance on demanding industrial benchmarks such as VisA and MVTec. Furthermore, adopting the visual-only architecture resulted in a dramatic reduction in trainable parameters, exceeding 99%. Crucially, the evaluation curves consistently revealed that the VisualAD prototype maintained a smooth and stable convergence trajectory, contrasting sharply with the original VLM-based AnomalyCLIP, which exhibited significant fluctuations and signs of optimization instability.

This compelling evidence validates two primary conclusions: 1) Within the VLM-based ZSAD pipeline, the Text Encoder primarily functions as an **indirect structural mechanism** for shaping a pair of discriminative visual prototypes. 2) Reliance on cross-modal alignment introduces unnecessary computational overhead and instability. This observation serves as the core motivation for VisualAD's design: a streamlined, language-free framework that maximizes efficiency, stability, and purely visual grounding by learning and refining these discriminative Normal and Abnormal prototypes directly within the latent feature space of the frozen ViT.

B.2. Instantiations of the Self-Alignment Function (SAF).

Table 8. Ablation on different instantiations of the Self-Alignment Function (SAF) for CLIP and DINOv2 backbones. Image-level and pixel-level metrics are reported as (AUROC, F_1-max, AP) and (AUROC, F_1-max, AP), respectively.

Backbone	Transform	Image-level	Pixel-level
CLIP	Linear	(83.5, 81.4, 86.7)	(94.5, 29.7, 23.7)
	MLP	**(84.7, 82.5, 87.6)**	**(95.8, 34.6, 28.4)**
	MLP-Residual	(83.4, 80.9, 86.7)	(94.5, 30.4, 24.0)
	Adapter	(77.1, 77.8, 82.1)	(94.0, 24.7, 18.7)
	LeakyReLU	(82.5, 80.7, 85.5)	(93.8, 29.4, 23.6)
DINOv2	Linear	(81.6, 80.9, 85.7)	(95.1, 33.2, 27.9)
	MLP	**(83.1, 81.4, 86.8)**	**(95.3, 35.2, 29.9)**
	MLP-Residual	(81.9, 80.8, 85.6)	(95.3, 32.9, 27.6)
	Adapter	(81.8, 81.1, 85.6)	(95.5, 33.2, 28.2)
	LeakyReLU	(81.9, 80.6, 86.3)	(95.1, 33.8, 28.4)

In the main experiments, the Self-Alignment Function (SAF) is implemented as a lightweight token-wise transformation that recalibrates patch features while preserving their dimensionality. To disentangle the effect of different architectural choices, we consider five instantiations of SAF, all sharing the same input–output interface but differing in expressiveness and parameterization:

- **Linear.** SAF is realized as a single linear projection, applying an affine transformation to each patch feature. Given an input feature $\mathbf{x}$, the transformation is

$$\mathbf{y} = W\mathbf{x} + \mathbf{b}, \tag{20}$$

where W and $\mathbf{b}$ denote the weight matrix and bias vector, respectively. This variant introduces the minimum number of additional parameters and serves as a baseline to quantify the benefit of more expressive SAF designs.
- **MLP.** SAF is instantiated as a two-layer feed-forward network with a non-linear hidden layer:

$$\mathbf{y} = W_2\,\sigma(W_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2, \tag{21}$$

where W_1, W_2 and $\mathbf{b}_1, \mathbf{b}_2$ are learnable parameters and $\sigma(\cdot)$ denotes a point-wise activation function. Compared with the linear baseline in Eq. (20), this design enables SAF to capture non-linear relations in the feature space and to learn more flexible re-mappings of patch representations.
- **MLP-Residual.** This variant augments the two-layer MLP with an explicit residual connection:

$$\mathbf{y} = \mathrm{MLP}(\mathbf{x}) + \mathbf{x}, \tag{22}$$

where $\mathrm{MLP}(\cdot)$ follows Eq. (21). By combining a non-linear transformation with an identity shortcut, the residual SAF preserves the original feature as a stable reference while allowing learnable corrections. This design typically improves optimization stability and mitigates the risk of over-smoothing or overfitting introduced by the MLP.
- **Adapter.** Motivated by parameter-efficient adaptation, SAF is implemented as a bottleneck adapter:

$$\mathbf{y} = \mathbf{x} + W_{\mathrm{up}}\,\sigma(W_{\mathrm{down}}\mathbf{x} + \mathbf{b}_{\mathrm{down}}) + \mathbf{b}_{\mathrm{up}}, \tag{23}$$

where $W_{\mathrm{down}} \in \mathbb{R}^{d_{\mathrm{b}} \times d}$ projects the feature into a lower-dimensional bottleneck, $W_{\mathrm{up}} \in \mathbb{R}^{d \times d_{\mathrm{b}}}$ projects it back, and $d_{\mathrm{b}} \ll d$. This structure maintains the input dimensionality while substantially reducing the number of additional parameters, providing a parameter-efficient realization of SAF analogous to adapter modules in NLP.
- **LeakyReLU.** This variant shares the same two-layer MLP architecture as Eq. (21) but replaces the standard activation with LeakyReLU, defined as

$$\sigma(z) = \begin{cases} z, & z \geq 0, \\ \alpha z, & z < 0, \end{cases} \quad \text{with } \alpha = 0.01. \tag{24}$$

By allowing a small negative slope, LeakyReLU alleviates the "dead neuron" effect and can improve gradient flow in regions where activations would otherwise saturate at zero. This instantiation isolates the impact of the activation choice on the effectiveness of the Self-Alignment Function.

All SAF variants share the same input–output interface and preserve the original feature dimensionality, which makes their performance directly comparable in the ablation study reported in Table 8.

B.3. Ablations on DINOv2

In addition to the CLIP-based experiments in the main paper, we further conduct ablation studies with a self-supervised DINOv2 backbone to assess the backbone-agnostic behavior of VisualAD. The results are summarized in the following three tables, focusing on the contributions of individual modules and loss components, the choice of ViT layers for anomaly aggregation, and the number of anchor queries used in SCA.

Table 9 investigates the role of SCA, SAF, and different loss terms. When disabling SCA but keeping SAF and all loss components, the model still achieves competitive performance, with only a slight drop in AUROC but the highest image-level AP and pixel-level AP, indicating that SAF alone can already provide strong self-alignment and localization ability. In contrast, keeping SCA while removing SAF, or disabling both modules, leads to substantial degradation at both the image and pixel levels, suggesting that SAF is the primary contributor to stable feature calibration and that SCA plays a complementary role by injecting spatially grounded evidence. On the loss side, removing any of the focal, Dice, or contrastive loss terms consistently harms at least one of the evaluation metrics. Compared with these variants, the full VisualAD configuration with all modules and all loss components yields the best or second-best AUROC and AP across both image-level and pixel-level metrics, confirming that the proposed design is effective and that all three loss components contribute to the final performance.

Table 9. Ablation study on different modules and loss components using DINOv2. Both Image-level and Pixel-level metrics are reported as (AUROC, AP).

Ablation	SCA	SAF	Focal	Dice	Ctr	Image-level	Pixel-level
Module	✗	✓	✓	✓	✓	(84.1, **88.4**)	(94.7, **28.7**)
	✓	✗	✓	✓	✓	(59.7, 67.0)	(90.5, 10.2)
	✗	✗	✓	✓	✓	(51.6, 60.7)	(83.0, 3.2)
Loss	✓	✓	✗	✓	✓	(82.9, 87.4)	(95.2, 25.8)
	✓	✓	✓	✗	✓	(82.6, 87.6)	(94.6, 28.2)
	✓	✓	✓	✓	✗	(80.4, 84.9)	(94.7, 27.4)
VisualAD	✓	✓	✓	✓	✓	(**84.7**, 87.6)	(**95.8**, 28.4)

Table 10 examines different layer combinations in DINOv2. Using a single shallow layer ($\{6\}$) produces clearly inferior results, while middle and deep layers ($\{12\}$, $\{18\}$, $\{24\}$) progressively improve both AUROC and AP, which aligns with the intuition that deeper layers encode more semantic and anomaly-relevant information. The best overall performance is obtained when aggregating features from all four layers $\{6, 12, 18, 24\}$, which achieves the highest pixel-level AUROC, F_1-max, and AP, together with strong image-level scores. This observation indicates that combining low-level structural cues from shallow layers with high-level semantics from deep layers is beneficial for robust anomaly detection, and the trend is consistent with the CLIP-based layer ablations in the main text.

Table 10. Ablation on different layer combinations using DINOv2. Pixel-level and image-level metrics follow (AUROC, F_1-max, AP). Best and second-best results are in **bold** and underline.

Layers	Pixel-level	Image-level
$\{6\}$	(84.4, 5.6, 2.7)	(59.1, 73.4, 66.5)
$\{12\}$	(93.7, 24.1, 19.2)	(74.6, 76.9, 79.9)
$\{18\}$	(95.0, 30.2, 25.2)	(79.8, 80.1, 84.4)
$\{24\}$	(94.4, 33.1, 26.6)	(**83.4**, 81.3, **87.2**)
$\{6, 12\}$	(92.8, 23.5, 18.2)	(75.7, 78.5, 80.1)
$\{6, 12, 18\}$	(94.4, 31.6, 26.5)	(80.6, 80.2, 85.6)
$\{6, 12, 18, 24\}$	(**95.3**, **35.2**, **29.9**)	(83.1, **81.4**, 86.8)

Table 11. Ablation on the number of anchor queries using DINOv2. Pixel-level and image-level metrics are reported as (AUROC, F_1-max, AP). The best and second-best results are highlighted in **bold** and underline, respectively.

Anchors	Pixel-level	Image-level
1	(94.7, 32.4, 26.7)	(**84.4**, **83.3**, **88.8**)
2	(94.4, 28.6, 23.6)	(81.4, 80.9, 86.2)
4	(**95.3**, **35.2**, **29.9**)	(83.1, 81.4, 86.8)
8	(94.5, 30.6, 25.4)	(82.3, 81.0, 86.9)
16	(93.7, 33.2, 26.9)	(83.4, 80.9, 87.5)
32	(94.3, 30.5, 25.2)	(82.6, 81.6, 87.3)

Finally, Table 11 studies the number of anchor queries used in SCA. With a single anchor, the model attains the best image-level AUROC, F_1-max, and AP, showing that a very compact set of queries can already capture discriminative global statistics for image-wise decisions. However, increasing the number of anchors to a moderate value (e.g., 4 or 16) leads to the best or second-best pixel-level performance, as more anchors provide finer spatial coverage and richer local context. When the number of anchors becomes too large, the gains saturate or slightly decline, suggesting a trade-off between spatial granularity and redundancy. Overall, these DINOv2-based ablations corroborate the conclusions drawn from the CLIP backbone, and demonstrate that VisualAD and its components generalize well across different ViT architectures.

B.4. Additional Training Dynamics and Cross-Dataset Evaluation

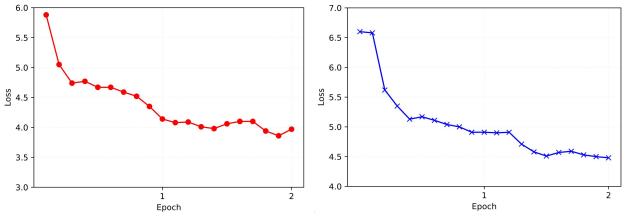

Figure 7. Training loss curves on MVTec-AD for the first two epochs. The left panel reports results with CLIP ViT-L/14@336px and the right panel reports DINOv2-ViT-L/14. In both cases, the loss drops sharply at the beginning of training and stabilizes rapidly, indicating that VisualAD reaches a near-optimal regime within a very small number of iterations.

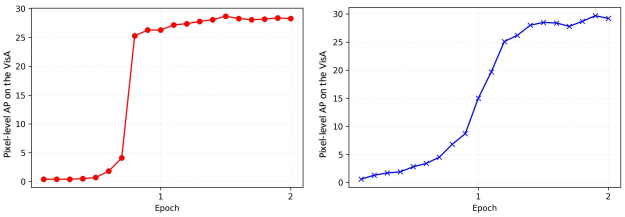

Figure 8. Pixel-level AP on VisA across the first two training epochs. Performance rises quickly for both CLIP ViT-L/14@336px (left) and DINOv2-ViT-L/14 (right), approaching a saturated level within the first epoch, which demonstrates the fast convergence and stable optimization.

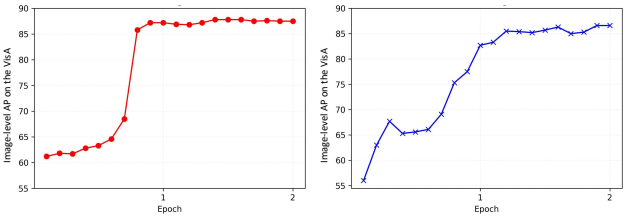

Figure 9. Image-level AP on VisA during the first two training epochs. Similar to the pixel-level results, both backbones exhibit a rapid increase in AP and converge early, with only marginal gains observed in the second epoch. This confirms that VisualAD achieves strong image-level discrimination without requiring prolonged training.

To further examine the optimization behavior of **VisualAD**, we report the training losses on MVTec-AD and the corresponding anomaly detection performance on VisA under two backbone configurations: CLIP's ViT-L/14@336px and DINOv2-ViT-L/14. Since our framework converges extremely quickly, we visualize only the first two training epochs, which already capture the full stabilization trend.

Fig. 7 shows the loss trajectories on MVTec-AD. For both backbones, the loss exhibits a steep decline within the first few hundred iterations, after which the curves flatten, indicating rapid stabilization. The ViT-L/14@336 backbone shows a slightly faster initial drop, whereas DINOv2-ViT-L/14 presents a smoother descent, consistent with its stronger pretraining.

The corresponding pixel-level average precision (AP) and image-level AP on VisA are reported in Fig. 8 and 9. In both cases, AP increases sharply at the beginning of training, rising from near-random initialization to near-saturated values within a single epoch. The second epoch yields only marginal improvements, confirming that VisualAD reaches its optimal regime almost immediately. This behavior is consistent across CLIP and DINOv2 backbones, demonstrating that the proposed combination of learnable tokens, SCA, and SAF provides a highly stable optimization landscape without the oscillatory behavior commonly observed in prompt-based or dual-encoder anomaly detection pipelines.

Moreover, preliminary evaluation on MVTec-AD also reflects this rapid convergence: the image-level and pixel-level metrics reach close-to-final values after the first epoch, and the performance gap between the two backbones remains small, suggesting that VisualAD generalizes robustly across both industrial (VisA) and texture/object-centric (MVTec) domains even with minimal training.

Overall, these results indicate that **VisualAD not only maintains strong detection accuracy but also exhibits exceptionally fast convergence and stable training dynamics**, which further supports its suitability for practical zero-shot anomaly detection scenarios where efficiency and reproducibility are essential.

C. Datasets

C.1. Industrial Domain

- **MVTec-AD** [6] is specifically designed for industrial anomaly detection, consisting of 15 different categories (e.g., bottle, wood). In this work, we use only its labeled test set, which includes 467 normal images and 1,258 anomalous images. It also includes data from both texture and object types, making it more comprehensive, and is widely used in our ablation experiments.
- **VisA** [60] is a challenging industrial dataset that includes 12 categories (e.g., candle, capsules), all of which are object types. Its test set contains 962 normal images and 1,200 anomalous images, and is primarily used for auxiliary evaluation.
- **BTAD** [37] includes three categories, all of which are object types, with resolutions ranging from 600 to 1600. The dataset contains 451 normal images and 290 anomalous images, used to evaluate the ZSAD performance.

- **KSDD2** [49] is a dataset designed for industrial defect detection. It includes 2,085 normal images and 246 abnormal images in the original training set, as well as 894 normal images and 110 abnormal images in the original test set. The image dimensions are similar, approximately 230 pixels in width and 630 pixels in height. In this work, we reconstruct the dataset for ZSAD. Specifically, all 356 abnormal images from the original training and test sets, along with an equal number of randomly selected normal images, are combined to form a new dataset. Note that this differs from the dataset processing approach in VCP-CLIP [41], where only all anomalous samples from the test set are used for zero-shot anomaly segmentation evaluation.
- **DAGM** [51] is a texture dataset designed for weakly supervised anomaly detection, consisting of 10 categories. It contains 6,996 normal images and 1,054 abnormal images. Since the original pixel-level annotations are weak labels in the form of ellipses, we manually re-annotate the DAGM dataset for anomaly segmentation.
- **DTD-Synthetic** [1] is a synthetic dataset designed for texture anomaly detection, comprising 12 categories. It contains 357 normal images and 947 anomalous images.

C.2. Medical Domain

- **OCT17** [26] is a large-scale dataset initially designed for classification tasks. It consists of retinal OCT images categorized into three types of anomalies: Choroidal Neovascularization (CNV), Diabetic Macular Edema (DME), and Drusen Deposits (DRUSEN). The images are continuous slices with a size of 512*496.
- **BrainMRI** [47] is a dataset for brain MRI analysis, comprising images of both healthy and abnormal brain scans, including conditions like tumors and lesions. It contains 98 anomalous images and 155 normal images, with only image-level labels. In this work, we directly adopt the dataset curated by AdaCLIP [9].
- **Brain_AD** [2, 3, 36] is built upon the BraTS2021 dataset, utilizing 3D FLAIR volumes. To account for variations in brain images at different depths, slices within the depth range of 60 to 100 of the 3D FLAIR volumes are selected. Each extracted 2D slice was saved in PNGformat and has an image size of 240×240 pixels. The training set encompasses 7,500 normal samples, while the test set comprises 3,715 samples with a balanced ratio of normal to anomaly instances.
- **HIS** [4] is derived from Camelyon16 and contains 400 hematoxylin- and eosin-stained whole-slide images of lymph node sections from breast-cancer patients. The training set incorporates 5,088 randomly extracted normal patches from the original training set. For testing, 1,003 normal and 997 abnormal patches from the 115 testing WSIs are utilized.
- **CVC-ClinicDB** [7] is a dataset for colorectal cancer detection in endoscopy images, containing 612 anomalous images with pixel-level annotations. Therefore, it is also used exclusively for anomaly segmentation tasks. In this work, we directly adopt the dataset curated by AdaCLIP [9].
- **Endo** [20] is another dataset similar to CVC-ColonDB, comprising 200 anomalous images with pixel-level annotations. While it is also used for colon polyp detection, differences in image acquisition devices and environments introduce a certain domain gap compared to other datasets. In this work, we directly adopt the dataset curated by AnomalyCLIP [58].
- **Kvasir** [20] is a larger medical dataset used for colon polyp detection in endoscopy images. It contains 1,000 anomalous images with pixel-level annotations and is used for anomaly segmentation task in the medical domain in this work. In this work, we directly adopt the dataset curated by AnomalyCLIP [58].

D. Detailed ZSAD results

From Fig. 10 to Fig. 18, we present detailed zero-shot anomaly detection (ZSAD) visualizations across all industrial, texture, and medical benchmarks. Each figure shows the input image, the ground-truth anomaly mask, and the corresponding anomaly score maps produced by VisualAD with CLIP and with DINOv2.

Overall, the results reveal that VisualAD produces sharp and spatially coherent anomaly maps, accurately capturing small, low-contrast, and irregular defects while suppressing background regions. The consistent behavior across diverse datasets and domains indicates that the proposed SCA and SAF modules provide effective spatial grounding and multi-layer alignment, supporting strong zero-shot generalization in both industrial and medical scenarios.

E. Limitations and Future Directions

Our VisualAD framework has already demonstrated state-of-the-art ZSAD performance across 13 industrial and medical datasets. However, it still faces several limitations in practical applications: 1) During the inference stage, the use of a fixed multi-layer feature ensemble introduces redundant computation, hindering the model's ability to achieve optimal inference efficiency; 2) The model's ultimate performance ceiling is constrained by the generic feature capacity of the frozen ViT backbone, which particularly challenges the detection of extremely subtle, fine-grained anomalies. In future work, we plan to investigate mechanisms like adaptive layer selection and structured tokens that may be better suited for achieving both high efficiency and robust, fine-grained feature refinement.

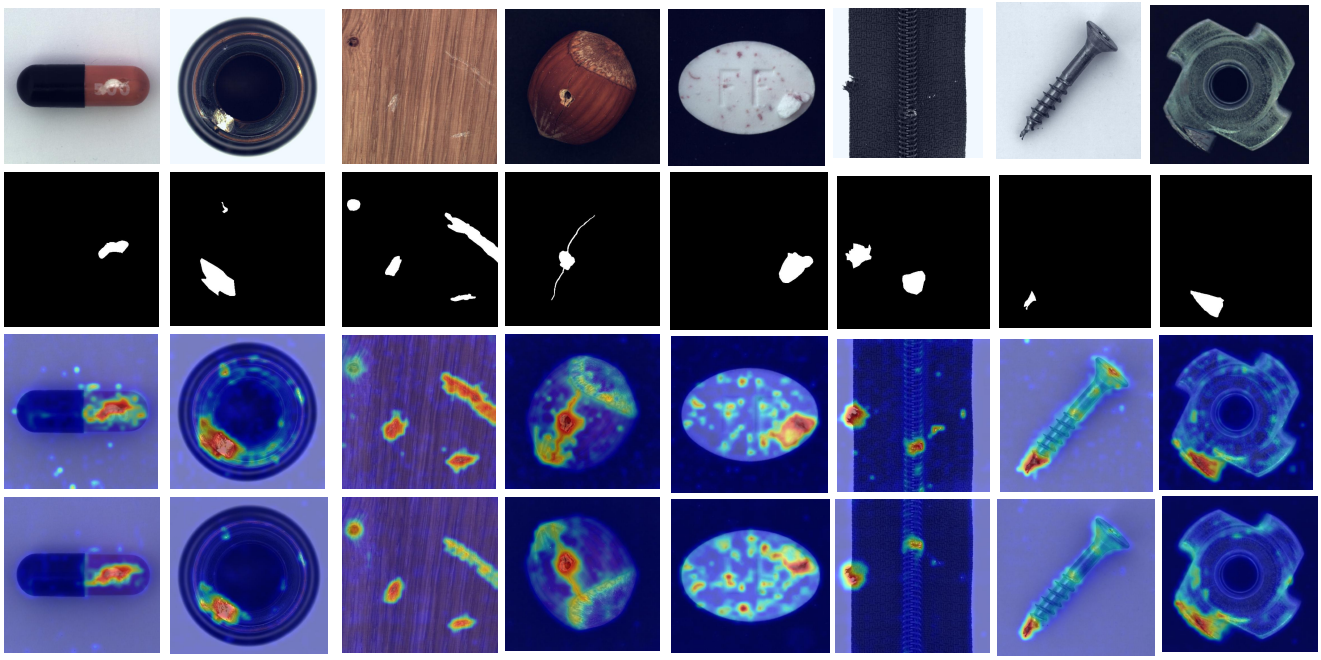

Figure 10. Anomaly score maps for samples from the MVTec dataset.The first row represents the input, the second row shows the ground-truth anomaly masks, the third row presents the segmentation results from VisualAD with CLIP, and the last row displays the segmentation results from VisualAD with DINOv2.

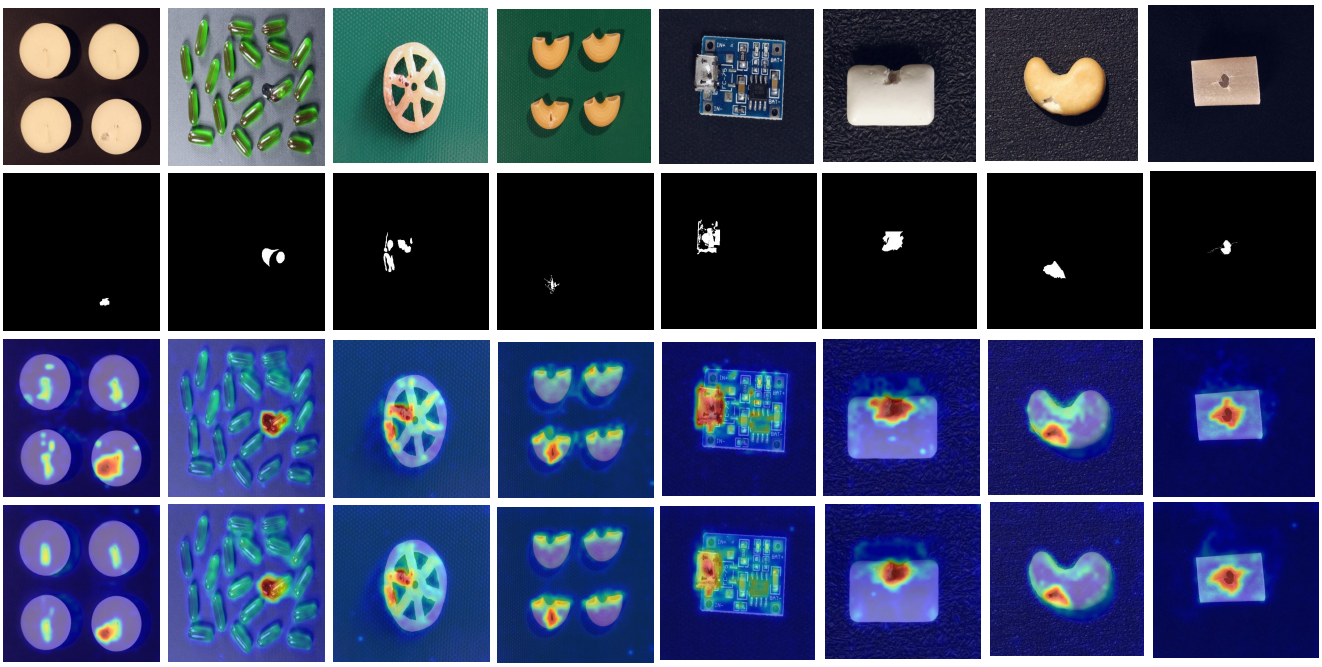

Figure 11. Anomaly score maps for samples from the VisA dataset.The first row represents the input, the second row shows the ground-truth anomaly masks, the third row presents the segmentation results from VisualAD with CLIP, and the last row displays the segmentation results from VisualAD with DINOv2.

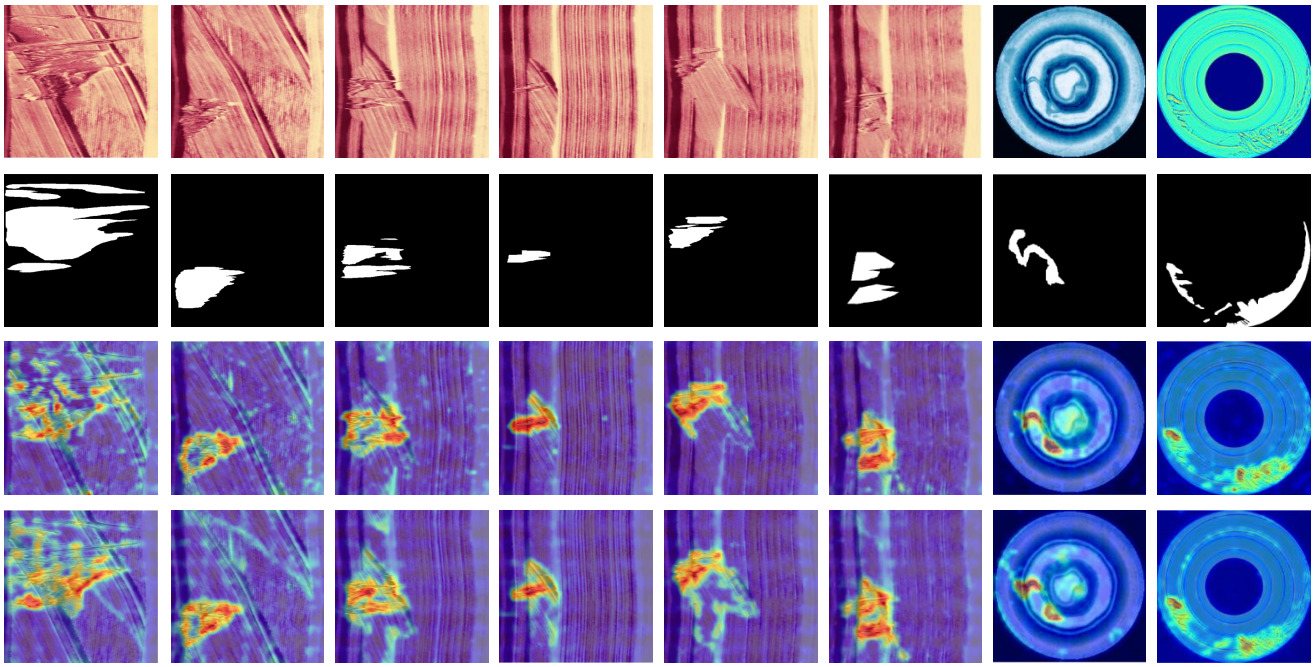

Figure 12. Anomaly score maps for samples from the BTAD dataset.The first row represents the input, the second row shows the ground-truth anomaly masks, the third row presents the segmentation results from VisualAD with CLIP, and the last row displays the segmentation results from VisualAD with DINOv2.

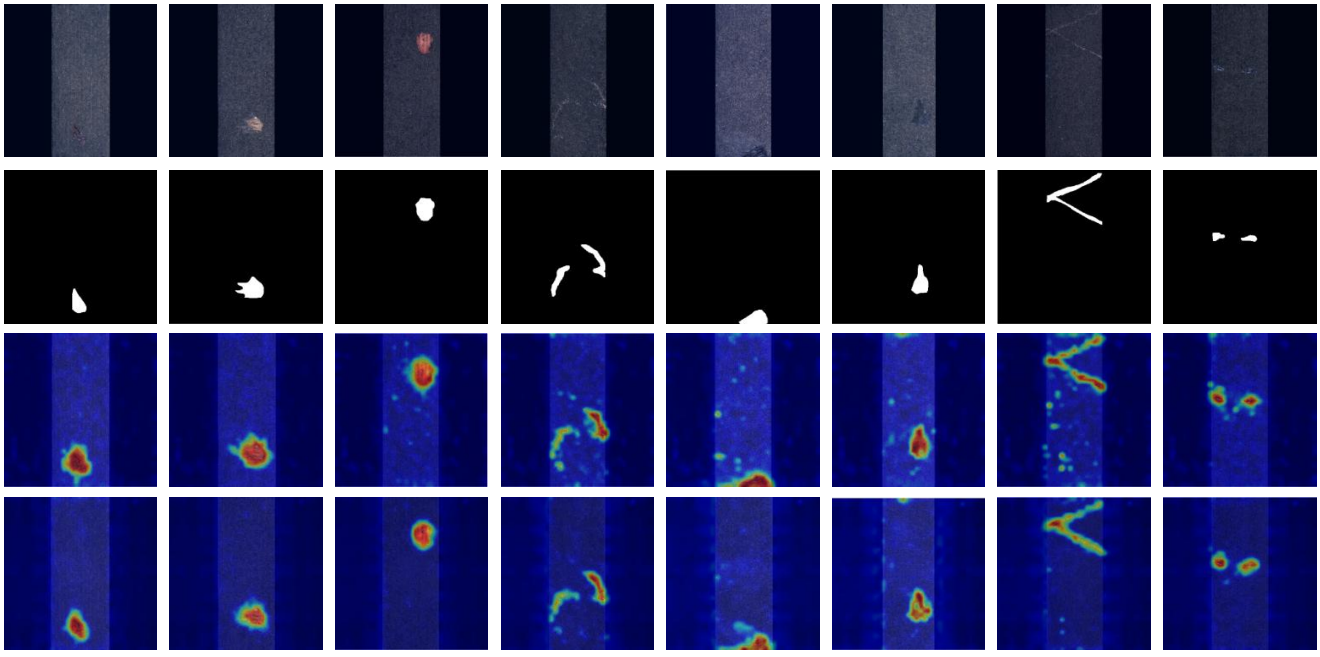

Figure 13. Anomaly score maps for samples from the KSDD2 dataset.The first row represents the input, the second row shows the ground-truth anomaly masks, the third row presents the segmentation results from VisualAD with CLIP, and the last row displays the segmentation results from VisualAD with DINOv2.

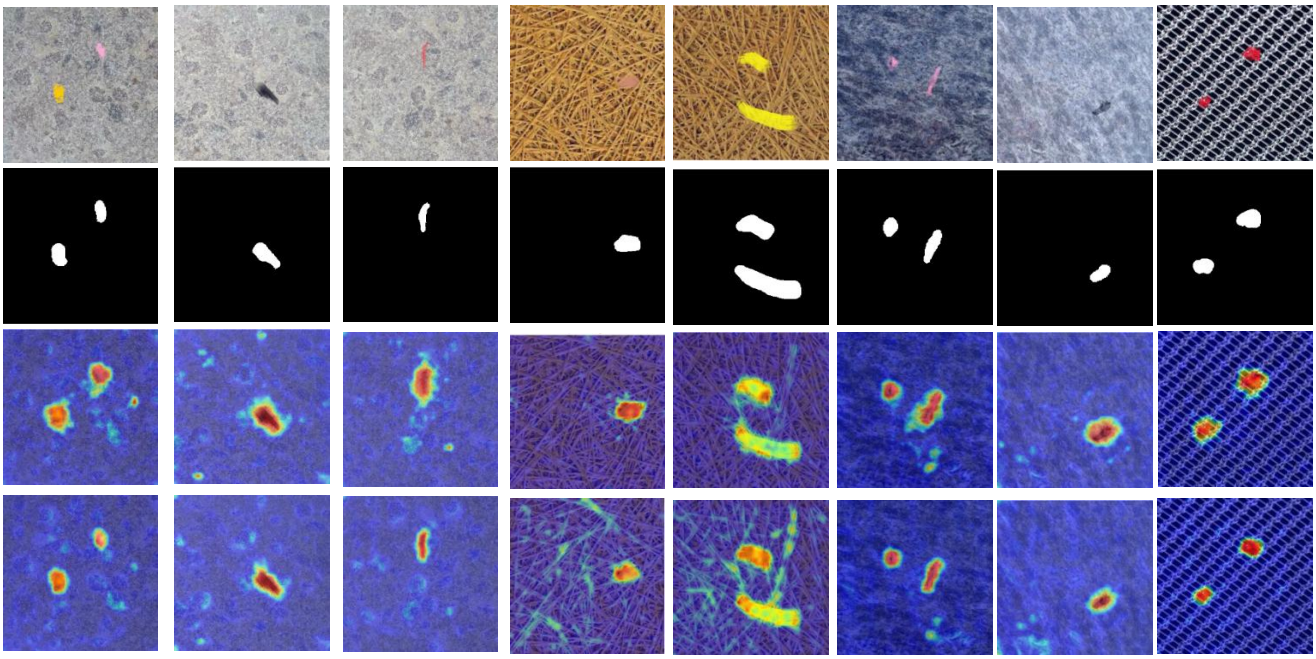

Figure 14. Anomaly score maps for samples from the DTD-Synthetic dataset.The first row represents the input, the second row shows the ground-truth anomaly masks, the third row presents the segmentation results from VisualAD with CLIP, and the last row displays the segmentation results from VisualAD with DINOv2.

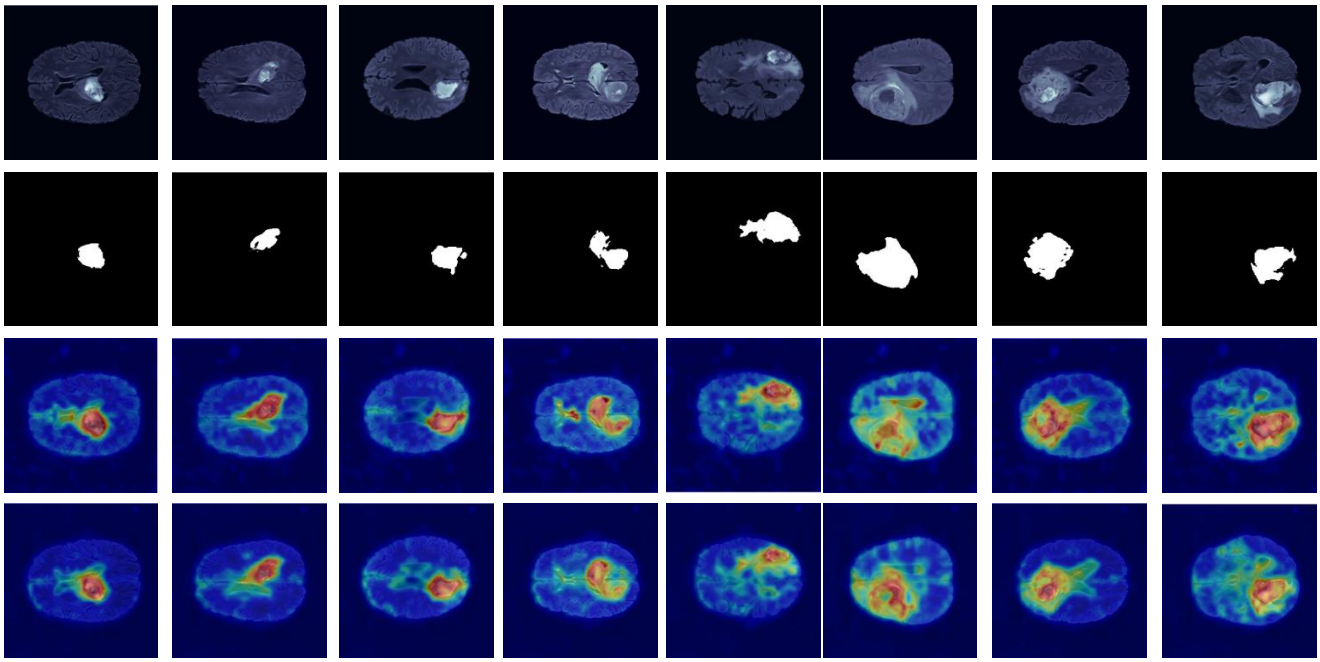

Figure 15. Anomaly score maps for the data subset brain.The first row represents the input, the second row shows the ground-truth anomaly masks, the third row presents the segmentation results from VisualAD with CLIP, and the last row displays the segmentation results from VisualAD with DINOv2.

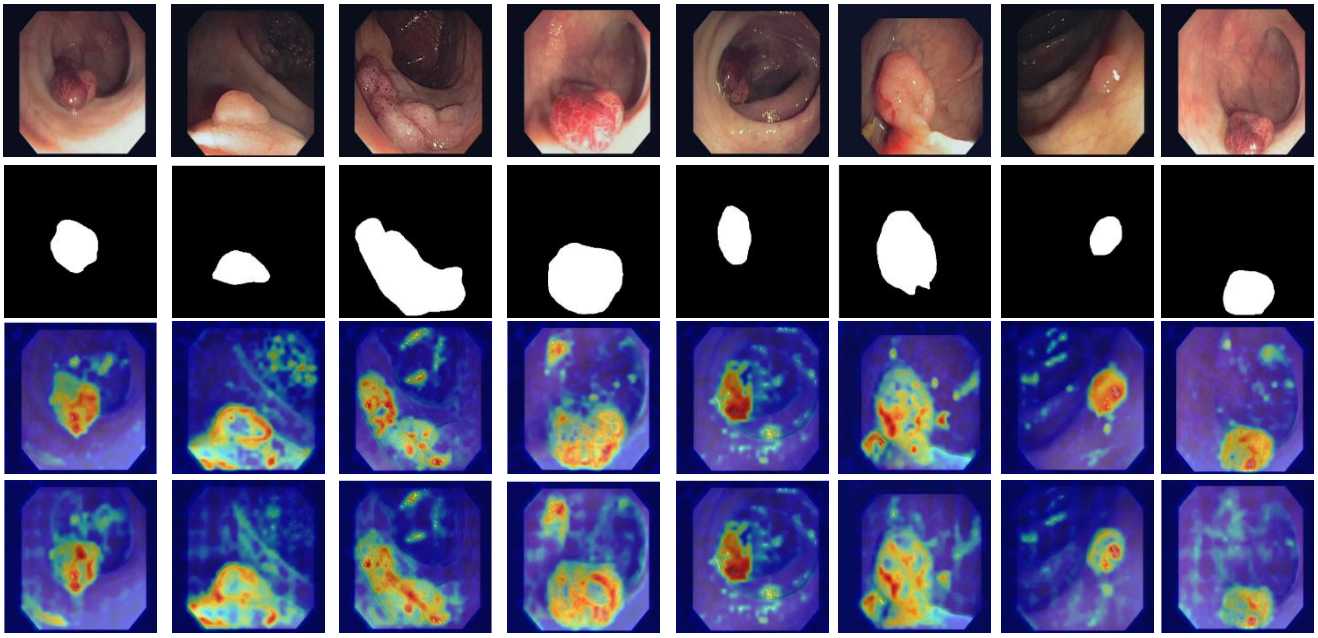

Figure 16. Anomaly score maps for samples from the CVC-ClinicDB dataset.The first row represents the input, the second row shows the ground-truth anomaly masks, the third row presents the segmentation results from VisualAD with CLIP, and the last row displays the segmentation results from VisualAD with DINOv2.

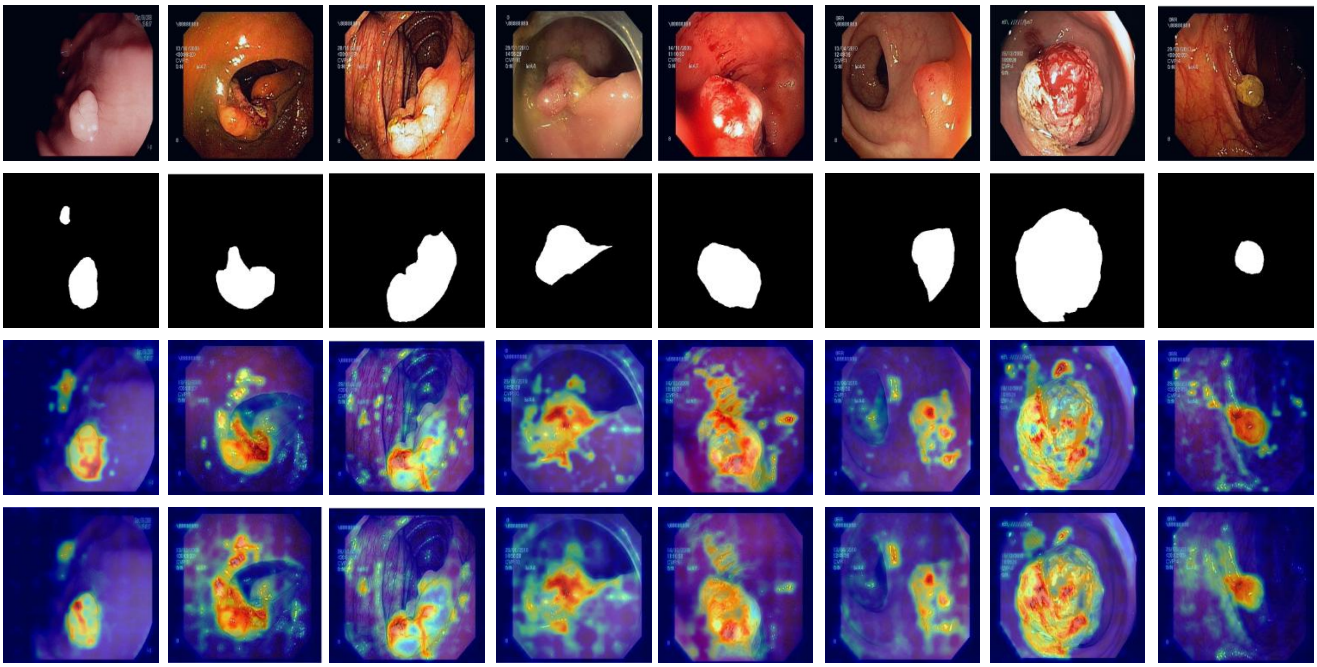

Figure 17. Anomaly score maps for samples from the Endo dataset. The first row represents the input, the second row shows the ground-truth anomaly masks, the third row presents the segmentation results from VisualAD with CLIP, and the last row displays the segmentation results from VisualAD with DINOv2.

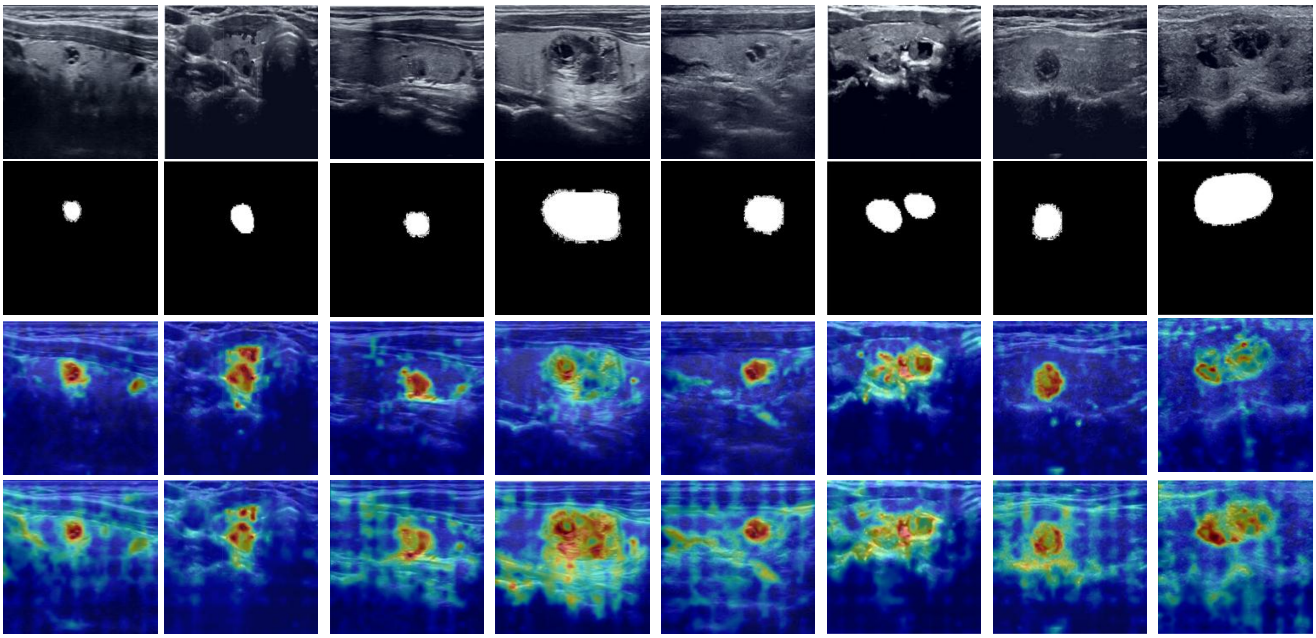

Figure 18. Anomaly score maps for the data subset thyroid. The first row represents the input, the second row shows the ground-truth anomaly masks, the third row presents the segmentation results from VisualAD with CLIP, and the last row displays the segmentation results from VisualAD with DINOv2.