arXiv:2508.13560v2 [cs.CV] 21 Aug 2025

DictAS: A Framework for Class-Generalizable Few-Shot Anomy Segmentation via Dictionary Lookup

Zhen Qu[1, 2] Xian Tao[1,2,3,4†] Xinyi Gong[5] ShiChen Qu[1, 2] Xiaopei Zhang[7]
Xingang Wang[1, 2] Fei Shen[1,2,3,4] Zhengtao Zhang[1,2,3] Mukesh Prasad[6] Guiguang Ding[8]
[1]Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences [3]Casivision
[4]Longmen Laboratory [5]HDU [6]UTS [7]UCLA [8]Tsinghua University

Abstract

Recent vision-language models (e.g., CLIP) have demonstrated remarkable class-generalizable ability to unseen classes in few-shot anomaly segmentation (FSAS), leveraging supervised prompt learning or fine-tuning on seen classes. However, their cross-category generalization largely depends on prior knowledge of real seen anomaly samples. In this paper, we propose a novel framework, namely DictAS, which enables a unified model to detect visual anomalies in unseen object categories without any retraining on the target data, only employing a few normal reference images as visual prompts. The insight behind DictAS is to transfer dictionary lookup capabilities to the FSAS task for unseen classes via self-supervised learning, instead of merely memorizing the normal and abnormal feature patterns from the training set. Specifically, DictAS mainly consists of three components: (1) ***Dictionary Construction*** *- to simulate the index and content of a real dictionary using features from normal reference images. (2)* ***Dictionary Lookup*** *- to retrieve queried region features from the dictionary via a sparse lookup strategy. When a query feature cannot be retrieved, it is classified as an anomaly. (3)* ***Query Discrimination Regularization*** *- to enhance anomaly discrimination by making abnormal features harder to retrieve from the dictionary. To achieve this, Contrastive Query Constraint and Text Alignment Constraint are further proposed. Extensive experiments on seven public industrial and medical datasets demonstrate that DictAS consistently outperforms state-of-the-art FSAS methods.*

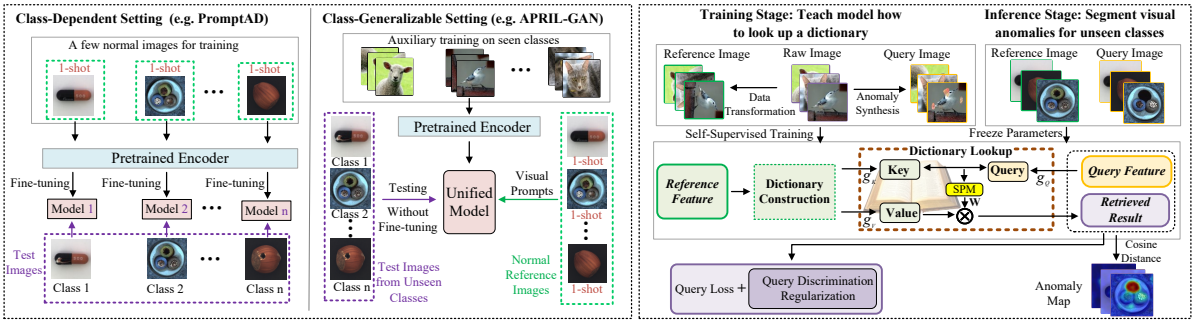
1. Introduction

Few-shot (few-normal-shot) anomaly segmentation (FSAS) aims to identify anomalous regions in images given only a limited number of normal samples. This task is particularly important in scenarios where training data is scarce and pixel-level annotations are limited, such as industrial defect detection [6, 26, 27] and medical image analysis [1, 22, 24].

Existing FSAS methods typically follow class-dependent or class-generalizable setting, as shown in Figure 1(a). Class-dependent approaches require fine-tuning on each unseen class with a limited number of normal samples. Since the training and testing image classes are identical, they primarily focus on modeling the distribution of normal images or learning the boundary between normal and abnormal content through regularization, without considering the substantial domain gaps across categories [2, 20, 21, 33, 34, 36]. Consequently, such methods encounter significant challenges when applied to privacy-sensitive medical applications or dynamic industrial scenarios with frequent production line changes.

In contrast, as shown in the right part of Figure 1(a), class-generalizable approaches aim to develop a unified model capable of detecting anomalies in unseen classes without retraining on target data, relying solely on a few normal samples as visual prompts [7, 11–13, 15, 16]. The earliest work, RegAD [15], introduces feature registration to align features but suffers from reduced inference efficiency due to heavy reference image augmentation. FastRecon [11] proposes feature reconstruction using linear regression but is prone to over-reconstruction. More recent methods have increasingly focused on pre-trained vision-language models (VLMs), particularly CLIP [30], to enhance zero/few-shot generalization for unseen classes [5, 7, 13, 16, 28, 29, 41]. Approaches such as WinCLIP [16] and APRIL-GAN [7] introduce memory bank-based visual priors through normal images and enhance FSAS performance by exploiting CLIP's image-text alignment capabilities. Despite their promising performance, these methods often rely on "*empirical knowledge*" learned from seen abnormal images during auxiliary training stage, which constrains their

†Corresponding Author: taoxian2013@ia.ac.cn.
‡Project Page: https://github.com/xiaozhen228/DictAS



(a) Two settings in few-shot anomaly segmentation

(b) Motivation of our DictAS

Figure 1. (a) Two settings in few-shot anomaly segmentation. (b) Motivation of our DictAS.

ability to generalize to novel classes.

However, even novice human inspectors can detect anomalies in unseen categories with just a few normal samples as references, without extensive prior experience. We approximate this intuition as a dictionary lookup: if a region in the query image cannot be retrieved from the dictionary, it is classified as anomalous; otherwise, it is normal. Inspired by this, a novel self-supervised framework built on CLIP, namely DictAS, is proposed for class-generalizable FSAS. The framework comprises three components: Dictionary Construction, Dictionary Lookup, and Query Discrimination Regularization, as depicted in Figure 1(b). Our motivation is to reformulate anomaly segmentation as a dictionary lookup task—determining whether a query feature exists in the dictionary. Through self-supervised training, the model acquires a feature-agnostic and dynamically adaptive dictionary lookup capability, enabling class-generalizable FSAS.

The proposed Dictionary Construction component organizes normal reference image features into a structured dictionary, where the *Dictionary Key* and *Dictionary Value* serve as the index and content, respectively. As demonstrated in Figure 1(b), given a query image, its extracted features are transformed into a *Dictionary Query*, which is then matched against the *Dictionary Key* to retrieve the most relevant normal patterns. To refine this retrieval process, we introduce a sparse lookup strategy within the Dictionary Lookup component. Specifically, the *Query-Key* matching results are processed by a Sparse Probability Module (SPM), which encourages sparsity in the retrieval process and prioritizes the most relevant *Dictionary Value*. Unlike prior VLM-based methods [7, 13, 16] that rely on memory-based visual priors or prior knowledge derived from text-image alignment, DictAS learns dynamically adaptive retrieval weights, enabling flexible adaptation to variations in both query and reference images. For self-supervised training, the raw images are processed using anomaly synthesis and data transformation algorithms to generate (query-reference) image pairs. To optimize the dictionary lookup task, a query loss is further introduced to minimize the average distance between the normal regions of input ***Query Feature*** and their counterparts in ***Retrieved Result*** generated by the dictionary lookup process.

To enhance the anomaly discrimination ability, the Query Discrimination Regularization component is proposed, which makes it harder for anomalous regions in the query image to be retrieved from the dictionary. It consists of two parts: the Contrastive Query Constraint (CQC) and the Text Alignment Constraint (TAC). The CQC explicitly enforces greater feature distances between the ***Query Feature*** and its ***Retrieved Result*** in anomalous regions compared to normal regions, ensuring that anomalies are effectively distinguished. Meanwhile, the TAC regularizes the retrieved global image representation by aligning it with the normal text embedding space, preventing the model from misinterpreting anomalous content as normal.

In summary, our contributions are threefold: 1) We propose DictAS, a novel self-supervised framework that reformulates anomaly segmentation as a dictionary lookup task, enabling models to learn an adaptive retrieval capability for class-generalizable FSAS; 2) We introduce two regularizations strategies—Contrastive Query Constraint and Text Alignment Constraint—to enhance the robustness of dictionary-based querying and improve anomaly discrimination; 3) DictAS achieves state-of-the-art performance on seven industrial and medical datasets, demonstrating superior FSAS performance even when trained on auxiliary datasets without pixel-level annotations.

2. Related Works

Class-Dependent FSAS methods fine-tune a separate model for each class using only its corresponding normal reference images. They primarily focus on: (1) estimating the distribution of a few normal samples [32, 34], or (2) learning a discriminative boundary using only normal data [2, 20, 21, 33]. Since these methods do not need to

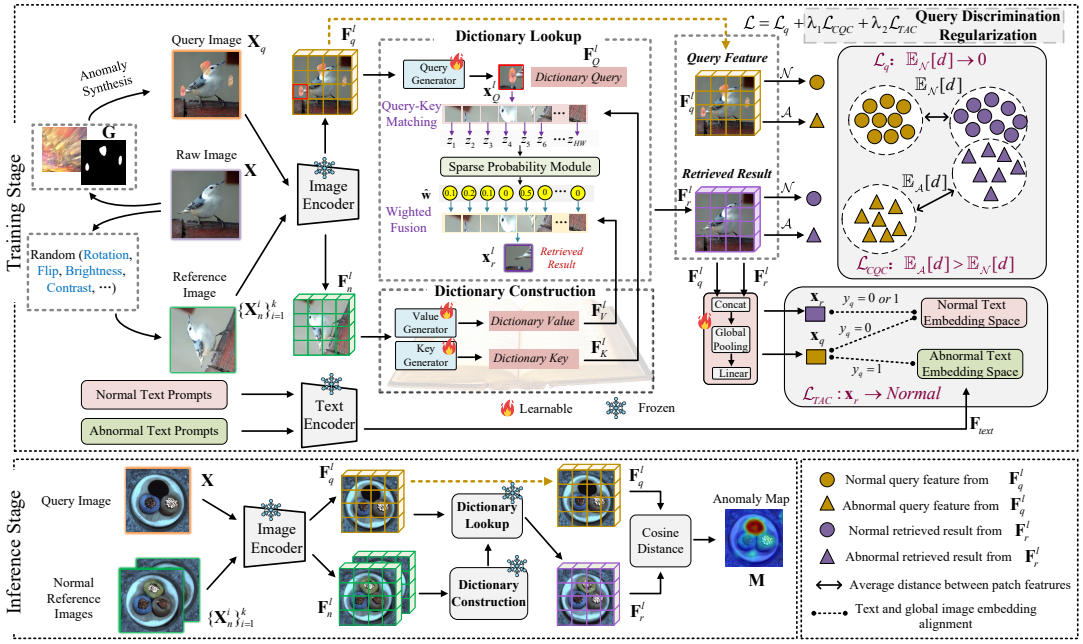


Figure 2. The framework of DictAS. It mainly consists of three components: Dictionary Construction, Dictionary Lookup and Query Discrimination Regularization. During training, the number of reference images is set to $k = 1$ for model efficiency. During inference, $k \geq 1$ normal reference images are used as visual prompts.

address domain shifts between training and testing classes, they typically achieve superior performance compared to class-generalizable approaches.

Class-Generalizable FSAS methods train a unified model on seen classes from an auxiliary dataset and directly generalize to unseen classes without additional fine-tuning. Early distance-based approaches [8, 9, 31, 37] detect anomalies by measuring the distance between query and reference images or features but are sensitive to image perturbations. Meta-learning-based methods [12, 15, 35] aim to achieve class-generalizable FSAS through task generalization but often require offline construction of auxiliary datasets and involve complex training processes. Feature residual-based methods [38, 42] mitigate inter-class variations by refining features based on the residuals between query and reference image features, though their effectiveness depends on the quality of residual feature extraction. Vision-language model-based approaches [7, 16] align text embeddings with image patch features in a joint embedding space, enabling language-guided anomaly segmentation. Building upon this, AnomalyGPT [13] further incorporates large language models to facilitate multi-turn user interaction, but introduces higher inference overhead.

Unlike the aforementioned class-generalizable methods, our DictAS reformulates FSAS as a dictionary lookup task, inspired by human inspectors. This enables us to leverage a large number of pixel-annotation-free images from seen classes for self-supervised training, guiding the model to learn adaptive querying and robust dictionary retrieval, thereby effectively addressing the FSAS task.

3. Method

Problem Statement. Class-generalizable FSAS is a challenging task that aims to achieve high performance on unseen classes without requiring fine-tuning on the target data. The seen classes C^s in the auxiliary training set and the unseen classes C^u in the test set must satisfy $C^s \cap C^u = \emptyset$. The ultimate goal is to develop a unified model capable of segmenting visual anomalies in novel classes C^u, relying solely on k-shot normal samples corresponding to the same class as the visual prompts.

3.1. Overview

The framework of our DictAS is illustrated in Figure 2. We employ the transformer-based CLIP model as the backbone, in line with recent FSAS approaches [7, 16, 20].

In training, we introduce a novel self-supervised learning strategy that eliminates the need for extensive pixel-labeled training samples. Given a raw image $\mathbf{X} \in \mathbb{R}^{h \times w \times 3}$ from a seen class C^s, we generate k reference images $\{\mathbf{X}_n^i\}_{i=1}^k$ by applying data transformations (e.g., random rotation, flipping) to simulate normal variations. Simultaneously, a query image $\mathbf{X}_q$ is created by synthesizing anomalies on $\mathbf{X}$ to mimic anomalous scenarios. A pretrained image encoder extracts multi-layer patch-level features from

both $\{\mathbf{X}_n^i\}_{i=1}^k$ and $\mathbf{X}_q$, denoted as $\mathbf{F}_n^l \in \mathbb{R}^{kHW\times C}$ and $\mathbf{F}_q^l \in \mathbb{R}^{HW\times C}$, where $l = 1, 2, \ldots, L$, with H, W, and C denoting the height, width, and feature dimension, respectively. These extracted features are utilized in three key components of our model: 1) **Dictionary Construction**: The reference image features $\mathbf{F}_n^l$ are used to construct a dictionary for retrieval; 2) **Dictionary Lookup**: The ***Query Feature*** $\mathbf{F}_q^l$ serve as queries for retrieval, producing the ***Retrieved Result*** $\mathbf{F}_r^l$; 3) **Query Discrimination Regularization**: Two regularization terms are added into the loss function to jointly optimize the model based on $(\mathbf{F}_r^l, \mathbf{F}_q^l)$, improving the discrimination between normal and anomalous patterns.

During inference, the model constructs a dictionary using features $\mathbf{F}_n^l$ extracted from a few normal reference images of unseen classes. Given a query image $\mathbf{X}_q$ from the same unseen classes, its features $\mathbf{F}_q^l$ are compared with the ***Retrieved Result*** $\mathbf{F}_r^l$. The computed distance guides the generation of the final anomaly map $\mathbf{M}$.

3.2. Dictionary Construction

A well-constructed dictionary typically consists of two components: an index and the corresponding content, referred to as the *Dictionary Key* and *Dictionary Value*, respectively. Motivated by this, we design a Key Generator g_K and a Value Generator g_V to transform the extracted normal reference image features $\mathbf{F}_n^l$ into structured dictionary representations. Meanwhile, the Query Generator g_Q processes the ***Query Feature*** $\mathbf{F}_q^l$ to obtain a *Dictionary Query*, which is then matched against the *Dictionary Key* for retrieval. The feature transformations are defined as follows:

$$\mathbf{F}_Q^l = g_Q(\mathbf{F}_q^l) = AttnBlock_Q(\mathbf{F}_q^l) \quad (1)$$

$$\mathbf{F}_K^l = g_K(\mathbf{F}_n^l) = AttnBlock_K(\mathbf{F}_n^l) \quad (2)$$

$$\mathbf{F}_V^l = g_V(\mathbf{F}_n^l) = \mathbf{F}_n^l + AttnBlock_V(\mathbf{F}_n^l) \quad (3)$$

where $\mathbf{F}_Q^l \in \mathbb{R}^{HW\times C}$, $\mathbf{F}_K^l, \mathbf{F}_V^l \in \mathbb{R}^{kHW\times C}$ are *Dictionary Query, Key and Value*, respectively. Here, $AttnBlock$ represents a self-attention-based transformer block designed for feature adaptation. The additional residual connection in g_V helps preserve the fine-grained details of $\mathbf{F}_n^l$, enhancing feature fidelity in the constructed dictionary.

Design of *AttnBlock*. To enable the dictionary representations to capture meaningful global relationships, we employ a self-attention mechanism within each $AttnBlock$. Given an input feature $\mathbf{F}_{in} \in \{\mathbf{F}_q^l, \mathbf{F}_n^l\}$, we first apply linear projections to generate the query, key, and value matrices: $\mathbf{Q} = \phi_Q(\mathbf{F}_{in}), \mathbf{K} = \phi_K(\mathbf{F}_{in}), \mathbf{V} = \phi_V(\mathbf{F}_{in})$. The transformed features are then passed through a multi-head self-attention module followed by a two-layer MLP:

$$\mathbf{F}_{out} = TwoLayerMLP(softmax(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{C}})\mathbf{V}) \quad (4)$$

By incorporating self-attention, each patch is able to perceive global contextual information, thereby improving the robustness of dictionary construction.

In the next subsection, we introduce the dictionary lookup process using individual patch features from $\mathbf{F}_q^l$. Specifically, we denote each patch feature as $\mathbf{x}_{q,p}^l \in \mathbb{R}^{1\times C}$, where $p = 1, 2, \cdots, HW$. The corresponding *Dictionary Query* in $\mathbf{F}_Q^l$ is represented as $\mathbf{x}_{Q,p}^l$. For simplicity, we omit the subscript p in the following discussion.

3.3. Dictionary Lookup

Intuitively, if a query image patch contains an anomaly, its features $\mathbf{x}_q^l$ cannot be well retrieved from the *Dictionary Value* $\mathbf{F}_V^l$. Based on this assumption, we propose three dictionary lookup strategies, each consisting of two main steps:

1) *Query-Key Matching*: The similarity $\mathbf{z}$ between the *Dictionary Query* $\mathbf{x}_Q^l$ and the *Dictionary Key* $\mathbf{F}_K^l$ is computed as: $\mathbf{z} = \mathbf{x}_Q^l {\mathbf{F}_K^l}^{\mathrm{T}}$.

2) *Weighted Fusion*: Using the computed similarity $\mathbf{z}$, the Retrieved Result $\mathbf{x}_r^l \in \mathbb{R}^{1\times C}$ is obtained by weighted fusion of the *Dictionary Value* $\mathbf{F}_V^l$: $\mathbf{x}_r^l = \hat{\mathbf{w}}\mathbf{F}_V^l$, where $\hat{\mathbf{w}} \in \mathbb{R}^{1\times kHW}$ is a weight vector derived from $\mathbf{z}$, and is determined by one of the following strategies:

$$\hat{\mathbf{w}} = \begin{cases} onehot(argmax(\mathbf{z})) & if \text{ Maximun Lookup}, \\ softmax(\mathbf{z}) & if \text{ Dense Lookup}, \\ SPM(\mathbf{z}) & if \text{ Sparse Lookup} \end{cases} \quad (5)$$

Specifically, for Maximum Lookup, the weight vector $\hat{\mathbf{w}}$ is a one-hot vector, selecting the *Dictionary Value* with the highest similarity score. For Dense Lookup, the weight vector $\hat{\mathbf{w}}$ is obtained by applying the softmax function to $\mathbf{z}$, distributing the weights across all *Dictionary Value* in a dense manner. For Sparse Lookup, we introduce a Sparse Probability Module (SPM), adapted from the approach proposed in [23], to sparsify $\mathbf{z}$ such that it emphasizes the most relevant *Dictionary Value* and reduces background redundancy as the number of reference images increases. We adopt Sparse Lookup as the default setting in this work.

Sparse Probability Module. The sparse fusion weight $\hat{\mathbf{w}}$ is computed by solving the optimization problem that sparsifies the similarity scores $\mathbf{z}$ and satisfies the probability simplex constraint:

$$\arg\min_{\triangle} \frac{1}{2}\|\mathbf{w} - \mathbf{z}\|^2, \triangle = \{\mathbf{w} | \mathbf{w}_u \geq 0, \sum_{u=1}^{kHW} \mathbf{w}_u = 1\} \quad (6)$$

where $\triangle$ is the probability simplex constraint. Solving the optimization problem yields $\hat{\mathbf{w}}_u = \max(\mathbf{z}_u - \tau, 0)$, where $\hat{\mathbf{w}}_u$ denotes the u-th element of $\hat{\mathbf{w}}$ and τ is a dynamic threshold determined by Algorithm 1. By repeating this process for all query patches, we obtain the final ***Retrieved Result*** $\mathbf{F}_r^l$ under the sparse lookup strategy.

Algorithm 1 The acquisition of the adaptive threshold τ

1: Sort $\mathbf{z}$ in descending order: $\mathbf{z}_1 \geq \mathbf{z}_2 \geq \cdots \mathbf{z}_{kHW}$
2: Compute the cumulative sum: $Cum_t = \sum_{u=1}^{t} \mathbf{z}_t$
3: Compute the candidate threshold: $\tau_t = (Cum_t - 1)/t$
4: Find the largest t (denoted as t^*) satisfying $\mathbf{z}_t > \tau_t$, then the final threshold is $\tau = \tau_{t^*}$

Query Loss. How can the model acquire a class-generalizable dictionary lookup ability? Our answer is to use the ***Query Feature*** $\mathbf{F}_q^l$ itself as a pseudo-label for self-supervised training. The core assumption is that normal patch features from $\mathbf{F}_q^l$ can always be retrieved from the constructed dictionary. To this end, we minimize the average distance $\mathbb{E}_{\mathcal{N}}[d]$ between the ***Query Feature*** $\mathbf{F}_q^l$ and the ***Retrieved Result*** $\mathbf{F}_r^l$ in normal regions. The query loss is defined as:

$$\mathcal{L}_q = \sum_l \mathbb{E}_{\mathcal{N}}[d] = \sum_l \Big(\frac{1}{|\mathcal{N}|} \sum_{j \in \mathcal{N}} d(\mathbf{F}_{q,j}^l, \mathbf{F}_{r,j}^l) \Big) \quad (7)$$

where $d(\mathbf{F}_{q,j}^l, \mathbf{F}_{r,j}^l) = 1 - \langle \mathbf{F}_{q,j}^l, \mathbf{F}_{r,j}^l \rangle$ denotes the cosine distance between the jth patch, and $\langle \cdot, \cdot \rangle$ is the cosine similarity. The anomaly mask $\mathbf{G} \in \{0,1\}^{HW}$ is obtained using the anomaly synthesis algorithm in DRÆM [39]. Here, $\mathcal{N} = \{j | \mathbf{G}_j = 0\}$ represents the index set of normal patches, and $|\mathcal{N}|$ is its cardinality.

3.4. Query Discrimination Regularization

The query loss $\mathcal{L}_q$ enables the model learn how to retrieve from the dictionary and detect anomalies by measuring the distance between query features and their retrieved counterparts. However, this strong retrieval capability is a double-edged sword. If the model becomes excessively powerful in both feature extraction and retrieval, it may find combinations of normal features in the dictionary that closely matches any query, including anomalous ones. This weakens segmentation, as the distance can no longer clearly separate normal from abnormal regions. To mitigate this issue, we introduce two constraints:

1) **Contrastive Query Constraint** enforces that the average distance between abnormal patches from $\mathbf{F}_q^l$ and their retrieved results from $\mathbf{F}_r^l$ is greater than that of normal patches, enhancing anomaly separability:

$$\mathcal{L}_{CQC} = \sum_l \max(0,\ \mathbb{E}_{\mathcal{N}}[d] - \mathbb{E}_{\mathcal{A}}[d]) \quad (8)$$

where $\mathbb{E}_{\mathcal{A}}[d]$ represents the average cosine distance between the ***Query Feature*** and ***Retrieved Result*** in abnormal regions, similar to $\mathbb{E}_{\mathcal{N}}[d]$ in Equation (7).

2) **Text Alignment Constraint** leverages CLIP's text-image matching capability to enforce alignment between global retrieved results and the embedding space of normal text descriptions, preventing the retrieval of anomalous features from the dictionary.

Inspired by WinCLIP [16], we adopt a two-class text prompt design to ensure effective CLIP-based alignment. Specifically, we construct a series of prompt templates and perform prompt ensembling by averaging the corresponding text embeddings. For example, the template "a photo of a [state] [class]", where [state] indicates an adjective describing normal or anomalous conditions (e.g., good / damaged), and [class] denotes the object category (e.g., wood, bottle). The resulting text embedding is denoted as $\mathbf{F}_{text} \in \mathbb{R}^{2 \times C}$. To obtain the global representation of the ***Query Feature***, $\mathbf{F}_q^l \in \mathbb{R}^{H \times W \times C}$ from different layers are concatenated along the channel dimension and global average pooling is applied over the spatial dimensions to obtain a compact representation. A linear layer is finally used to map the pooled features into $\mathbf{x}_q$ in the joint embedding space. The same process is applied to the ***Retrieved Result*** $\mathbf{F}_r^l$ to yield the global feature $\mathbf{x}_r$. The final constraint is formulated as:

$$\mathcal{L}_{TAC} = CE(\widetilde{\mathbf{x}}_r \widetilde{\mathbf{F}}_{text}^{\mathrm{T}}, 0) + CE(\widetilde{\mathbf{x}}_q \widetilde{\mathbf{F}}_{text}^{\mathrm{T}}, y_q) \quad (9)$$

where $CE(\cdot)$ represents the cross-entropy loss function [19] and $y_q \in \{0,1\}$ is the image-level label derived from the synthesized query image. $\widetilde{(\cdot)}$ denotes the L2-normalized version along the embedding dimension. Details of the text prompt design are provided in Appendix A.3.

3.5. Training and Inference

Training. We train DictAS in a self-supervised manner, allowing any image $\mathbf{X}$ without pixel annotations as auxiliary data. The total loss function is defined as:

$$\mathcal{L} = \mathcal{L}_q + \lambda_1 \mathcal{L}_{CQC} + \lambda_2 \mathcal{L}_{TAC} \quad (10)$$

where λ_1 and λ_2 are weighting coefficients for the regularization terms. Note that the number of simulated reference images is set to $k = 1$ for training efficiency.

Inference. During this stage, the dictionary is constructed using $k (k \geq 1)$ real normal reference images. The cosine distance between the query features and their retrieved counterparts is computed to generate the anomaly map. The final anomaly map $\mathbf{M} \in \mathbb{R}^{h \times w}$ is obtained by aggregating the maps from L layers:

$$\mathbf{M} = Up\Big(\frac{1}{2L} \sum_{l=1}^{L} \Big(1 - \langle \mathbf{F}_q^l, \mathbf{F}_r^l \rangle \Big) \Big) \quad (11)$$

where $Up(\cdot)$ represents the upsampling operation used to restore the original input resolution.

4. Experiments

4.1. Experimental Settings

Datasets. To evaluate the class-generalizable FSAS performance, experiments on seven real-world datasets from

Table 1. Comparison with existing state-of-the-art methods under the 4-shot setting. The best results are highlighted in red, and the second-best results are marked in blue. The symbol † denotes methods based on CLIP, and (a,b,c) represents pixel-level (AUROC, PRO, AP). To ensure a fair comparison, all methods use the same normal reference images, and all CLIP-based methods employ the same backbone (ViT-L-14-336) and input resolution (336×336).

Datasets	RegAD [15] (ECCV 22)	AnomalyGPT [13] (AAAI 24)	FastRecon [11] (ICCV 23)	† FastRecon+ [11] (ICCV 23)	† WinCLIP [16] (CVPR 23)	† APRIL-GAN [7] (CVPR 23)	† PromptAD [20] (CVPR 24)	† DictAS (Ours)
Industrial Datasets (AUROC, PRO, AP)								
MVTecAD [3]	(95.7, 86.0, 46.5)	(96.4, 91.2, 52.9)	(95.9, 79.9, 47.0)	(96.3, 92.2, 53.9)	(92.4, 83.8, 39.2)	(92.2, 86.6, 46.6)	(96.0, 92.4, 57.5)	(98.6, 95.1, 66.8)
VisA [43]	(94.7, 72.8, 21.4)	(96.5, 65.4, 20.8)	(96.0, 77.7, 31.1)	(97.0, 86.2, 32.5)	(96.0, 86.5, 25.7)	(96.2, 86.6, 30.6)	(97.9, 89.5, 37.5)	(98.8, 91.9, 41.8)
MVTec3D [4]	(96.9, 89.2, 13.3)	(96.6, 87.4, 27.8)	(95.6, 83.6, 12.9)	(97.1, 91.8, 39.2)	(96.6, 87.9, 24.0)	(96.4, 89.1, 33.1)	(97.7, 92.1, 36.9)	(98.4, 94.9, 44.2)
MPDD [17]	(94.9, 83.3, 16.4)	(97.7, 93.2, 40.8)	(97.0, 87.5, 25.7)	(97.4, 93.1, 37.8)	(97.0, 90.7, 29.3)	(95.3, 86.9, 31.4)	(97.3, 94.0, 40.5)	(98.4, 95.8, 42.9)
BTAD [25]	(97.3, 75.5, 44.1)	(96.2, 73.5, 50.6)	(88.7, 62.1, 35.5)	(97.4, 80.8, 62.2)	(90.3, 64.7, 28.5)	(93.3, 74.6, 50.9)	(96.6, 80.1, 62.5)	(98.0, 83.3, 66.8)
Average	(95.9, 81.3, 28.3)	(96.7, 82.1, 38.6)	(94.6, 78.2, 30.4)	(97.0, 88.8, 45.1)	(94.5, 82.7, 29.3)	(94.7, 84.8, 38.5)	(97.1, 89.6, 47.0)	(98.4, 92.2, 52.5)
Medical Datasets (AUROC, PRO, AP)								
RESC [14]	(87.9, 60.0, 18.1)	(86.7, 60.0, 28.5)	(91.7, 71.7, 30.3)	(95.8, 82.8, 68.5)	(93.1, 75.7, 38.4)	(93.7, 77.6, 57.3)	(96.8, 86.8, 71.3)	(97.5, 89.7, 74.9)
BrasTS [24]	(93.8, 70.2, 24.8)	(95.4, 73.6, 41.8)	(92.5, 63.8, 24.0)	(96.1, 73.8, 43.9)	(93.3, 64.0, 33.4)	(91.3, 63.0, 40.0)	(96.6, 77.0, 54.4)	(97.3, 77.2, 59.3)
Average	(90.8, 65.1, 21.5)	(91.0, 66.8, 35.2)	(92.1, 67.8, 27.1)	(96.0, 78.3, 56.2)	(93.2, 69.8, 35.9)	(92.5, 70.3, 48.7)	(96.7, 82.2, 62.9)	(97.4, 83.4, 67.1)

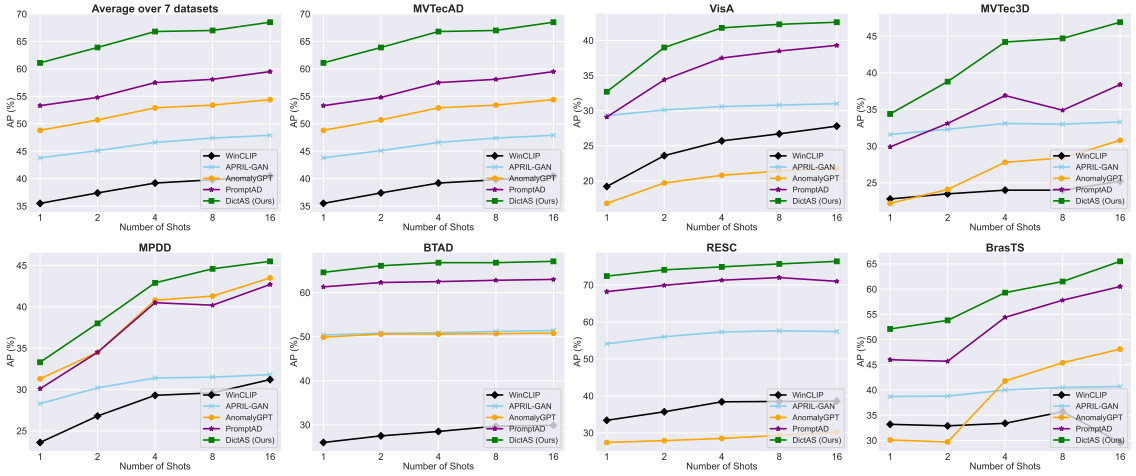


Figure 3. Comparison with four representative VLM-based methods under different numbers of shots. The pixel-level AP on seven datasets is reported. DictAS demonstrates consistent improvements over the compared methods for all shots. As the number of reference images increases, DictAS shows a greater advantage, unlike some methods that exhibit instability or even a decline (e.g., WinCLIP in BrasTS).

the industrial and medical domain are conducted. Specifically, five commonly used industrial anomaly segmentation datasets are adopted: MVTecAD [3], VisA [43], MVTec3D [4], MPDD [17], and BTAD [25]. Additionally, two medical datasets are included: BraTS [24] for brain tumor segmentation and RESC [14] for retinal lesion detection. For MVTec3D, only RGB images in the dataset are adopted. Following [7, 13, 38], we utilize all normal images from the industrial dataset VisA [43] as auxiliary data for self-supervised training and directly conduct few-normal-shot testing on remaining industrial and medical datasets. To evaluate VisA itself, MVTecAD [3] is adopted for auxiliary training. Notably, since a self-supervised paradigm is employed, any image without pixel-level annotations can serve as auxiliary training data, including those from natural scenes. In Appendix C.2, we further analyze the impact of different auxiliary training sets through ablation studies.

Evaluation Metrics. Since this paper primarily focuses on anomaly segmentation, we report three pixel-level metrics in the main text: Area Under the Receiver Operating Characteristic (AUROC), Per-Region Overlap (PRO), and the Average Precision (AP). As a supplement, we also report anomaly classification metrics in Appendix E.1, where the classification score for each image is obtained following the same strategy as APRIL-GAN [7]. We present the performance for different values of the few-shot normal samples, with k set to 1, 2, 4, 8, and 16.

Implementation Details. Similar to recent state-of-the-art FSAS methods [7, 16, 20], we adopt the CLIP model (ViT-L-14-336), pretrained by OpenAI [30], as the default backbone in DictAS. All input images are uniformly resized to 336×336 before being fed into the model. During training, we extract the 6th, 12th, 18th, and 24th layers from the frozen image encoder as patch-level features, following APRIL-GAN [7]. This multi-level feature selection balances low-level appearance and high-level semantics, and facilitates a fair comparison with prior work. The balancing coefficients λ_1 and λ_2 for the regularization loss are both

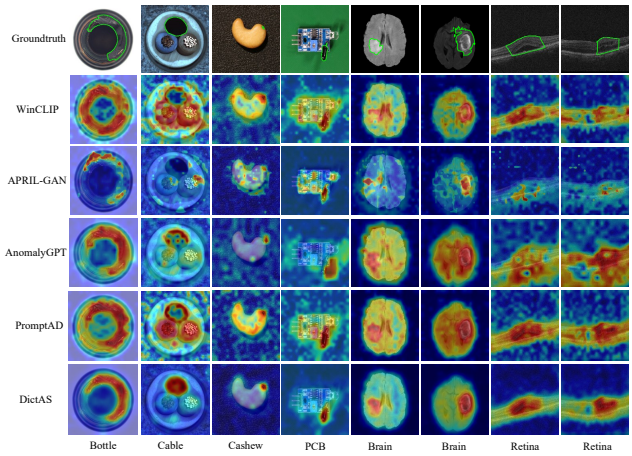


Figure 4. Qualitative comparison of anomaly segmentation results across different FSAS methods.

Table 2. The comparison of model efficiency on MVTecAD under the 4-shot setting (mean±std for AP).

Method	Backbone	Resolution	GPU Cost (GB)	Time (ms)	AP (%)
RegAD [38]	ResNet18	224 × 224	7.1	5790.1	46.5±1.2
AnomalyGPT [13]	ImageBind-Huge	224 × 224	19.1	1555.2	52.9±0.5
FastRecon [11]	Wide-ResNet50	336 × 336	**1.0**	110.9	47.0±0.7
FastRecon+ [11]	ViT-L-14-336	336 × 336	3.4	150.5	53.9±0.6
WinCLIP [16]	ViT-L-14-336	336 × 336	15.8	8227.5	39.2±1.1
APRIL-GAN [7]	ViT-L-14-336	336 × 336	3.8	227.6	46.6±0.5
PromptAD [20]	ViT-L-14-336	336 × 336	2.1	81.3	57.5±0.6
DictAS	ViT-L-14-336	336 × 336	3.4	**73.5**	**66.8±0.4**

set to 0.1 by default. During the auxiliary training stage, two types of data transformations—Geometric Transformations (e.g., Random Rotation) and Occlusion Augmentations (e.g., Random GridDropout)—are applied to the raw images to generate reference images. We use the Adam optimizer [18] to train DictAS for 30 epochs, with an initial learning rate of 0.0001 and a batch size of 24. All experiments were conducted on a single NVIDIA 3090 with 24GB of GPU memory. More details can be found in Appendix A.

4.2. Comparison with State-of-the-art methods

Competing Methods. In this study, we compare our DictAS with six state-of-the-art (SOTA) methods: RegAD [15], AnomalyGPT [13], FastRecon [11], WinCLIP [16], APRIL-GAN [7], and PromptAD [20]. Among these, PromptAD follows a class-dependent FSAS setting, while the other methods adopt a class-generalizable setting. To ensure a fair comparison, all methods were evaluated using the same k normal reference images and repeated five times. Additionally, we introduce a variant of FastRecon that incorporates CLIP image encoder as the feature extractor, referred to as FastRecon+. All CLIP-based methods [7, 16, 20] utilize the same backbone and input image resolution as our DictAS.

Quantitative Comparison. Table 1 presents the quantitative results of ZSAS across different datasets under the 4-shot setting in both industrial and medical domains. DictAS consistently outperforms existing methods, achieving state-of-the-art FSAS performance. Compared to the suboptimal method, DictAS improves AUROC, PRO, and AP by 1.3%, 2.6% and 5.5%, respectively, on industrial datasets, and by 0.7%, 1.2%, 4.2% on medical datasets. Notably, most of the suboptimal results are achieved on PromptAD, which is a class-dependent method that requires fine-tuning a new model for each unseen class, limiting its scalability. In contrast, our DictAS removes this dependency, enabling class-generalizable FSAS in a unified model, which is more flexible and efficient. In addition, DictAS outperforms other class-generalizable methods (e.g., WinCLIP, APRIL-GAN) by a large margin across all metrics. This is because it adopts a self-supervised training paradigm to learn cross-category dictionary lookup capabilities, rather than relying on prior knowledge from previous real anomaly samples or pretraining. Figure 3 further illustrates the variations in AP across seven datasets under different numbers of shots. It can be observed that DictAS demonstrates consistent improvements over the compared methods for all shots. This improvement is attributed to our sparse lookup strategy, which helps reduce feature redundancy at higher shots.

Qualitative Comparison. In Figure 4, we present visualization results from industrial and medical datasets under the 4-shot setting. Overall, our DictAS achieves more precise and complete anomaly localization, attributed to the global lookup capability learned during self-supervised training via the designed *Dictionary Key/Query/Value Generator*.

Efficiency Comparison. Table 2 compares the efficiency of different methods, including the average inference time per image and the maximum GPU memory usage per image during inference. The experimental results show that the proposed DictAS achieves the fastest inference speed and the best FSAS performance. Although few-shot methods are susceptible to variations in reference images, experimental results show that DictAS exhibits greater stability with lower standard deviation compared to other methods.

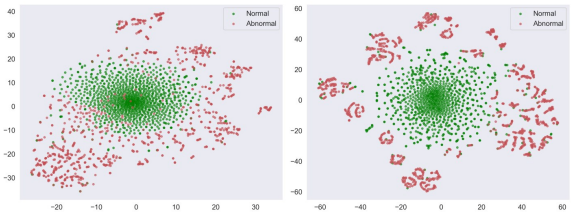
4.3. Ablation

Unless otherwise specified, all ablation experiments in this subsection are conducted on the MVTecAD [3] dataset under the 4-shot setting.

The effects of components. Ablation studies on both modules and loss functions are conducted as shown in Table 3. In the module ablation, it can be observed that when any generator is removed and the original image features are employed instead, the performance of DictAS drops to varying degrees. This is mainly due to the fact that the

Table 3. Ablation on different components (%).

		AUROC	PRO	AP
Module Ablation	w/o *Query Generator*	97.5	94.2	63.5
	w/o *Key Generator*	97.9	94.5	63.8
	w/o *Value Generator*	98.0	94.6	64.2
Loss Ablation	w/o $\mathcal{L}_{CQC}$	97.4	94.1	64.6
	w/o $\mathcal{L}_{TAC}$	98.0	94.6	65.0
	w/o $\mathcal{L}_{CQC}$ and $\mathcal{L}_{TAC}$	97.1	93.5	63.7
	DictAS	**98.6**	**95.1**	**66.8**



(a) Without Query Discrimination Regularization (b) With Query Discrimination Regularization

Figure 5. Feature t-SNE visualization. (a) / (b) represent the cases without / with Query Discrimination Regularization, respectively.

Table 4. Ablation on dictionary lookup strategies. The pixel level AP (%) is employed to evaluate under different shots.

Lookup Strategy	1-shot	2-shot	4-shot	8-shot	16-shot
Maximum Lookup	52.2	56.5	59.1	59.7	60.6
Dense Lookup	60.2	62.9	63.7	63.6	63.8
Sparse Lookup	**61.1**	**63.9**	**66.8**	**67.0**	**68.5**

designed *AttnBlock* in the generator provides global attention for dictionary lookup, which facilitates sparse matching of *Query-Key* pairs and weighted fusion of the *Dictionary Value*. For the loss function, we performed ablation on the two regularization constraints and found that they jointly influence the final FSAS performance. Compared to the constraint $\mathcal{L}_{CQC}$, which affects the global feature space, constraint $\mathcal{L}_{TAC}$ enhances the discriminability of anomalous regions in the fine-grained feature space, resulting in greater performance gains (e.g. 2.2% ↑AP vs. 1.8% ↑AP).

To further visualize the impact of the proposed Query Discrimination Regularization on the results, we randomly selected 40 images from the category *cable* of MVTecAD and performed t-SNE visualization on the feature map $(\mathbf{F}_r - \mathbf{F}_q)$, which is obtained by taking the residual between the ***Query Feature*** and ***Retrieved Result*** from the same layer. As shown in Figure 5, without the Query Discrimination Regularization (w/o $\mathcal{L}_{CQC}$ and $\mathcal{L}_{TAC}$), the boundary between normal and anomalous regions in the residual features remains ambiguous. However, when the regularization is applied, the residual features become more distinguishable between normal and anomalous regions, thereby facilitating easier anomaly discrimination.

The effects of dictionary lookup strategies. Table 4 presents an evaluation of various dictionary lookup strategies. The experimental results indicate that the proposed Sparse Lookup achieves the highest AP across all shot settings. Moreover, compared to other strategies, Sparse Lookup demonstrates greater advantages when the number of reference images increases (e.g., in the 8-shot and 16-shot settings). This is primarily because it maintains query sparsity while utilizing a larger set of Dictionary Values, effectively mitigating feature redundancy in the dictionary.

Table 5. Ablation on backbone and input resolution (%).

Backbone	Resolution	AUROC	PRO	AP	Time (ms)
ViT-B-16-224	336×336	98.1	93.3	64.8	43.3
ViT-L-14-224	336×336	98.3	94.3	66.2	73.5
ViT-L-14-336	336×336	98.6	95.1	66.8	73.5
ViT-L-14-336	420×420	98.6	95.4	67.6	130.2
ViT-L-14-336	518×518	**98.7**	**95.6**	**68.7**	235.6

The effects of backbone and resolution. Table 5 evaluates the impact of different input resolutions and CLIP backbones on DictAS. Despite the significant differences among the pre-trained backbones (top three rows), the anomaly segmentation performance of DictAS remains relatively stable, especially in terms of AUROC. This indicates that our method is not highly dependent on the original visual representations. The reason is that self-supervised training on the auxiliary dataset equips the model with category-agnostic dictionary lookup capabilities, thereby reducing reliance on original features. Additionally, with the increase in input resolution, we observe a significant improvement in AP, but the inference time per image also increases considerably. To strike a balance, we adopt a default setting of 336^2 resolution with the ViT-L-14-336 backbone.

5. Conclusion

In this paper, we introduce DictAS, a novel framework for class-generalizable few-shot anomaly segmentation (FSAS). Inspired by human inspectors, we reformulate FSAS as a dictionary lookup task. An anomaly is detected when the query feature cannot be retrieved from the dictionary constructed from normal reference image features. Through self-supervised training on seen classes from an auxiliary dataset, DictAS learns a transferable dictionary lookup ability, enabling it to generalize effectively to unseen classes in FSAS. To further enhance anomaly discrimination, we introduce query discrimination regularization, which is jointly optimized with the query loss to make anomalous features less retrievable from the dictionary. The final anomaly map is computed based on the cosine distance between the query feature and its retrieval result. Extensive experiments on seven industrial and medical datasets demonstrate that DictAS surpasses SOTA methods in FSAS performance while also achieving the fastest inference speed with comparable stability.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under Grant Nos. 62373350 and 62371179; in part by the Youth Innovation Promotion Association CAS (2023145); in part by the Beijing Nova Program 20240484687; in part by the Beijing Municipal Natural Science Foundation (China) 4252053; in part by the Longmen Laboratory "*Research and Development Project of General-Purpose AI Platform Software Based on Industrial Quality Inspection Large Model*".

References

[1] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, 42:1–13, 2018. 1

[2] Niamh Belton, Misgina Tsighe Hagos, Aonghus Lawlor, and Kathleen M Curran. Fewsome: One-class few shot anomaly detection with siamese networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2977–2986, 2023. 1, 2

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019. 6, 7, 11, 14, 16, 19

[4] Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization. *arXiv preprint arXiv:2112.09045*, 2021. 6, 11, 14, 19

[5] Yunkang Cao, Jiangning Zhang, Luca Frittoli, Yuqi Cheng, Weiming Shen, and Giacomo Boracchi. Adaclip: Adapting clip with hybrid learnable prompts for zero-shot anomaly detection. In *European Conference on Computer Vision*, pages 55–72. Springer, 2024. 1

[6] Qiyu Chen, Huiyuan Luo, Haiming Yao, Wei Luo, Zhen Qu, Chengkan Lv, and Zhengtao Zhang. Center-aware residual anomaly synthesis for multiclass industrial anomaly detection. *IEEE Transactions on Industrial Informatics*, pages 1–11, 2025. 1

[7] Xuhai Chen, Yue Han, and Jiangning Zhang. April-gan: A zero-/few-shot anomaly classification and segmentation method for cvpr 2023 vand workshop challenge tracks 1&2: 1st place on zero-shot ad and 4th place on few-shot ad. *arXiv preprint arXiv:2305.17382*, 2023. 1, 2, 3, 6, 7, 13, 16, 17, 18, 19

[8] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *CoRR*, abs/2005.02357, 2020. 3

[9] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International conference on pattern recognition*, pages 475–489. Springer, 2021. 3

[10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 15, 16

[11] Zheng Fang, Xiaoyang Wang, Haocheng Li, Jiejie Liu, Qiugui Hu, and Jimin Xiao. Fastrecon: Few-shot industrial anomaly detection via fast feature reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17481–17490, 2023. 1, 6, 7, 14, 18, 19

[12] Bin-Bin Gao. Metauas: Universal anomaly segmentation with one-prompt meta-learning. *Advances in Neural Information Processing Systems*, 37:39812–39836, 2025. 3, 14

[13] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. Anomalygpt: Detecting industrial anomalies using large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1932–1940, 2024. 1, 2, 3, 6, 7, 14, 16, 18, 19

[14] Junjie Hu, Yuanyuan Chen, and Zhang Yi. Automated segmentation of macular edema in oct using deep neural networks. *Medical image analysis*, 55:216–227, 2019. 6, 11, 19

[15] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. Registration based few-shot anomaly detection. In *European Conference on Computer Vision*, pages 303–319. Springer, 2022. 1, 3, 6, 7, 14, 18, 19

[16] 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. 1, 2, 3, 5, 6, 7, 13, 18, 19

[17] Stepan Jezek, Martin Jonak, Radim Burget, Pavel Dvorak, and Milos Skotak. Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions. In *2021 13th International congress on ultra modern telecommunications and control systems and workshops (ICUMT)*, pages 66–71. IEEE, 2021. 6, 11, 19

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7, 13

[19] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 5

[20] 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. 1, 2, 3, 6, 7, 13, 14, 18, 19

[21] Jingyi Liao, Xun Xu, Manh Cuong Nguyen, Adam Goodge, and Chuan Sheng Foo. Coft-ad: Contrastive fine-tuning for few-shot anomaly detection. *arXiv preprint arXiv:2402.18998*, 2024. 1, 2

[22] Dwarikanath Mahapatra, Behzad Bozorgtabar, and Zongyuan Ge. Medical image classification using generalized zero shot learning. In *Proceedings of the IEEE/CVF*

international conference on computer vision, pages 3344–3353, 2021. 1

[23] Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR, 2016. 4

[24] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014. 1, 6, 11, 14, 15, 16, 19

[25] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. Vt-adl: A vision transformer network for image anomaly detection and localization. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, pages 01–06. IEEE, 2021. 6, 11, 14, 16, 19

[26] Shichen Qu, Xian Tao, Xinyi Gong, Zhen Qu, Mukesh Prasad, Fei Shen, Zhengtao Zhang, and Guiguang Ding. Lscad: A large-small model collaboration framework for unsupervised industrial anomaly detection. *IEEE Transactions on Instrumentation and Measurement*, 2025. 1

[27] Zhen Qu, Xian Tao, Fei Shen, Zhengtao Zhang, and Tao Li. Investigating shift equivalence of convolutional neural networks in industrial defect segmentation. *IEEE Transactions on Instrumentation and Measurement*, 72:1–17, 2023. 1

[28] Zhen Qu, Xian Tao, Mukesh Prasad, Fei Shen, Zhengtao Zhang, Xinyi Gong, and Guiguang Ding. Vcp-clip: A visual context prompting model for zero-shot anomaly segmentation. In *European Conference on Computer Vision*, pages 301–317. Springer, 2024. 1

[29] Zhen Qu, Xian Tao, Xinyi Gong, Shichen Qu, Qiyu Chen, Zhengtao Zhang, Xingang Wang, and Guiguang Ding. Bayesian prompt flow learning for zero-shot anomaly detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30398–30408, 2025. 1

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 1, 6, 13

[31] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14318–14328, 2022. 3

[32] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1907–1916, 2021. 2

[33] Eli Schwartz, Assaf Arbelle, Leonid Karlinsky, Sivan Harary, Florian Scheidegger, Sivan Doveh, and Raja Giryes. Maeday: Mae for few-and zero-shot anomaly-detection. *Computer Vision and Image Understanding*, page 103958, 2024. 1, 2

[34] Shelly Sheynin, Sagie Benaim, and Lior Wolf. A hierarchical transformation-discriminating generative model for few shot anomaly detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8495–8504, 2021. 1, 2

[35] Jhih-Ciang Wu, Ding-Jie Chen, Chiou-Shann Fuh, and Tyng-Luh Liu. Learning unsupervised metaformer for anomaly detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4369–4378, 2021. 3

[36] Jhih-Ciang Wu, Ding-Jie Chen, Chiou-Shann Fuh, and Tyng-Luh Liu. Learning unsupervised metaformer for anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4369–4378, 2021. 1

[37] Guoyang Xie, Jinbao Wang, Jiaqi Liu, Feng Zheng, and Yaochu Jin. Pushing the limits of fewshot anomaly detection in industry vision: Graphcore. *arXiv preprint arXiv:2301.12082*, 2023. 3

[38] Xincheng Yao, Zixin Chen, Chao Gao, Guangtao Zhai, and Chongyang Zhang. Resad: A simple framework for class generalizable anomaly detection. *Advances in Neural Information Processing Systems*, 37:125287–125311, 2025. 3, 6, 7, 14, 16

[39] 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. 5, 11, 12

[40] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 15, 16

[41] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. AnomalyCLIP: Object-agnostic prompt learning for zero-shot anomaly detection. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 13

[42] Jiawen Zhu and Guansong Pang. Toward generalist anomaly detection via in-context residual learning with few-shot sample prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17826–17836, 2024. 3

[43] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pretraining for anomaly detection and segmentation. In *European Conference on Computer Vision*, pages 392–408. Springer, 2022. 6, 11, 14, 15, 16, 19

Appendix for DictAS: A Framework for Class-Generalizable Few-Shot Anomaly Segmentation via Dictionary Lookup

This appendix includes the following five parts: 1) More experimental details (e.g. datasets, self-supervised training) in Section A; 2) Detailed description of SOTA methods and comparison with contemporaneous approaches (e.g., MetaUAS, ResAD) in Section B; 3) Additional ablation studies (e.g., hyperparameters, auxiliary datasets, data transformations) in Section C; 4) Limitations of our methods in Section D; 5) Presentation of more detailed quantitative and qualitative results of few-shot anomaly classification / segmentation in Section E.

A. Experimental Details

A.1. Details of the Datasets

Table 6. Key statistics of the datasets. (a, b) in the training/testing sets denotes the number of normal and abnormal samples, respectively. $|\mathcal{C}|$ is the number of categories. Note that anomaly segmentation datasets have only normal images in the training set.

Domain	Dataset	Category	Modality	$\lvert\mathcal{C}\rvert$	Testing Set	Training Set	Usage
Industrial	MVTecAD [3]	Obj &texture	Photography	15	(467, 1258)	(3629, 0)	Industrial defect detection
	VisA [43]	Obj	Photography	12	(962, 1200)	(8659, 0)	Industrial defect detection
	MVTec3D [4]	Obj	Photography+Depth	10	(249, 948)	(2656, 0)	Industrial defect detection
	MPDD [17]	Obj	Photography	6	(176, 282)	(888, 0)	Industrial defect detection
	BTAD [25]	Obj	Photography	3	(451, 290)	(1799, 0)	Industrial defect detection
Medical	RESC [14]	Retina	Photography	1	(1041, 764)	(4297, 0)	Retinal Lesion Detection
	BrasTS [24]	Brain	Radiology(MRI)	1	(828, 1948)	(4211, 0)	Brain Tumor Segmentation

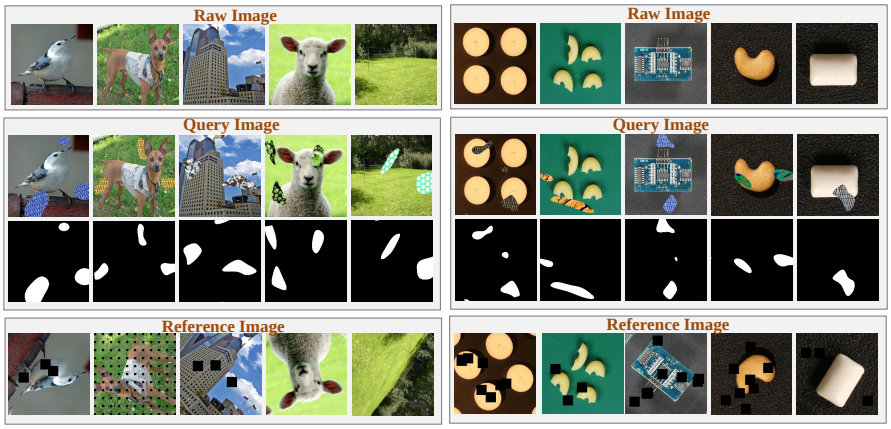
In this study, we conduct extensive experiments on 7 public datasets covering industrial and medical domains to assess the effectiveness of our methods, including MVTecAD [3], VisA [43], MVTec3D [4], MPDD [17], BTAD [25], RESC [14] and BrasTS [24]. The key statistics for these datasets are demonstrated in Table 6. In this study, normal reference images are randomly selected from the training set, and all samples from the testing set are used to evaluate the model's performance. By default, all samples in the VisA training set are treated as seen classes for self-supervised training and are subsequently tested on other datasets. For VisA itself, the training set in MVTeAD is used as an auxiliary training dataset.

A.2. Details of Self-Supervised Training

This subsection further elaborates on the online construction of auxiliary data for self-supervised training.

In the self-supervised training stage,both query and reference images are dynamically constructed from raw images belonging to any seen class. Note that this process is conducted online. Specifically, given a raw image $\mathbf{X}$, we apply random transformations (e.g., random rotation) to generate a corresponding reference image, simulating the few normal reference images $\mathbf{X}_n$ available in the real anomaly segmentation process. In DictAS, we by default use Geometric Transformations and Occlusion Transformations as shown in Figure 6. Detailed descriptions and parameters for each transformation type are provided in Listing 1. Additional ablation studies investigating the effect of different transformation types can be found in Section C.

For the query image $\mathbf{X}_q$, it is derived from the raw image using the anomaly synthesis algorithm proposed in DRAEM [39]. The detailed strategy for synthesizing the query image during self-supervised training is described in Algorithm A. Alongside the synthesized image, the pixel-level pseudo-label $\mathbf{G}$ and the image-level pseudo-label y_q are also generated using the Berlin noise mask. These pseudo-labels are used to compute the query contrastive loss and the text alignment loss, both of which act as regularization terms during self-supervised training.



(a) Acquisition of auxiliary training data from natural images

(b) Acquisition of auxiliary training data from industrial images

Figure 6. **Acquisition of the auxiliary training data.** Given a raw image without pixel-level annotations, the query image is generated using an anomaly synthesis algorithm [39], while the normal reference image is obtained via data transformations (e.g., random rotation). Both natural images shown in (a) and industrial images shown in (b) can be utilized as sources to construct auxiliary training data.

```python
import albumentations as A
import cv2

img_trans_for_reference = A.Compose([
        A.RandomRotate90(p = 1),
        A.Rotate(limit=[30, 270], p=1.0),
        A.HorizontalFlip(p=0.5),
        A.VerticalFlip(p=0.5),
        A.GridDropout(ratio=0.3, p=0.5),
        A.CoarseDropout(max_holes=8, max_height=32, max_width=32, p=0.5),
        ], is_check_shapes=False)
X_raw = cv2.imread("raw_img_path")   # Read the raw image
# Perform data transformation on the raw image to simulate the reference image.
X_reference = img_trans_for_reference(img = X_raw)
```

Listing 1. Data transformation for generating the reference image in the self-supervised training stage.

Algorithm 2 Anomaly synthesis strategy for generating the query image in the self-supervised training stage.

Input: Raw image $\mathbf{X}$; Anomaly source image $\mathbf{A}$; Perlin noise generator P; Image size H and W; Noise resolution r_x and r_y; Blending parameter γ; Binarization threshold λ

Output: Query image $\mathbf{X}_q$, pixel-level pseudo-label $\mathbf{G}$, image-level pseudo-label y_q.

while True **do**
  $\mathbf{G} \leftarrow \text{where}(P(H, w, r_x, r_y) > \lambda)$
  $\mathbf{M}_A \leftarrow G \times \mathbf{X}$
  $\overline{\mathbf{M}}_A \leftarrow 1 - \mathbf{M}_A$
  $\mathbf{X}_q \leftarrow \gamma(\mathbf{M}_A \odot \mathbf{A}) + (1-\gamma)(\mathbf{M}_A \odot \mathbf{X}) + \overline{\mathbf{M}}_A \odot \mathbf{X}$
end while
if SUM($\mathbf{G}$) is 0 **then**
  $y_q \leftarrow 0$
else
  $y_q \leftarrow 1$
end if
return $\mathbf{X}_q$, $\mathbf{G}$, y_q

A.3. Details of Text Prompt Design

In this work, two types of text prompts (normal descriptions and anomaly descriptions) are fed into the text encoder of CLIP to generate text embeddings. The global image representation obtained from the Retrieved Result $\mathbf{F}_r^l$ is constrained to align with the normal text embedding space, thereby enhancing anomaly discrimination capability. Since the design of text prompts is not the focus of this study, we directly follow the design principles of WinCLIP [16] (i.e. text prompt ensemble). Specifically, to obtain normal text embeddings, the object category name (e.g., bottle) and state are inserted into predefined prompt templates to generate multiple semantically similar normal prompts. These prompts are encoded by the text encoder, and the resulting embeddings are averaged to form the final normal text representation. Similarly, the abnormal text embeddings are constructed in the same manner by replacing the state with an anomalous one. The details of the prompt template and the settings of normal/abnormal [state] are illustrated in Figure 7.

State-level (normal)	**State-level (abnormal)**
"flawless [class]"	"damaged [class]"
"perfect [class]"	"broken [class]"
"unblemished [class]"	"abnormal [class]"
"normal [class]"	"imperfect [class]"
"[class] without flaw"	"[class] with flaw"
"[class] without defect"	"[class] with defect"
"[class] without damage"	"[class] with damage"

Template-level	
"a photo of a/the [state][class]."	"a cropped photo of a/the [state][class]."
"a good photo of the [state][class]."	"a bright photo of a/the [state][class]."
"a photo of my [state][class]."	"a dark photo of a/the [state][class]."
"a photo of the [state][class]."	"a black and white photo of a/the [state][class]."
"a photo of a/the [state][class]."	"a jpeg corrupted photo of a/the [state][class]."
"a photo of a/the small [state][class]."	"a close-up photo of the [state][class]."
"a bad photo of a/the [state][class]."	"There is a/the [state][class] in the scene."
"a low resolution photo of a/the [state][class]."	"This is one [state][class] in the scene."

Figure 7. Detailed design of prompt template and normal/abnormal [state] words for text prompt ensemble.

A.4. Details of Implementation

Similar to recent state-of-the-art FSAS methods [7, 16, 20], we adopt the CLIP model (ViT-L-14-336), pretrained by OpenAI [30], as the default backbone for our DictAS. All input images are uniformly resized to 336×336 before being fed into the model. During training, we extract the 6th, 12th, 18th, and 24th layers from the frozen image encoder as patch-level features similar to [7]. To increase the receptive field, average pooling with a kernel size of 3 is applied to the patch-level features extracted from the CLIP image encoder. The regularization loss balancing coefficients, λ_1 and λ_2, are both set to 0.1 by default. During the auxiliary training phase, two types of data transformations—Geometric Transformations (e.g., Random Rotation) and Occlusion Augmentations (e.g., Random GridDropout)—are applied to the raw images to generate reference images. For computational efficiency, the number of reference images is set to $k = 1$ during training. During inference, $k \geq 1$ normal reference images are used as visual prompts. To ensure a fair comparison, all methods are evaluated using the same k normal reference images. Each experiment is repeated five times with different random seeds. DictAS is trained for 30 epochs using the Adam optimizer [18], with an initial learning rate of 0.0001 and a batch size of 24. All experiments are conducted on a single NVIDIA RTX 3090 GPU with 24 GB of memory.

B. State-of-the-art Methods

B.1. Method Introduction and Comparison Details

- **WinCLIP** [16] is one of the earliest works based on CLIP for the zero/few shot anomaly segmentation task. Since the vanilla CLIP [30] 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 language-guided anomaly segmentation results are derived by harmoniously aggregating the classification outcomes of sub-images corresponding to the same spatial locations. To leverage the few normal reference images, it also employs memory bank-based nearest neighbor retrieval to obtain visually guided anomaly maps. For a fair comparison, we report the results using ViT-L-14-336 as the backbone with an input resolution of 336×336, based on the reproduced code from [41].
- **APRIL-GAN** [7] 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 being trained on real anomalous samples with pixel-level, it can directly generalize to unseen classes. A memory-bank strategy like WinCLIP [16] is also adopted to enhance text-image alignment results. For a fair comparison, we retrained the model using the official code on ViT-L-14-336 with a resolution of 336×336 and re-evaluated it across all industrial and medical datasets.

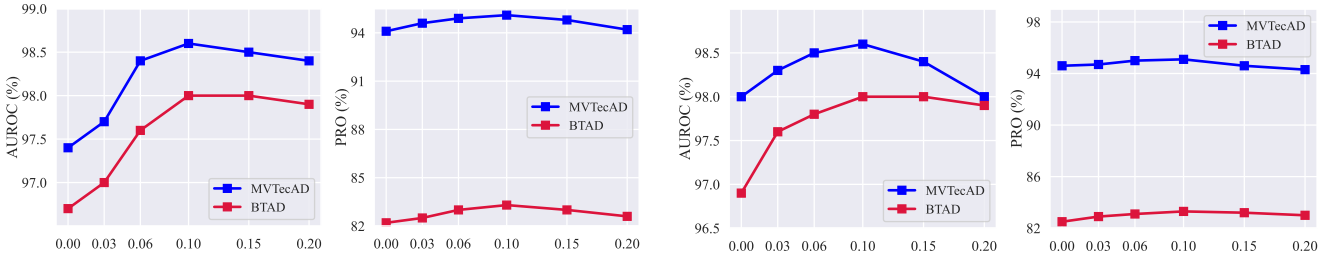
- **RegAD** [15] first proposed a feature registration strategy using a spatial transformer network for class-generalizable FSAS. With a meta-learning training approach, it demonstrates strong generalization to unseen classes. However, its performance on unseen category objects heavily depends on extensive augmentation of normal reference images and utilizes distribution estimation to generate the final anomaly map, making it less memory-efficient. In this work, we retrained RegAD using the same auxiliary dataset as our DictAS, i.e., trained on all classes of the VisA training set and tested on other datasets. For evaluation on VisA, the weights were obtained using MVTecAD as the auxiliary training set. Since RegAD has a specific backbone-dependent network structure, the backbone and resolution from the original paper were adopted (ResNet-18, 224×224).
- **Fastrecon** [11] models class-generalizable FSAS as a feature reconstruction problem based on linear regression. By designing a distribution regularization term and solving the analytical solution, it demonstrates excellent cross-category generalization in a training-free manner. However, as the number of reference images increases, the linear model may theoretically overfit arbitrary features, which means that Fastrecon still faces the challenge of over-reconstruction. In this version, we used the official code and backbone (wide-resnet50, 336×336) from the original paper and tested it across all datasets.
- **Fastrecon+** [11] is a reimplementation of Fastrecon, utilizing the CLIP image encoder as the feature extractor. For a fair comparison, ViT-L-14-336 with a resolution of 336×336 is adopted. Following their original paper, we extracted the two intermediate patch-level features (the 12th and 18th layers) and concatenated them along the embedding dimension to construct new features. The other experimental hyperparameters are set to be the same as those in the original paper.
- **AnomalyGPT** [13] is a class-generalizable FSAS method that integrates a large language model for anomaly segmentation and supports multi-turn dialogues with users. It employs supervised training using synthetic anomaly data, allowing the model to generalize to new classes. We conducted experiments using the official code and evaluated the model's FSAS performance in the same way as our DictAS. To use the officially pre-trained weights, the original backbone and input image resolution were adopted (ImageBind-Huge, 224×224).
- **PromptAD** [20] is a class-dependent FSAS method, which is different from other CLIP-based approaches. It directly trains on normal reference images for each class and evaluates on the test set of the same object category. Moreover, it proposes a one-class prompt learning method for few-shot anomaly segmentation. Although it outperforms most FSAS methods, the need for fine-tuning on each category limits its practicality in scenarios involving data privacy or rapidly changing environments. For fairness in comparison, we retrained the model on ViT-L-14-336 using an input resolution of 336×336.
- **MetaUAS** [12] proposes viewing FSAS as a segmentation change problem. By leveraging meta-learning training on a synthetic dataset, it enables the acquisition of a universal model capable of detecting anomalies in unseen classes. However, it is only applicable to situations where a single normal sample is used as the visual prompt (i.e., 1-shot). In this paper, we use it as a concurrent method and compare it with our DictAS.
- **ResAD** [38] proposes using learned residual feature distributions to reduce feature variations across different classes for class-generalizable FSAS. It ultimately transforms the anomaly segmentation problem into an out-of-distribution detection problem using a Feature Distribution Estimator, achieving strong performance on unseen classes. In this paper, we employ it as a concurrent method and compare its performance with our DictAS.

B.2. Comparison with Concurrent Methods

Table 7. Comparison with the concurrent state-of-the-art methods. The pixel-level AUROC (%) is reported, and the best results are highlighted in **bold**. The experimental results of MetaUAS and ResAD are taken from their original papers.

		Backbone	MVTecAD [3]	VisA [43]	BTAD [25]	MVTec3D [4]	BrasTS [24]
1-shot	MetaUAS [12]	EfficientNet-b4	94.6	92.2	—	—	—
	DictAS	ViT-B-16	97.1	97.3	—	—	—
	DictAS	ViT-L-14-336	**97.7**	**98.0**	**97.4**	**97.5**	**96.5**
2-shot	ResAD [38]	ImageBind-Huge	95.6	95.1	96.4	97.5	94.3
	DictAS	ViT-L-14-336	**98.2**	**98.5**	**97.9**	**97.9**	**96.4**
4-shot	ResAD [38]	ImageBind-Huge	96.9	97.5	96.8	97.9	96.1
	DictAS	ViT-L-14-336	**98.6**	**98.8**	**98.0**	**98.4**	**97.3**

Table 7 compares our DictAS with two contemporary state-of-the-art methods, MetaAUS [12] and ResAD [38]. As our method currently applies only to transformer-based architectures, we selected the CLIP pre-trained backbones with



(a) λ_1 (b) λ_2

Figure 8. (a) Ablation study on the weight coefficient λ_1 of the query contrastive constraint. (b) Ablation study on the weight coefficient λ_2 of the text alignment constraint. The experiments are conducted on the MVTecAD and BTAD datasets under the 4-shot setting and the metric pixel-level AUROC and PRO are reported.

Table 8. Ablation on different auxiliary datasets under 4-shot setting (%).

Auxiliary Dataset	MVTecAD			BTAD		
	AUROC	PRO	AP	AUROC	PRO	AP
VisA [43]	**98.6**	**95.1**	**66.8**	98.0	83.3	66.8
BrasTS [24]	98.3	95.0	66.2	97.9	83.0	66.2
Ade20K [40]	98.4	**95.1**	66.5	98.1	**83.5**	66.8
VOC2012 [10]	98.3	94.9	66.4	**98.2**	83.4	**66.9**

Table 9. Ablation on the scale of auxiliary dataset VisA under 4-shot setting (%).

Scale	MVTecAD			BTAD		
	AUROC	PRO	AP	AUROC	PRO	AP
15%	96.8	92.0	62.8	96.1	80.2	63.3
35%	97.3	92.9	63.5	96.6	81.7	63.8
55%	98.0	93.5	64.6	97.1	82.0	64.2
75%	98.3	94.6	66.0	97.6	82.9	66.5
95%	98.5	95.1	66.6	98.0	83.2	66.7
100%	**98.6**	**95.1**	**66.8**	**98.0**	**83.3**	**66.8**

the smallest (ViT-B-16) and largest (ViT-L-14-336) parameter counts for comparison. The experimental results show that, among the reported results, our DictAS achieves state-of-the-art performance in FSAS. Notably, despite using fewer backbone parameters than ResAD (which adopts ImageBird-Huge), our ViT-L-14-336-based DictAS performs better, highlighting its effectiveness.

C. Additional Ablations

C.1. Ablation on Hyperparameters

In this subsection, we conduct an ablation study on the weighting coefficients λ_1 and λ_2, which correspond to the two query discriminative regularization terms in our method. As shown in Figure 8, the model achieves optimal performance when both hyperparameters are set to approximately 0.1. As the weighting coefficients gradually increase to the equilibrium point (0.1), both AUROC and AP exhibit an upward trend. Beyond this point, the model's performance on unseen classes begins to gradually decline.

Reason Analysis. Before analyzing the reasons, it is crucial to clarify the pseudo-labels used during the training process of DictAS under the self-supervised learning paradigm. The main loss, i.e., the query loss, is computed using all normal patches in the query image, where the query image feature itself serves as the pseudo-label. In contrast, the two query discriminative regularization terms use the synthesized mask $\mathbf{G}$ as the pseudo-label, which indicates the location of the synthesized anomaly within the query image.From this perspective, the query loss enables the model to acquire a category-agnostic dictionary querying capability, thereby facilitating generalization to unseen categories. Meanwhile, the two regularization losses leverage the synthetic anomaly information to enhance anomaly discriminability, making the boundary between normal and anomalous regions more distinguishable. Therefore, moderate regularization (e.g., $\lambda_1 = \lambda_2 = 0.1$) proves beneficial in the early training stages, as it improves the model's ability to distinguish anomalies without overwhelming the dictionary querying mechanism. However, as the influence of the regularization losses increases, the model's reliance on the dictionary-based querying diminishes. This shift causes the model to focus more on discriminating the synthesized anomalies during training, leading to a loss of generalization capability.

Table 10. The details of different types of data transformations .

Type	Transformation	Parameters
Geo. Trans.	RandomRotate90	$p = 1.0$
	Rotate	$30° \sim 270°, p = 1.0$
	HorizontalFlip	$p = 0.5$
	VerticalFlip	$p = 0.5$
Color Trans.	RandomBrightnessContrast	$p = 0.5$
	HueSaturationValue	$hue = 20, sat = 30, val = 20, p = 0.5$
Noise Dist.	GaussNoise	$var = 10.0 \sim 50.0, p = 0.5$
	MotionBlur	$blur_limit = 5, p = 0.5$
Occl. Aug.	GridDropout	$ratio = 0.3, p = 0.5$
	CoarseDropout	$max_holes = 8, max_size = 32 \times 32, p = 0.5$

Table 11. Ablation on different types of data transformations.

Geo. Trans.	Color Trans.	Noise Dist.	Occl. Aug.	AUROC	PRO	AP
✔	✘	✘	✘	98.3	94.9	66.0
✘	✔	✘	✘	98.2	94.7	64.7
✘	✘	✔	✘	98.1	94.7	64.8
✘	✘	✘	✔	98.2	94.8	64.9
✘	✔	✔	✘	97.9	94.1	64.0
✔	✘	✘	✔	**98.6**	**95.1**	**66.8**
✔	✔	✘	✘	98.2	94.7	64.6
✔	✘	✔	✘	98.2	94.6	64.5
✔	✔	✔	✔	98.3	94.8	65.5

C.2. Ablation on Auxiliary Datasets

As mentioned above, our DictAS by default uses the industrial dataset VisA [43] as an auxiliary dataset for self-supervised training and then directly performs few-shot anomaly segmentation on unseen classes in other datasets. This setup is designed to follow the settings of existing methods for a fairer comparison [7, 13, 38]. Can our method use a more general dataset for auxiliary training? If so, how does the scale of the auxiliary data affect the model's FSAS performance? We will address these two questions in the following discussion.

Domain of Auxiliary Datasets. In Table 8, we investigate the impact of using auxiliary datasets from different domains for self-supervised training and evaluate their 4-shot performance on MVTecAD [3] and BTAD [25]. Specifically, the VisA dataset [43] from the industrial domain, the BrasTS dataset [24] from the medical domain, and the Ade20K [40] and VOC2012 [10] datasets from natural scenes are used as auxiliary datasets. For the natural scene datasets Ade20K [40] and VOC2012 [10], we randomly select samples identical to those in the VisA training set for auxiliary training. Since each natural image may contain multiple categories, we use *object* to replace *[class]* in the text prompts. Note that our auxiliary datasets do not require pixel-level annotations. Experimental results show that our DictAS is not sensitive to the auxiliary datasets and demonstrates strong robustness across industrial, medical, and natural scene domains. It is attributed to the use of the self-supervised learning paradigm, which demonstrates that DictAS has learned a generalizable dictionary lookup capability and successfully transferred this ability to the class-generalizable FSAS task.

Scale of Auxiliary Datasets. In Table 9, we evaluate the impact of the auxiliary dataset size on model performance. Specifically, (15%, 35%, 55%, 75%, 95%) of the VisA training set samples are randomly selected for self-supervised training, and the FSAS performance on MVTecAD and BTAD is assessed under the 4-shot setting. The experimental results show that as the dataset size increases, the FSAS performance of the proposed DictAS also improves. Even when trained on only

half or less of the auxiliary data, the proposed DictAS already achieves satisfactory results, highlighting the efficiency of our training strategy. Moreover, DictAS demonstrates promising potential with larger-scale training data, which will be explored in our future work.

C.3. Ablation on Types of Data Transformations

In this subsection, we conduct an ablation study on the types of data transformations used to generate reference images in the self-supervised training process.

Specifically, we predefined four types of data transformations: Geometric Transformations (Geo. Trans.), Color Transformations (Color Trans.), Noise Disturbance (Noise Dist.), and Occlusion Augmentation (Occl. Aug). The details and hyperparameters of different types of data transformations are presented in Table 10. To investigate the impact of different transformation types on the experiment, we conducted an ablation study on these four types of transformations, as shown in Table 11. It can be observed that when only a single transformation type is used, Geometric Transformations provide the greatest performance gain for FSAS, especially in terms of pixel-level AP (66.0%). This is because applying geometric transformations to raw images, such as random rotation and random flipping, simulates the most significant variations among normal reference images in real-world anomaly segmentation. During training, self-supervised learning enables the model to capture the correspondence between query and reference images under geometric transformations, which helps DictAS enhance its robustness to different reference images. Furthermore, among different transformation combinations, the combination of Geometric Transformations and Occlusion Augmentation achieved the best results, with scores of 98.6% in AUROC, 95.1% in PRO and 66.8% in AP. We attribute this to the occlusion simulating missing parts in real scenarios, further enhancing the robustness of the dictionary lookup.

Considering the model's performance, this work defaults to using Geometric Transformations and Occlusion Augmentation as the data transformation methods.

D. Limitations

Our DictAS has demonstrated the state-of-the-art ZSAD performance in seven industrial and medical datasets. However, it still faces several limitations in practical applications: 1) Our method aims to learn the dictionary lookup ability of human inspectors when encountering unseen classes. While this enables generalization to novel categories, the dictionary lookup task imposes a limitation, requiring a few normal reference images to construct the dictionary, making it unsuitable for zero-shot tasks; 2) This work does not investigate the impact of larger-scale auxiliary datasets on the model's FSAS performance. However, ablation studies on the VisA dataset suggest that DictAS has the potential to leverage large-scale datasets (even at an internet scale) for self-supervised training, enabling continuous performance improvement. In the future, we will further enhance the FSAS capability of DictAS by incorporating human prior knowledge, while enabling zero-shot generalization. Moreover, larger-scale auxiliary data will be leveraged to enhance the dictionary lookup capability of DictAS.

E. Detailed FSAS Results

In this section, we present a detailed comparison of different SOTA methods under the 1-, 2-, and 4-shot settings. As mentioned in the main text, since DictAS primarily focuses on anomaly segmentation, pixel-level AUROC, PRO, and AP are used as the default evaluation metrics. As a complement, this section also reports image-level AUROC, F1-Max, and AP to assess the performance of few-shot anomaly classification. The classification score for each image is obtained following the same strategy as APRIL-GAN [7].

E.1. Detailed few-shot anomaly classification results

Table 12. **Performance comparison of anomaly classification with other SOTA methods under the 1-shot setting**. The best results are highlighted in red, and the second-best results are marked in blue. The symbol † denotes methods based on CLIP, and (a,b,c) represents image-level (AUROC, F1-max, AP). To ensure a fair comparison, all methods use the same normal reference images, and all CLIP-based methods employ the same backbone (ViT-L-14-336) and input resolution (336×336).

Datasets	RegAD [15] (ECCV 22)	AnomalyGPT [13] (AAAI 24)	FastRecon [11] (ICCV 23)	† FastRecon+ [11] (ICCV 23)	† WinCLIP [16] (CVPR 23)	† APRIL-GAN [7] (CVPR 23)	† PromptAD [20] (CVPR 24)	† DictAS (Ours)
Industrial Datasets (AUROC, F1-Max, AP)								
MVTecAD	(73.3, 87.1, 87.2)	(92.8, 94.3, 96.1)	(83.7, 90.9, 91.6)	(92.0, 93.4, 95.6)	(92.6, 92.0, 96.1)	(91.1, 90.9, 95.6)	(93.0, 93.7, 96.6)	(96.1, 94.4, 98.3)
VisA	(69.3, 76.2, 72.2)	(86.4, 84.4, 87.4)	(80.1, 82.3, 83.1)	(81.0, 81.4, 82.3)	(84.8, 82.8, 87.0)	(87.1, 83.1, 90.5)	(85.2, 83.3, 86.8)	(89.5, 85.9, 91.0)
MVTec3D	(54.0, 88.4, 81.7)	(76.0, 90.3, 91.9)	(63.5, 89.6, 86.5)	(72.8, 90.8, 89.9)	(79.8, 90.3, 93.1)	(75.3, 89.8, 91.1)	(71.2, 90.1, 89.8)	(78.6, 91.1, 93.4)
MPDD	(47.9, 72.9, 61.5)	(72.4, 79.3, 75.9)	(62.2, 77.5, 67.9)	(76.5, 80.3, 76.1)	(79.9, 80.9, 82.5)	(75.1, 80.1, 80.8)	(79.3, 81.6, 83.5)	(81.3, 83.5, 82.6)
BTAD	(84.4, 78.2, 80.5)	(93.6, 89.7, 94.6)	(86.2, 77.7, 81.7)	(93.7, 92.0, 95.0)	(89.5, 81.8, 86.3)	(86.5, 84.0, 88.5)	(93.4, 90.4, 94.4)	(96.2, 92.8, 97.3)
Average	(65.8, 80.5, 76.6)	(84.2, 87.6, 89.2)	(75.1, 83.6, 82.1)	(83.2, 87.6, 87.8)	(85.3, 85.5, 89.2)	(83.0, 85.6, 89.3)	(84.4, 87.8, 90.2)	(88.3, 89.5, 92.5)
Medical Datasets (AUROC, F1-Max, AP)								
RESC	(55.9, 60.4, 46.6)	(86.8, 76.2, 83.4)	(76.8, 72.5, 59.4)	(82.8, 71.3, 80.4)	(57.4, 60.7, 48.1)	(77.3, 69.5, 69.7)	(87.4, 78.2, 84.3)	(89.9, 79.0, 89.6)
BrasTS	(58.4, 83.0, 73.2)	(73.1, 85.8, 82.0)	(61.8, 84.3, 75.7)	(76.2, 86.5, 85.1)	(86.6, 87.4, 92.5)	(86.8, 88.9, 92.5)	(81.7, 87.5, 88.5)	(85.8, 88.0, 92.9)
Average	(57.2, 71.7, 59.9)	(79.9, 81.0, 82.7)	(69.3, 78.4, 67.6)	(79.5, 78.9, 82.7)	(72.0, 74.0, 70.7)	(82.1, 79.2, 81.1)	(84.6, 82.8, 86.4)	(87.8, 83.5, 91.2)

Table 13. **Performance comparison of anomaly classification with other SOTA methods under the 2-shot setting**. The best results are highlighted in red, and the second-best results are marked in blue. The symbol † denotes methods based on CLIP, and (a,b,c) represents image-level (AUROC, F1-max, AP). To ensure a fair comparison, all methods use the same normal reference images, and all CLIP-based methods employ the same backbone (ViT-L-14-336) and input resolution (336×336).

Datasets	RegAD [15] (ECCV 22)	AnomalyGPT [13] (AAAI 24)	FastRecon [11] (ICCV 23)	† FastRecon+ [11] (ICCV 23)	† WinCLIP [16] (CVPR 23)	† APRIL-GAN [7] (CVPR 23)	† PromptAD [20] (CVPR 24)	† DictAS (Ours)
Industrial Datasets (AUROC, F1-Max, AP)								
MVTecAD	(76.6, 88.8, 88.9)	(94.4, 95.0, 97.0)	(88.9, 93.6, 94.7)	(94.2, 94.5, 96.5)	(93.8, 93.0, 96.6)	(90.1, 91.0, 95.5)	(95.4, 95.1, 97.7)	(97.4, 96.6, 98.9)
VisA	(70.4, 75.8, 73.6)	(87.2, 84.1, 88.8)	(84.6, 82.9, 86.7)	(81.1, 81.8, 81.3)	(83.5, 81.3, 85.9)	(86.6, 82.6, 90.4)	(85.1, 83.0, 87.0)	(90.2, 86.6, 91.3)
MVTec3D	(55.1, 88.5, 82.0)	(81.2, 91.5, 94.2)	(65.5, 89.6, 88.0)	(76.9, 91.2, 92.2)	(81.4, 90.5, 94.5)	(75.8, 90.0, 91.5)	(75.6, 90.8, 92.0)	(82.4, 91.2, 94.7)
MPDD	(52.5, 73.6, 62.2)	(79.7, 82.1, 81.1)	(67.0, 78.4, 70.6)	(81.8, 83.0, 81.4)	(81.5, 81.0, 83.3)	(75.1, 79.4, 80.2)	(83.3, 83.6, 88.2)	(84.9, 86.4, 85.4)
BTAD	(88.9, 89.2, 92.1)	(93.4, 89.9, 95.0)	(89.4, 83.2, 86.4)	(93.8, 90.3, 94.9)	(90.7, 84.2, 87.6)	(86.1, 84.2, 88.5)	(92.7, 89.0, 94.4)	(95.6, 92.3, 96.6)
Average	(68.7, 83.2, 79.8)	(87.2, 88.5, 91.2)	(79.1, 85.5, 85.3)	(85.6, 88.2, 89.3)	(86.2, 86.0, 89.6)	(82.7, 85.4, 89.2)	(86.4, 88.3, 91.9)	(90.1, 90.6, 93.4)
Medical Datasets (AUROC, F1-Max, AP)								
RESC	(59.4, 62.0, 48.0)	(87.8, 78.2, 83.5)	(77.6, 72.7, 60.6)	(87.6, 75.7, 84.9)	(60.3, 61.0, 50.6)	(78.3, 70.8, 71.0)	(89.2, 79.6, 85.9)	(91.6, 80.6, 90.9)
BrasTS	(57.4, 83.5, 72.0)	(74.9, 86.6, 83.3)	(65.4, 84.6, 77.4)	(75.8, 87.2, 83.6)	(87.0, 88.0, 93.4)	(87.5, 89.2, 93.0)	(83.0, 88.1, 89.3)	(85.5, 88.6, 92.4)
Average	(58.4, 72.8, 60.0)	(81.3, 82.4, 83.4)	(71.5, 78.7, 69.0)	(81.7, 81.5, 84.3)	(73.6, 74.5, 72.0)	(82.9, 80.0, 82.0)	(86.1, 83.8, 87.6)	(88.6, 84.6, 91.7)

Table 14. **Performance comparison of anomaly classification with other SOTA methods under the 4-shot setting**. The best results are highlighted in red, and the second-best results are marked in blue. The symbol † denotes methods based on CLIP, and (a,b,c) represents image-level (AUROC, F1-max, AP). To ensure a fair comparison, all methods use the same normal reference images, and all CLIP-based methods employ the same backbone (ViT-L-14-336) and input resolution (336×336).

Datasets	RegAD [15] (ECCV 22)	AnomalyGPT [13] (AAAI 24)	FastRecon [11] (ICCV 23)	† FastRecon+ [11] (ICCV 23)	† WinCLIP [16] (CVPR 23)	† APRIL-GAN [7] (CVPR 23)	† PromptAD [20] (CVPR 24)	† DictAS (Ours)
Industrial Datasets (AUROC, F1-Max, AP)								
MVTec-AD	(83.4, 89.8, 91.7)	(97.0, 95.9, 98.0)	(94.2, 90.9, 90.4)	(96.2, 95.3, 97.2)	(95.5, 94.0, 97.3)	(91.0, 91.6, 95.9)	(95.9, 95.2, 97.5)	(98.8, 98.2, 99.5)
VisA	(72.0, 77.1, 73.9)	(91.4, 87.2, 92.6)	(68.5, 77.1, 72.6)	(84.4, 82.6, 85.1)	(85.7, 82.8, 87.8)	(87.2, 83.3, 91.1)	(87.5, 83.9, 89.2)	(92.3, 88.5, 93.6)
MVTec3D	(57.7, 88.4, 84.1)	(83.4, 91.6, 95.1)	(57.9, 88.4, 83.7)	(81.4, 91.4, 93.7)	(81.3, 90.8, 94.3)	(76.4, 90.0, 91.8)	(79.5, 91.1, 93.5)	(84.5, 91.6, 95.3)
MPDD	(61.1, 75.9, 66.9)	(85.9, 88.5, 89.0)	(79.8, 78.5, 75.7)	(81.9, 82.8, 81.0)	(84.0, 83.1, 86.1)	(76.5, 80.7, 81.6)	(88.0, 87.6, 92.6)	(87.3, 87.8, 89.2)
BTAD	(91.3, 91.2, 94.5)	(93.5, 91.0, 95.9)	(68.3, 79.1, 75.1)	(94.4, 91.8, 96.2)	(91.7, 84.3, 88.0)	(86.1, 83.9, 88.4)	(92.6, 91.0, 94.4)	(96.5, 92.7, 97.2)
Average	(73.1, 84.5, 82.2)	(90.2, 90.8, 94.2)	(73.7, 82.8, 79.5)	(87.6, 88.8, 90.6)	(87.6, 87.0, 90.7)	(83.4, 85.9, 89.8)	(88.7, 89.8, 93.4)	(91.9, 91.8, 94.9)
Medical Datasets (AUROC, F1-Max, AP)								
RESC	(64.2, 63.7, 51.0)	(88.5, 78.8, 85.4)	(70.8, 65.6, 56.5)	(87.5, 76.4, 84.6)	(63.8, 62.6, 54.0)	(78.3, 70.7, 71.2)	(90.2, 81.0, 87.3)	(91.2, 79.6, 90.6)
BrasTS	(63.3, 83.9, 75.5)	(79.4, 86.2, 87.8)	(54.9, 82.6, 72.4)	(78.6, 87.6, 86.4)	(87.0, 88.0, 93.4)	(88.0, 89.1, 93.5)	(86.4, 88.2, 92.4)	(88.4, 88.9, 94.3)
Average	(63.7, 73.8, 63.2)	(83.9, 82.5, 86.6)	(62.9, 74.1, 64.4)	(83.0, 82.0, 85.5)	(75.4, 75.3, 73.7)	(83.2, 79.9, 82.4)	(88.3, 84.6, 89.8)	(89.8, 84.3, 92.5)

E.2. Detailed few-shot anomaly segmentation results

Table 15. **Performance comparison of anomaly segmentation with other SOTA methods under the 1-shot setting**. The best results are highlighted in red, and the second-best results are marked in blue. The symbol † denotes methods based on CLIP, and (a,b,c) represents pixel-level (AUROC, PRO, AP). To ensure a fair comparison, all methods use the same normal reference images, and all CLIP-based methods employ the same backbone (ViT-L-14-336) and input resolution (336×336).

Datasets	RegAD [15] (ECCV 22)	AnomalyGPT [13] (AAAI 24)	FastRecon [11] (ICCV 23)	† FastRecon+ [11] (ICCV 23)	† WinCLIP [16] (CVPR 23)	† APRIL-GAN [7] (CVPR 23)	† PromptAD [20] (CVPR 24)	† DictAS (Ours)
Industrial Datasets (AUROC, PRO, AP)								
MVTecAD	(92.3, 76.8, 36.1)	(95.3, 89.0, 48.8)	(93.9, 82.6, 48.2)	(95.1, 90.8, 50.5)	(91.6, 82.0, 35.5)	(91.2, 84.5, 43.8)	(95.2, 90.9, 53.3)	(97.7, 92.5, 61.1)
VisA	(93.3, 68.7, 17.9)	(87.4, 65.3, 16.8)	(96.5, 81.6, 31.8)	(96.1, 84.3, 26.0)	(95.3, 85.2, 19.2)	(95.9, 87.0, 29.3)	(97.2, 88.4, 29.1)	(98.0, 89.6, 32.7)
MVTec3D	(95.4, 84.5, 8.4)	(95.5, 84.3, 22.2)	(95.4, 83.6, 15.9)	(96.7, 89.3, 30.8)	(96.2, 86.4, 22.8)	(96.1, 88.4, 31.6)	(97.0, 89.7, 29.9)	(97.5, 92.1, 34.4)
MPDD	(93.2, 74.6, 8.4)	(96.6, 89.9, 31.3)	(95.5, 84.1, 20.9)	(96.2, 90.1, 30.7)	(95.6, 86.7, 23.6)	(94.9, 85.1, 28.3)	(96.0, 90.4, 30.1)	(97.4, 92.8, 33.3)
BTAD	(95.6, 68.9, 33.1)	(95.7, 71.7, 49.9)	(95.9, 67.9, 42.8)	(96.8, 80.6, 60.0)	(88.9, 61.7, 26.0)	(93.0, 73.4, 50.4)	(96.1, 79.4, 61.3)	(97.6, 82.1, 64.6)
Average	(94.0, 74.7, 20.8)	(94.1, 80.0, 33.8)	(95.4, 80.0, 31.9)	(96.2, 87.0, 39.6)	(93.5, 80.4, 25.4)	(94.2, 83.7, 36.7)	(96.3, 87.8, 40.8)	(97.6, 89.8, 45.2)
Medical Datasets (AUROC, PRO, AP)								
RESC	(84.6, 53.2, 14.6)	(86.0, 58.5, 27.4)	(93.0, 76.5, 31.7)	(96.0, 83.6, 66.5)	(92.3, 73.3, 33.4)	(93.0, 74.9, 54.1)	(96.4, 85.8, 68.2)	(97.2, 88.8, 72.4)
BrasTS	(91.3, 62.6, 17.5)	(94.2, 69.9, 30.1)	(93.4, 66.8, 25.8)	(95.4, 71.4, 39.0)	(93.1, 64.2, 33.2)	(90.9, 62.7, 38.7)	(95.9, 74.8, 46.0)	(96.5, 74.5, 52.1)
Average	(88.0, 57.9, 16.0)	(90.1, 64.2, 28.8)	(93.2, 71.6, 28.7)	(95.7, 77.5, 52.8)	(92.7, 68.7, 33.3)	(92.0, 68.8, 46.4)	(96.2, 80.3, 57.1)	(96.9, 81.6, 62.3)

Table 16. **Performance comparison of anomaly segmentation with other SOTA methods under the 2-shot setting**. The best results are highlighted in red, and the second-best results are marked in blue. The symbol † denotes methods based on CLIP, and (a,b,c) represents pixel-level (AUROC, PRO, AP). To ensure a fair comparison, all methods use the same normal reference images, and all CLIP-based methods employ the same backbone (ViT-L-14-336) and input resolution (336×336).

Datasets	RegAD [15] (ECCV 22)	AnomalyGPT [13] (AAAI 24)	FastRecon [11] (ICCV 23)	† FastRecon+ [11] (ICCV 23)	† WinCLIP [16] (CVPR 23)	† APRIL-GAN [7] (CVPR 23)	† PromptAD [20] (CVPR 24)	† DictAS (Ours)
Industrial Datasets (AUROC, PRO, AP)								
MVTecAD	(94.5, 82.7, 42.1)	(95.9, 90.2, 50.7)	(95.3, 85.8, 50.5)	(95.5, 91.5, 51.9)	(91.9, 82.7, 37.4)	(91.6, 85.5, 45.1)	(95.6, 91.5, 54.8)	(98.2, 94.2, 63.9)
VisA	(94.3, 70.2, 21.6)	(87.7, 65.0, 19.7)	(97.5, 83.9, 37.5)	(96.6, 85.2, 30.6)	(95.7, 85.9, 23.6)	(96.1, 86.8, 30.1)	(97.7, 89.4, 34.4)	(98.5, 91.1, 39.0)
MVTec3D	(95.9, 86.2, 10.0)	(95.8, 85.5, 24.1)	(95.8, 85.0, 16.9)	(96.8, 90.4, 35.5)	(96.4, 87.0, 23.5)	(96.3, 88.8, 32.3)	(97.2, 90.6, 33.1)	(97.9, 93.4, 38.8)
MPDD	(94.0, 79.3, 13.1)	(97.3, 91.8, 34.5)	(96.8, 89.1, 26.2)	(96.8, 92.7, 35.7)	(96.5, 89.4, 26.8)	(95.1, 86.6, 30.2)	(96.8, 92.6, 34.5)	(97.9, 94.6, 38.0)
BTAD	(96.9, 74.1, 42.3)	(96.0, 72.4, 50.6)	(96.4, 71.1, 45.1)	(97.2, 80.5, 61.6)	(89.6, 63.4, 27.5)	(93.2, 73.2, 50.8)	(96.4, 79.6, 62.3)	(97.9, 82.4, 66.1)
Average	(95.1, 78.5, 25.8)	(94.5, 81.0, 35.9)	(96.4, 83.0, 35.2)	(96.6, 88.1, 43.1)	(94.0, 81.7, 27.8)	(94.5, 84.2, 37.7)	(96.7, 88.7, 43.8)	(98.1, 91.1, 49.2)
Medical Datasets (AUROC, PRO, AP)								
RESC	(85.9, 54.5, 15.1)	(86.3, 59.0, 27.9)	(93.5, 75.6, 32.9)	(96.2, 84.7, 68.4)	(92.7, 74.6, 35.7)	(93.4, 76.5, 56.0)	(96.7, 86.6, 69.9)	(97.4, 89.6, 74.1)
BrasTS	(92.7, 66.0, 20.6)	(94.1, 70.2, 29.7)	(93.4, 67.3, 25.6)	(95.2, 71.7, 35.3)	(93.0, 63.6, 32.9)	(90.9, 63.1, 38.8)	(95.8, 75.2, 45.7)	(96.4, 73.8, 53.8)
Average	(89.3, 60.3, 17.9)	(90.2, 64.6, 28.8)	(93.4, 71.4, 29.2)	(95.7, 78.2, 51.9)	(92.8, 69.1, 34.3)	(92.2, 69.8, 47.4)	(96.3, 80.9, 57.8)	(96.9, 81.7, 62.0)

Table 17. **Performance comparison of anomaly segmentation with other SOTA methods under the 4-shot setting**. The best results are highlighted in red, and the second-best results are marked in blue. The symbol † denotes methods based on CLIP, and (a,b,c) represents pixel-level (AUROC, PRO, AP). To ensure a fair comparison, all methods use the same normal reference images, and all CLIP-based methods employ the same backbone (ViT-L-14-336) and input resolution (336×336).

Datasets	RegAD [15] (ECCV 22)	AnomalyGPT [13] (AAAI 24)	FastRecon [11] (ICCV 23)	† FastRecon+ [11] (ICCV 23)	† WinCLIP [16] (CVPR 23)	† APRIL-GAN [7] (CVPR 23)	† PromptAD [20] (CVPR 24)	† DictAS (Ours)
Industrial Datasets (AUROC, PRO, AP)								
MVTecAD [3]	(95.7, 86.0, 46.5)	(96.4, 91.2, 52.9)	(95.9, 79.9, 47.0)	(96.3, 92.2, 53.9)	(92.4, 83.8, 39.2)	(92.2, 86.6, 46.6)	(96.0, 92.4, 57.5)	(98.6, 95.1, 66.8)
VisA [43]	(94.7, 72.8, 21.4)	(96.5, 65.4, 20.8)	(96.0, 77.7, 31.1)	(97.0, 86.2, 32.5)	(96.0, 86.5, 25.7)	(96.2, 86.6, 30.6)	(97.9, 89.5, 37.5)	(98.8, 91.9, 41.8)
MVTec3D [4]	(96.9, 89.2, 13.3)	(96.6, 87.4, 27.8)	(95.6, 83.6, 12.9)	(97.1, 91.8, 39.2)	(96.6, 87.9, 24.0)	(96.4, 89.1, 33.1)	(97.7, 92.1, 36.9)	(98.4, 94.9, 44.2)
MPDD [17]	(94.9, 83.3, 16.4)	(97.7, 93.2, 40.8)	(97.0, 87.5, 25.7)	(97.4, 93.1, 37.8)	(97.0, 90.7, 29.3)	(95.3, 86.9, 31.4)	(97.3, 94.0, 40.5)	(98.4, 95.8, 42.9)
BTAD [25]	(97.3, 75.5, 44.1)	(96.2, 73.5, 50.6)	(88.7, 62.1, 35.5)	(97.4, 80.8, 62.2)	(90.3, 64.7, 28.5)	(93.3, 74.6, 50.9)	(96.6, 80.1, 62.5)	(98.0, 83.3, 66.8)
Average	(95.9, 81.3, 28.3)	(96.7, 82.1, 38.6)	(94.6, 78.2, 30.4)	(97.0, 88.8, 45.1)	(94.5, 82.7, 29.3)	(94.7, 84.8, 38.5)	(97.1, 89.6, 47.0)	(98.4, 92.2, 52.5)
Medical Datasets (AUROC, PRO, AP)								
RESC [14]	(87.9, 60.0, 18.1)	(86.7, 60.0, 28.5)	(91.7, 71.7, 30.3)	(95.8, 82.8, 68.5)	(93.1, 75.7, 38.4)	(93.7, 77.6, 57.3)	(96.8, 86.8, 71.3)	(97.5, 89.7, 74.9)
BrasTS [24]	(93.8, 70.2, 24.8)	(95.4, 73.6, 41.8)	(92.5, 63.8, 24.0)	(96.1, 73.8, 43.9)	(93.1, 64.0, 33.4)	(91.3, 63.0, 40.0)	(96.6, 77.0, 54.4)	(97.3, 77.2, 59.3)
Average	(90.8, 65.1, 21.5)	(91.0, 66.8, 35.2)	(92.1, 67.8, 27.1)	(96.0, 78.3, 56.2)	(93.1, 69.8, 35.9)	(92.5, 70.3, 48.7)	(96.7, 82.2, 62.9)	(97.4, 83.4, 67.1)

Table 18. Anomaly segmentation performance of our DictAS on **MVTecAD** for each object category. Pixel-level AUROC, PRO and AP are reported.

Object	1-shot			2-shot			4-shot		
	AUROC	PRO	AP	AUROC	PRO	AP	AUROC	PRO	AP
bottle	99.1±0.1	96.7±0.3	87.2±0.9	99.2±0.0	96.7±0.3	87.5±0.5	99.2±0.0	96.6±0.1	87.2±0.6
cable	97.8±0.3	90.1±0.6	66.7±2.5	98.7±0.4	93.7±1.1	76.5±4.2	98.9±0.3	94.8±0.9	78.7±2.1
capsule	97.7±0.2	93.1±0.9	37.3±9.4	98.5±0.3	95.6±1.2	43.0±9.2	98.6±0.3	95.6±0.8	45.0±4.4
carpet	99.7±0.0	98.4±0.1	85.1±0.3	99.7±0.0	98.5±0.1	85.1±0.3	99.7±0.0	98.4±0.0	85.4±0.3
grid	96.3±0.7	88.4±2.1	33.2±0.5	96.9±0.6	90.2±1.6	33.0±2.3	97.7±0.6	92.9±2.2	36.4±1.0
hazelnut	98.4±0.2	94.2±1.2	62.0±1.8	98.8±0.3	95.5±0.8	64.8±2.5	99.1±0.1	96.2±0.2	67.0±1.7
leather	99.6±0.0	98.8±0.1	57.9±0.4	99.6±0.0	98.8±0.1	57.5±0.5	99.6±0.0	98.7±0.1	58.5±1.0
metal_nut	95.8±0.9	93.3±1.0	74.7±4.2	96.0±0.6	94.3±1.3	75.3±3.2	97.5±0.1	96.3±0.3	82.5±1.0
pill	98.5±0.1	97.8±0.1	77.4±0.8	98.7±0.1	97.9±0.1	80.1±0.9	98.9±0.1	98.0±0.1	81.8±0.9
screw	98.3±0.6	92.0±1.7	30.7±0.7	98.6±0.7	93.5±2.4	26.2±0.2	99.1±0.8	94.7±3.1	37.8±1.0
tile	98.5±0.1	95.8±0.3	82.1±1.2	98.6±0.1	96.1±0.2	83.0±0.6	98.8±0.0	96.3±0.2	85.0±0.1
toothbrush	97.3±0.8	85.4±3.1	40.2±6.0	99.0±0.6	91.2±2.5	52.7±4.3	99.2±0.4	91.4±4.2	56.7±4.5
transistor	93.3±2.1	75.9±4.1	56.1±5.7	95.7±1.2	82.8±3.5	63.5±4.4	96.5±0.7	87.2±1.9	66.1±3.0
wood	97.1±0.1	94.5±0.2	70.9±0.3	97.3±0.1	94.5±0.1	71.8±0.3	97.4±0.1	94.5±0.2	72.5±0.7
zipper	97.8±0.0	93.9±0.1	55.2±0.3	98.0±0.2	94.4±0.5	58.0±0.7	98.3±0.1	95.0±0.3	60.8±1.2
Average	97.7±0.1	92.5±0.3	61.1±0.5	98.2±0.1	94.2±0.2	63.9±1.2	98.6±0.0	95.1±0.3	66.8±0.4

Table 19. Anomaly segmentation performance of our DictAS on **VisA** for each object category. Pixel-level AUROC, PRO and AP are reported.

Object	1-shot			2-shot			4-shot		
	AUROC	PRO	AP	AUROC	PRO	AP	AUROC	PRO	AP
candle	99.3±0.1	96.3±0.1	23.6±0.9	99.4±0.1	96.4±0.1	23.7±0.5	99.5±0.0	96.7±0.1	24.5±0.6
capsules	97.8±0.2	84.6±2.3	37.0±1.2	98.4±0.2	86.0±1.4	39.7±1.1	98.7±0.1	87.6±2.1	40.0±0.7
cashew	99.4±0.1	96.1±0.6	60.3±2.8	99.5±0.1	96.2±0.4	66.4±2.8	99.5±0.0	95.7±0.3	67.5±2.1
chewinggum	99.6±0.0	93.0±0.4	78.1±0.5	99.6±0.0	91.8±0.6	78.8±0.5	99.6±0.0	92.4±0.3	78.1±0.3
fryum	97.5±0.2	89.5±0.5	41.5±1.1	97.8±0.2	90.1±0.8	42.9±1.4	97.9±0.1	90.8±0.8	44.0±0.3
macaroni1	99.2±0.2	96.5±1.7	10.4±0.7	99.5±0.1	97.5±0.5	12.1±0.7	99.6±0.0	97.6±0.2	15.1±0.4
macaroni2	96.7±0.6	86.7±1.8	2.7±1.3	96.6±0.5	87.5±0.4	5.4±0.8	97.5±0.3	91.0±1.2	7.1±0.8
pcb1	98.4±0.5	91.3±3.7	43.3±4.6	99.4±0.1	93.7±1.8	73.3±4.0	99.6±0.1	94.9±1.3	81.1±3.8
pcb2	96.2±0.3	76.0±3.4	12.6±3.7	97.2±0.2	81.8±2.6	19.1±2.7	97.5±0.2	80.1±1.7	20.5±1.9
pcb3	95.4±0.4	79.5±3.0	13.6±1.4	97.1±0.4	85.7±2.0	25.9±1.4	97.9±0.1	87.8±2.0	30.8±1.1
pcb4	97.5±0.4	89.0±2.5	18.4±3.7	98.1±0.2	89.8±0.6	29.9±6.3	98.6±0.3	91.8±1.0	40.6±8.8
pipe_fryum	99.1±0.1	97.0±0.3	50.8±1.6	99.2±0.1	96.8±0.2	50.4±2.5	99.2±0.0	96.8±0.2	51.9±0.7
Average	98.0±0.1	89.6±0.7	32.7±0.9	98.5±0.1	91.1±0.4	39.0±2.0	98.8±0.1	91.9±0.3	41.8±1.7

Table 20. Anomaly segmentation performance of our DictAS on **MVTec3D** for each object category. Pixel-level AUROC, PRO and AP are reported.

Object	1-shot			2-shot			4-shot		
	AUROC	PRO	AP	AUROC	PRO	AP	AUROC	PRO	AP
cookie	98.7±0.1	94.5±0.2	60.3±2.8	98.9±0.1	95.2±0.3	65.0±1.3	99.1±0.1	96.1±0.4	69.1±1.1
dowel	98.1±0.2	92.4±0.7	24.1±2.9	98.4±0.3	94.0±1.2	24.5±3.0	99.1±0.3	96.0±1.2	32.7±6.1
cable_gland	96.1±0.7	88.1±1.7	11.8±2.2	97.2±0.5	91.6±1.5	16.4±5.3	99.1±0.6	97.4±1.5	32.6±3.7
rope	99.2±0.1	96.7±0.3	45.8±1.6	99.2±0.1	96.7±0.3	44.3±1.2	99.3±0.0	97.4±0.1	46.8±0.8
peach	98.5±0.5	94.5±1.8	26.6±15.5	99.1±0.5	96.6±1.8	39.5±16.6	99.6±0.0	98.5±0.2	56.0±1.7
potato	99.4±0.1	97.3±0.2	29.8±2.1	99.3±0.0	97.0±0.2	30.3±1.9	99.5±0.1	97.7±0.3	35.2±2.5
bagel	99.5±0.0	98.3±0.2	67.1±1.7	99.5±0.0	98.4±0.2	66.5±0.9	99.6±0.0	98.6±0.3	65.7±1.8
carrot	99.4±0.1	97.7±0.2	31.3±1.0	99.4±0.1	98.0±0.2	34.4±2.0	99.5±0.0	98.2±0.2	35.1±1.4
foam	88.6±1.0	69.6±2.1	30.9±0.6	88.8±0.4	70.8±1.1	31.0±0.3	90.0±0.2	72.1±0.6	30.9±0.1
tire	98.0±0.1	91.5±0.3	16.2±0.8	99.1±0.1	95.6±0.4	35.6±0.9	99.3±0.0	96.5±0.3	37.5±0.8
Average	97.5±0.1	92.1±0.1	34.4±1.5	97.9±0.1	93.4±0.3	38.8±1.7	98.4±0.1	94.9±0.2	44.2±1.1

Table 21. Anomaly segmentation performance of our DictAS on **MPDD** for each object category. Pixel-level AUROC, PRO and AP are reported.

Object	1-shot			2-shot			4-shot		
	AUROC	PRO	AP	AUROC	PRO	AP	AUROC	PRO	AP
bracket_brown	94.8±0.4	90.7±0.7	5.3±0.3	95.7±0.3	92.9±1.3	7.1±0.7	96.5±0.3	94.3±1.1	9.7±0.9
connector	97.3±0.4	90.9±1.4	23.0±3.9	97.8±0.2	92.3±0.8	28.8±3.7	98.5±0.2	94.8±0.7	51.2±3.3
tubes	99.2±0.1	97.0±0.3	73.0±1.6	99.4±0.1	97.8±0.3	75.5±1.1	99.5±0.1	98.2±0.3	75.8±0.8
metal_plate	98.3±0.0	95.0±0.1	89.8±0.1	99.0±0.0	96.4±0.1	93.3±0.2	99.2±0.1	96.8±0.2	94.4±0.4
bracket_black	95.0±1.5	89.6±5.7	4.0±3.6	95.7±0.7	91.9±2.0	11.8±0.9	96.7±1.4	93.7±4.2	13.4±5.5
bracket_white	99.4±0.1	93.4±3.1	4.8±0.6	99.8±0.1	96.3±1.9	11.8±0.4	99.8±0.2	97.0±0.6	12.7±2.5
Average	97.4±0.2	92.8±0.5	33.3±0.2	97.9±0.1	94.6±0.3	38.0±1.7	98.4±0.3	95.8±0.8	42.9±1.7

Table 22. Anomaly segmentation performance of our DictAS on **BTAD** for each object category. Pixel-level AUROC, PRO and AP are reported.

Object	1-shot			2-shot			4-shot		
	AUROC	PRO	AP	AUROC	PRO	AP	AUROC	PRO	AP
01	97.0±0.2	77.2±1.5	60.5±0.9	97.3±0.1	78.9±0.5	61.4±0.5	97.5±0.1	80.6±0.6	61.8±0.4
02	97.0±0.0	73.1±1.5	74.2±0.4	97.1±0.1	71.5±1.3	74.5±0.8	97.2±0.0	72.2±0.6	74.4±0.3
03	99.0±0.1	96.0±0.1	59.1±1.4	99.1±0.1	96.6±0.3	62.5±1.6	99.3±0.0	97.2±0.1	64.1±1.8
Average	97.6±0.1	82.1±0.6	64.6±0.7	97.9±0.1	82.4±0.3	66.1±0.8	98.0±0.0	83.3±0.4	66.8±0.7

F. Detailed Qualitative Results

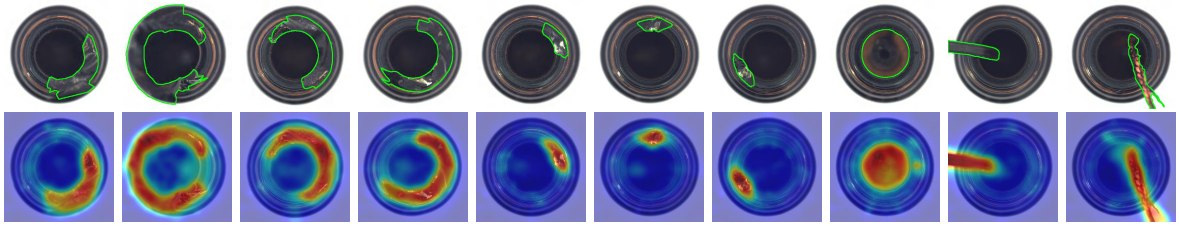


Figure 9. Visualization of segmentation results for the **bottle** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

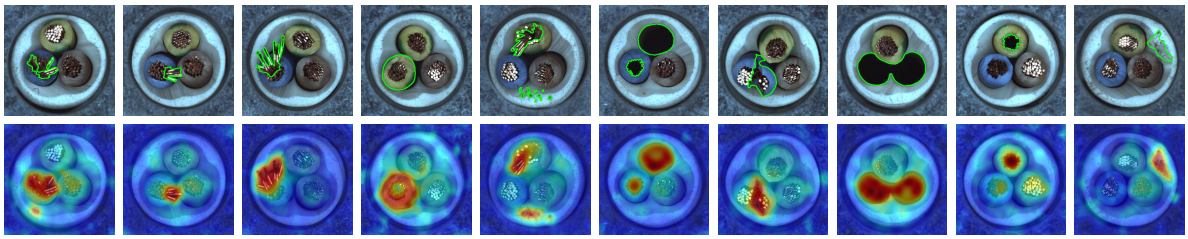


Figure 10. Visualization of segmentation results for the **cable** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

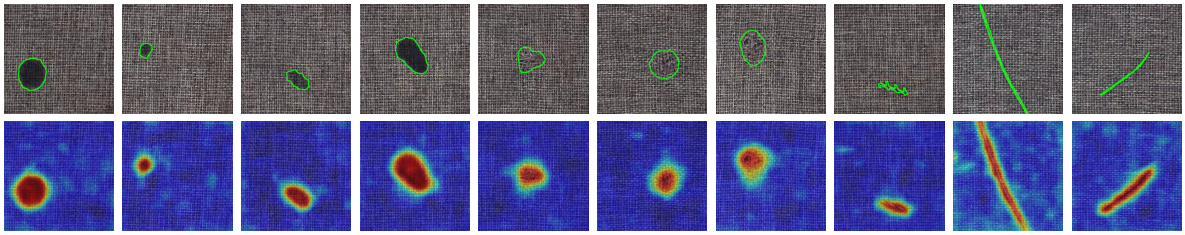


Figure 11. Visualization of segmentation results for the **carpet** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

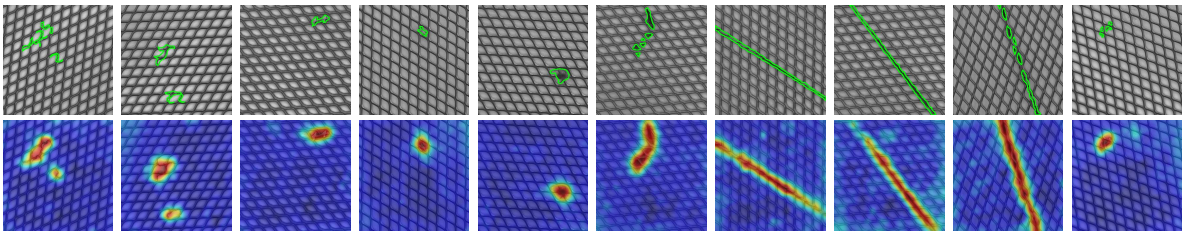


Figure 12. Visualization of segmentation results for the **grid** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

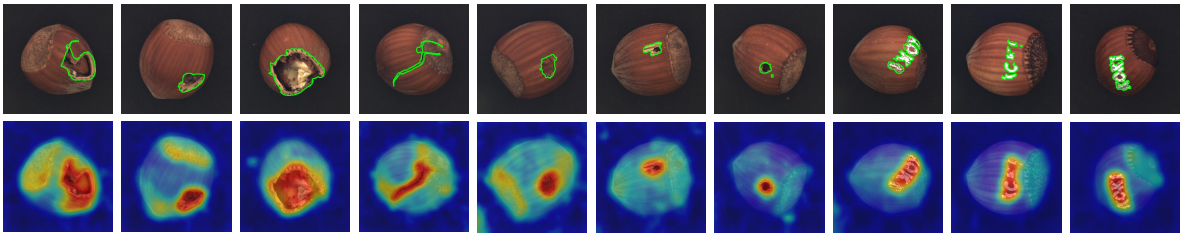


Figure 13. Visualization of segmentation results for the **hazelnut** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

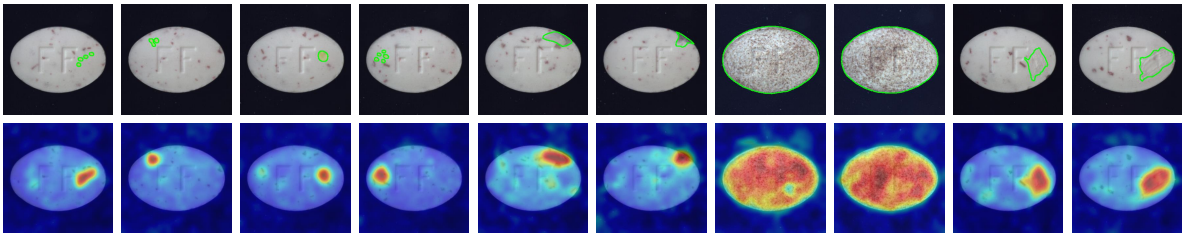


Figure 14. Visualization of segmentation results for the **pill** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

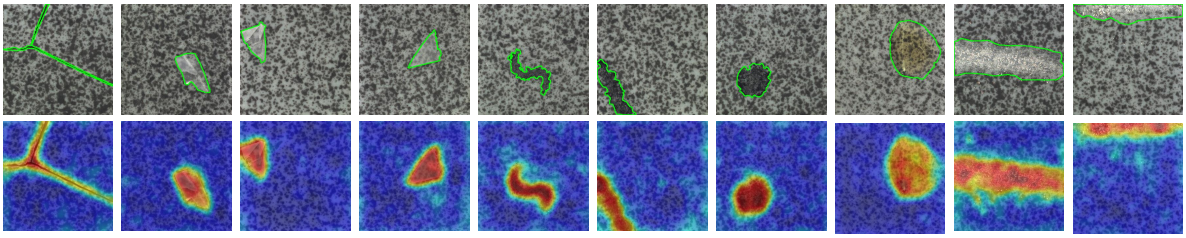


Figure 15. Visualization of segmentation results for the **tile** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

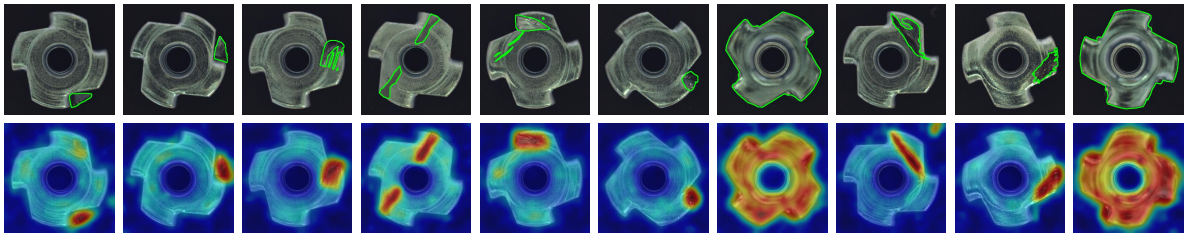


Figure 16. Visualization of segmentation results for the **metal_nut** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

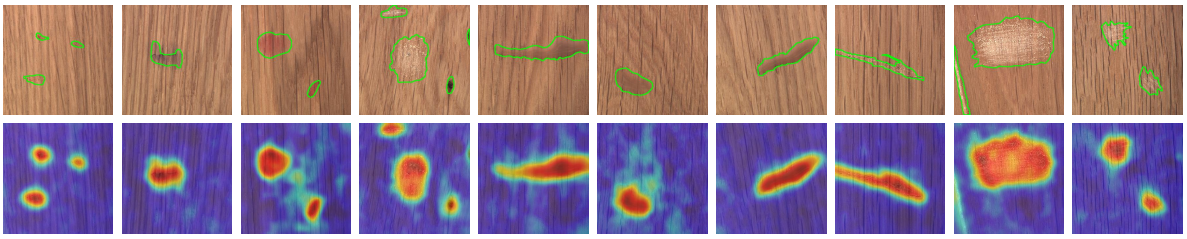


Figure 17. Visualization of segmentation results for the **wood** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

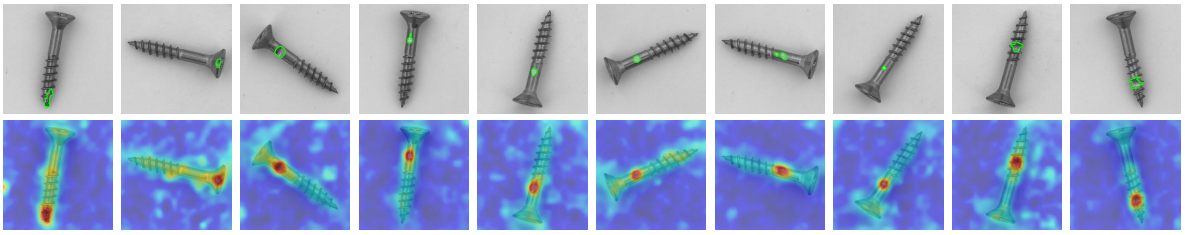


Figure 18. Visualization of segmentation results for the **screw** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

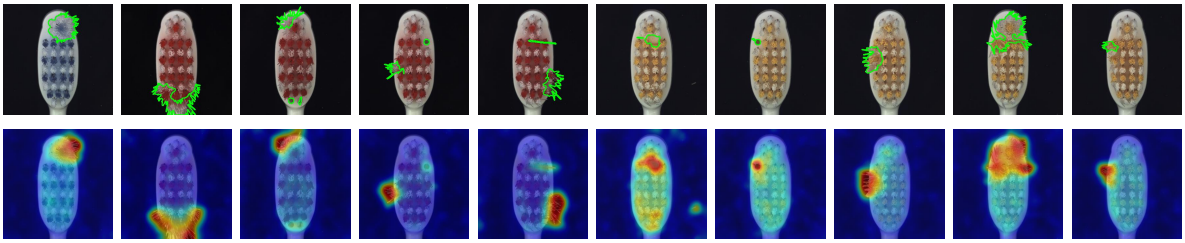


Figure 19. Visualization of segmentation results for the **toothbrush** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

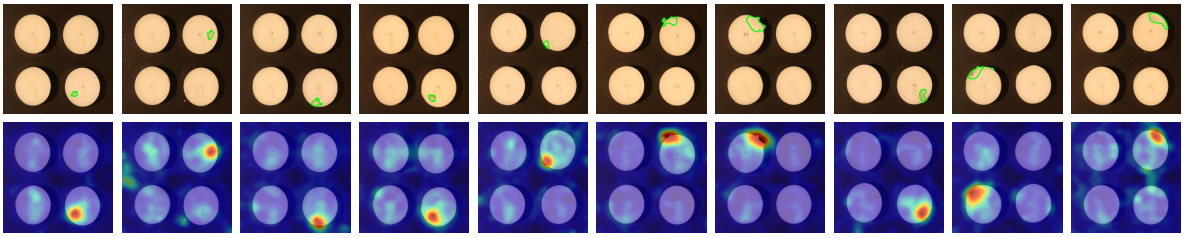


Figure 20. Visualization of segmentation results for the **candle** class on **VisA** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

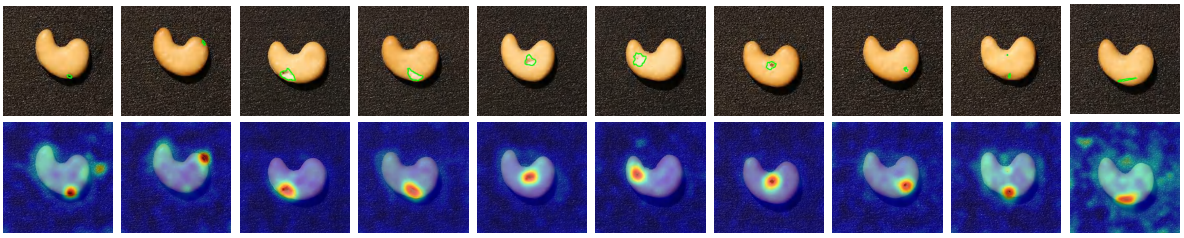


Figure 21. Visualization of segmentation results for the **cashew** class on **VisA** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

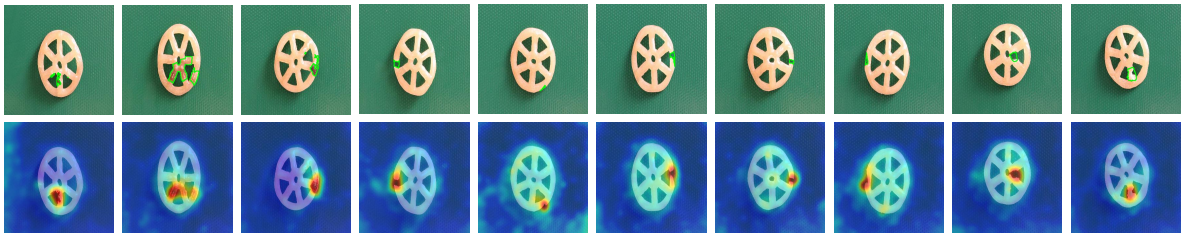


Figure 22. Visualization of segmentation results for the **fryum** class on **VisA** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

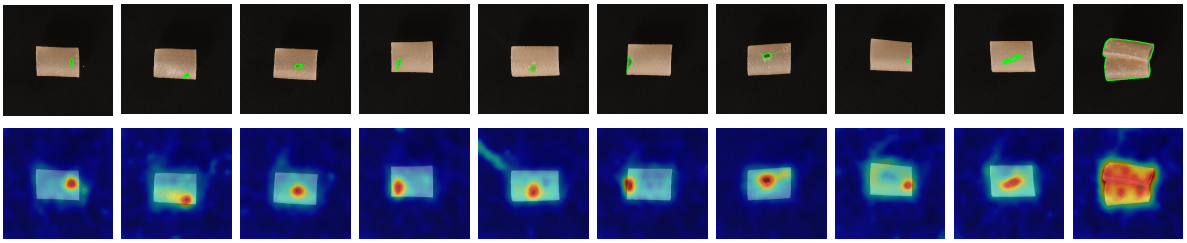


Figure 23. Visualization of segmentation results for the **pipe_fryum** class on **VisA** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

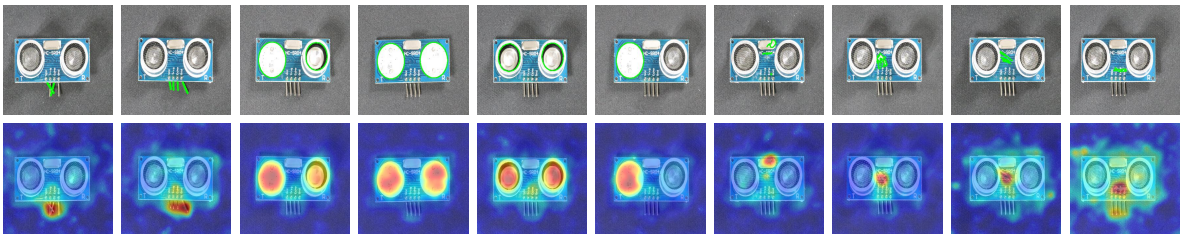


Figure 24. Visualization of segmentation results for the **PCB1** class on **VisA** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

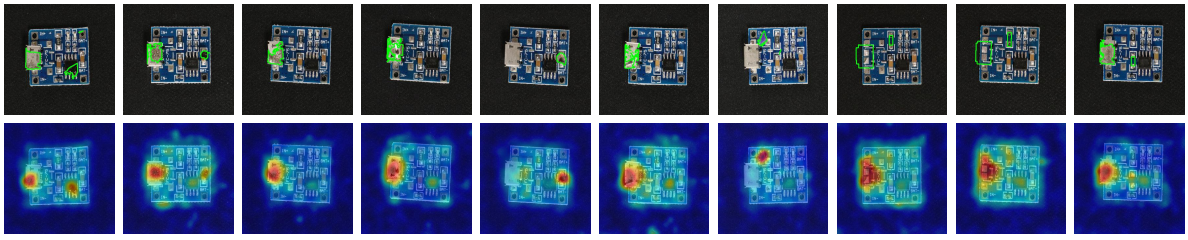


Figure 25. Visualization of segmentation results for the **PCB4** class on **VisA** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

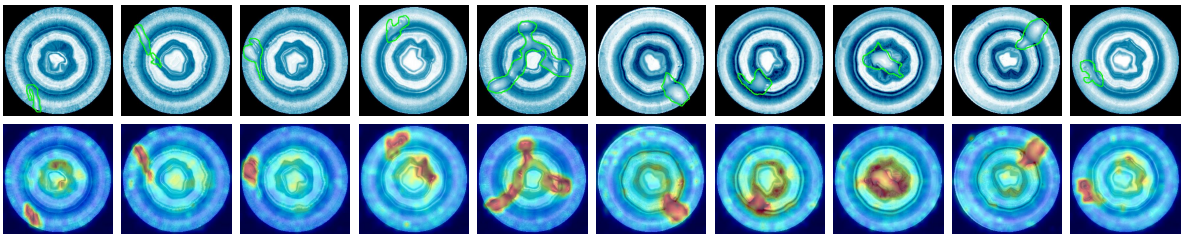


Figure 26. Visualization of segmentation results on BTAD under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results..

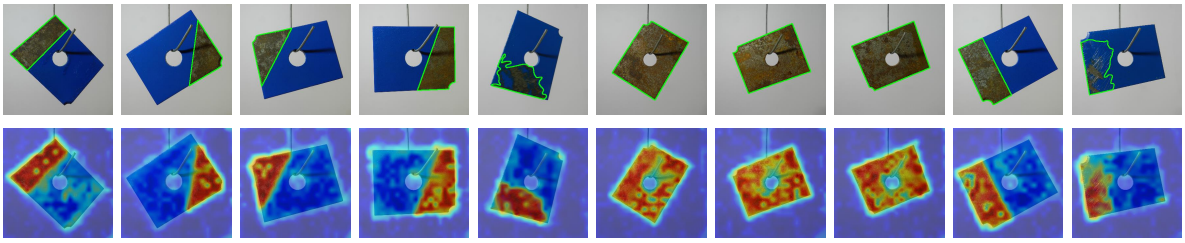


Figure 27. Visualization of segmentation results for the **metal_plate** class on **MPDD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

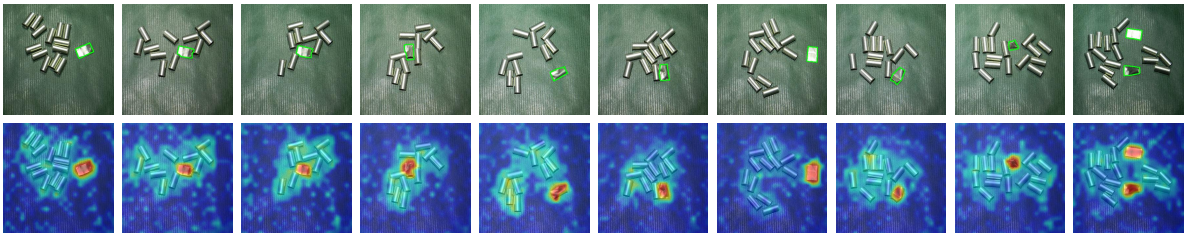


Figure 28. Visualization of segmentation results for the **tubes** class on **MPDD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

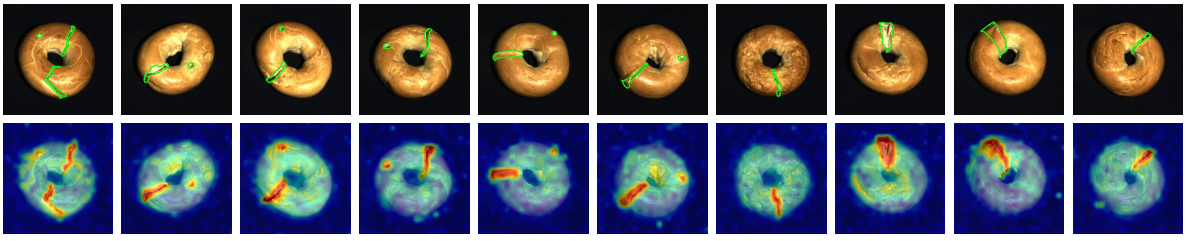


Figure 29. Visualization of segmentation results for the **bangel** class on **MVTec3D** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

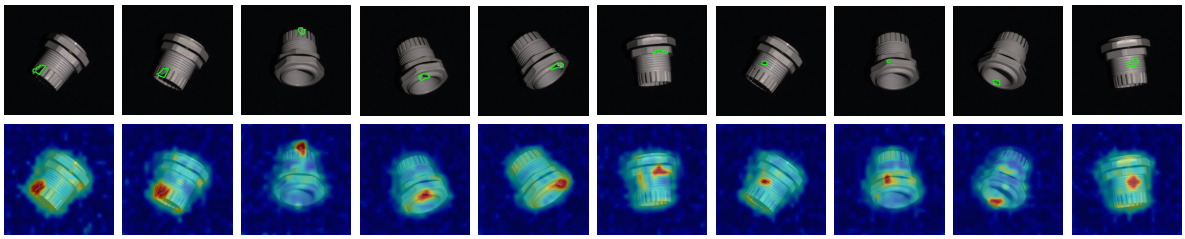


Figure 30. Visualization of segmentation results for the **cable_gland** class on **MVTec3D** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

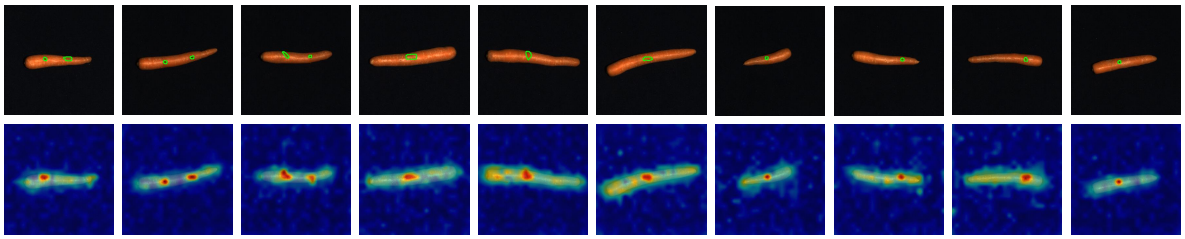


Figure 31. Visualization of segmentation results for the **carrot** class on **MVTec3D** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

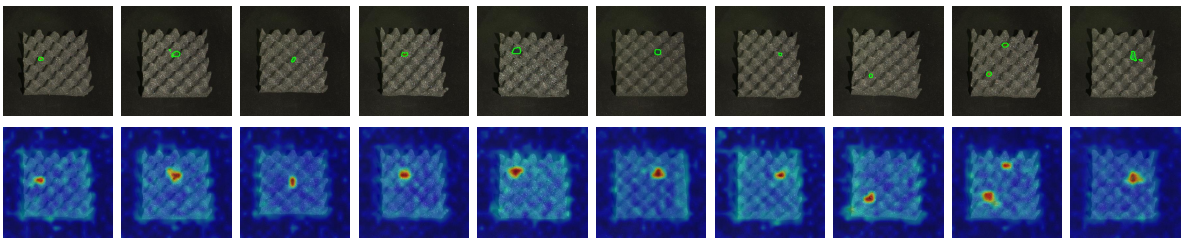


Figure 32. Visualization of segmentation results for the **foam** class on **MVTec3D** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

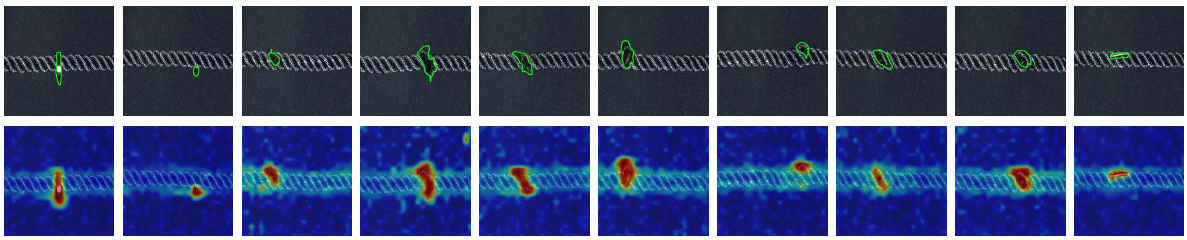


Figure 33. Visualization of segmentation results for the **rope** class on **MVTec3D** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

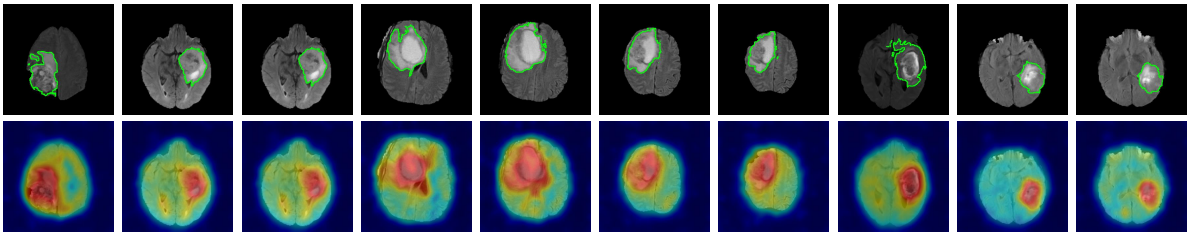


Figure 34. Visualization of segmentation results for the **brain** class on **RESC** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

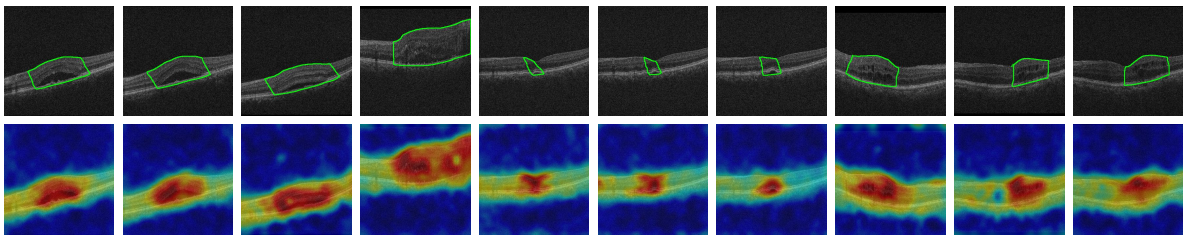


Figure 35. Visualization of segmentation results for the **retina** class on **BrasTS** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.