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 A.1. 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	$\|\mathcal{C}\|$	Testing Set	Training Set	Usage
Industrial	MVTecAD [1]	Obj &texture	Photography	15	(467, 1258)	(3629, 0)	Industrial defect detection
	VisA [21]	Obj	Photography	12	(962, 1200)	(8659, 0)	Industrial defect detection
	MVTec3D [2]	Obj	Photography+Depth	10	(249, 948)	(2656, 0)	Industrial defect detection
	MPDD [11]	Obj	Photography	6	(176, 282)	(888, 0)	Industrial defect detection
	BTAD [15]	Obj	Photography	3	(451, 290)	(1799, 0)	Industrial defect detection
Medical	RESC [8]	Retina	Photography	1	(1041, 764)	(4297, 0)	Retinal Lesion Detection
	BrasTS [14]	Brain	Radiology(MRI)	1	(828, 1948)	(4211, 0)	Brain Tumor Segmentation

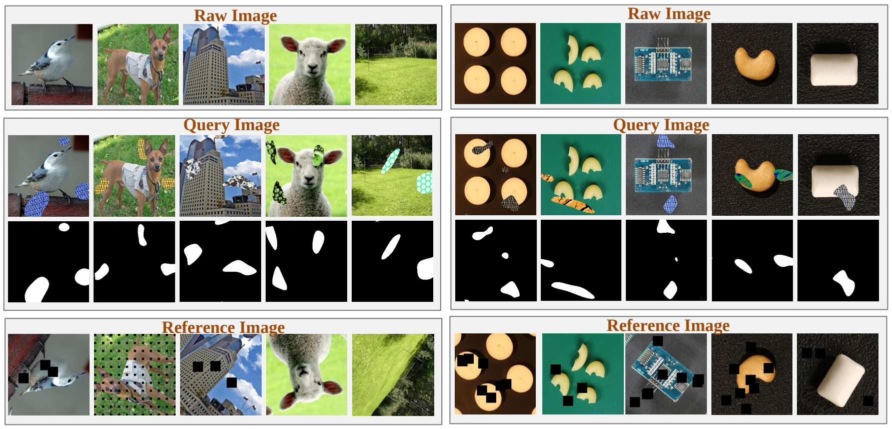
In this study, we conduct extensive experiments on 7 public datasets covering industrial and medical domains to assess the effectiveness of our methods, including MVTecAD [1], VisA [21], MVTec3D [2], MPDD [11], BTAD [15], RESC [8] and BrasTS [14]. The key statistics for these datasets are demonstrated in Table A.1. 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 A.1. 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 [18]. 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 A.1. **Acquisition of the auxiliary training data.** Given a raw image without pixel-level annotations, the query image is generated using an anomaly synthesis algorithm [18], 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 A 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.

```
1:  while True do
2:      $\mathbf{G} \leftarrow$ where($P(H, w, r_x, r_y) > \lambda$)
3:      $\mathbf{M}_A \leftarrow G \times \mathbf{X}$
4:      $\overline{\mathbf{M}}_A \leftarrow$ 1 - $\mathbf{M}_A$
5:      $\mathbf{X}_q \leftarrow \gamma(\mathbf{M}_A \odot \mathbf{A}) + (1-\gamma)(\mathbf{M}_A \odot \mathbf{X}) + \overline{\mathbf{M}}_A \odot \mathbf{X}$
6:  end while
7:  if SUM($\mathbf{G}$) is 0 then
8:      $y_q \leftarrow 0$
9:  else
10:     $y_q \leftarrow 1$
11: end if
12: 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 [10] (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 A.2.

State-level (normal)

"flawless [class]"
"perfect [class]"
"unblemished [class]"
"normal [class]"
"[class] without flaw"
"[class] without defect"
"[class] without damage"

State-level (abnormal)

"damaged [class]"
"broken [class]"
"abnormal [class]"
"imperfect [class]"
"[class] with flaw"
"[class] with defect"
"[class] with damage"

Template-level

"a photo of a/the [state][class]."
"a good photo of the [state][class]."
"a photo of my [state][class]."
"a photo of the [state][class]."
"a photo of a/the [state][class]."
"a photo of a/the small [state][class]."
"a bad photo of a/the [state][class]."
"a low resolution photo of a/the [state][class]."

"a cropped photo of a/the [state][class]."
"a bright photo of a/the [state][class]."
"a dark photo of a/the [state][class]."
"a black and white photo of a/the [state][class]."
"a jpeg corrupted photo of a/the [state][class]."
"a close-up photo of the [state][class]."
"There is a/the [state][class] in the scene."
"This is one [state][class] in the scene."

Figure A.2. 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 [3, 10, 13], we adopt the CLIP model (ViT-L-14-336), pretrained by OpenAI [16], 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 [3]. 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 [12], 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** [10] is one of the earliest works based on CLIP for the zero/few shot anomaly segmentation task. Since the vanilla CLIP [16] 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 [20].
- **APRIL-GAN** [3] 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 [10] 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.

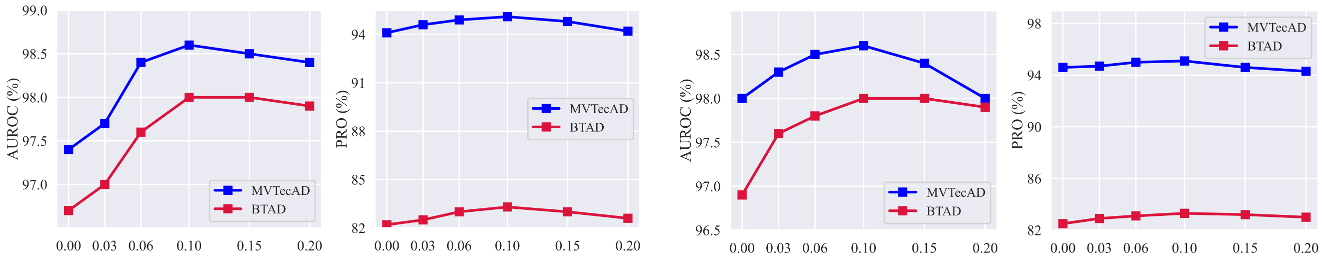
- **RegAD** [9] first proposed a feature registration strategy using a spatial transformer network for class-generalizable FSAS. With a meta-learning training approach, it demonstrates strong generalization to unseen classes. However, its performance on unseen category objects heavily depends on extensive 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** [5] 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+** [5] 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** [7] 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** [13] 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** [6] 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** [17] 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 A.2. 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 [1]	VisA [21]	BTAD [15]	MVTec3D [2]	BrasTS [14]
1-shot	MetaUAS [6]	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 [17]	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 [17]	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 A.2 compares our DictAS with two contemporary state-of-the-art methods, MetaAUS [6] and ResAD [17]. As our method currently applies only to transformer-based architectures, we selected the CLIP pre-trained backbones with the smallest (ViT-B-16) and largest (ViT-L-14-336) parameter counts for comparison. The experimental results show that,



(a) λ_1

(b) λ_2

Figure A.3. (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 A.3. Ablation on different auxiliary datasets under 4-shot setting (%).

Auxiliary Dataset	MVTecAD			BTAD		
	AUROC	PRO	AP	AUROC	PRO	AP
VisA [21]	**98.6**	**95.1**	**66.8**	98.0	83.3	66.8
BrasTS [14]	98.3	95.0	66.2	97.9	83.0	66.2
Ade20K [19]	98.4	**95.1**	66.5	98.1	**83.5**	66.8
VOC2012 [4]	98.3	94.9	66.4	**98.2**	83.4	**66.9**

Table A.4. 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**

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 A.3, 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 A.5. 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 A.6. 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 [21] 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 [3, 7, 17]. 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 A.3, we investigate the impact of using auxiliary datasets from different domains for self-supervised training and evaluate their 4-shot performance on MVTecAD [1] and BTAD [15]. Specifically, the VisA dataset [21] from the industrial domain, the BrasTS dataset [14] from the medical domain, and the Ade20K [19] and VOC2012 [4] datasets from natural scenes are used as auxiliary datasets. For the natural scene datasets Ade20K [19] and VOC2012 [4], 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 A.4, 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 A.5. 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 A.6. 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 [3].

E.1. Detailed few-shot anomaly classification results

Table A.7. **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 [9]	AnomalyGPT [7]	FastRecon [5]	† FastRecon+ [5]	† WinCLIP [10]	† APRIL-GAN [3]	† PromptAD [13]	† DictAS
	(ECCV 22)	(AAAI 24)	(ICCV 23)	(ICCV 23)	(CVPR 23)	(CVPR 23)	(CVPR 24)	(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 A.8. **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 [9]	AnomalyGPT [7]	FastRecon [5]	† FastRecon+ [5]	† WinCLIP [10]	† APRIL-GAN [3]	† PromptAD [13]	† DictAS
	(ECCV 22)	(AAAI 24)	(ICCV 23)	(ICCV 23)	(CVPR 23)	(CVPR 23)	(CVPR 24)	(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 A.9. **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 [9]	AnomalyGPT [7]	FastRecon [5]	† FastRecon+ [5]	† WinCLIP [10]	† APRIL-GAN [3]	† PromptAD [13]	† DictAS
	(ECCV 22)	(AAAI 24)	(ICCV 23)	(ICCV 23)	(CVPR 23)	(CVPR 23)	(CVPR 24)	(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 A.10. **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 [9] (ECCV 22)	AnomalyGPT [7] (AAAI 24)	FastRecon [5] (ICCV 23)	† FastRecon+ [5] (ICCV 23)	† WinCLIP [10] (CVPR 23)	† APRIL-GAN [3] (CVPR 23)	† PromptAD [13] (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 A.11. **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 [9] (ECCV 22)	AnomalyGPT [7] (AAAI 24)	FastRecon [5] (ICCV 23)	† FastRecon+ [5] (ICCV 23)	† WinCLIP [10] (CVPR 23)	† APRIL-GAN [3] (CVPR 23)	† PromptAD [13] (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 A.12. **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 [9] (ECCV 22)	AnomalyGPT [7] (AAAI 24)	FastRecon [5] (ICCV 23)	† FastRecon+ [5] (ICCV 23)	† WinCLIP [10] (CVPR 23)	† APRIL-GAN [3] (CVPR 23)	† PromptAD [13] (CVPR 24)	† DictAS (Ours)
Industrial Datasets (AUROC, PRO, AP)								
MVTecAD [1]	(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 [21]	(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 [2]	(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 [11]	(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 [15]	(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 [8]	(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 [14]	(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 A.13. 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 A.14. 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 A.15. 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 A.16. 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 A.17. 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

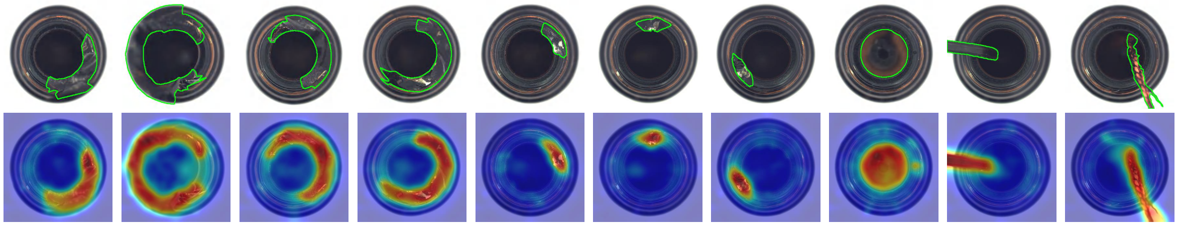


Figure A.4. Visualization of segmentation results for the **bottle** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

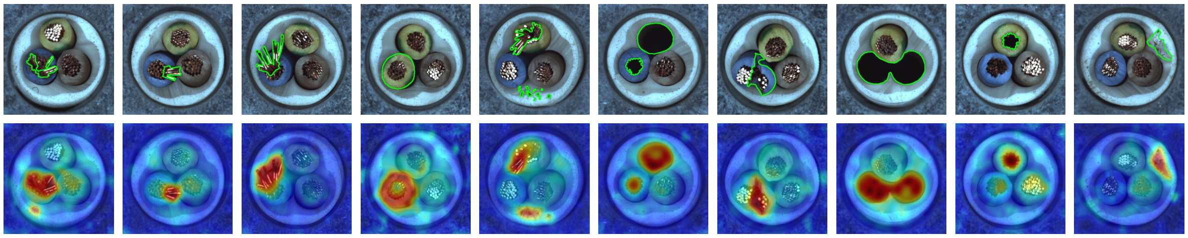


Figure A.5. Visualization of segmentation results for the **cable** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

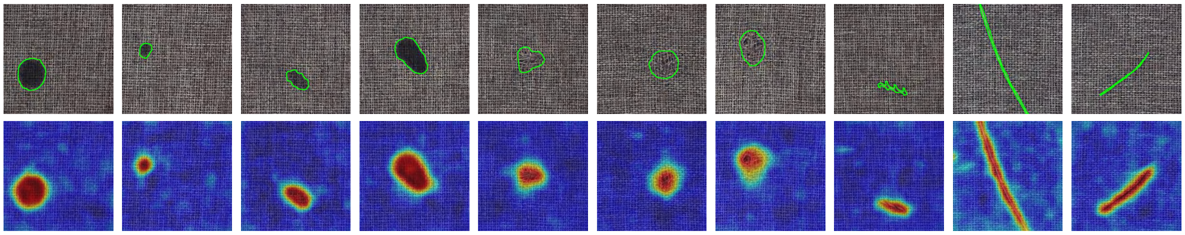


Figure A.6. Visualization of segmentation results for the **carpet** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

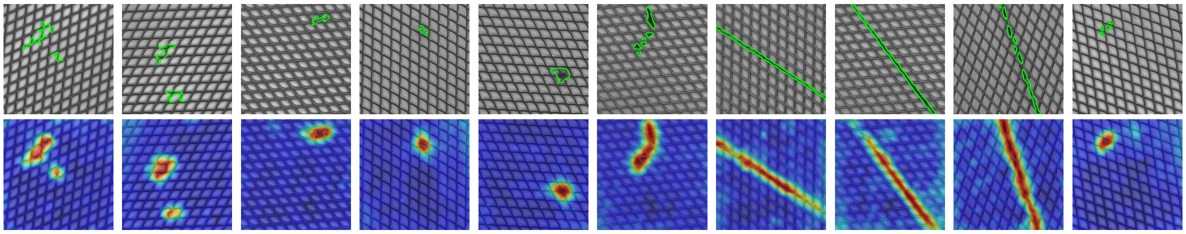


Figure A.7. Visualization of segmentation results for the **grid** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

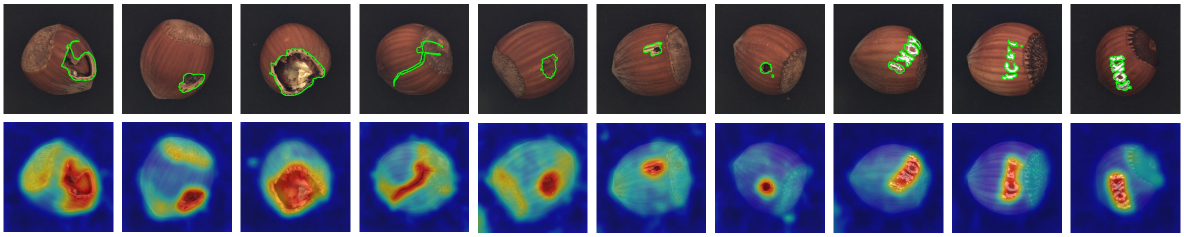


Figure A.8. Visualization of segmentation results for the **hazelnut** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

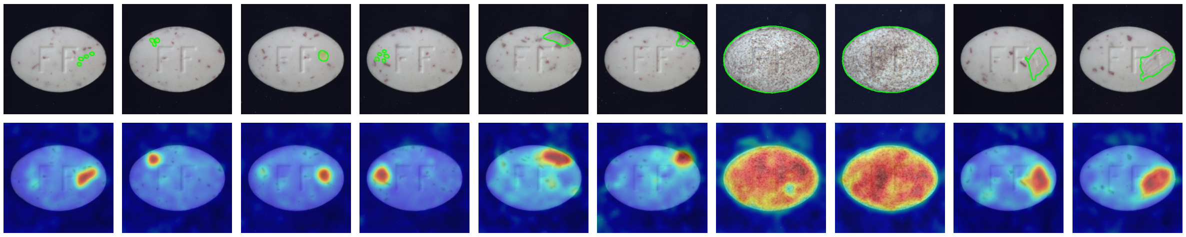


Figure A.9. Visualization of segmentation results for the **pill** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

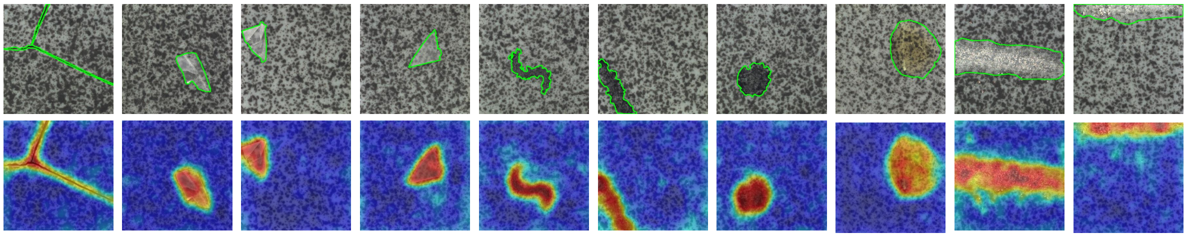


Figure A.10. Visualization of segmentation results for the **tile** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

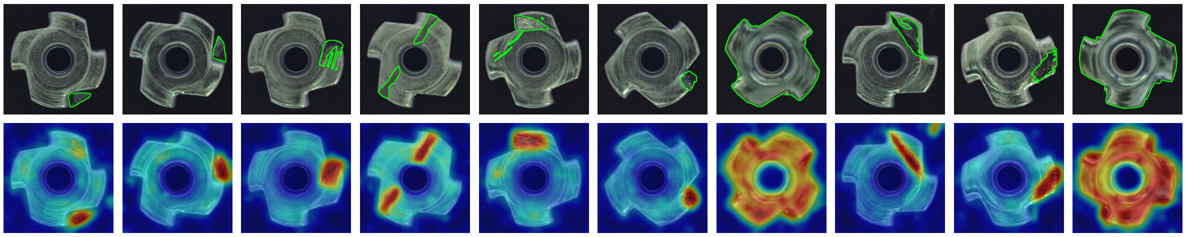


Figure A.11. Visualization of segmentation results for the **metal_nut** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

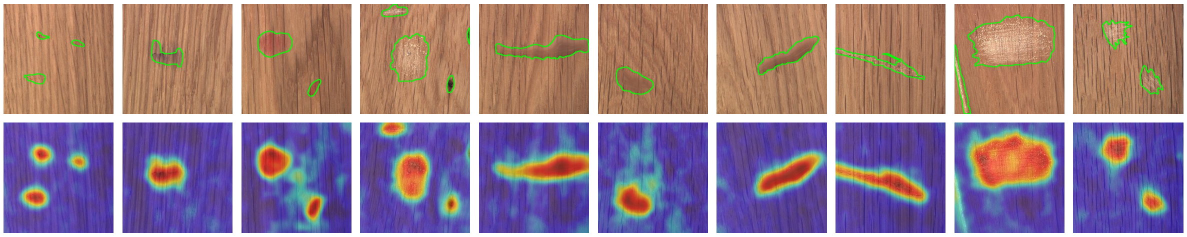


Figure A.12. Visualization of segmentation results for the **wood** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

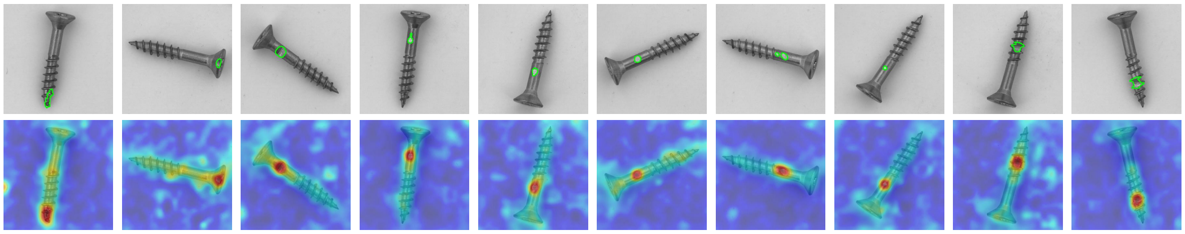


Figure A.13. Visualization of segmentation results for the **screw** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

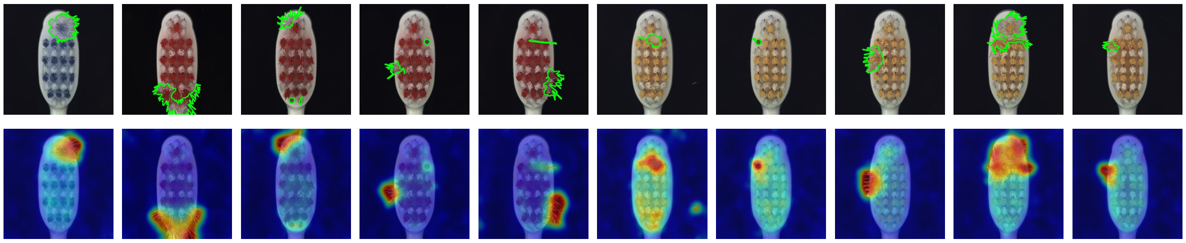


Figure A.14. Visualization of segmentation results for the **toothbrush** class on **MVTecAD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

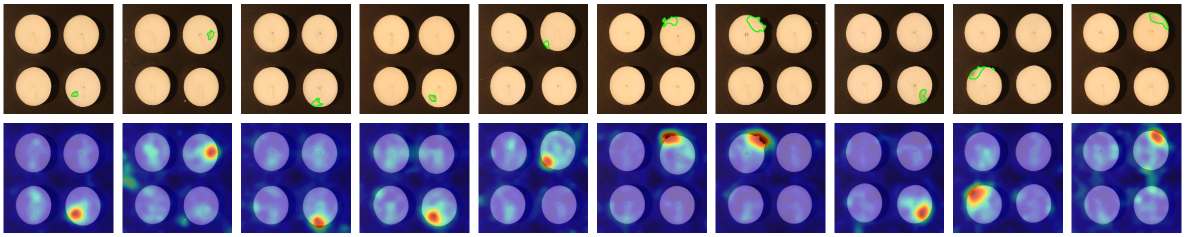


Figure A.15. Visualization of segmentation results for the **candle** class on **VisA** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

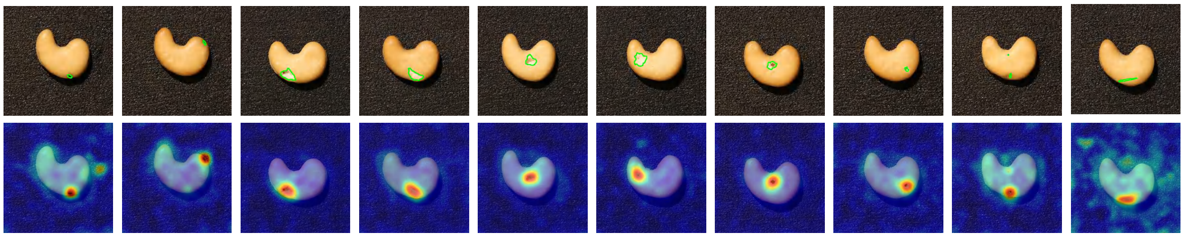


Figure A.16. Visualization of segmentation results for the **cashew** class on **VisA** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

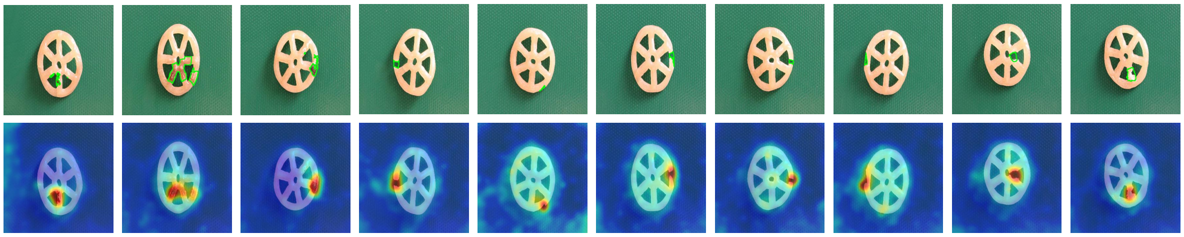


Figure A.17. Visualization of segmentation results for the **fryum** class on **VisA** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

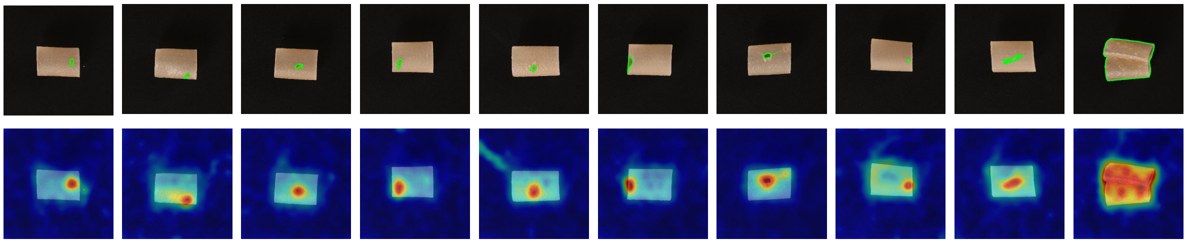


Figure A.18. Visualization of segmentation results for the **pipe_fryum** class on **VisA** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

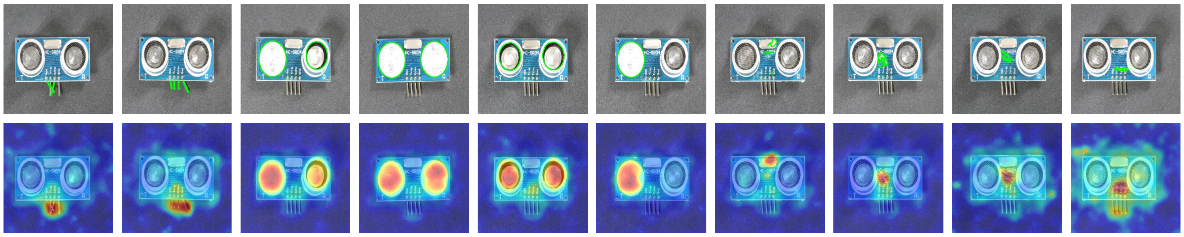


Figure A.19. Visualization of segmentation results for the **PCB1** class on **VisA** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

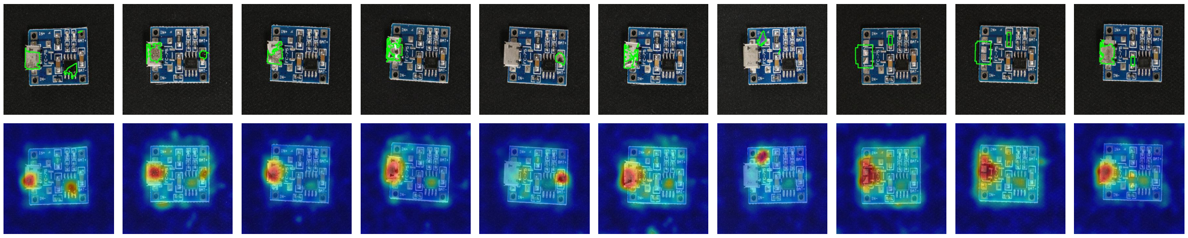


Figure A.20. Visualization of segmentation results for the **PCB4** class on **VisA** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

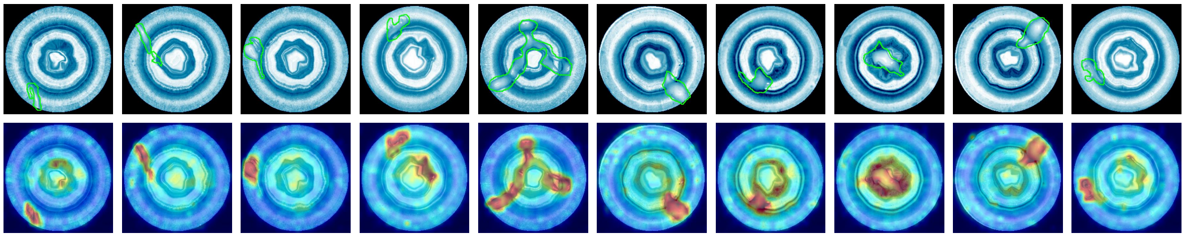


Figure A.21. Visualization of segmentation results on BTAD under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results..

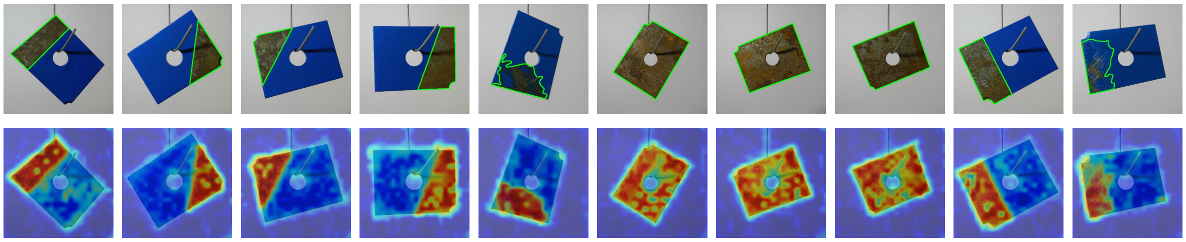


Figure A.22. Visualization of segmentation results for the **metal_plate** class on **MPDD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

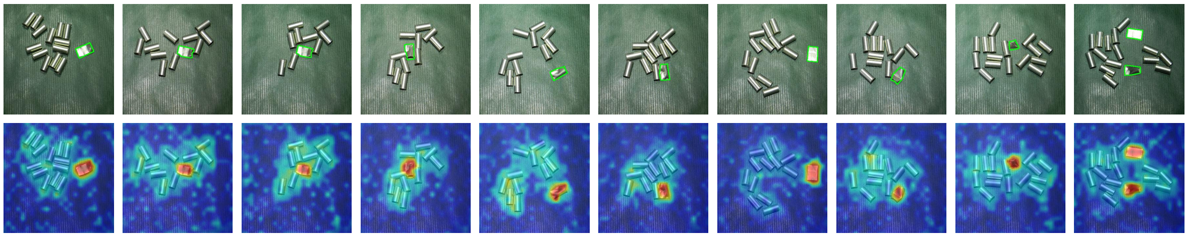


Figure A.23. Visualization of segmentation results for the **tubes** class on **MPDD** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

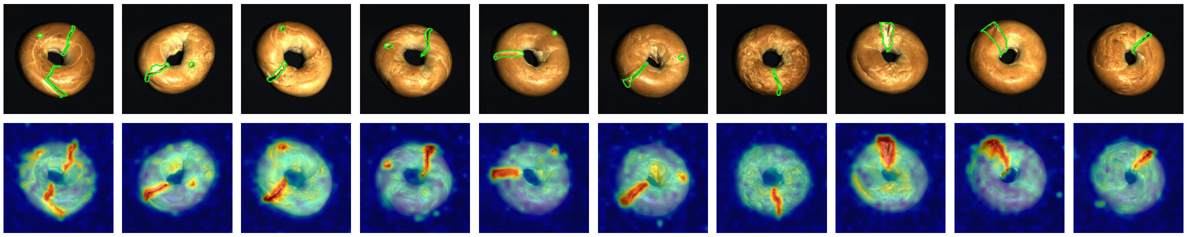


Figure A.24. Visualization of segmentation results for the **bangel** class on **MVTec3D** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

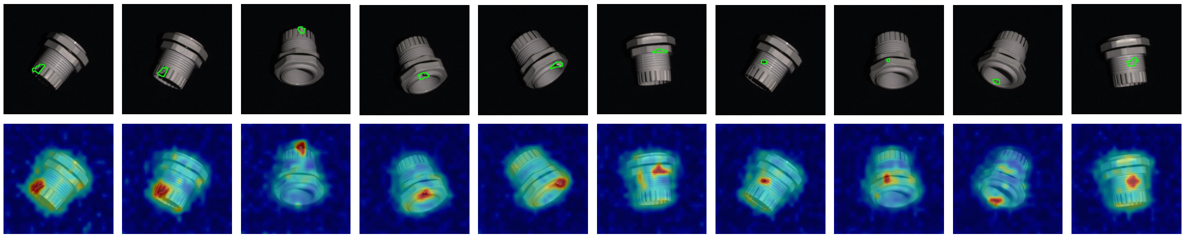


Figure A.25. Visualization of segmentation results for the **cable_gland** class on **MVTec3D** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

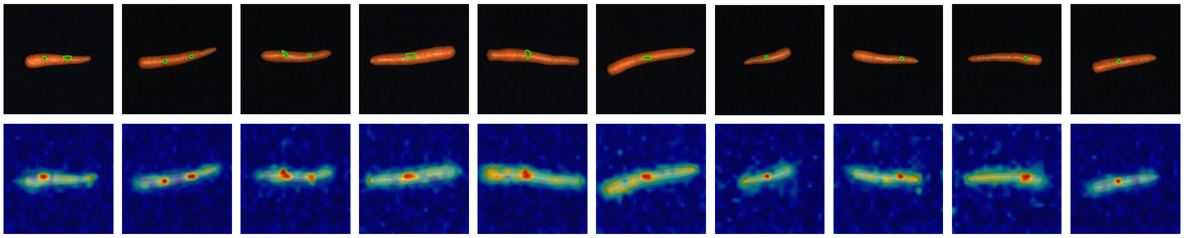


Figure A.26. Visualization of segmentation results for the **carrot** class on **MVTec3D** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

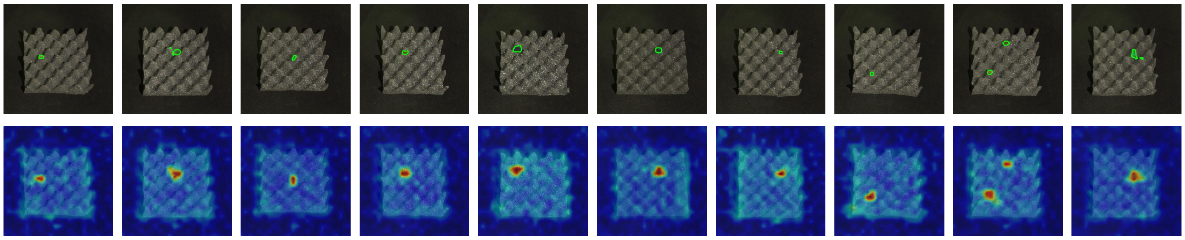


Figure A.27. Visualization of segmentation results for the **foam** class on **MVTec3D** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

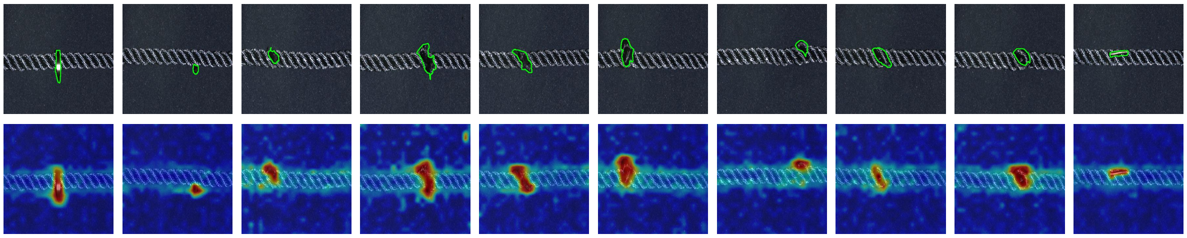


Figure A.28. Visualization of segmentation results for the **rope** class on **MVTec3D** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

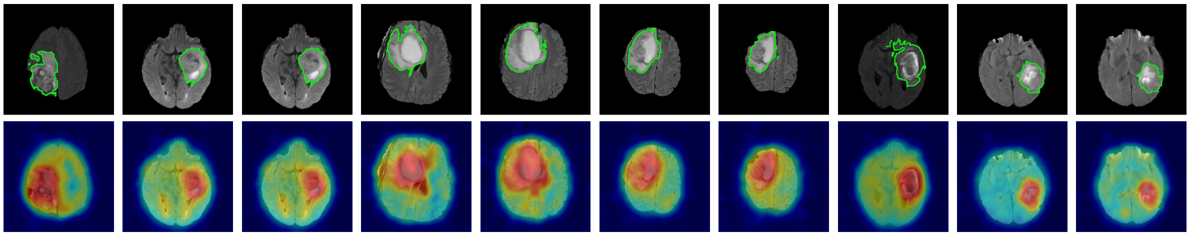


Figure A.29. Visualization of segmentation results for the **brain** class on **RESC** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

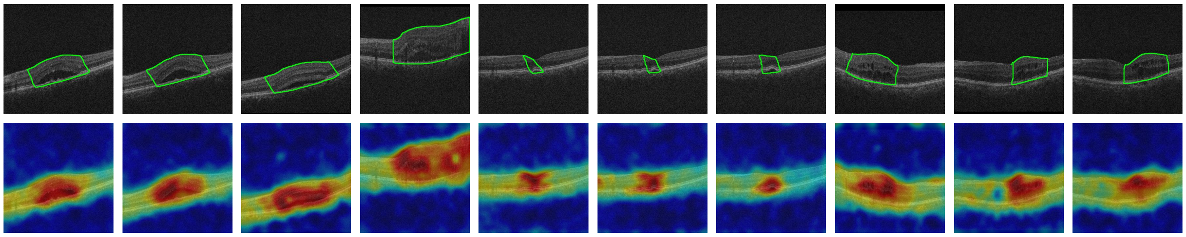


Figure A.30. Visualization of segmentation results for the **retina** class on **BrasTS** under the 4-shot setting. The first row displays the input images, with green outlines indicating the ground truth regions. The second row presents the anomaly segmentation results.

References

[1] 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. 1, 4, 6, 9

[2] 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. 1, 4, 9

[3] 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. 3, 6, 7, 8, 9

[4] 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. 5, 6

[5] 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. 4, 8, 9

[6] Bin-Bin Gao. Metauas: Universal anomaly segmentation with one-prompt meta-learning. *Advances in Neural Information Processing Systems*, 37:39812–39836, 2025. 4

[7] 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. 4, 6, 8, 9

[8] 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. 1, 9

[9] 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. 4, 8, 9

[10] 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. 3, 8, 9

[11] 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. 1, 9

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[13] 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. 3, 4, 8, 9

[14] 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, 4, 5, 6, 9

[15] 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. 1, 4, 6, 9

[16] 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. 3

[17] 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. 4, 6

[18] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8330–8339, 2021. 1, 2

[19] 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. 5, 6

[20] 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. 3

[21] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*, pages 392–408. Springer, 2022. 1, 4, 5, 6, 9