AdaCLIP: Adapting CLIP with Hybrid Learnable Prompts for Zero-Shot Anomaly Detection

Yunkang Cao[1,2], Jiangning Zhang[3,4], Luca Frittoli[2], Yuqi Cheng[1], Weiming Shen[1]✉, and Giacomo Boracchi[2]

[1] Huazhong University of Science and Technology
{cyk_hust,yuqicheng,shenwm}@hust.edu.cn
[2] Politecnico di Milano
{luca.frittoli,giacomo.boracchi}@polimi.it
[3] Zhejiang University
[4] Youtu Lab, Tencent
186368@zju.edu.cn

In this appendix, we present more details about the dataset (Section 1), the proposed AdaCLIP (Section 2), and the selected baselines (Section 3). Section 4 provides a fair and comprehensive comparison between the proposed AdaCLIP and another popular ZSAD method, AnomalyCLIP. Section 5 presents comparison results between the proposed AdaCLIP and other popular full-shot unsupervised anomaly detection methods, demonstrating the potential practical applicability of the proposed AdaCLIP. Sections 6 and 7 offer more quantitative and qualitative comparisons.

1 Dataset Details

In this study, we conduct extensive experiments on 14 public datasets covering industrial and medical domains across three modalities (photography, radiology, and endoscopy) to assess the effectiveness of our methods. We solely utilize the test data from these datasets, and their relevant information is presented in Table 1. We default to using two datasets, MVTec AD [2] and ClinicDB [3], as auxiliary data for training. Additionally, for evaluations on MVTec AD and ClinicDB, we employ VisA [17] and ColonDB [14] for training.

Table 1: Key statistics the utilized datasets. $|\mathcal{C}|$ denotes to the number of categories in individual datasets.

Domain	Anomaly Detection		Dataset	Category	Modality	$\lvert\mathcal{C}\rvert$	Normal and anomalous samples
	Image-level	Pixel-level					
Industrial	✔	✔	MVTec AD	Obj & Texture	Photography	15	(467,1258)
	✔	✔	VisA	Obj	Photography	12	(962,1200)
	✔	✔	MPDD	Obj	Photography	6	(176,282)
	✔	✔	BTAD	Obj	Photography	3	(451,290)
	✔	✔	KSDD	Obj	Photography	1	(181,74)
	✔	✔	DAGM	Texture	Photography	10	(6996,1054)
	✔	✔	DTD-Synthetic	Texture	Photography	12	(357,947)
Medical	✔	✘	HeadCT	Brain	Radiology (CT)	1	(100,100)
	✔	✘	BrainMRI	Brain	Radiology (MRI)	1	(98,155)
	✔	✘	Br35H	Brain	Radiology (MRI)	1	(1500,1500)
	✘	✔	ISIC	Skin	Photography	1	(0,379)
	✘	✔	ClinicDB	Colon	Endoscopy	1	(0,612)
	✘	✔	ColonDB	Colon	Endoscopy	1	(0,380)
	✘	✔	TN3K	Thyroid	Radiology (Ultralsound)	1	(0,614)

2 Module Details

2.1 Hybrid Learnable Prompts

We introduce hybrid learnable prompts for adapting the pre-trained CLIP [12] for the ZSAD task. Figure 1 presents the details of hybrid learnable prompts. In particular, we utilize a pre-trained and frozen CLIP image encoder to extract image embeddings that contain high-level semantic information. Then for image and text encoders, we employ a simple linear layer to project the image embeddings into dynamic prompts, respectively. These dynamic prompts are then summed with static prompts from the initial J layers as final hybrid prompts for the image and text encoders. While CoCoOp [15] employs a similar design of hybrid (static+dynamic) prompts, our proposed AdaCLIP prompts both image and text encoders for improved adaptation. We further examine the impact of prompting encoders, with results presented in Table 2. The data indicates that solely prompting the text encoder, as CoCoOp does, results in a performance decrease of 7.2% (0.1%) in image (pixel)-level AUROCs for the medical domain and 2.0% (0.7%) for the industrial domain. Therefore, our multimodal prompting approach more effectively leverages the multimodal capabilities of CLIP, enhancing its potential for zero-shot anomaly detection.

2.2 Hybrid Semantic Fusion

Previous maximum value-based image-level anomaly detection methods [4, 8] may exhibit sensitivity to prediction noise. In contrast, we propose a Hybrid Semantic Fusion (HSF) module aimed at fusing region-level anomalies into a semantic-rich image embedding to enhance image-level anomaly detection performance. Specifically, patch embeddings from individual hierarchies are clustered using the KMeans++ algorithm [1]. We hypothesize that these clusters should represent different regions within the image, with clusters having the highest average anomaly scores likely corresponding to abnormal regions. To validate this assumption, we visualize the clustering results in Fig. 2. It is apparent that the clusters delineate distinct regions within the image. Also, the cluster with the highest average anomaly score typically denotes the abnormal region. Consequently, the HSF module aggregates the centroids of these clusters with the highest average anomaly scores into the semantic-rich image embedding, which encapsulates multi-hierarchy context pertaining to region-level anomalies, thereby significantly enhancing image-level anomaly detection. As shown in Table 3, we investigated anomaly detection performance with varying $K \in [10, 20, 40, 80]$. HSF consistently enhances image-level detection results compared to the maximum value-based method, achieving improvements of 2.4% (2.9%), 2.8% (4.2%), 3.8% (2.4%), and 0.3% (4.1%) in image-level AUROCs for medical (industrial) domains, respectively. A larger K results in smaller clusters, and when clusters become sufficiently small, HSF degrades to the maximum value-based method. Optimal

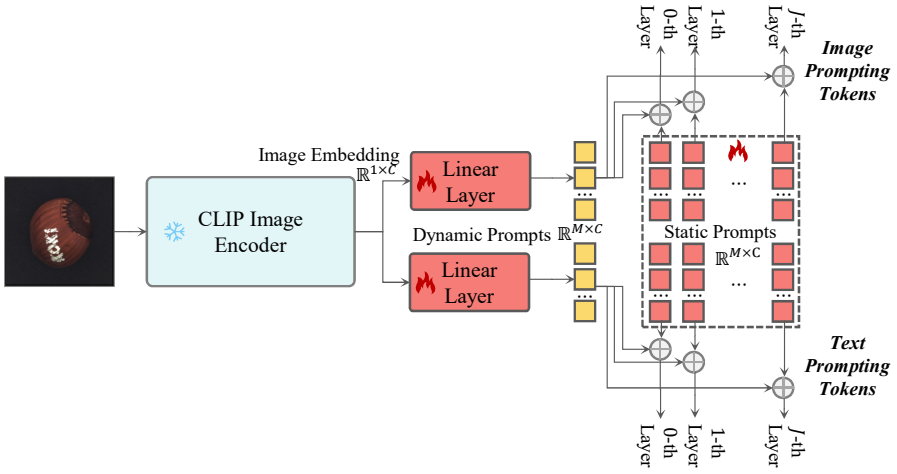

Fig. 1: Illustration of Hybrid Learnable Prompts. This illustration depicts the utilization of two linear layers in conjunction with a shared pre-trained CLIP image encoder to generate dynamic prompts for both the image and text encoders. These dynamic prompts, along with the static prompts from the initial J layers, are then combined to prompt the encoders effectively.

Table 2: Influence of prompting encoders. The best performance is in **bold**.

Prompting Encoder		Medical Domain		Industrial Domain	
Image	Text	Image-level	Pixel-level	Image-level	Pixel-level
✔	✘	(86.6, 49.9)	(80.6, 42.9)	(87.6, 85.1)	(93.9, 48.2)
✘	✔	(87.4, 59.0)	(85.2, 57.0)	(88.2, 86.9)	(93.5, 49.8)
✔	✔	(**94.6**, **89.6**)	(**85.3**, **57.4**)	(**90.2**, **89.6**)	(**94.2**, **50.2**)

performance is ideally obtained when clusters match the size of testing anomalies. However, due to the variability in anomaly sizes across testing categories and samples, achieving an optimal K for both medical and industrial domains is challenging, as indicated in Table 3. Therefore, we set $K = 20$ by default.

3 Comparison Method Details

We compare the proposed AdaCLIP with several SOTA methods. Table 4 highlights the key differences between these methods. Notably, AnomalyGPT [7] and AnomalyCLIP [16] are the most relevant concurrent works. In comparison to AdaCLIP, AnomalyGPT uses learnable static prompts but lacks zero-shot anomaly detection capability. While AnomalyCLIP also utilizes prompting learning to enhance ZSAD performance, it solely adds static prompts to the text

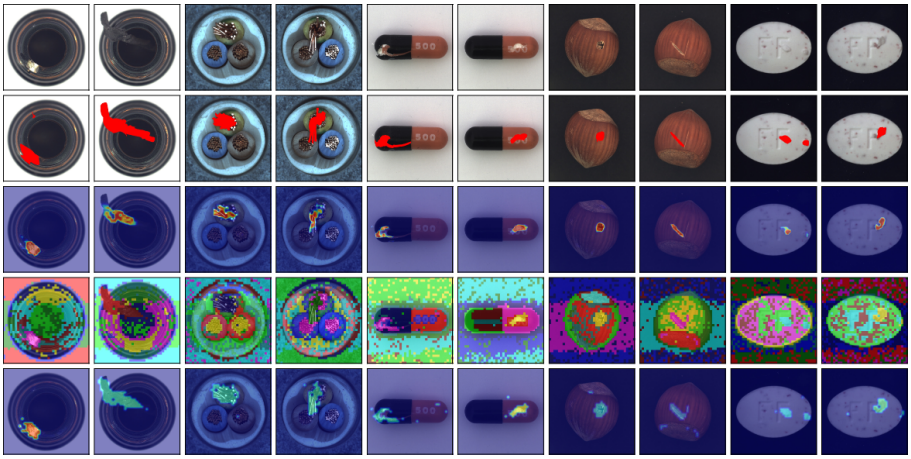

Fig. 2: Illustration of HSF. Top to bottom: input images, ground truths, anomaly maps, clustering results, and clusters with average anomaly scores. It clearly demonstrates that HSF can identify abnormal regions and then extract region-level features. The resulting semantic-rich image embedding comprises multi-hierarchy region-level features, enhancing robust image-level anomaly detection.

Table 3: Ablation on the number (K) of clusters in HSF.

HSF	Medical Domain		Industrial Domain	
	Image-level	Pixel-level	Image-level	Pixel-level
✘	(91.8, 88.7)	(85.7, 57.7)	(86.0, 85.8)	(93.8, 49.9)
K=10	(94.2, 89.6)	(86.9, **59.4**)	(88.9, 87.3)	(93.9, 50.1)
K=20	(94.6, 89.6)	(85.3, 57.4)	(**90.2**, **89.6**)	(**94.2**, 50.2)
K=40	(**95.6**, **90.5**)	(85.4, 56.0)	(88.4, 87.8)	(93.3, 48.4)
K=80	(92.1, 88.9)	(**87.1**, 58.9)	(90.1, 89.4)	(94.1, **50.8**)

encoder of CLIP. In the proposed AdaCLIP, both static and dynamic prompts for both text and image encoders are developed. Due to the unavailability of a publicly accessible implementation for AnomalyCLIP, we report the comparison results against AnomalyCLIP in Section 4. Implementation and reproduction details of other comparison methods are given as follows:

- SAA [5]: SAA is a novel ZSAD model that integrates Grounding DINO [10] and SAM [9] for anomaly detection without any training. Various manual prompts can be adjusted to enhance ZSAD performance. In the case of MVTec AD and VisA datasets, we adhere to the officially provided prompts [5]. For datasets not covered in the original implementation, default prompts in SAA are utilized for ZSAD.
- WinCLIP [8]: WinCLIP represents a SOTA ZSAD method. It devises an extensive array of manual text prompts tailored specifically for anomaly

[5] https://github.com/caoyunkang/Segment-Any-Anomaly

detection and employs a window scaling strategy for anomaly segmentation. We strictly adhere to the text prompts outlined in the original paper.
- APRIL-GAN [6]: APRIL-GAN enhances WinCLIP by employing training on auxiliary AD datam. We adopt the official implementation [6] and adhere to the training settings outlined in the paper, specifically training on both industrial and medical datasets concurrently to improve generalization ability.
- DINOV2 [11]: DINOV2 represents a recent advancement in visual foundation models. We adapt DINOV2 for the ZSAD task by training on auxiliary data like AdaCLIP. Specifically, we utilize the ViT-S/14 architecture[7] as the backbone. Similar to AdaCLIP, we incorporate additional learnable projection layers after the multi-hierarchy patch embeddings and employ the same training set for optimizations. Patch embeddings from the 3rd, 6th, 9th, and 12th layers are selected for multi-hierarchy representations.
- SAM [9]: SAM is recognized as another prominent visual foundation model, primarily crafted for image segmentation tasks. We repurpose SAM for ZSAD by training on auxiliary AD data as well. Specifically, we discard the prompting encoder and mask decoder of SAM and only utilize the backbone of ViT-L architecture[8] for patch embedding extraction. Similar to AdaCLIP, we append trainable projection layers to the patch embeddings from the 6th, 12th, 18th, and 24th layers.
- **AdaCLIP**: As mentioned in the main body, we use the publicly available pre-trained CLIP (ViT-L/14@336px)[9] as the default backbone. We apply the data pre-processing pipeline officially given by CLIP to all images.

Table 4: Comparison between ZSAD-related methods. The proposed AdaCLIP introduces both static and dynamic prompts for the text and image encoders for enhanced ZSAD performace.

Method	Zero-shot Capacity	Supervised Training	Manual Prompts	Learnable Prompts		Prompting Encoder	
				Static	Dynamic	Text	Image
AnomalyGPT [7]	✘	✔	✔	✔	✘	✔	✘
SAA [5]	✔	✘	✔	✘	✘	✘	✘
WinCLIP [8]	✔	✘	✔	✘	✘	✔	✘
APRIL-GAN [6]	✔	✔	✔	✘	✘	✘	✘
AnomalyCLIP [16]	✔	✔	✔	✔	✘	✔	✘
DINOV2 [11]	✔	✔	✘	✘	✘	✘	✘
SAM [9]	✔	✔	✘	✘	✘	✘	✘
AdaCLIP	✔	✔	✔	✔	✔	✔	✔

[6] https://github.com/ByChelsea/VAND-APRIL-GAN
[7] https://github.com/facebookresearch/dinov2
[8] https://github.com/facebookresearch/segment-anything
[9] https://github.com/mlfoundations/open_clip

Table 5: Comparisons between frozen and learnable projection layers. The best performance is in **bold**.

Proj	Medical Domain		Industrial Domain	
	Image-level	Pixel-level	Image-level	Pixel-level
Frozen	(84.4, 81.1)	(79.3, 47.4)	(82.9, 82.6)	(90.1, 41.0)
Learnable	(**94.6**, **89.6**)	(**85.3**, **57.4**)	(**90.2**, **89.6**)	(**94.2**, **50.2**)

4 Comparison with AnomalyCLIP

AnomalyCLIP [16] represents a concurrent ZSAD method, introducing learnable object-agnostic prompts for ZSAD, under the assumption of the existence of generic normality and abnormality in an image from whatever category. Due to differences in experimental settings between AnomalyCLIP and our study, as well as the unavailability of publicly available code for AnomalyCLIP (before our submission date), we opt to evaluate AdaCLIP within the framework of AnomalyCLIP for fair comparisons. Specifically, we employ MVTec [2] as the default auxiliary dataset, whereas evaluations on MVTec AD are conducted using VisA [17] for training. The results are depicted in Table 6 and Table 7. The results clearly demonstrate that the proposed AdaCLIP outperforms AnomalyCLIP in average image-level anomaly detection performance across both industrial and medical domains, primarily attributed to the proposed Hybrid Semantic Fusion module. It should be noted that AdaCLIP slightly lags behind AnomalyCLIP in pixel-level detection performance, as AnomalyCLIP incorporates specific design elements to enhance pixel-level anomaly localization, such as a Diagonally Prominent Attention Map mechanism, V-V self-attention, and improved loss functions. In summary, AdaCLIP achieves comparable performance to AnomalyCLIP, while also providing a more thorough investigation into learnable prompts and emphasizing the importance of tailored prompts for individual images.

In addition, we found that AnomalyCLIP differs from AdaCLIP regarding the design of projection layers. Specifically, AnomalyCLIP utilizes the pre-trained and frozen projection layer from CLIP, whereas our proposed AdaCLIP introduces learnable projection layers. To study their differences, we replaced the original learnable projection layers with frozen pre-trained layers, and the comparison results are presented in Table 5. The results clearly show that frozen projection layers lead to significant drops in all metrics. We attribute these drops to the smaller number of learnable parameters with frozen layers, which may limit the adaptation of CLIP to zero-shot anomaly detection.

5 Comparison with SOTA Full-shot Methods

In this section, we are interested in the performance gap between AdacLIP and the recently published SOTA full-shot methods, such as PatchCore [13] and

Table 6: Comparisons between the proposed AdaCLIP and AnomalyCLIP [16] within the experimental setting of AnomalyCLIP. The results of AnomalyCLIP are directly taken from the original reports. The best performance is in **bold**.

Metric	Dataset	AnomalyCLIP	AdaCLIP
Image-level (AUROC)	MVTec AD	**91.5**	89.6
	VisA	82.1	**83.9**
	MPDD	**77.0**	76.8
	BTAD	88.3	**88.6**
	KSDD	84.7	**94.1**
	DAGM	97.5	**98.3**
	DTD	93.5	**95.5**
	Average	87.8	**89.5**
Pixel-level (AUROC)	MVTec AD	**91.1**	90.3
	VisA	95.5	**95.6**
	MPDD	**96.5**	96.4
	BTAD	**94.2**	92.1
	SDD	90.6	**96.7**
	DAGM	**95.6**	91.0
	DTD	**97.9**	96.9
	Average	**94.5**	94.1

CDO [4]. Since some datasets do not provide normal training data, we conduct experiments on seven public industrial datasets. As Table 8 shows, AdaCLIP achieves comparable anomaly detection and localization performance compared to PatchCore and CDO, and it even outperforms them in some datasets. This illustrates that AdaCLIP can effectively detect anomalies even in unseen categories by training on auxiliary data. With more extensive data and advanced adapting techniques, future ZSAD methods have opportunities to surpass these SOTO full-shot methods, making ZSAD a viable generic anomaly detection solution.

6 Category-Level Quantitative Results

Some datasets contain several categories. In this section, their category-level quantitative results are presented from Table 9 to Table 14 in details.

7 Additional Qualitative Results

Table 7: Comparisons between the proposed AdaCLIP and AnomalyCLIP [16] within the experimental setting of AnomalyCLIP. The results of AnomalyCLIP are directly taken from the original reports. The best performance is in **bold**.

Metric	Dataset	AnomalyCLIP	AdaCLIP
Image-level (AUROC)	HeadCT	**93.4**	91.5
	BrainMRI	90.3	**94.8**
	Br35H	94.6	**97.7**
	Average	92.8	**94.7**
Pixel-level (AUROC)	ISIC	**89.7**	88.3
	ColonDB	**81.9**	79.1
	ClinicDB	82.9	**84.4**
	TN3K	**81.5**	77.4
	Average	**84.0**	82.3

Fig. 3: Failure cases of AdaCLIP. Three categories are illustrated with anomaly detection failures. Each category is depicted with its normal state in the left column, and two cases of logical anomalies in the middle and right columns. The top row presents the input images, while the second row shows the ground truth. The bottom row displays the anomaly maps generated by AdaCLIP.

7.1 Failure Cases

While the proposed AdaCLIP can achieve promising detection results for arbitrary categories without any references, it may fail to detect anomalies lacking structural deviations. Specifically, the anomalies depicted in Figure 3 exhibit no evident structural deviations. Their abnormality stems from their departure from the expected contextual norms, such as the normal positioning of transistors, among others. However, detecting these anomalies without references poses significant challenges. In the future, it may be worthwhile to explore the integration of more intricate textual prompts describing the normal state to enhance the detection of such anomalies in the absence of references.

Table 8: Comparisons between the proposed ZSAD method AdaCLIP and full-shot unsupervised AD methods PatchCore and CDO. The best performance is in **bold**, and the second-best is underlined.

Metric	Dataset	PatchCore [13]	CDO [4]	**AdaCLIP**
Image-level (AUROC, max-F1)	MVTec AD	(**98.8**, **98.3**)	(97.1, 97.0)	(89.2, 90.6)
	VisA	(92.7, 89.8)	(**95.0**, **91.4**)	(85.8, 83.1)
	MPDD	(94.4, 93.5)	(**95.8**, **94.1**)	(76.0, 82.5)
	BTAD	(94.4, **96.6**)	(**97.6**, 94.3)	(88.6, 88.2)
	SDD	(93.6, 76.4)	(96.0, 83.5)	(**97.1**, **90.7**)
	DAGM	(95.0, 93.6)	(95.1, 92.5)	(**99.1**, **97.5**)
	DTD	(**97.5**, **96.4**)	(96.8, 95.7)	(95.5, 94.7)
	Average	(95.2, 92.1)	(**96.2**, **92.6**)	(90.2, 89.6)
Pixel-level (AUROC, max-F1)	MVTec AD	(**98.4**, **62.2**)	(98.2, 60.1)	(88.7, 43.4)
	VisA	(98.6, **43.9**)	(**99.0**, 43.5)	(95.5, 37.7)
	MPDD	98.8, **47.7**	(**99.0**, 46.9)	(96.1, 34.9)
	BTAD	(97.5, 54.4)	(**98.1**, **60.4**)	(92.1, 51.7)
	SDD	(95.6, 36.9)	(**97.9**, 35.7)	(97.7, **54.5**)
	DAGM	(97.2, **59.4**)	(**97.3**, 58.3)	(91.5, 57.5)
	DTD	(98.2, 56.8)	(**98.3**, 59.9)	(97.9, **71.6**)
	Average	(97.7, 51.6)	(**98.2**, **52.1**)	(94.2, 50.2)

7.2 Results in the Industrial Domain

In this section, we provide additional qualitative results in the industrial domain. Further details can be observed in Figure 4 to Figure 23.

7.3 Results in the Medical Domain

This section showcases additional qualitative results in the medical domain, spanning from Figure 24 to Figure 26.

Table 9: Comparisons of ZSAD methods on MVTec AD. The best performance is in **bold**, and the second-best is underlined.

Metric	Category	w/o supervised training		w/i supervised training			
		SAA [5]	WinCLIP [8]	DINOV2 [11]	SAM [9]	APRIL-GAN [6]	**AdaCLIP**
Image-level (AUROC, max-F1)	bottle	(75.5, 89.4)	(99.2, 97.6)	(**99.4**, **98.4**)	(94.6, 96.0)	(92.9, 94.0)	(94.4, 91.6)
	cable	(63.7, 76.0)	(86.5, 84.5)	(49.9, 76.0)	(59.6, 76.0)	(62.6, 77.5)	(**90.6**, **87.8**)
	capsule	(42.0, 90.5)	(72.9, 91.4)	(65.2, 90.5)	(54.5, 90.5)	(79.1, 90.8)	(**91.5**, **92.4**)
	carpet	(99.5, 98.3)	(**100.0**, **99.4**)	(87.6, 90.8)	(80.5, 88.0)	(99.2, 98.3)	(82.1, 86.5)
	grid	(83.7, 86.4)	(**98.8**, **98.2**)	(97.7, 96.4)	(90.4, 90.3)	(89.3, 89.3)	(90.0, 90.8)
	hazelnut	(83.2, 83.3)	(**93.9**, **89.7**)	(43.8, 77.8)	(58.0, 79.1)	(76.8, 81.2)	(80.2, 82.6)
	leather	(99.3, 97.8)	(**100.0**, **100.0**)	(100.0, 99.4)	(90.4, 90.9)	(99.7, 98.9)	(99.8, 99.5)
	metal_nut	(34.8, 89.4)	(**97.1**, **96.3**)	(44.3, 89.4)	(60.6, 89.4)	(45.6, 89.4)	(83.5, 90.5)
	pill	(50.6, 91.6)	(79.1, 91.6)	(69.7, 91.6)	(68.2, 92.5)	(**90.4**, 92.5)	(82.9, **92.8**)
	screw	(46.4, 85.9)	(83.3, 87.4)	(77.5, 85.6)	(68.6, 86.2)	(70.1, 86.3)	(**87.0**, **89.7**)
	tile	(95.7, 93.9)	(**100.0**, **99.4**)	(74.2, 83.6)	(41.9, 83.6)	(93.4, 93.9)	(90.5, 91.4)
	toothbrush	(22.2, 83.3)	(87.5, 87.9)	(64.0, 83.3)	(68.3, 85.7)	(72.2, 84.9)	(**93.6**, **95.2**)
	transistor	(37.0, 57.1)	(**88.0**, **79.5**)	(51.7, 57.1)	(53.3, 57.1)	(72.8, 68.2)	(82.1, 77.5)
	wood	(**99.8**, **99.2**)	(99.4, 98.3)	(97.0, 96.6)	(96.4, 96.7)	(96.8, 95.2)	(98.3, 96.7)
	zipper	(19.4, 88.2)	(91.5, 92.9)	(**93.9**, **93.9**)	(76.0, 88.5)	(93.4, 92.7)	(91.5, 93.9)
	Average	(63.5, 87.4)	(**91.8**, **92.9**)	(74.4, 87.4)	(70.7, 86.0)	(82.3, 88.9)	(89.2, 90.6)
Pixel-level (AUROC, max-F1)	bottle	(66.5, 37.7)	(89.5, 58.1)	(81.6, 57.8)	(**90.5**, 51.0)	(80.8, **60.5**)	(90.4, 54.3)
	cable	(69.2, **30.0**)	(77.0, 19.7)	(60.7, 9.2)	(76.3, 18.1)	(71.7, 18.8)	(**79.8**, 19.6)
	capsule	(62.1, 17.4)	(86.9, 21.7)	(80.3, 23.4)	(**90.4**, 16.0)	(72.8, 29.6)	(82.3, **31.8**)
	carpet	(83.7, 57.8)	(95.4, 49.7)	(**99.3**, **72.6**)	(92.9, 43.5)	(97.1, 70.8)	(97.3, 61.4)
	grid	(63.3, 25.5)	(82.2, 18.6)	(92.3, 35.5)	(86.5, 25.1)	(84.6, 34.1)	(**96.9**, **43.7**)
	hazelnut	(89.8, 47.1)	(94.3, 37.6)	(91.7, 23.0)	(92.8, 26.9)	(95.8, 32.8)	(**97.8**, **51.8**)
	leather	(89.7, **68.8**)	(96.7, 39.7)	(98.3, 54.3)	(89.3, 40.1)	(99.0, 56.5)	(**99.2**, 53.4)
	metal_nut	(64.0, **36.1**)	(61.0, 32.4)	(67.2, 32.6)	(**75.5**, 33.4)	(65.5, 28.8)	(74.3, 34.8)
	pill	(**91.7**, **53.6**)	(80.0, 17.6)	(90.4, 32.8)	(88.3, 24.0)	(87.0, 36.6)	(86.4, 37.6)
	screw	(68.8, 15.0)	(89.6, 13.5)	(98.1, **52.7**)	(97.0, 25.5)	(96.4, 22.1)	(**98.4**, 41.9)
	tile	(86.6, **71.0**)	(77.6, 32.6)	(82.6, 56.8)	(61.2, 21.9)	(80.2, 63.3)	(**88.5**, 61.9)
	toothbrush	(66.8, 8.0)	(86.9, 17.1)	(87.6, 11.5)	(85.7, 10.4)	(90.6, 21.6)	(**94.9**, **31.9**)
	transistor	(66.9, 20.1)	(**74.7**, **30.5**)	(65.2, 17.1)	(70.6, 17.2)	(60.5, 16.5)	(63.2, 17.5)
	wood	(84.3, 63.0)	(93.4, 51.5)	(**95.9**, 62.8)	(91.7, 57.1)	(89.1, **64.2**)	(87.9, 57.9)
	zipper	(78.4, 19.7)	(91.6, 34.4)	(**97.1**, 51.3)	(92.3, 30.5)	(84.3, 41.2)	(93.8, **52.1**)
	Average	(75.5, 38.1)	(85.1, 31.6)	(85.9, 39.6)	(85.4, 29.4)	(83.7, 39.8)	(**88.7**, **43.4**)

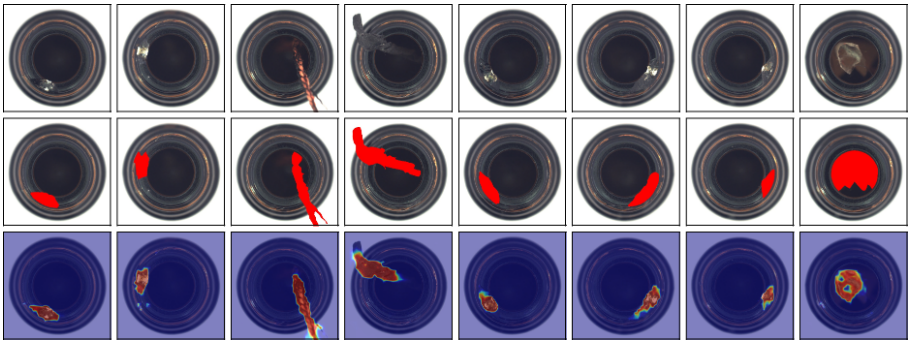

Fig. 4: Visualization of anomaly maps generated by AdaCLIP for the bottle category in MVTec AD. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

Table 10: Comparisons of ZSAD methods on VisA. The best performance is in **bold**, and the second-best is underlined.

Metric	Category	w/o supervised training		w/i supervised training			
		SAA [5]	WinCLIP [8]	DINOV2 [11]	SAM [9]	APRIL-GAN [6]	**AdaCLIP**
Image-level (AUROC, max-F1)	candle	(63.8, 68.6)	(95.4, 89.4)	(81.3, 75.4)	(58.4, 68.1)	(81.0, 74.7)	(**96.0**, **90.2**)
	capsules	(58.1, 76.9)	(85.0, 83.9)	(**92.1**, 88.4)	(38.3, 77.5)	(91.8, **89.2**)	(85.1, 82.1)
	cashew	(87.0, 86.1)	(**92.1**, **88.4**)	(56.0, 80.0)	(43.0, 80.0)	(89.4, 88.3)	(91.8, 86.6)
	chewinggum	(91.9, 88.4)	(96.5, 94.8)	(85.4, 85.8)	(78.7, 80.0)	(**97.0**, **95.9**)	(96.4, 94.8)
	fryum	(39.6, 80.0)	(80.3, 82.7)	(75.5, 84.1)	(71.7, 81.3)	(78.1, 83.2)	(**93.0**, **91.0**)
	macaroni1	(88.7, 82.2)	(76.2, 74.2)	(89.1, **83.4**)	(50.1, 66.7)	(82.2, 78.0)	(**91.6**, 83.1)
	macaroni2	(67.3, 67.6)	(63.7, 69.8)	(**78.7**, **72.6**)	(47.7, 66.7)	(58.7, 66.7)	(64.1, 70.3)
	pcb1	(53.4, 66.9)	(73.6, 71.0)	(65.0, 73.1)	(69.3, 71.2)	(67.5, 68.7)	(**81.1**, **76.9**)
	pcb2	(59.2, 66.7)	(51.2, 67.1)	(54.3, 67.3)	(56.6, 66.7)	(73.9, 71.6)	(**75.3**, **73.7**)
	pcb3	(54.0, 66.5)	(**73.4**, **71.0**)	(57.3, 66.7)	(63.0, 66.9)	(69.1, 67.1)	(64.7, 67.2)
	pcb4	(46.9, 66.5)	(79.6, 74.9)	(72.1, 71.6)	(77.3, 74.1)	(**94.9**, **90.6**)	(93.4, 87.2)
	pipe_fryum	(95.8, **94.6**)	(69.7, 80.7)	(96.1, 93.8)	(88.3, 87.3)	(**96.6**, 94.2)	(**96.6**, 94.4)
	Average	(67.1, 75.9)	(78.1, 79.0)	(75.2, 78.5)	(61.9, 73.9)	(81.7, 80.7)	(**85.8**, **83.1**)
Pixel-level (AUROC, max-F1)	candle	(54.1, 12.8)	(88.9, 22.5)	(98.5, 42.2)	(97.1, 14.6)	(98.5, 41.3)	(**98.9**, **46.6**)
	capsules	(81.5, 39.8)	(81.6, 9.2)	(**98.6**, **62.2**)	(88.7, 6.4)	(97.5, 49.0)	(98.6, 52.8)
	cashew	(56.4, 13.8)	(84.7, 13.2)	(90.7, 10.9)	(90.2, 13.1)	(92.2, 22.7)	(**95.9**, **39.2**)
	chewinggum	(94.9, **83.3**)	(93.3, 41.1)	(99.6, 77.6)	(98.4, 59.3)	(99.4, 78.4)	(**99.6**, 77.9)
	fryum	(92.6, **42.8**)	(88.5, 22.1)	(92.8, 25.7)	(93.4, 26.1)	(93.4, 29.6)	(**94.4**, 30.5)
	macaroni1	(84.1, **42.3**)	(70.9, 7.0)	(98.9, 27.1)	(96.1, 7.4)	(98.8, 29.1)	(**99.5**, 35.0)
	macaroni2	(81.5, **29.9**)	(59.3, 1.0)	(98.0, 21.7)	(95.5, 3.9)	(97.2, 4.6)	(**98.8**, 10.2)
	pcb1	(73.7, **42.1**)	(61.2, 2.4)	(91.3, 10.8)	(89.1, 5.2)	(92.1, 13.1)	(**93.7**, 19.8)
	pcb2	(80.7, 3.5)	(71.6, 4.7)	(**91.3**, 12.1)	(89.3, 8.4)	(90.6, 24.2)	(84.3, **27.7**)
	pcb3	(71.9, 11.2)	(85.3, 10.3)	(89.8, 12.0)	(83.4, 9.4)	(91.0, 23.5)	(**91.8**, **32.2**)
	pcb4	(66.7, 10.2)	(94.4, 32.0)	(94.8, 30.2)	(92.8, 26.9)	(94.7, 37.3)	(**96.1**, **43.3**)
	pipe_fryum	(79.7, **47.1**)	(75.4, 12.3)	(96.1, 31.8)	(**97.1**, 38.1)	(96.7, 35.2)	(94.6, 37.4)
	Average	(76.5, 31.6)	(79.6, 14.8)	(95.0, 30.3)	(92.6, 18.2)	(95.2, 32.3)	(**95.5**, **37.7**)

Table 11: Comparisons of ZSAD methods on MPDD. The best performance is in **bold**, and the second-best is underlined.

Metric	Category	w/o supervised training		w/i supervised training			
		SAA [5]	WinCLIP [8]	DINOV2 [11]	SAM [9]	APRIL-GAN [6]	**AdaCLIP**
Image-level (AUROC, max-F1)	bracket_black	(37.2, 74.6)	(40.7, 74.6)	(**70.2**, 79.3)	(59.2, 75.9)	(50.7, 75.2)	(62.0, **81.4**)
	bracket_brown	(63.4, 81.0)	(33.2, 79.7)	(42.1, 79.7)	(48.8, 80.0)	(70.9, 80.7)	(**71.3**, **81.7**)
	bracket_white	(73.1, 74.1)	(41.8, 67.4)	(57.4, 68.2)	(56.7, 71.6)	(68.0, 71.8)	(**74.7**, **75.0**)
	connector	(31.9, 48.3)	(**78.6**, 65.1)	(35.7, 50.0)	(76.0, 61.1)	(48.1, 50.0)	(69.9, **66.3**)
	metal_plate	(36.9, 84.5)	(**95.5**, 95.1)	(84.9, 87.9)	(93.3, 91.4)	(65.1, 85.4)	(84.6, **95.4**)
	tubes	(13.5, 81.2)	(78.4, 83.1)	(84.3, 84.0)	(44.2, 81.7)	(**93.4**, 93.2)	(93.3, **95.2**)
	Average	(42.7, 73.9)	(61.4, 77.5)	(62.4, 74.9)	(63.0, 77.0)	(66.0, 76.0)	(**76.0**, **82.5**)
Pixel-level (AUROC, max-F1)	bracket_black	(93.9, 1.8)	(46.4, 0.2)	(**96.9**, **22.3**)	(96.0, 4.4)	(96.5, 13.8)	(93.2, 9.1)
	bracket_brown	(66.9, 5.3)	(56.4, 1.4)	(92.0, 13.3)	(89.7, 11.2)	(89.9, 9.0)	(**93.8**, **15.9**)
	bracket_white	(97.1, **30.5**)	(72.2, 1.0)	(97.6, 1.9)	(99.3, 5.4)	(**99.3**, 9.0)	(97.1, 3.9)
	connector	(71.5, 8.2)	(78.8, 10.7)	(93.2, 14.9)	(93.1, 17.7)	(93.5, 26.2)	(**97.4**, **37.7**)
	metal_plate	(73.8, 56.9)	(95.7, 69.7)	(95.9, 72.9)	(**96.9**, **77.2**)	(92.5, 60.8)	(95.8, 72.9)
	tubes	(87.3, 10.6)	(77.6, 9.5)	(98.1, 61.5)	(94.0, 16.8)	(99.0, 64.8)	(**99.2**, **70.1**)
	Average	(81.7, 18.9)	(71.2, 15.4)	(95.6, 31.1)	(94.8, 22.1)	(95.1, 30.6)	(**96.1**, **34.9**)

Table 12: Comparisons of ZSAD methods on BTAD. The best performance is in **bold**, and the second-best is underlined.

Metric	Category	w/o supervised training		w/i supervised training			
		SAA [5]	WinCLIP [8]	DINOV2 [11]	SAM [9]	APRIL-GAN [6]	**AdaCLIP**
Image-level (AUROC, max-F1)	Class01	(6.6, 82.4)	(89.3, 87.6)	(80.3, 84.3)	(**96.2**, **93.9**)	(87.3, 88.2)	(91.6, 90.8)
	Class02	(72.3, 93.0)	(72.2, 93.0)	(**88.2**, 94.3)	(76.1, 93.0)	(75.2, 93.0)	(78.0, **94.6**)
	Class03	(**98.2**, **93.7**)	(43.0, 22.2)	(69.5, 29.2)	(96.1, 70.1)	(93.0, 64.7)	(96.3, 79.1)
	Average	(59.0, **89.7**)	(68.2, 67.6)	(79.3, 69.3)	(**89.4**, 85.7)	(85.2, 82.0)	(88.6, 88.2)
Pixel-level (AUROC, max-F1)	Class01	(49.6, 6.6)	(84.0, 21.8)	(86.0, 44.0)	(**90.6**, 43.7)	(83.9, 41.2)	(87.1, **55.3**)
	Class02	(73.7, 26.4)	(86.4, 33.1)	(**96.0**, **68.7**)	(94.7, 59.5)	(92.2, 58.3)	(92.9, 59.8)
	Class03	(74.0, 11.5)	(47.5, 0.7)	(93.6, 17.4)	(96.2, 37.4)	(92.3, 15.7)	(**96.2**, **40.1**)
	Average	(65.8, 14.8)	(72.6, 18.5)	(91.9, 43.4)	(**93.8**, 46.9)	(89.5, 38.4)	(92.1, **51.7**)

Table 13: Comparisons of ZSAD methods on DAGM. The best performance is in **bold**, and the second-best is underlined.

Metric	Category	w/o supervised training		w/i supervised training			
		SAA [5]	WinCLIP [8]	DINOV2 [11]	SAM [9]	APRIL-GAN [6]	**AdaCLIP**
Image-level (AUROC, max-F1)	Class1	(96.2, 90.9)	(68.4, 67.8)	(81.0, 76.5)	(**96.3**, 93.3)	(91.3, 85.6)	(96.2, **93.8**)
	Class2	(**100.0**, **100.0**)	(99.8, 99.0)	(99.4, 98.0)	(**100.0**, **100.0**)	(99.8, 99.0)	(**100.0**, 99.7)
	Class3	(100.0, 99.3)	(99.0, 95.6)	(**100.0**, **100.0**)	(94.2, 89.1)	(**100.0**, **100.0**)	(**100.0**, **100.0**)
	Class4	(36.8, 66.7)	(89.0, 80.7)	(55.9, 67.1)	(45.1, 66.7)	(67.2, 69.2)	(**96.6**, **89.5**)
	Class5	(**100.0**, 99.7)	(95.2, 89.0)	(99.7, 98.0)	(88.0, 83.5)	(99.7, 99.0)	(**100.0**, **100.0**)
	Class6	(72.0, 66.9)	(99.8, 98.7)	(91.0, 85.7)	(86.3, 80.8)	(99.9, 99.0)	(**100.0**, **100.0**)
	Class7	(99.7, 98.2)	(96.3, 90.3)	(99.3, 98.7)	(98.5, 93.8)	(**100.0**, 99.3)	(**100.0**, **100.0**)
	Class8	(**100.0**, **99.7**)	(74.2, 9.9)	(81.7, 74.3)	(73.2, 70.8)	(97.2, 93.7)	(99.3, 97.8)
	Class9	(**100.0**, **100.0**)	(96.4, 90.4)	(99.5, 96.7)	(49.2, 67.2)	(97.8, 94.6)	(99.6, 96.9)
	Class10	(66.6, 66.7)	(98.9, 94.5)	(99.4, 96.9)	(96.8, 90.4)	(81.9, 78.3)	(**99.5**, **97.4**)
	Average	(87.1, 88.8)	(91.7, 87.6)	(90.7, 89.2)	(82.7, 83.6)	(93.5, 91.8)	(**99.1**, **97.5**)
Pixel-level (AUROC, max-F1)	Class1	(63.9, 39.4)	(76.0, 12.3)	(84.3, 32.1)	(**90.5**, 42.9)	(83.3, 42.0)	(85.4, **47.6**)
	Class2	(74.9, 55.5)	(80.9, 9.3)	(94.3, 57.2)	(**98.9**, 65.8)	(96.7, 63.9)	(97.8, **66.7**)
	Class3	(57.4, 25.9)	(86.8, 19.4)	(87.7, 59.4)	(87.8, 40.0)	(88.1, 65.5)	(**89.8**, **65.7**)
	Class4	(50.0, 2.7)	(**85.6**, 17.4)	(83.6, 18.5)	(79.4, 13.5)	(79.6, 21.0)	(84.8, **23.9**)
	Class5	(61.3, 37.1)	(83.4, 15.4)	(92.3, 64.4)	(90.2, 42.5)	(92.3, 69.5)	(**95.0**, **69.7**)
	Class6	(73.0, 35.3)	(76.9, 19.6)	(**97.8**, 71.2)	(95.0, 70.6)	(96.8, **79.6**)	(94.5, 77.3)
	Class7	(65.5, 45.7)	(85.9, 24.5)	(92.2, 66.1)	(83.2, 51.0)	(89.9, 70.8)	(**94.1**, **72.1**)
	Class8	(48.7, 16.8)	(69.7, 3.5)	(84.5, 21.9)	(83.8, 17.7)	(**88.1**, 56.4)	(87.2, **60.2**)
	Class9	(61.8, 38.2)	(80.1, 2.0)	(**97.6**, 62.4)	(80.3, 3.7)	(94.7, **62.4**)	(91.3, 48.3)
	Class10	(70.4, 29.2)	(87.9, 15.1)	(94.5, **66.7**)	(**97.2**, 59.0)	(93.9, 48.1)	(94.7, 43.8)
	Average	(62.7, 32.6)	(81.3, 13.9)	(90.9, 52.0)	(88.6, 40.7)	(90.3, **57.9**)	(**91.5**, 57.5)

Table 14: Comparisons of ZSAD methods on DTD-Synthetic. The best performance is in **bold**, and the second-best is underlined.

Metric	Category	w/o supervised training		w/i supervised training			
		SAA [5]	WinCLIP [8]	DINOV2 [11]	SAM [9]	APRIL-GAN [6]	**AdaCLIP**
Image-level (AUROC, max-F1)	Blotchy_099	(**100.0**, **100.0**)	(99.3, 99.4)	(80.6, 88.9)	(58.3, 88.9)	(**100.0**, **100.0**)	(**100.0**, 99.4)
	Fibrous_183	(99.1, 98.7)	(97.0, 94.9)	(52.5, 88.9)	(64.6, 88.9)	(99.3, 97.6)	(99.9, 99.4)
	Marbled_078	(98.3, 97.5)	(98.4, 97.5)	(66.4, 90.9)	(89.4, 91.7)	(**100.0**, **100.0**)	(99.8, 99.4)
	Matted_069	(99.3, 98.1)	(97.5, 96.1)	(59.3, 88.8)	(46.8, 88.8)	(**99.4**, **98.1**)	(90.9, 93.8)
	Mesh_114	(82.5, 81.6)	(76.0, 82.5)	(**95.0**, **92.1**)	(81.8, 83.1)	(93.0, 91.1)	(83.2, 84.3)
	Perforated_037	(98.4, 98.1)	(99.5, 98.8)	(**100.0**, **100.0**)	(96.3, 96.8)	(97.1, 95.6)	(92.5, 83.6)
	Stratified_154	(96.3, 96.3)	(97.6, 96.2)	(99.0, 98.1)	(98.5, 98.1)	(**100.0**, **100.0**)	(**100.0**, **100.0**)
	Woven_001	(98.1, 96.4)	(95.7, 93.6)	(99.8, 99.3)	(95.2, 91.9)	(**100.0**, **100.0**)	(**100.0**, **100.0**)
	Woven_068	(94.6, 91.6)	(96.6, 94.3)	(96.7, 94.9)	(**99.7**, **99.4**)	(99.2, 97.4)	(91.8, 93.2)
	Woven_104	(90.2, 93.0)	(98.1, 98.1)	(91.1, 94.1)	(**99.9**, **99.4**)	(99.4, 98.1)	(92.6, 91.1)
	Woven_125	(98.9, 97.5)	(99.4, 98.7)	(94.3, 95.0)	(99.9, 99.4)	(99.9), 99.4	(**100.0**, **100.0**)
	Woven_127	(77.5, 72.9)	(86.1, 78.5)	(94.3, 90.5)	(52.4, 66.7)	(90.6, 84.0)	(**95.8**, **92.8**)
	Average	(94.4, 93.5)	(95.1, 94.1)	(85.8, 93.5)	(81.9, 91.1)	(**98.1**, **96.8**)	(95.5, 94.7)
Pixel-level (AUROC, max-F1)	Blotchy_099	(84.0, 80.3)	(67.3, 11.4)	(97.0, 60.8)	(97.1, 44.9)	(**99.7**, 77.8)	(99.3, **81.1**)
	Fibrous_183	(81.8, 76.1)	(87.2, 28.2)	(96.0, 45.3)	(94.5, 54.2)	(99.5, **78.8**)	(**99.6**, 67.3)
	Marbled_078	(79.7, 71.7)	(78.0, 14.9)	(95.9, 47.2)	(98.1, 71.0)	(99.6, **78.7**)	(**99.7**, 78.4)
	Matted_069	(70.0, 55.9)	(90.2, 17.8)	(89.5, 22.9)	(84.7, 12.4)	(**99.2**, **72.3**)	(96.9, 67.9)
	Mesh_114	(68.7, 50.8)	(76.1, 9.5)	(96.7, **72.6**)	(93.6, 49.7)	(94.7, 66.1)	(**97.3**, 68.7)
	Perforated_037	(80.9, 59.0)	(76.9, 8.4)	(**99.0**, **75.3**)	(95.1, 67.6)	(95.8, 68.0)	(95.8, 70.0)
	Stratified_154	(81.6, 70.4)	(71.8, 26.9)	(99.1, **81.1**)	(98.0, 71.3)	(99.0, 77.4)	(**99.3**, 66.9)
	Woven_001	(80.0, 70.4)	(83.0, 10.2)	(**99.7**, 77.2)	(98.3, 71.9)	(99.6, 77.7)	(99.5, **78.8**)
	Woven_068	(73.4, 49.8)	(92.1, 21.9)	(98.4, 66.5)	(**99.0**, **76.4**)	(97.5, 71.2)	(96.4, 64.6)
	Woven_104	(84.2, 52.8)	(79.4, 18.2)	(98.4, 66.8)	(98.6, **72.0**)	(96.3, 69.2)	(**98.7**, 70.8)
	Woven_125	(75.3, 64.6)	(84.8, 20.2)	(**99.7**, 82.5)	(99.7, 79.2)	(99.7, 82.3)	(99.5, **83.1**)
	Woven_127	(60.3, 25.5)	(66.7, 6.2)	(**94.6**, **62.1**)	(83.9, 10.0)	(93.1, 53.4)	(93.3, 62.0)
	Average	(76.7, 60.6)	(79.5, 16.1)	(97.0, 63.4)	(95.0, 56.7)	(97.8, **72.7**)	(**97.9**, 71.6)

Fig. 5: Visualization of anomaly maps generated by AdaCLIP for the capsule category in MVTec AD. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

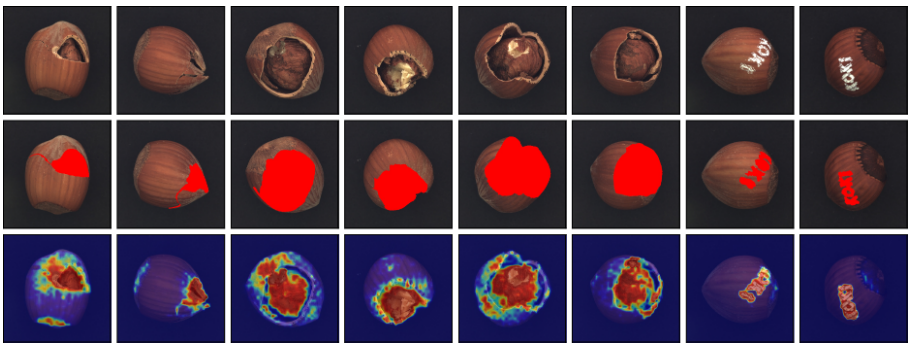

Fig. 6: Visualization of anomaly maps generated by AdaCLIP for the hazelnut category in MVTec AD. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

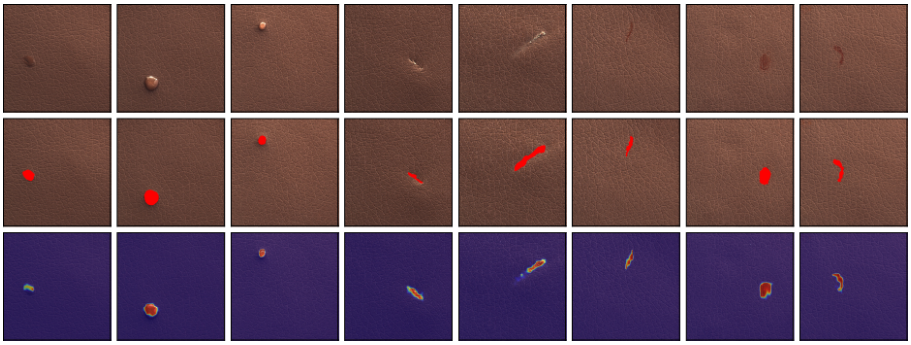

Fig. 7: Visualization of anomaly maps generated by AdaCLIP for the leather category in MVTec AD. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

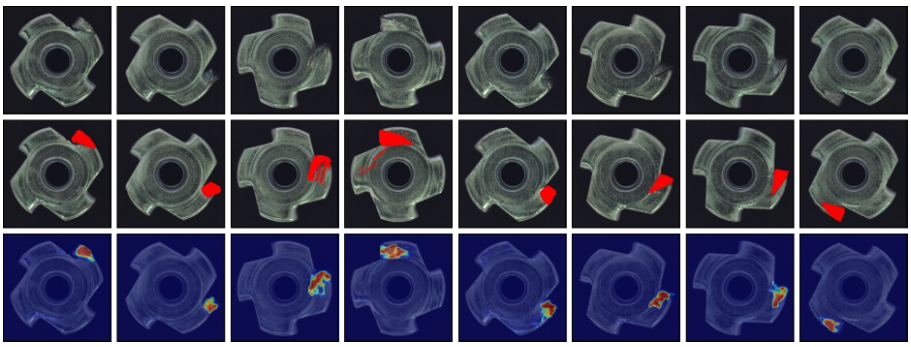

Fig. 8: Visualization of anomaly maps generated by AdaCLIP for the metal_nut category in MVTec AD. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

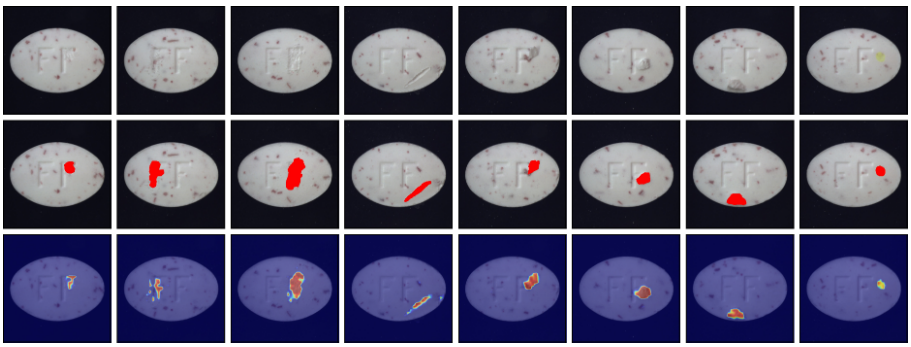

Fig. 9: Visualization of anomaly maps generated by AdaCLIP for the pill category in MVTec AD. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

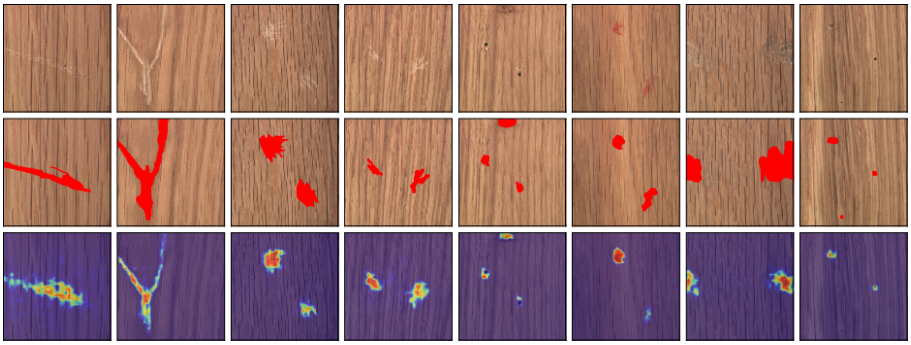

Fig. 10: Visualization of anomaly maps generated by AdaCLIP for the wood category in MVTec AD. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

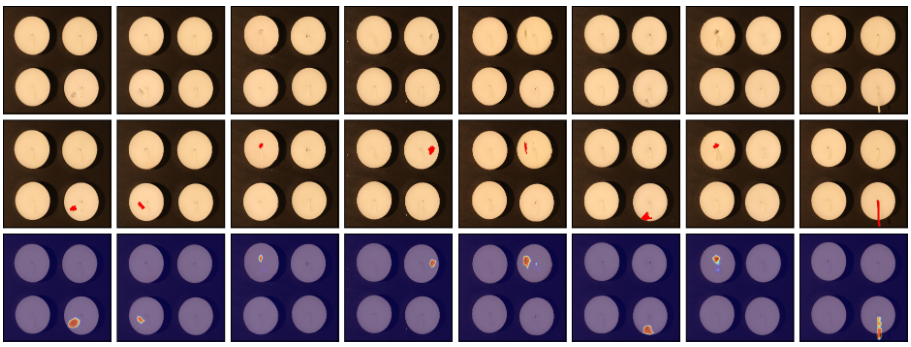

Fig. 11: Visualization of anomaly maps generated by AdaCLIP for the candle category in VisA. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

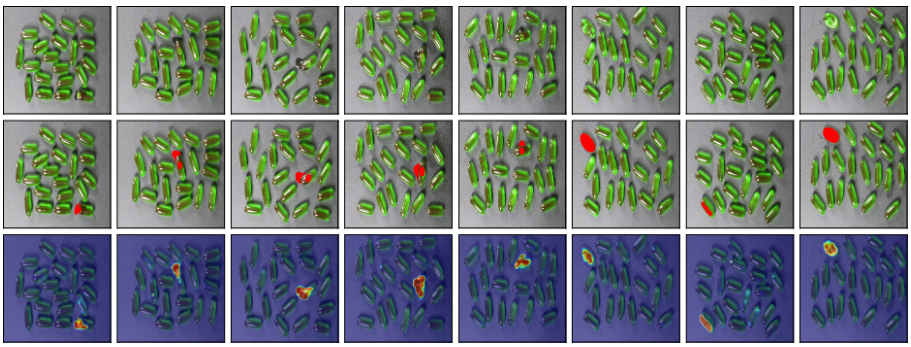

Fig. 12: Visualization of anomaly maps generated by AdaCLIP for the capsules category in VisA. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

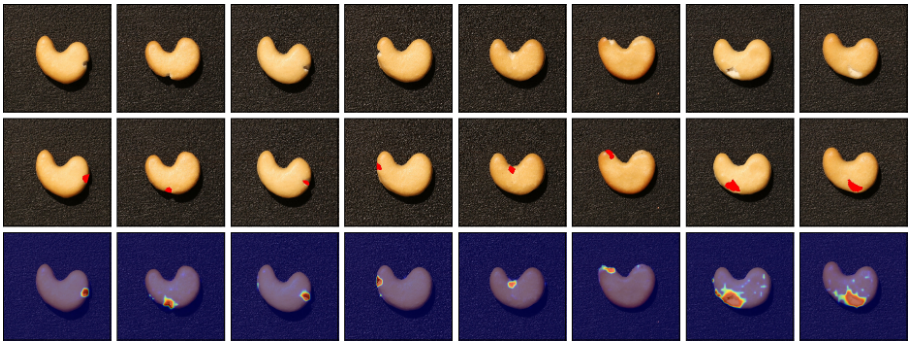

Fig. 13: Visualization of anomaly maps generated by AdaCLIP for the cashew category in VisA. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

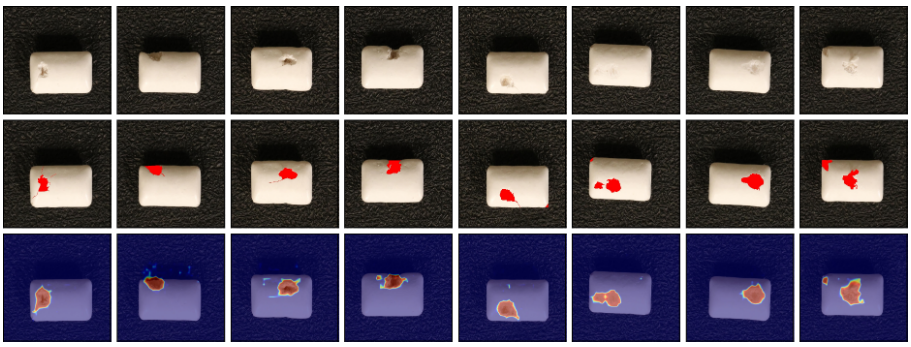

Fig. 14: Visualization of anomaly maps generated by AdaCLIP for the chewinggum category in VisA. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

Fig. 15: Visualization of anomaly maps generated by AdaCLIP for the macaroni1 category in VisA. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

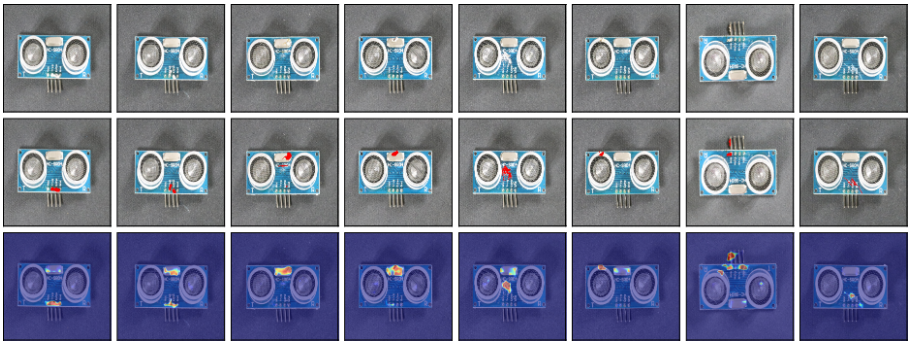

Fig. 16: Visualization of anomaly maps generated by AdaCLIP for the pcb1 category in VisA. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

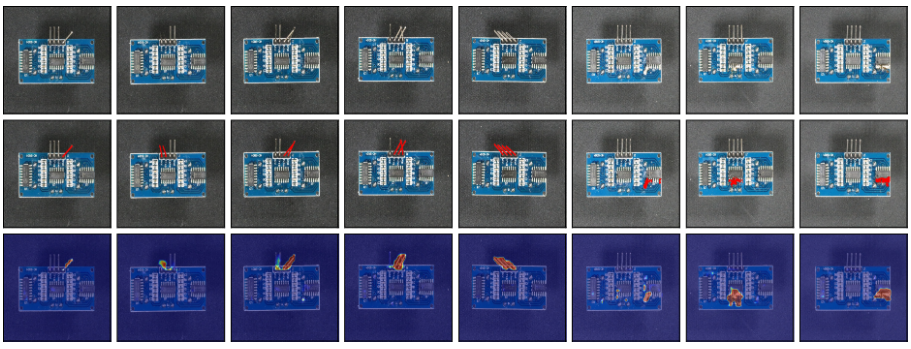

Fig. 17: Visualization of anomaly maps generated by AdaCLIP for the pcb2 category in VisA. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

Fig. 18: Visualization of anomaly maps generated by AdaCLIP for the pcb3 category in VisA. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

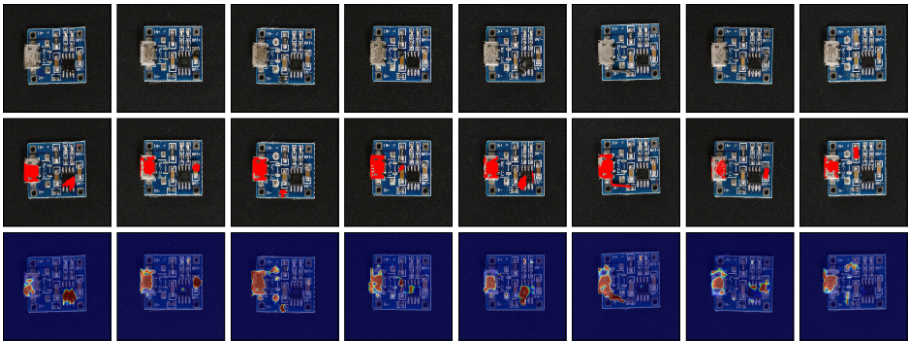

Fig. 19: Visualization of anomaly maps generated by AdaCLIP for the pcb4 category in VisA. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

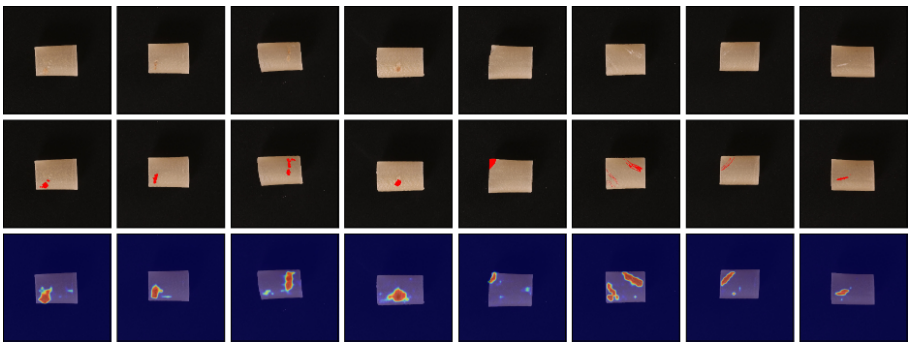

Fig. 20: Visualization of anomaly maps generated by AdaCLIP for the pipe_fryum category in VisA. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

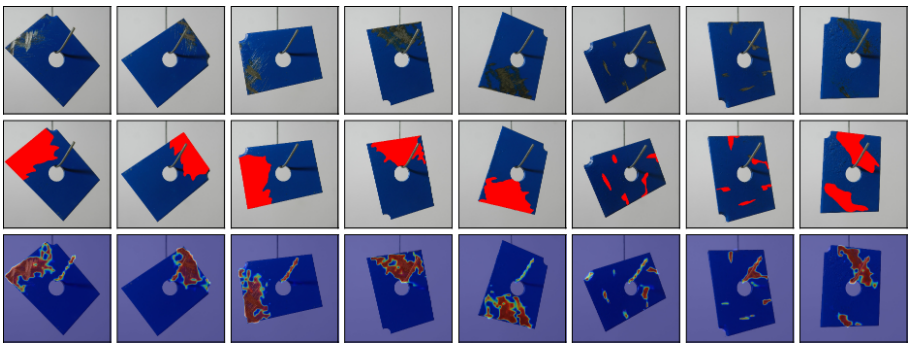

Fig. 21: Visualization of anomaly maps generated by AdaCLIP for the metal plate category in MPDD. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

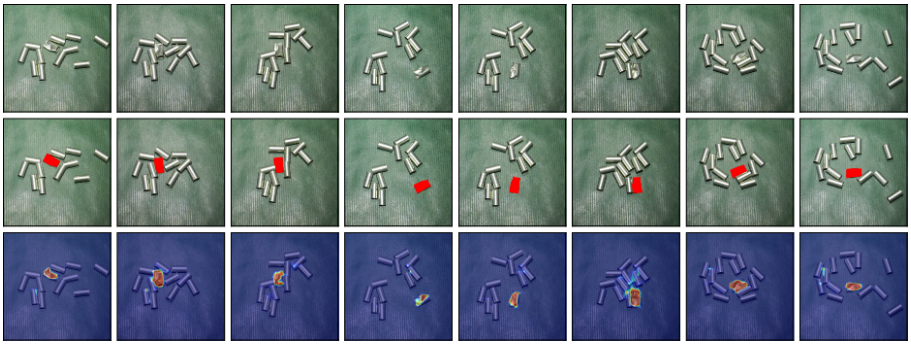

Fig. 22: Visualization of anomaly maps generated by AdaCLIP for the tubes category in MPDD. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

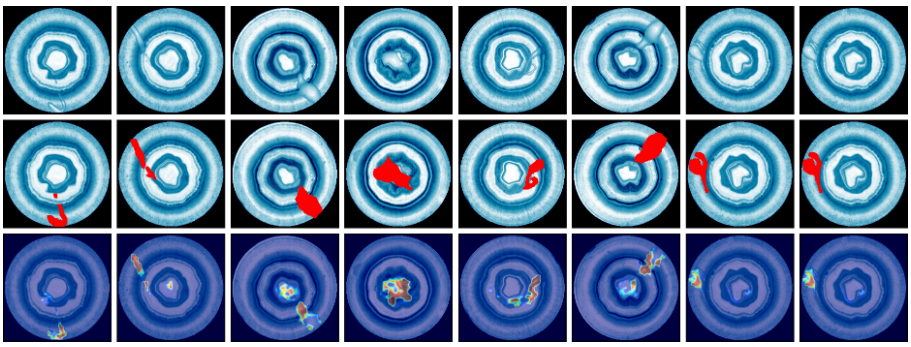

Fig. 23: Visualization of anomaly maps generated by AdaCLIP for the class03 category in BTAD. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

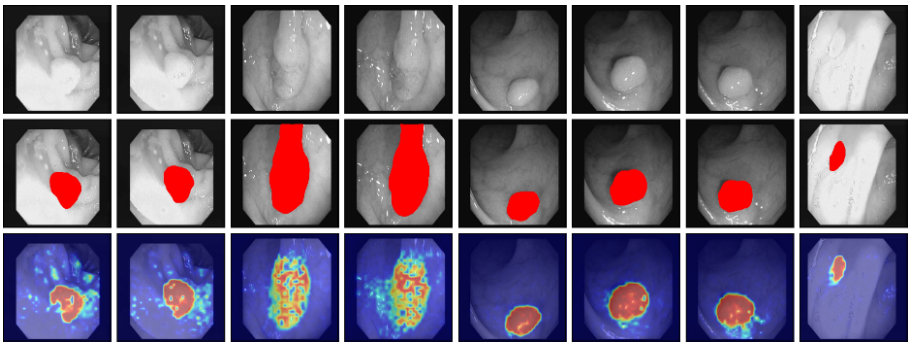

Fig. 24: Visualization of anomaly maps generated by AdaCLIP for the Clinicdb dataset. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

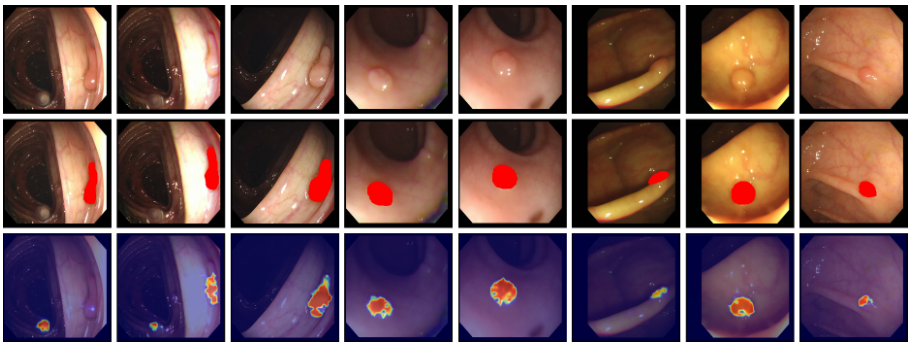

Fig. 25: Visualization of anomaly maps generated by AdaCLIP for the Colondb dataset. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

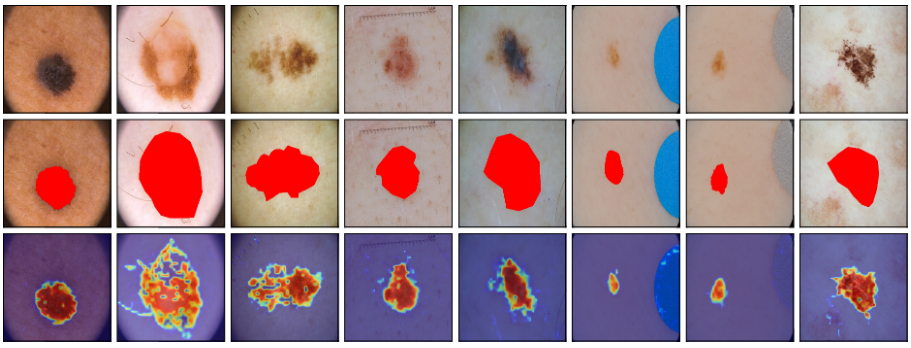

Fig. 26: Visualization of anomaly maps generated by AdaCLIP for the ISIC dataset. The first row displays the input images, while the second row shows the ground truth. The bottom row illustrates the anomaly maps generated by AdaCLIP.

References

1. Arthur, D., Vassilvitskii, S., et al.: k-means++: The advantages of careful seeding. In: Soda. vol. 7, pp. 1027–1035 (2007)
2. Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., Steger, C.: The MVTec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection. International Journal of Computer Vision **129**(4), 1038–1059 (2021)
3. Bernal, J., Sánchez, F.J., Fernández-Esparrach, G., Gil, D., Rodríguez, C., Vilariño, F.: Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. Computerized medical imaging and graphics **43**, 99–111 (2015)
4. Cao, Y., Xu, X., Liu, Z., Shen, W.: Collaborative discrepancy optimization for reliable image anomaly localization. IEEE Transactions on Industrial Informatics pp. 1–10 (2023)
5. Cao, Y., Xu, X., Sun, C., Cheng, Y., Du, Z., Gao, L., Shen, W.: Segment any anomaly without training via hybrid prompt regularization. arXiv preprint arXiv:2305.10724 (2023)
6. Chen, X., Han, Y., Zhang, J.: 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)
7. Gu, Z., Zhu, B., Zhu, G., Chen, Y., Tang, M., Wang, J.: Anomalygpt: Detecting industrial anomalies using large vision-language models. In: Proceedings of the AAAI conference on artificial intelligence (2024)
8. Jeong, J., Zou, Y., Kim, T., Zhang, D., Ravichandran, A., Dabeer, O.: Winclip: Zero-/few-shot anomaly classification and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19606–19616 (June 2023)
9. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollar, P., Girshick, R.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4015–4026 (October 2023)
10. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
11. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision. arXiv:2304.07193 (2023)
12. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
13. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14318–14328 (2022)
14. Tajbakhsh, N., Gurudu, S.R., Liang, J.: Automated polyp detection in colonoscopy videos using shape and context information. IEEE Transactions on Medical Imaging **35**(2), 630–644 (2016)

15. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16795–16804 (2022)
16. Zhou, Q., Pang, G., Tian, Y., He, S., Chen, J.: Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. In: International Conference on Learning Representations (2024)
17. Zou, Y., Jeong, J., Pemula, L., Zhang, D., Dabeer, O.: Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In: European Conference on Computer Vision. pp. 392–408. Springer (2022)