



Figure A1: Qualitative comparisons with state-of-the-art methods on MVTec, VisA and Goods. In both two sub-figures (left and right), (b) and (g) represent query images and their anomaly masks, while (a) represent the corresponding normal image prompts. The predicted anomaly maps are shown using different methods, including (c) WinCLIP+ [26], (d) AnomalyCLIP [76], (e) UniAD [70] and (f) our MetaUAS. Best viewed in color and zoom-in.

Table A1: Quantitative results on **MVTec** with MetaUAS, MetaUAS $\star$  and MetaUAS $\star\star$ .

Methods	Categories	Anomaly Classification			Anomaly Segmentation			
		I-ROC	I-PR	I-F1 <sub>max</sub>	P-ROC	P-PR	P-F1 <sub>max</sub>	P-PRO
<b>MetaUAS</b>	bottle	98.3 $\pm$ 0.8	99.5 $\pm$ 0.2	97.9 $\pm$ 0.8	97.6 $\pm$ 1.6	85.9 $\pm$ 2.6	77.9 $\pm$ 1.5	94.4 $\pm$ 1.5
	cable	90.8 $\pm$ 1.5	95.1 $\pm$ 0.9	86.5 $\pm$ 1.8	95.2 $\pm$ 0.4	64.1 $\pm$ 1.3	63.0 $\pm$ 1.6	85.8 $\pm$ 1.8
	capsule	67.1 $\pm$ 5.2	89.8 $\pm$ 3.0	91.4 $\pm$ 0.5	94.2 $\pm$ 0.7	23.6 $\pm$ 6.5	33.7 $\pm$ 4.3	54.3 $\pm$ 7.0
	carpet	99.8 $\pm$ 0.3	99.9 $\pm$ 0.1	99.3 $\pm$ 0.7	97.4 $\pm$ 0.4	73.7 $\pm$ 1.5	68.1 $\pm$ 1.7	94.4 $\pm$ 0.4
	grid	94.6 $\pm$ 1.2	98.1 $\pm$ 0.5	92.7 $\pm$ 1.3	89.0 $\pm$ 1.2	25.1 $\pm$ 2.7	33.8 $\pm$ 1.6	70.0 $\pm$ 2.9
	hazelnut	97.9 $\pm$ 2.2	98.9 $\pm$ 1.2	95.0 $\pm$ 3.2	98.1 $\pm$ 0.7	66.2 $\pm$ 9.8	60.3 $\pm$ 8.4	87.9 $\pm$ 3.5
	leather	99.9 $\pm$ 0.2	100.0 $\pm$ 0.0	99.7 $\pm$ 0.3	99.7 $\pm$ 0.0	71.2 $\pm$ 0.9	65.4 $\pm$ 0.8	95.8 $\pm$ 0.8
	metal nut	94.4 $\pm$ 3.9	98.6 $\pm$ 1.1	95.0 $\pm$ 1.2	95.0 $\pm$ 0.7	76.9 $\pm$ 4.0	70.9 $\pm$ 2.6	87.1 $\pm$ 1.2
	pill	92.3 $\pm$ 1.4	98.5 $\pm$ 0.2	94.2 $\pm$ 1.1	96.4 $\pm$ 0.6	70.1 $\pm$ 2.9	63.8 $\pm$ 2.3	88.6 $\pm$ 2.1
	screw	63.5 $\pm$ 5.0	84.4 $\pm$ 3.4	85.5 $\pm$ 0.3	92.1 $\pm$ 3.1	8.1 $\pm$ 2.9	14.4 $\pm$ 3.9	72.4 $\pm$ 5.7
	tile	95.6 $\pm$ 0.6	98.5 $\pm$ 0.1	94.4 $\pm$ 1.1	95.3 $\pm$ 0.5	84.6 $\pm$ 1.0	79.5 $\pm$ 0.7	92.0 $\pm$ 0.8
	toothbrush	92.2 $\pm$ 1.8	97.2 $\pm$ 0.8	91.5 $\pm$ 2.9	98.9 $\pm$ 0.2	70.2 $\pm$ 1.6	69.2 $\pm$ 0.4	81.0 $\pm$ 3.6
	transistor	79.7 $\pm$ 6.6	79.3 $\pm$ 6.8	71.9 $\pm$ 4.2	82.4 $\pm$ 3.2	37.2 $\pm$ 5.1	37.7 $\pm$ 5.0	67.1 $\pm$ 3.6
	wood	98.5 $\pm$ 0.3	99.5 $\pm$ 0.1	96.7 $\pm$ 0.8	94.1 $\pm$ 0.4	70.0 $\pm$ 1.7	65.9 $\pm$ 2.1	89.0 $\pm$ 1.1
	zipper	95.9 $\pm$ 2.5	98.5 $\pm$ 1.3	96.1 $\pm$ 1.2	94.5 $\pm$ 1.3	62.1 $\pm$ 2.2	59.5 $\pm$ 1.7	78.7 $\pm$ 2.2
	mean	90.7 $\pm$ 0.7	95.7 $\pm$ 0.6	92.5 $\pm$ 0.3	94.6 $\pm$ 0.2	59.3 $\pm$ 1.4	57.5 $\pm$ 1.1	82.6 $\pm$ 0.6
<b>MetaUAS<math>\star</math></b>	bottle	99.6	99.9	98.4	97.5	85.6	77.5	95.4
	cable	95.3	97.6	91.9	96.3	67.5	65.9	90.2
	capsule	80.1	94.9	93.5	95.8	40.5	48.3	57.6
	carpet	99.6	99.9	98.9	97.0	73.9	68.7	93.2
	grid	96.2	98.7	94.8	90.8	28.7	37.1	75.6
	hazelnut	99.3	99.6	97.9	98.8	74.7	68.0	89.1
	leather	100	100	100	99.7	70.9	65.5	96.4
	metal nut	96.2	99.1	95.2	96.3	81.4	73.3	91.0
	pill	95.3	99.2	94.7	94.8	64.8	59.9	86.3
	screw	84.2	94.5	87.6	95.0	29.4	33.4	61.7
	tile	95.1	98.3	93.4	94.6	83.3	78.8	91.2
	toothbrush	93.6	97.6	92.3	98.9	70.3	70.5	78.6
	transistor	91.0	88.3	79.2	86.0	47.9	48.0	72.8
	wood	98.8	99.6	96.8	94.3	73.0	68.4	88.2
	zipper	89.3	96.3	93.7	94.2	63.7	61.5	79.0
	mean	94.2	97.6	93.9	95.3	63.7	61.6	83.1
<b>MetaUAS<math>\star\star</math></b>	bottle	99.6	99.9	98.4	98.8	87.5	78.1	96.8
	cable	95.5	97.7	91.9	97.1	67.4	66.4	91.6
	capsule	83.4	95.7	92.7	97.8	43.3	49.4	90.0
	carpet	99.8	100	98.9	99.5	80.6	71.0	98.0
	grid	99.6	99.9	98.2	98.2	36.5	39.4	94.7
	hazelnut	100	100	100	99.1	79.1	74.1	92.7
	leather	100	100	100	99.7	71.6	65.5	98.9
	metal nut	97.8	99.5	96.3	96.5	82.1	73.7	92.1
	pill	95.8	99.3	95.0	96.8	68.5	60.9	94.1
	screw	88.2	95.6	91.3	98.4	34.4	33.9	90.5
	tile	96.1	98.6	94.0	98.1	88.4	79.3	95.4
	toothbrush	94.4	97.9	92.3	99.4	72.6	70.9	91.7
	transistor	91.1	88.5	80.5	91.6	51.0	50.6	78.6
	wood	99.0	99.7	96.8	96.7	77.4	69.9	95.0
	zipper	89.4	96.4	93.4	96.0	64.7	61.0	87.9
	mean	95.3	97.9	94.6	97.6	67.0	62.9	92.5

## A Implementation Details.

Following UniAD [70], we extract multi-scale features from all 5 stages of EfficientNet-b4 [57] encoder. In the feature alignment module, the three highest-level features are used to perform query-prompt alignment, and the channel number is reduced to half of one of the original channels before calculating the similarity between query and prompt. Therefore, we derive three aligned features of query and prompt using the feature alignment module. Finally, these three aligned features and two original low-level query features from the first and second stages are fed into the decoder and segmentation head for change segmentation. The model is trained with 30 epochs on 8 Tesla V100 GPUs with batch size 128. We freeze the encoder and optimize the feature alignment module, the

Table A2: Quantitative results on **VisA** with MetaUAS, MetaUAS $\star$  and MetaUAS $\star+$ .

Methods	Categories	Anomaly Classification			Anomaly Segmentation			
		I-ROC	I-PR	I-F1 <sub>max</sub>	P-ROC	P-PR	P-F1 <sub>max</sub>	P-PRO
<b>MetaUAS</b>	candle	84.7 $\pm$ 1.1	85.2 $\pm$ 1.4	79.7 $\pm$ 1.1	99.3 $\pm$ 0.1	60.0 $\pm$ 2.3	57.3 $\pm$ 1.7	63.0 $\pm$ 3.2
	capsules	77.7 $\pm$ 3.9	86.4 $\pm$ 2.3	79.7 $\pm$ 1.7	96.5 $\pm$ 0.7	40.5 $\pm$ 4.1	44.8 $\pm$ 3.4	76.9 $\pm$ 2.5
	cashew	78.9 $\pm$ 5.1	90.1 $\pm$ 2.4	82.2 $\pm$ 1.7	91.1 $\pm$ 1.9	49.4 $\pm$ 3.7	50.4 $\pm$ 2.4	51.8 $\pm$ 4.2
	chewinggum	95.8 $\pm$ 0.2	98.2 $\pm$ 0.1	93.5 $\pm$ 1.1	98.5 $\pm$ 0.4	85.2 $\pm$ 1.5	79.6 $\pm$ 1.1	69.6 $\pm$ 0.9
	fryum	83.5 $\pm$ 2.4	91.9 $\pm$ 1.4	84.0 $\pm$ 1.4	65.2 $\pm$ 5.4	14.9 $\pm$ 5.5	23.0 $\pm$ 6.6	23.9 $\pm$ 2.5
	macaroni1	73.0 $\pm$ 6.0	77.0 $\pm$ 5.4	71.2 $\pm$ 2.0	82.4 $\pm$ 2.1	13.1 $\pm$ 6.3	21.2 $\pm$ 8.0	31.5 $\pm$ 4.0
	macaroni2	60.8 $\pm$ 4.2	59.8 $\pm$ 3.5	68.0 $\pm$ 0.7	89.5 $\pm$ 5.7	2.3 $\pm$ 1.1	7.5 $\pm$ 2.9	56.4 $\pm$ 13.7
	pcb1	75.4 $\pm$ 13.6	76.0 $\pm$ 9.8	75.1 $\pm$ 9.6	98.2 $\pm$ 0.6	66.1 $\pm$ 5.8	62.9 $\pm$ 4.1	71.4 $\pm$ 6.3
	pcb2	76.0 $\pm$ 2.9	76.6 $\pm$ 3.0	72.9 $\pm$ 3.5	94.5 $\pm$ 0.2	30.8 $\pm$ 2.7	39.0 $\pm$ 2.7	66.4 $\pm$ 4.0
	pcb3	77.1 $\pm$ 3.4	79.8 $\pm$ 2.6	72.8 $\pm$ 2.7	97.0 $\pm$ 0.4	42.7 $\pm$ 3.4	42.9 $\pm$ 1.9	58.5 $\pm$ 1.5
	pcb4	95.2 $\pm$ 2.4	95.0 $\pm$ 2.1	89.3 $\pm$ 4.1	97.1 $\pm$ 0.8	41.3 $\pm$ 3.2	45.6 $\pm$ 2.4	69.3 $\pm$ 3.4
	pipe fryum	95.8 $\pm$ 1.4	97.5 $\pm$ 1.2	93.9 $\pm$ 1.3	96.9 $\pm$ 1.0	66.0 $\pm$ 3.4	62.7 $\pm$ 2.7	86.2 $\pm$ 4.3
mean		81.2 $\pm$ 1.7	84.5 $\pm$ 1.4	80.2 $\pm$ 0.7	92.2 $\pm$ 0.7	42.7 $\pm$ 0.8	44.7 $\pm$ 0.6	60.4 $\pm$ 1.5
<b>MetaUAS<math>\star</math></b>	candle	84.4	85.4	78.8	98.9	59.8	57.5	55.9
	capsules	83.4	90.0	82.3	97.1	48.3	50.6	74.7
	cashew	84.3	92.1	85.6	88.8	43.5	45.6	48.8
	chewinggum	95.0	98.0	93.3	98.6	85.9	80.1	70.4
	fryum	84.1	92.8	83.4	67.1	13.7	20.6	22.4
	macaroni1	71.6	74.3	71.1	81.0	4.7	10.4	24.6
	macaroni2	60.3	57.9	67.6	91.0	2.8	9.6	65.1
	pcb1	86.9	84.8	80.8	98.6	78.8	74.5	63.8
	pcb2	79.9	78.7	75.0	95.9	34.9	41.0	64.5
	pcb3	79.7	81.6	73.9	96.4	46.4	46.4	52.5
	pcb4	96.1	95.3	91.1	95.4	43.7	46.9	62.8
	pipe fryum	95.6	97.8	92.5	95.1	64.8	63.5	82.6
mean		83.4	85.7	81.3	92.0	43.9	45.6	57.3
<b>MetaUAS<math>\star+</math></b>	candle	85.8	86.3	79.8	98.3	58.5	57.5	92.9
	capsules	84.5	91.0	82.3	98.3	51.5	51.8	80.4
	cashew	87.7	93.5	88.9	98.5	55.9	50.6	88.1
	chewinggum	95.8	98.3	93.3	99.5	86.0	80.2	85.1
	fryum	89.6	94.9	88.2	96.6	38.6	44.5	81.9
	macaroni1	73.1	76.3	70.8	96.9	7.8	12.5	81.1
	macaroni2	62.6	64.4	67.6	97.7	4.6	10.5	89.6
	pcb1	87.9	86.0	81.7	99.3	81.8	75.7	82.4
	pcb2	80.4	79.1	75.4	97.4	35.1	41.8	77.4
	pcb3	80.7	82.3	75.1	96.8	46.7	47.2	85.7
	pcb4	96.6	95.8	91.6	97.2	43.6	47.1	84.3
	pipe fryum	96.5	98.3	93.0	99.0	66.9	63.5	96.6
mean		85.1	87.2	82.3	98.0	48.1	48.6	85.5

decoder, and the segmentation head with AdamW [25] using weight decay 0.0005 and learning rate 0.0001. We conduct experiments based on the open-source framework PyTorch.

We follow CYWS [50] and use the same procedure for synthesizing the change segmentation dataset. Specifically, given a labeled image from an existing instance segmentation dataset, i.e., MS-COCO, we randomly selected one or several instances and then could make it disappear from the image by inpainting the mask region [56]. It is worth noting that the binary change mask between the inpainted and original images can be freely available because these selected instances have been manually annotated at the pixel level. We keep the dataset setup as similar to CYWS [50] as possible. Specifically, the change segmentation dataset is synthesized using the randomly selected 60,000 images from the MS-COCO training set. For each image, a synthesized image is generated by inpainting a union mask of a random set of labeled instances. Then, all these 60,000 samples are divided into training and validation sets with a ratio of 0.95:0.05. During training, we randomly employ object-level change and local-region change with a probability of 0.5.

## B Competing Methods.

To demonstrate the superiority of MetaUAS, we compare MetaUAS and its variants (MetaUAS $\star$  and MetaUAS $\star+$ ) with diverse state-of-the-art methods. Implementation and reproduction details are summarized as follows:

Table A3: Quantitative results on **Goods** with MetaUAS, MetaUAS $\star$  and MetaUAS $\star\star$ .

Methods	Categories	Anomaly Classification			Anomaly Segmentation			
		I-ROC	I-PR	I-F1 <sub>max</sub>	P-ROC	P-PR	P-F1 <sub>max</sub>	P-PRO
MetaUAS	cigarette box	58.9 $\pm$ 3.8	63.2 $\pm$ 3.6	74.2 $\pm$ 0.5	88.4 $\pm$ 1.3	21.1 $\pm$ 3.3	28.9 $\pm$ 2.9	62.9 $\pm$ 3.3
	drink bottle	55.1 $\pm$ 1.2	59.3 $\pm$ 1.1	70.6 $\pm$ 0.2	92.4 $\pm$ 0.7	7.3 $\pm$ 1.6	12.8 $\pm$ 2.0	64.6 $\pm$ 0.7
	drink can	52.1 $\pm$ 3.4	48.4 $\pm$ 2.0	66.7 $\pm$ 0.0	86.6 $\pm$ 1.4	7.6 $\pm$ 1.1	14.0 $\pm$ 0.8	58.4 $\pm$ 0.5
	food bottle	55.0 $\pm$ 1.4	64.4 $\pm$ 0.5	75.0 $\pm$ 0.1	89.9 $\pm$ 0.3	8.4 $\pm$ 0.6	14.2 $\pm$ 0.6	59.4 $\pm$ 1.0
	food box	52.9 $\pm$ 2.6	65.6 $\pm$ 2.5	77.7 $\pm$ 0.3	86.4 $\pm$ 1.7	4.4 $\pm$ 0.7	8.3 $\pm$ 1.1	57.2 $\pm$ 2.3
	food package	52.7 $\pm$ 1.7	50.4 $\pm$ 1.8	64.8 $\pm$ 0.1	87.5 $\pm$ 1.5	2.8 $\pm$ 0.6	5.7 $\pm$ 1.4	51.6 $\pm$ 3.2
	mean	54.5 $\pm$ 1.0	58.5 $\pm$ 0.4	71.5 $\pm$ 0.1	88.5 $\pm$ 0.6	8.6 $\pm$ 0.7	14.0 $\pm$ 0.7	59.0 $\pm$ 1.3
MetaUAS $\star$	cigarette box	98.9	99.2	96.0	98.7	78.0	73.8	88.0
	drink bottle	85.2	86.7	80.9	98.9	62.2	61.3	68.1
	drink can	96.7	97.1	91.5	93.8	44.9	53.7	57.9
	food bottle	90.1	93.1	86.2	97.1	50.5	52.1	70.1
	food box	86.9	92.4	84.5	98.3	54.6	54.8	67.5
	food package	82.7	81.8	74.8	97.5	32.1	37.0	73.6
	mean	90.1	91.7	85.7	97.4	53.7	55.5	70.8
MetaUAS $\star\star$	cigarette box	97.5	96.3	96.4	98.6	74.9	74.0	95.3
	drink bottle	85.4	86.8	81.3	98.8	58.7	61.4	87.6
	drink can	97.2	97.5	91.8	96.7	42.8	54.9	86.5
	food bottle	90.4	92.9	86.7	97.5	44.1	52.2	88.6
	food box	85.2	87.4	84.1	97.8	46.5	54.6	85.5
	food package	83.6	78.1	76.9	97.9	27.0	37.4	84.4
	mean	89.9	89.9	86.2	97.9	49.0	55.8	88.0

**CLIP** [42] is a powerful vision-language model, and it has a strong zero-shot generalization ability. Following previous works, we use two classes of text prompt templates, “A photo of a normal [cls]” and “A photo of an anomalous [cls]”, where “cls” denotes the target class name. The anomaly score is computed by cosine similarity between textual features and the class token of a query image. For anomaly segmentation, we extend the above computation from class tokens to local patch tokens.

**WinCLIP** [26] is a zero-shot anomaly segmentation method based on CLIP. A large set of hand-crafted textual prompts is designed for anomaly classification. A window scaling strategy is used to obtain better anomaly segmentation. We keep all parameters the same as in their paper. Note that no official implementation of WinCLIP is available, our results are based on an unofficial implementation<sup>1</sup>.

**WinCLIP+** [26] combines the complementary prediction from both language-guided and visual-based for better anomaly classification and segmentation. The language-guided prediction is the same as WinCLIP. For visual-based prediction, it first simply stores multi-scale features for given few-shot normal images and retrieves the memory features based on the cosine similarity. The final anomaly score is derived by averaging these two scores.

**AnomalyCLIP** [76] learns object-agnostic text prompts that capture generic normality and abnormality in an image regardless of its foreground objects. But AnomalyCLIP requires fine-tuning on an auxiliary domain dataset including normal and anomaly images. AnomalyCLIP is a zero-shot anomaly classification and segmentation method, and it is capable of recognizing any anomalies. We use the official model to report performance for anomaly classification and segmentation.

**UniAD** [70] is a unified unsupervised anomaly segmentation method for addressing multi-classes anomalies with a single model. Different from most zero-/few-shot anomaly segmentation models, UniAD learns feature reconstruction with a transformer-based encoder-decoder architecture on all normal training images. We use the official code to train the specific model for each dataset.

**PatchCore** [47] is a popular unsupervised anomaly classification method that enjoys training-free. For a fair comparison, we modify the official implementation in two folds. First, we replace the original WideResNet-50 backbone with EfficientNet-b4. Second, the memory-bank construction is limited to only one normal image for each class.

<sup>1</sup><https://github.com/zqhang/Accurate-WinCLIP-pytorch>