Appendix of AG-VAS: Anchor-Guided Zero-Shot Visual Anomaly Segmentation with Large Multimodal Models

This appendix includes the following five parts: 1) the construction of the Anomaly-Instruct20K dataset in Section A; 2) additional experimental details in Section B; 3) further ablation studies and analyses in Section C; 4) more detailed qualitative and quantitative results in Section D; 5) limitations of our method in Section E.

A. Construction of Anomaly-Instruct20K

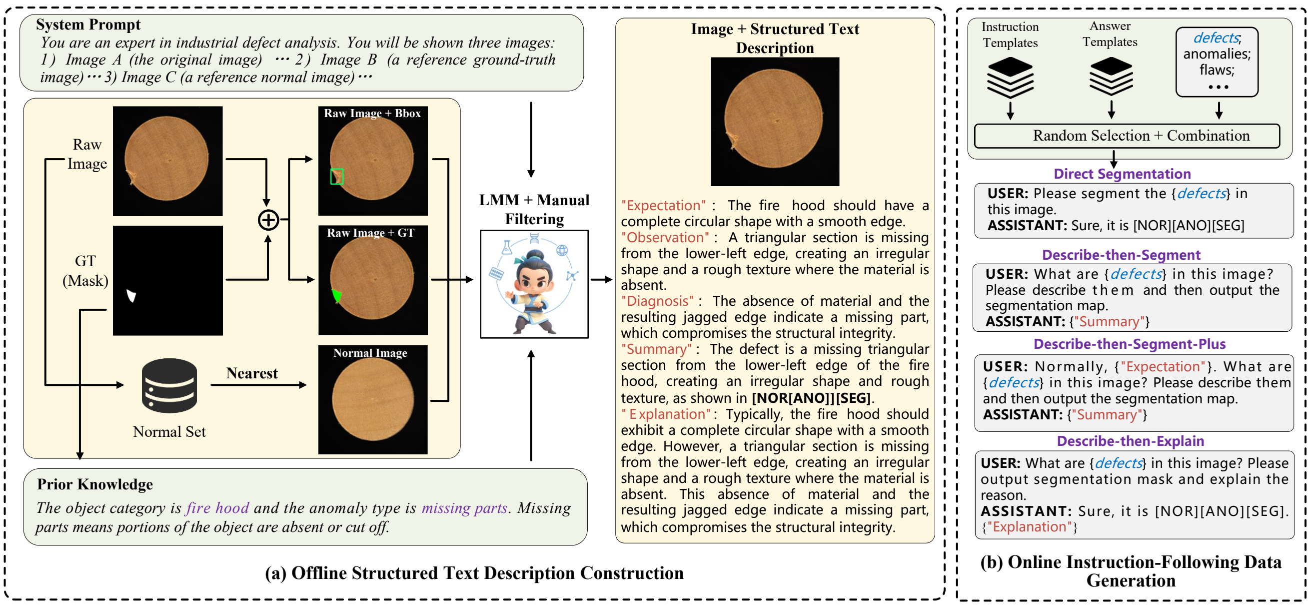

Figure A.1. The Anomaly-Instruct20K generation pipeline for ZSAS.

A.1. Motivation of Anomaly-Instruct20K

In this work, we construct Anomaly-Instruct20K, an instruction-tuning dataset specifically designed for zero-shot anomaly segmentation (ZSAS). The motivation for developing this dataset is twofold. First, it introduces rich world knowledge into existing large multimodal models (LMMs), enabling them to recognize anomalies, including their names, shapes, and locations, and to encode such information into the proposed anchor embeddings through semantic interaction flows within large language models (LLMs). Second, it equips the model with instruction-guided segmentation capability, allowing it to understand user intent, perform interactive reasoning, and ultimately produce accurate anomaly localization.

To construct Anomaly-Instruct20K, we use samples from three public datasets as data sources: RealIAD [34], GoodsAD [39], and DTD-Synthetic [2]. Among these, RealIAD [34] offers a large-scale industrial anomaly detection benchmark comprising 30 categories and multiple anomaly types, which serves as a key foundation for building our dataset. The GoodsAD [39] dataset focuses on industrial goods, and the brand-dependent variations in product appearance contribute to a large expansion in the number of object categories. DTD-Synthetic [2] is a specialized texture synthesis dataset that enhances the model's ability to understand and handle texture-level anomalies.

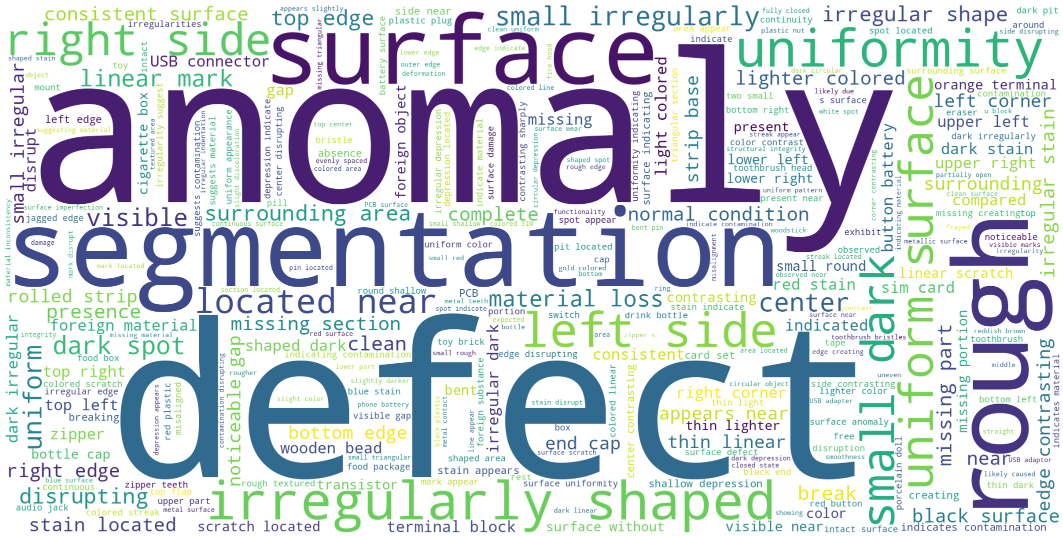

Figure A.2. Word cloud of Anomaly-Instruct20K. Note that the special anchor words [NOR], [ANO], and [SEG] have been removed to reflect the true data distribution.

A.2. Pipeline for Constructing Anomaly-Instruct20K

Table A.1 illustrates the construction pipeline of the proposed Anomaly-Instruct20K dataset, which consists of two stages: 1) offline generation of structured text descriptions from raw images using LMMs (e.g., fields such as "Expectation" and "Observation") to provide source data; and 2) online generation of instruction-following data during training, where carefully designed templates are applied to the source data to produce realistic fine-tuning samples for various segmentation tasks.

Offline Structured Text Description Construction. Unlike existing approaches [15, 29] that use LMMs to generate a single text description and then convert it into task-specific datasets, we aim to construct structured annotations for each image, as illustrated in Figure A.1 (a). Specifically, the structured data is designed to cover the following five aspects: *Expectation*, which describes the object's ideal normal appearance; *Observation*, detailing concrete visual deviations (e.g., location, shape, size, or texture irregularities); *Diagnosis*, explaining why these deviations violate normal material or structural consistency; *Summary*, converting the detected cues into a concise segmentation instruction and explicitly encoding the spatial and appearance attributes required to ground the anchor tokens [NOR], [ANO], and [SEG]; and *Explanation*, synthesizing all components into a coherent reasoning paragraph.

Our method fully leverages the image understanding and text generation capabilities of the science-oriented LMM Intern-S1-241B [3]. We employ carefully designed language prompts together with human filtering to ensure the reliability of the generated content. Compared to commercial closed-source models such as GPT-4V [1], Intern-S1 is not only open-source but also demonstrates stronger capability in understanding scientific data, making it particularly suitable for our task. To further enhance anomaly understanding, we provide Intern-S1 with both visual and textual prompts. For visual inputs, three components are jointly fed into the LMM: the original image with the anomalous region highlighted by a green bounding box, the original image with the anomalous region masked, and a normal reference image from the same category. The normal reference image is selected following MMAD [15], using the SSIM score [35] and Bhattacharyya distance [7] to retrieve the most similar sample. For textual inputs, in addition to the system prompt, we provide prior knowledge in the form of a single sentence that specifies the object category, anomaly type, and corresponding explanation.

In designing the system prompt, we impose explicit constraints on the LMM's textual outputs, such as prohibiting any direct reference to ground truth annotations or prior knowledge. In addition, we provide an in-context example to guide the model toward generating higher-quality and rule-compliant responses. Detailed formulations of the system prompt are presented in Figure A.5. The final structured data further undergoes an additional round of manual verification, involving 10 annotators and requiring over 120 working hours.

Table A.2 illustrates the word cloud of Anomaly-Instruct20K. The results show that the dataset contains rich descriptions of anomaly categories (e.g., material loss), appearance attributes (e.g., dark spot), and spatial locations (e.g., left side). Such diverse semantic information facilitates LMMs in better understanding anomaly characteristics, thereby enabling more accurate localization.

Table A.1. Construction of the Training Dataset.

Dataset	Name	Type	Task	Number	Sampling Rate
General Segmentation Datasets	ADE20K [40] COCO-Stuff [9]	Semantic Segmentation	Direct Segmentation Direct Segmentation	∼200k	0.4
	PACO-LVIS [27] PartImageNet [13] PASCAL-Part [11]	Part Semantic Segmentation	Direct Segmentation Direct Segmentation Direct Segmentation		
	refCLEF [16] refCOCO [16] refCOCO+ [16] refCOCOg [23]	Referring Segmentation	Direct Segmentation Direct Segmentation Direct Segmentation Direct Segmentation		
Anomaly-Instruct20K	RealIAD [34] DTD-Synthetic [2] GoodsAD [39]	Anomaly Segmentation	Segment-then-explain Describe-then-segment	∼20k	0.25
Anomaly-Seg20K	Road [30] MIAD [4] GC [22] DTD-Synthetic [2] ZJU-Leaper [38] RealIAD [34]	Anomaly Segmentation	Direct Segmentation	∼20k	0.25
Visual Question Answering Datasets	LLaVA-150k [20] WebAD [36]	Conversation	General Conversation Anomaly Conversation	∼222k	0.1

Online Instruction-Following Data Generation. After generating structured text descriptions for each image in the first step, how can we leverage these source data to train a model with instruction-following zero-shot anomaly segmentation (ZSAS) capability? As shown in Fig. A.1(b), instruction-following data are generated online during training by combining the source data with randomly selected templates. For the three segmentation settings (direct segmentation, describe-then-segment, and segment-then-explain), we design dedicated sets of instruction templates together with their corresponding answer templates. To further enrich the variability of both instructions and responses, GPT-4 is employed to expand the template collections.

Taking the describe-then-segment setting as an example, the instruction template `"What are the {class_name} in this image? Please describe them and then output the segmentation map"` is randomly selected from the instruction template set. The placeholder {*class_name*} is filled with a term randomly sampled from the anomaly target set $\mathcal{S}$, which includes entries such as *defects*, *anomalies*, and *flaws*. For this task, the corresponding response is directly replaced with the *Summary* from the source data, which inherently contains the anchor words [NOR],[ANO] and [SEG]. Subsequently, the image–instruction–response triplet is used as a training sample for instruction tuning. Detailed instruction and response templates can be found in Figures A.5,A.6,A.7.

B. Experimental Details

B.1. Details of the Datasets

Training Datasets. Table A.1 presents the auxiliary datasets used during the training of the proposed AG-VAS, which consist of four components: general segmentation datasets, Anomaly-Instruct20K, Anomaly-Seg20K, and visual question answering (VQA) datasets. For the general segmentation datasets, a setup similar to LISA [18] is adopted, including semantic segmentation datasets [9, 40], part semantic segmentation datasets [11, 13, 27], and referring segmentation datasets [16, 23]. They provide abundant samples for training the absolute semantic anchor [SEG], enabling it to more easily align with specific anomaly semantics, such as *holes*, *scratches*, and *pits*. As described in Section A, Anomaly-Instruct20K provides the model with world knowledge, allowing it to comprehend what anomalies are, where they appear, and their visual characteristics. Additionally, it equips the LMM to perform both Segment-then-Explain and Describe-then-Segment tasks. For Anomaly-Seg20K, we randomly sampled 20k images from several industrial anomaly segmentation datasets [2, 4, 22, 30, 34, 38], ensuring a balanced distribution of normal and anomalous samples. These datasets are used for direct segmentation tasks without requiring any additional annotations. Anomaly-Seg20K and Anomaly-Instruct20K are employed together to train the relative semantic anchors [NOR]/[ANO] and the absolute semantic anchor [SEG]. *This allows the model to utilize im-*

age background information to detect relative anomalies, while simultaneously exploiting world knowledge and contextual semantics to precisely localize absolute anomalous regions or objects (e.g., holes, bent wires). Finally, the VQA datasets are further utilized to preserve the model's language description and dialogue capabilities, with LLaVA-150k [20] used for general conversation and WebAD [36] for anomaly-specific conversation.

Testing Datasets. Following [25], we evaluate AG-VAS on six datasets spanning industrial and medical domains under the zero-shot anomaly segmentation (ZSAS) setting. In the industrial domain, MVTec-AD [5], KSDD2 [8], and RSDD [37] are used for comprehensive evaluation. MVTec-AD, a widely adopted benchmark, contains 15 categories of textures and objects. KSDD2 and RSDD are real-world industrial datasets, where KSDD2 evaluates anomaly localization under low-contrast backgrounds, and RSDD focuses on defect detection on railway surfaces. In the medical domain, ISIC [12] is used for skin lesion analysis, while ClinicDB [6] and ColonDB [32] are used for colon polyp segmentation. Notably, under the zero-shot setting, the training and testing categories are disjoint, with substantial domain gaps.

B.2. Details of Implementations

Training details. By default, we adopt LLaVA-OneVision-7B [19] as our LMM backbone, equipped with a semantic image encoder (siglip-patch14-384), while SAM-ViT-H [17] serves as the pixel image encoder for mask prediction. Training is implemented using DeepSpeed [28] with the AdamW optimizer [21] at a learning rate of 0.0003. The learning rate is scheduled using WarmupDecayLR, with the warmup period configured to 100 iterations. The weights λ_{bce} and λ_{dic} for the segmentation losses are empirically set to 0.5 and 2.0, respectively. During instruction tuning, the LLM is fine-tuned using LoRA [14] with rank 16 and a global batch size of 80. The trainable parameters include the LoRA weights, the proposed SPAM and AGMD modules, as well as the LLM token embeddings and output head. To address the imbalance among the four training data sources, we assign sampling probabilities of 0.4, 0.25, 0.25, and 0.1 at each iteration, as summarized in Table A.1. Due to the imbalance in the sizes of the four main training datasets, we assign sampling probabilities of 0.4, 0.25, 0.25, and 0.1 for each step to balance the different data types and training tasks, as summarized in Table A.1. In addition, to enable the relative semantic anchors [NOR]/[ANO] to learn how to localize anomaly regions based on the surrounding image context, we train them exclusively on Anomaly-Instruct20K and Anomaly-Seg20K. The absolute semantic anchors, in contrast, are trained jointly using all segmentation datasets. The full training of the 7B model requires approximately 30 hours on four A100 GPUs with 40 GB memory each.

Testing details. During inference, we by default employ the direct-segmentation approach, corresponding to the *Implicit Mode* described in Section 3.1 of the main text. In this mode, the model uses its learned world knowledge and contextual semantics to perform implicit internal reasoning and directly output the segmentation mask of the target anomalous regions. Specifically, the default instruction used during inference is `"Please segment the anomalies in this image"`. After the LLM autoregressively generates a response containing the anchor tokens [NOR], [ANO], and [SEG], we extract the final-layer hidden states corresponding to these tokens to obtain the anchor embeddings. These embeddings are then fed into the proposed AGMD module to produce the segmentation probability maps P_{ano} and P_{seg}. The two probability maps are then averaged using a weighting coefficient of $\alpha = 0.5$ to obtain the final anomaly map. Finally, a threshold of 0.5 is applied to obtain the binary segmentation map $\mathbf{M}$.

B.3. Details of Evaluation Metrics

In this work, four metrics are employed to evaluate the model's ZSAS performance on novel categories. This includes two threshold-independent metrics, which have been widely adopted in previous CLIP-based methods [10, 25, 41]: average precision (AP) and the maximum F1 score at the optimal threshold (F1-Max). However, in practical industrial and medical anomaly segmentation scenarios, the objective is not merely to produce anomaly score maps, but to obtain accurate binary masks of anomalous regions. To this end, we further introduce the pixel-level intersection over union (IoU_{ano}) to quantify the overlap between the predicted binary mask $\mathbf{M}$ and the ground-truth mask $\mathbf{G}$ for anomalous samples:

$$\text{IoU}_{\text{ano}} = \frac{1}{N_{\text{ano}}} \sum_{i=1}^{N_{\text{ano}}} \frac{|\mathbf{M}_i \cap \mathbf{G}_i|}{|\mathbf{M}_i \cup \mathbf{G}_i|}, \tag{A.1}$$

where N_{ano} denotes the total number of anomalous samples in the test set, and $\mathbf{M}_i, \mathbf{G}_i \in \{0,1\}^{H\times W}$ are the binarized prediction mask and ground-truth mask for the i-th anomalous image, with H and W being the height and width of the image, respectively.

To assess the model's ability to reject normal samples by predicting an empty mask, we introduce an additional metric,

Table A.2. Ablation on different pixel image encoders on the MVTec-AD.

Pixel Image Encoder	AP	F1-Max	IoU_{nor}	IoU_{ano}
CLIP(ViT-L-14-336)	48.7	51.8	75.2	39.4
SigLIP(ViT-L-14-384)	46.8	48.5	72.1	37.1
DINOv2 (ViT-L-14)	53.2	53.8	56.5	44.1
DINOv2 (ViT-G-14)	57.8	**57.3**	64.2	49.2
DINOv3 (ViT-H-16)	**59.6**	58.9	74.6	**49.4**
SAM-FT (ViT-H-16)	39.9	40.6	60.4	32.3
SAM (ViT-H-16)	51.0	52.7	**87.7**	44.8

Table A.3. Comparison of average inference time and maximum GPU cost on MVTec-AD. The best results are shown in **bold**.

Method	Base Model	AP	IoU_{ano}	GPU (GB)	Time (s)
PaDT	Qwen2.5-VL-7B	6.8	16.4	19.4	1.2
PixelLM	LLaVA-13B	13.6	11.9	52.8	6.9
LISA	LLaVA-7B	9.8	10.2	**15.6**	**0.6**
LISA*	LLaVA-13B	37.6	31.8	27.8	1.5
LISA*	LLaVA-OneVision-7B	41.0	32.3	19.9	0.97
AG-VAS	LLaVA-OneVision-7B	**51.0**	**44.8**	20.8	1.0

IoU_{nor}, which equals 1 if the predicted mask for a normal sample is empty, and 0 otherwise:

$$\text{IoU}_{\text{nor}} = \frac{1}{N_{\text{nor}}} \sum_{i=1}^{N_{\text{nor}}} \mathbf{1}\{\mathbf{M}_i = \mathbf{0}\}, \tag{A.2}$$

where N_{nor} denotes the number of normal samples in the test set, $\mathbf{M}_i$ is the predicted mask for the i-th normal image, $\mathbf{0}$ represents an empty mask (all pixels are zero), and $\mathbf{1}\{\cdot\}$ is the indicator function that returns 1 if the condition is true and 0 otherwise.

B.4. Details of the Model Architecture

The proposed AG-VAS adopts LLaVA-OneVision-7B [19] as its default base LMM, with Qwen2 [33] as an alternative LLM. Since Qwen2's text embedding space has a default dimensionality of 3584, we use a Token Refiner composed of two linear layers to map the extracted anchor embeddings into the 256-dimensional SAM [17] image embedding space. The refined anchor embeddings $h_{\text{seg}}, h_{\text{nor}}, h_{\text{ano}}$ are concatenated with three learnable vectors $t_{\text{seg}}, t_{\text{nor}}, t_{\text{ano}}$ and then fed into the proposed AGMD, enabling the segmentation targets to be located in the pixel embedding space based on the anchors. Figure A.3 illustrates the architecture of the proposed AGMD. It primarily relies on two bidirectional attention blocks adapted from SAM [17] to facilitate interaction between the anchor and pixel embeddings. This design allows the anchor embeddings from the LMM to more effectively absorb information from the pixel embedding space. Conversely, the pixel embeddings become better aligned with the semantic embedding space through cross-modal attention. Finally, the three output learnable vectors $t'_{\text{seg}}, t'_{\text{nor}}, t'_{\text{ano}}$, which have fully integrated information from both the anchor and pixel embeddings, are multiplied with the refined pixel features f'_p and then passed through softmax and sigmoid activation functions to obtain the relative probability map $[P_{\text{nor}}, P_{\text{ano}}]$ and the absolute probability map P_{seg}.

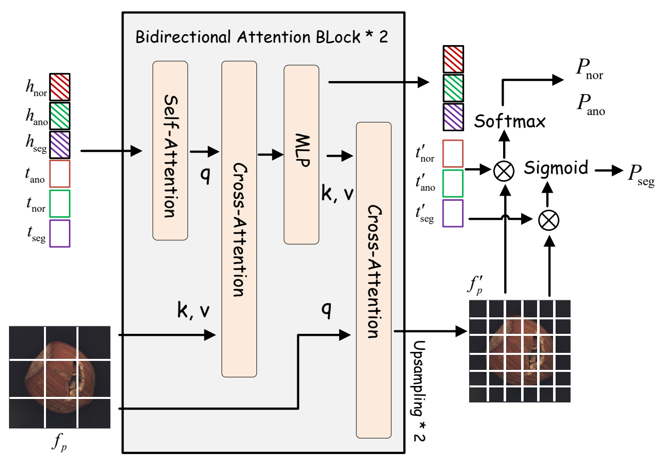

Figure A.3. The proposed AGMD architecture.

C. Additional Ablations and Analysis

C.1. Ablation on the Pixel Image Encoder

Table A.2 reports the performance obtained by replacing the default SAM (ViT-H) [17] with alternative pixel-level encoders. Here, SAM-FT denotes the variant where a linear layer is appended to the encoder output and fine-tuned during training. Results show that, in addition to SAM, other encoders such as CLIP [26], DINOv2 [24], and DINOv3 [31] also achieve strong performance, with DINOv2 and DINOv3 even outperforming SAM in terms of ZSAS. This suggests that our method is not tied to a specific pixel encoder and can flexibly adapt to different backbone choices. Moreover, as the representational capacity of the encoder increases, the zero-shot performance of AG-VAS consistently improves, highlighting the scalability of our framework. Notably, fine-tuning SAM with an additional linear layer leads to a significant performance drop, indicating that preserving the original pre-trained representations is crucial for maintaining generalization.

C.2. Analysis of Inference Efficiency

Table A.3 compares different LMM-based methods in terms of ZSAS performance, maximum GPU memory consumption per image, and average inference time on the MVTec-AD dataset. Our method achieves the best AP and IoU_{ano} while maintaining competitive inference efficiency. In particular, AG-VAS significantly outperforms prior LMM-based approaches such as LISA and PixelLM, demonstrating the effectiveness of the proposed modules. Despite the improved segmentation accuracy, AG-VAS introduces only a modest increase in GPU memory usage and inference time compared to the LISA baseline, indicating an efficient design. Overall, these results show that our method not only improves ZSAS performance but also remains practical for real-world deployment.

C.3. Analysis of Failure Cases

Figure A.4 presents failure cases of the proposed AG-VAS in the ZSAS task. It can be observed that AG-VAS tends to produce more conservative predictions; that is, for anomalous regions without clear boundaries, the binarized segmentation masks are smaller than the ground-truth regions. In addition, we find that AG-VAS performs poorly when segmenting multiple anomalous regions within a single image, especially when the anomaly types differ significantly. For example, in the second column (hazelnut), the crack outside the hole is ignored by the model, and in the third column (wood), the small crack in the lower-left region is not correctly localized. In future work, we will further improve the model's ability to simultaneously segment multiple anomalous regions.

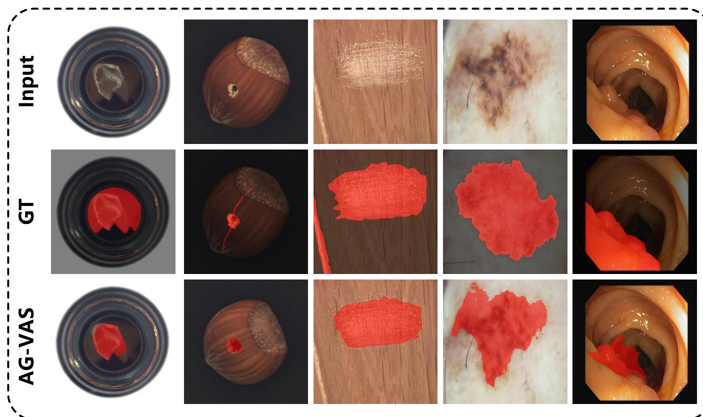

Figure A.4. Failure Cases.

D. Detailed ZSAS results

In Tables A.4 to A.7, we present the detailed quantitative results for each specific category of the MVTec-AD datasets. In Figures A.14 to A.26, we provide more extensive qualitative results for various categories across different industrial and medical datasets. Moreover, we provide additional examples of the three instruction types: Direct Segmentation, Describe-then-Segment, and Segment-then-Explain, in Figures A.10 through A.13.

E. Limitations

Although AG-VAS achieves state-of-the-art ZSAS performance across multiple medical and industrial datasets and produces highly accurate binary segmentation masks, it still has two main limitations. First, despite its ability to leverage richer contextual semantics and world knowledge during inference to enhance zero-shot generalization, LMM-based methods inherently suffer from slower inference speed, which constrains their applicability in real-time industrial environments. Second, while the segment-then-explain and describe-then-segment modes improve human–model interaction and provide greater interpretability, they do not yield additional performance gains without more expert priors. The extra textual generation tends to divert the model's focus from the segmentation objective, which is why AG-VAS primarily adopts implicit reasoning mode to directly output segmentation results.

```
SYSTEM_PROMPT = """
You are an expert in industrial defect analysis.

You will be shown three images:
- Image A (the original image): It has a green bounding box overlaid to indicate the defect location; this bounding box fully contain the defects and is only for rough localization, not part of the scene itself.
- Image B (a reference ground-truth image): It has green regions that fully cover all defect areas of the original image with pixel-level precision.
- Image C (a reference normal image): It is a normal image of the same category as the original image A.

You may use Image B and Image C to understand what defective and normal regions look like,
but your final reasoning must only describe what is visually different in Image A.

You must produce a **machine-readable JSON block**, enclosed between <RESPONSE_JSON> and </RESPONSE_JSON>.
The JSON must have the following schema:

{
  "exists": true|false,              // whether a defect is present
  "Expectation": "<string>",         // normal appearance (short)
  "Observation": "<string>",         // observed difference (short)
  "Diagnosis": "<string>",           // why it is defective (short)
  "Summary": "<string>",             // concise location & appearance (must include [NOR][ANO][SEG])
  "Explanation": "<string>"          // natural-language summary combining Expectation, Observation, Diagnosis, Summary
}

Rules and constraints:

- JSON must be syntactically valid (double quotes, no trailing commas).
- Keep each field concise (1-3 sentences).
- Observation must describe at least TWO visual cues (choose from: location relative to the object, color/brightness contrast, shape, size, texture, or difference from normal).
- Diagnosis must explain why those visual cues indicate a defect (e.g., breaks texture/uniformity, implies material loss, suggests foreign object).
- Do not mention or refer to the bounding box, green mask, Image B, Image C, or image identifiers (e.g., "Image A", "Image B", "Image C").
- Do not use wording such as "green box", "ground truth", "mask", "annotation", "normal image", or "reference image".
- Output **only** the JSON block, nothing else.

Example output format:

<RESPONSE_JSON>
{
  "exists": true,
  "Expectation": "The orange terminal block should have a smooth and uniform surface.",
  "Observation": "A small dark pit appears on the right side near the edge, breaking the texture.",
  "Diagnosis": "This irregular depression indicates a surface defect.",
  "Summary": "The defect is a small dark pit located on the right side of the orange terminal block near the edge, as indicated by [NOR][ANO][SEG].",
  "Explanation": "Normally, the orange terminal block should have a smooth and uniform surface. However, a small, dark pit appears on the right side near the edge, breaking the texture. Therefore, this irregular depression indicates a surface defect. In conclusion, the defect is a small dark pit located on the right side of the orange terminal block near the edge."
}
</RESPONSE_JSON>
"""
```

Figure A.5. The design of system prompts.

{**class_name**}: ["defects", "anomalies", "visual defects", "surface defects", "appearance anomalies", "product defects", "defective areas", "abnormal regions", "flaws", "irregular patterns", "visual imperfections", "damaged regions", "defective areas"]

Instruction Templates For Direct Segmentation:

- "Can you segment the {**class_name**} in this image?",
- "Can you segment the {**class_name**} for the object in this image?",
- "Please segment the {**class_name**} in this image.",
- "Please segment the {**class_name**} for the object in this image.",
- "What are {**class_name**} in this image? Please respond with segmentation mask.",
- "What are {**class_name**} in this image? Please output segmentation mask.",
- "Please highlight the {**class_name**} in the image.",
- "Please highlight the {**class_name**} for the object in the image.",
- "Identify and segment the {**class_name**} present in this image.",
- "Segment all visible {**class_name**} in this picture.",
- "Segment all visible {**class_name**} for the object in this picture.",
- "Could you localize the {**class_name**} in this image by segmentation?",
- "Find and segment any {**class_name**} you can detect.",
- "Draw a segmentation map for the {**class_name**} in the picture.",
- "Segment the region that looks different from the rest of the surface.",
- "Locate and segment the abnormal region on the surface.",
- "Locate and segment the abnormal region for the object on the surface.",
- "Highlight the area that appears defective or inconsistent.",
- "Segment the surface area that does not match the normal pattern or texture.",
- "Please segment any regions that seem irregular, damaged, or abnormal.",
- "Locate the defective regions and output a segmentation mask.",
- "Find and segment the area that looks inconsistent with the surrounding material.",
- "Segment the part of the surface showing visual anomalies or defects.",
- "Please mark and segment the defective areas on this surface.",

Answer Templates For Direct Segmentation:

- "It is [NOR][ANO][SEG].",
- "Sure, [NOR][ANO][SEG].",
- "Sure, it is [NOR][ANO][SEG].",
- "Sure, the segmentation result is [NOR][ANO][SEG].",
- "[NOR][ANO][SEG].",

Figure A.6. The design of instruction and answer templates for direct segmentation.

Instruction Templates For Describe-then-Segment:

- "What are {class_name} in this image? Please describe the {class_name} and then output the corresponding segmentation map.",
- "What are {class_name} in this image? Describe the {class_name} and provide its segmentation map.",
- "Please locate the {class_name} in the image. Provide a concise description about it, and generate the segmentation map.",
- "Please segment the {class_name} in the image. Provide a concise description about it, and generate the segmentation map.",
- "Please segment the {class_name} in this image. Provide a concise description, and then output the segmentation map.",
- "Please segment the {class_name} for the object in this image. Provide a concise description for the {class_name}, and generate the segmentation map.",
- "Please segment the {class_name} for the object in this image. Provide a concise description, and generate the segmentation map."

Instruction Templates For Rejection:

- "No {class_name} detected. ",
- "It looks normal to me.",
- "The image appears to be normal.",
- "I don't see any {class_name} in this image.",
- "I don't see any {class_name} in this image.",
- "No visible {class_name} are present.",
- "There are no {class_name} that I can detect.",
- "This image appears free of {class_name}.",
- "No abnormalities detected.",
- "Nothing unusual is found in this image.",
- "I couldn't find any signs of {class_name}.",
- "There don't seem to be any {class_name} here.",
- "This sample looks clean and defect-free.",
- "I didn't detect any suspicious areas.",
- "All regions in this image look normal.",
- "The product surface is defect-free.",
- "Inspection result: No {class_name} detected.",
- "No segmentation required — image is normal.",
- "Visual inspection shows no {class_name}.",
- "No anomalies found during analysis.",
- "I don't observe any defects requiring segmentation.",
- "No defect masks needed for this image."

Figure A.7. The design of instruction templates for Describe-then-Segment and Rejection.

Table A.4. Comparison of different categories in terms of pixel-level AP on the MVTec-AD.

object	AnomalyCLIP	Bayes-PFL	PaDT	PixelLM	LISA-7B	LISA-13B	LISA-13B*	LISA-OV	LISA-OV*	AG-VAS
bottle	**55.3**	64.7	10.1	22.9	14.2	15.6	34.3	35.0	47.5	45.8
cable	12.3	15.1	6.1	9.7	8.8	22.6	10.8	25.3	**27.8**	27.1
capsule	27.7	34.4	4.0	7.0	15.1	15.7	14.4	14.9	17.1	**38.4**
carpet	56.6	**82.2**	1.4	40.8	1.1	11.3	59.4	30.4	54.5	72.8
grid	24.1	41.1	0.7	0.7	0.4	0.5	60.4	23.4	57.5	**64.1**
hazelnut	43.4	70.9	5.8	7.6	11.5	11.7	36.0	34.8	55.9	**70.2**
leather	22.8	58.9	0.8	0.5	3.1	2.5	**77.0**	58.6	69.9	76.7
metal_nut	**26.5**	24.3	25.0	14.5	19.9	21.2	19.9	**38.5**	19.3	23.3
pill	34.1	29.4	9.6	31.6	15.6	**40.8**	20.6	32.0	25.3	25.5
screw	27.5	**41.6**	2.3	2.9	4.9	4.0	4.2	13.0	12.2	33.7
tile	61.7	75.9	8.5	8.1	3.7	4.9	85.3	69.1	79.8	**89.6**
toothbrush	19.4	34.0	4.8	21.7	5.9	5.8	28.0	9.3	25.0	**53.4**
transistor	15.6	12.9	9.2	16.9	6.7	7.4	11.1	12.7	15.6	**19.6**
wood	52.7	70.5	4.5	15.5	31.5	42.6	**80.8**	69.2	84.1	74.1
zipper	38.7	69.0	8.5	3.1	4.5	11.9	21.8	20.0	22.9	**51.2**
mean	34.5	48.3	6.8	13.6	9.8	14.6	37.6	32.4	41.0	**51.0**

Table A.5. Comparison of different categories in terms of pixel-level F1-Max on the MVTec-AD.

object	AnomalyCLIP	Bayes-PFL	PaDT	PixelLM	LISA-7B	LISA-13B	LISA-13B*	LISA-OV	LISA-OV*	AG-VAS
bottle	51.6	**61.3**	18.9	34.6	27.4	30.5	38.3	38.3	46.4	47.0
cable	18.9	23.5	14.8	16.4	13.8	27.3	17.1	36.9	38.0	**34.5**
capsule	31.0	40.5	7.9	13.2	27.2	26.3	27.8	26.7	27.8	**46.5**
carpet	57.1	74.9	3.7	43.7	3.2	18.6	57.2	37.7	52.4	**66.3**
grid	32.0	41.6	1.6	1.5	1.4	1.4	61.7	29.1	59.2	**64.2**
hazelnut	47.6	**66.2**	11.3	15.5	23.0	23.2	44.3	36.3	56.9	65.0
leather	33.2	56.9	1.9	1.4	4.9	5.3	**70.4**	62.1	64.4	69.2
metal_nut	33.1	35.5	36.2	36.3	37.9	39.1	21.0	**53.4**	21.0	22.6
pill	35.5	32.9	17.4	28.7	24.1	43.8	27.0	33.2	31.4	33.9
screw	33.4	44.1	4.8	5.9	10.5	8.7	9.6	23.2	26.9	**44.7**
tile	64.9	71.2	18.4	15.4	13.2	13.5	82.6	68.2	77.5	**85.3**
toothbrush	29.0	38.0	10.5	27.4	11.9	11.7	34.7	15.6	31.8	**58.7**
transistor	18.8	17.3	20.0	21.9	9.3	14.2	18.3	16.9	21.6	**28.7**
wood	55.2	65.5	10.6	26.4	34.8	41.3	74.8	67.8	77.2	**72.4**
zipper	45.0	**66.9**	15.9	6.3	10.3	22.4	27.8	29.6	28.4	52.2
mean	39.1	49.1	12.9	19.6	16.9	21.8	40.9	38.3	44.1	**52.7**

Table A.6. Comparison of different categories in terms of IoU_{ano} on the MVTec-AD.

object	AnomalyCLIP	Bayes-PFL	PaDT	PixelLM	LISA-7B	LISA-13B	LISA-13B*	LISA-OV	LISA-OV*	AG-VAS
bottle	27.3	25.2	20.0	26.7	20.0	25.4	30.7	36.7	23.8	**41.4**
cable	10.3	7.5	11.5	5.0	3.4	8.9	4.0	23.4	16.0	**29.2**
capsule	16.8	7.6	4.7	4.9	5.4	8.3	16.0	14.1	12.1	**36.3**
carpet	32.5	48.9	16.4	29.6	3.3	13.6	46.5	39.0	43.6	52.1
grid	20.5	24.4	12.9	2.0	0.8	0.5	**43.7**	32.8	47.9	41.7
hazelnut	28.0	9.4	14.5	13.2	17.6	23.0	45.8	44.7	45.7	**64.4**
leather	13.3	16.3	26.8	1.5	5.3	1.7	**56.0**	54.0	58.4	48.0
metal_nut	21.7	24.0	26.2	23.0	22.7	13.1	38.5	39.8	34.6	**46.8**
pill	20.5	9.2	10.8	10.5	9.0	19.4	18.7	29.4	31.2	**54.5**
screw	13.9	1.9	2.9	2.9	2.8	4.5	6.0	12.2	9.3	**31.9**
tile	43.2	51.7	46.9	19.1	5.3	14.9	74.1	58.7	61.6	**75.0**
toothbrush	20.4	6.9	7.2	4.8	7.3	6.6	13.1	7.7	24.3	**43.7**
transistor	14.7	13.8	9.9	8.6	5.5	0.5	7.0	12.9	13.0	**18.8**
wood	33.8	39.5	22.7	23.7	41.1	51.9	63.9	59.7	50.9	**62.3**
zipper	22.1	**46.9**	13.0	3.6	2.7	13.7	12.7	18.5	11.7	26.0
mean	22.6	22.2	16.4	11.9	10.2	13.7	31.8	32.3	32.3	**44.8**

Table A.7. Comparison of different categories in terms of IoU_{nor} on the MVTec-AD.

object	AnomalyCLIP	Bayes-PFL	PaDT	PixelLM	LISA-7B	LISA-13B	LISA-13B*	LISA-OV	LISA-OV*	AG-VAS
bottle	0.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	75.0	**85.0**
cable	17.2	0.0	0.0	34.5	25.9	17.2	0.0	0.0	56.9	**55.2**
capsule	73.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	91.3	**95.7**
carpet	78.6	96.4	0.0	0.0	0.0	3.6	**100.0**	57.1	89.3	**100.0**
grid	76.2	95.2	0.0	0.0	0.0	0.0	**95.2**	0.0	**95.2**	**95.2**
hazelnut	7.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	65.0	**97.5**
leather	28.1	53.1	0.0	0.0	0.0	0.0	90.6	0.0	**96.9**	**96.9**
metal_nut	9.1	0.0	0.0	0.0	0.0	0.0	36.4	0.0	95.5	**95.5**
pill	11.5	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0	53.9	96.2
screw	48.8	0.0	0.0	0.0	2.4	2.4	4.9	0.0	90.2	53.7
tile	78.8	12.1	0.0	0.0	0.0	0.0	78.8	3.0	81.8	**97.0**
toothbrush	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	50.0	**100.0**
transistor	5.0	0.0	0.0	1.7	0.0	71.7	0.0	0.0	90.0	**53.3**
wood	36.8	5.3	0.0	0.0	0.0	0.0	68.4	0.0	89.5	**100.0**
zipper	**100.0**	43.8	0.0	0.0	0.0	0.0	68.8	0.0	93.8	93.8
mean	38.1	20.4	0.0	2.4	1.9	6.3	43.5	4.0	81.0	**87.7**

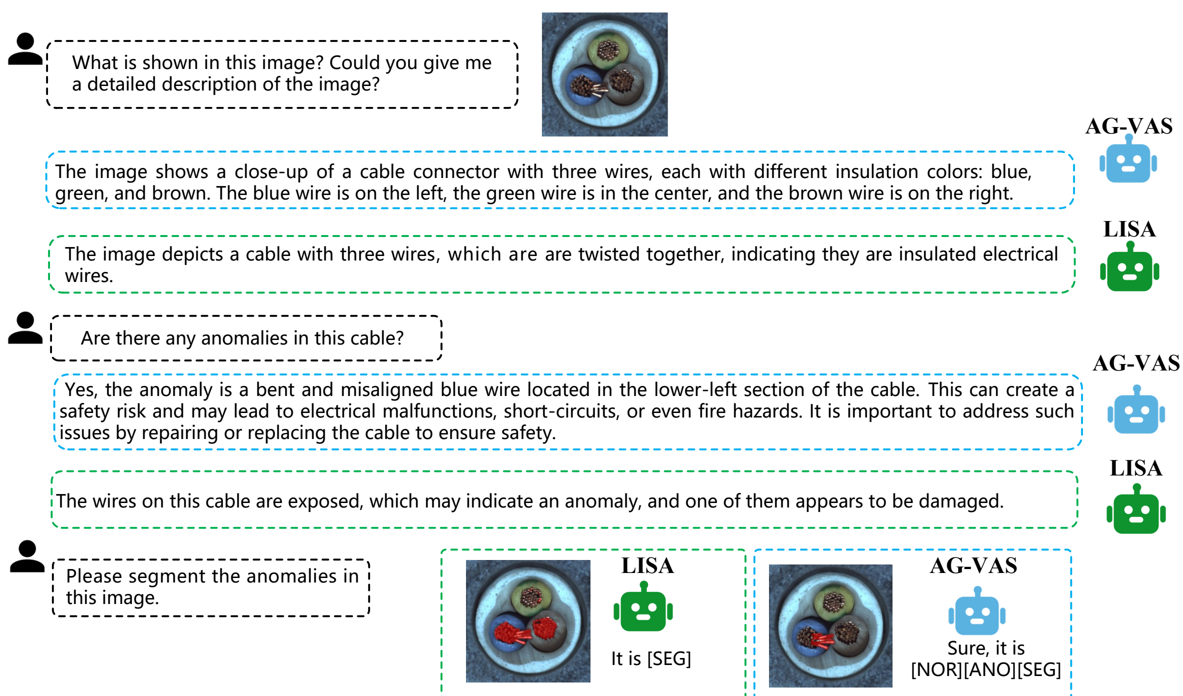

Figure A.8. Comparison of the dialogue and segmentation capabilities between the proposed AG-VAS and LISA [18].

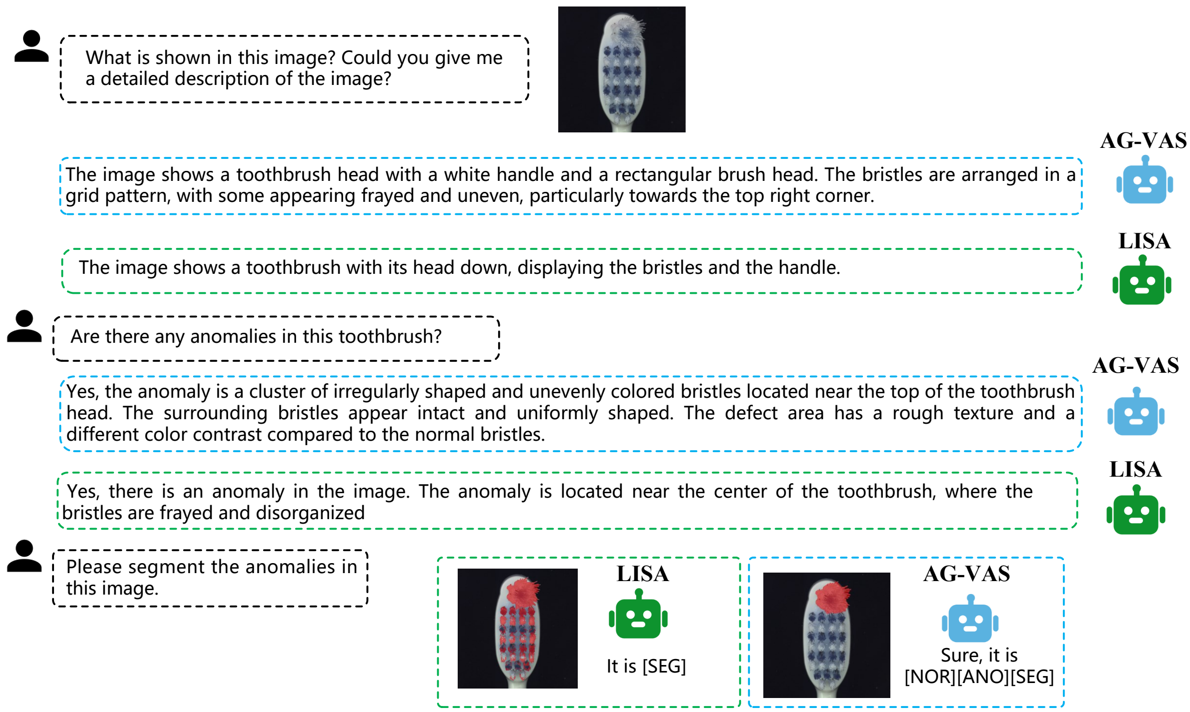

Figure A.9. Comparison of the dialogue and segmentation capabilities between the proposed AG-VAS and LISA [18].

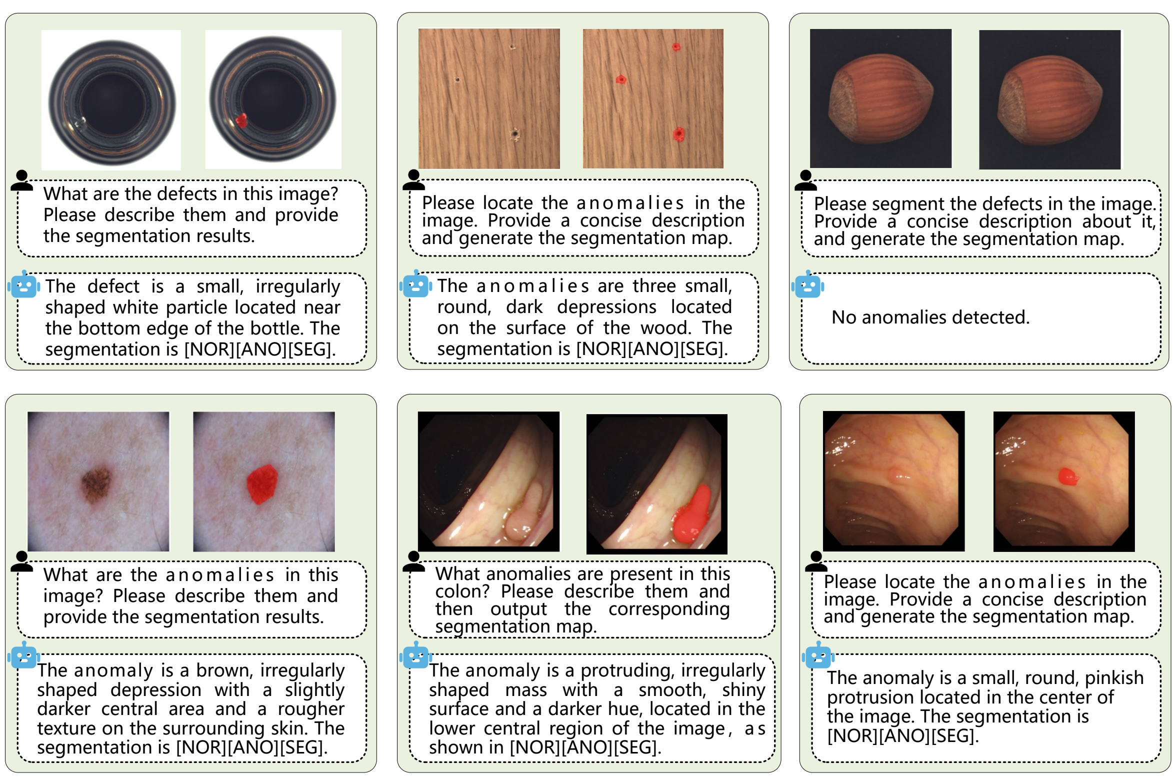

Figure A.10. Some examples under the describe-then-segment task. Left: original images; Right: AG-VAS segmentation results.

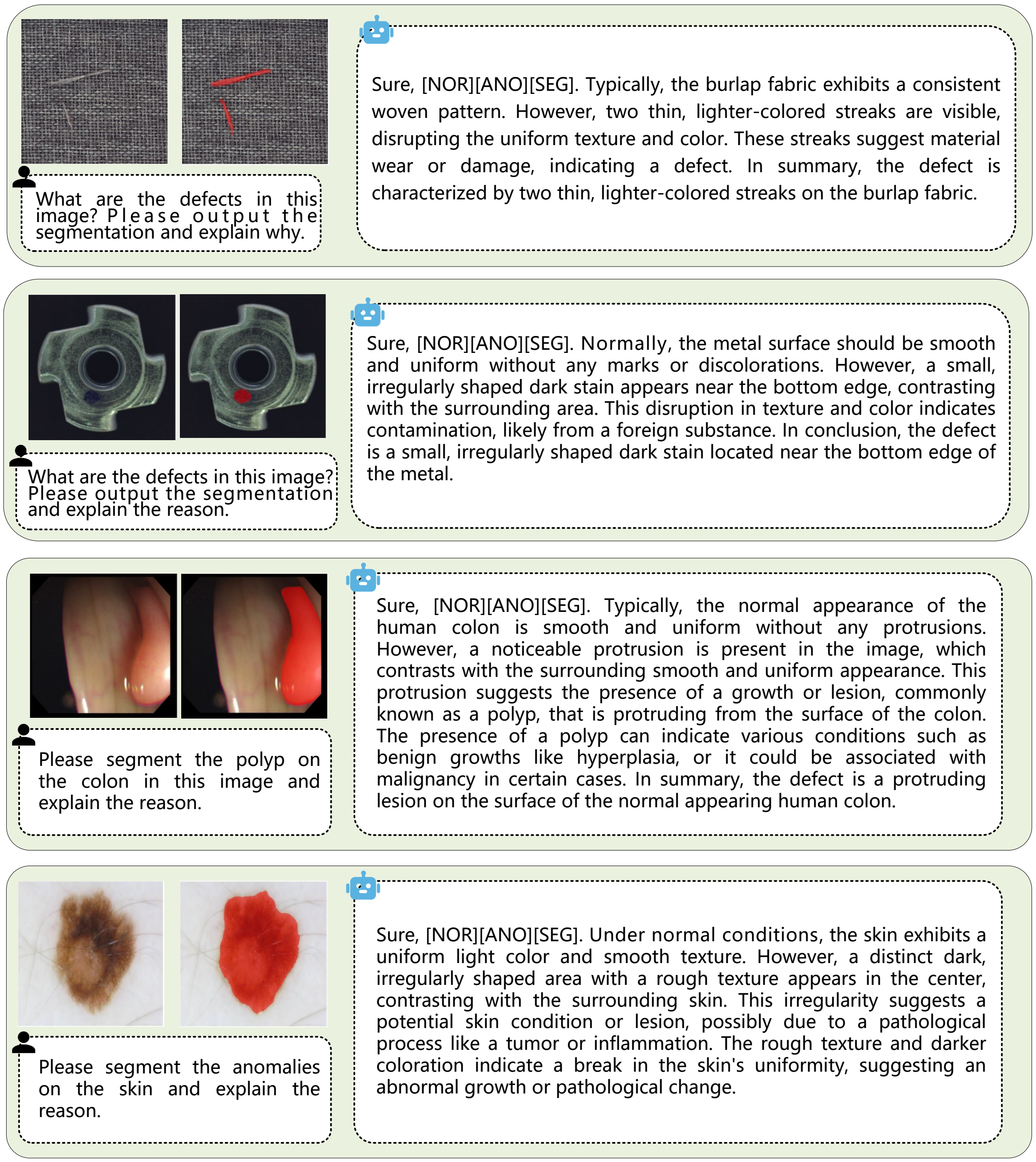

Figure A.11. Some examples under the segment-then-explain task. Left: original images; Right: AG-VAS segmentation results.

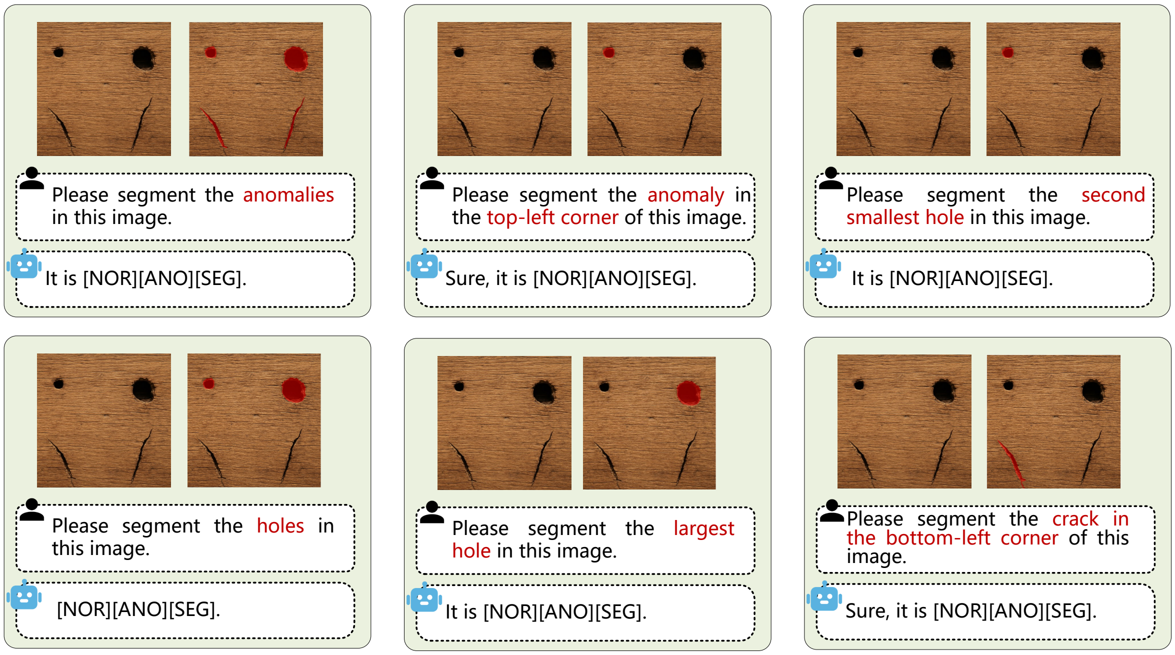

Figure A.12. Generalization of AG-VAS to fine-grained segmentation instructions. Left: original images; Right: AG-VAS segmentation results. AG-VAS effectively captures both positional cues (e.g., “top-left”) and attribute-specific descriptions (e.g., “the smallest”), achieving precise localization of anomalous regions.

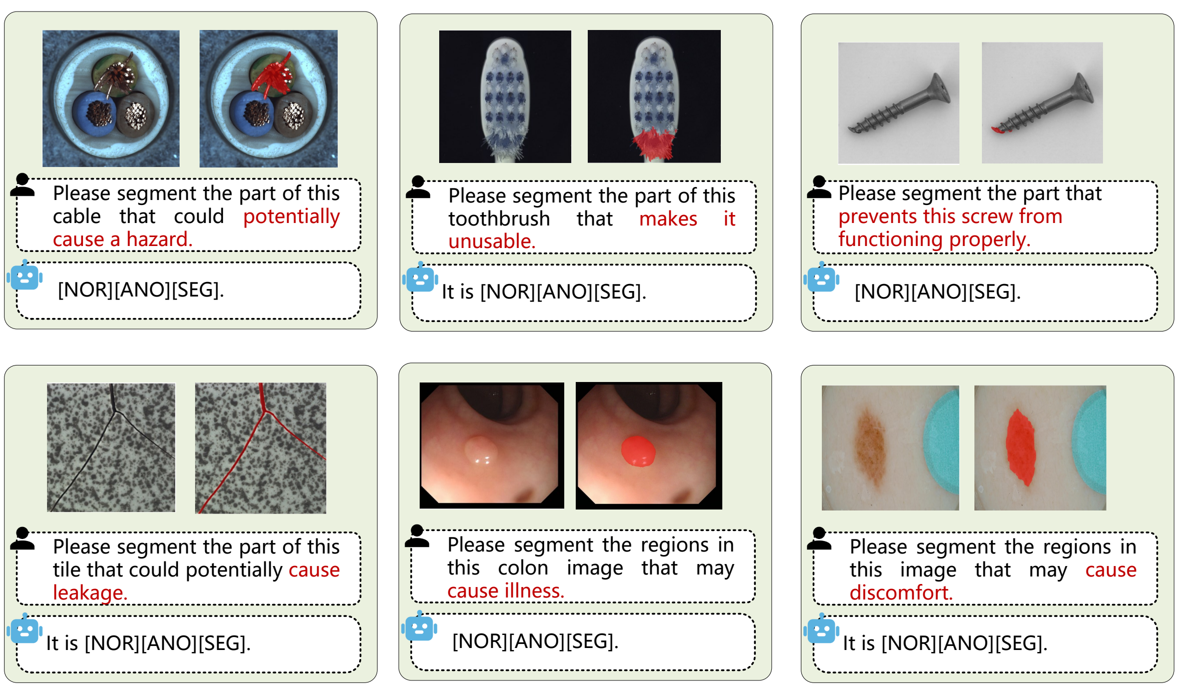

Figure A.13. Examples showcasing AG-VAS’s ability to leverage world knowledge for implicit reasoning and direct segmentation. Left: original images; Right: AG-VAS segmentation results. AG-VAS exploits its learned world knowledge to conduct implicit reasoning and precisely localize anomalous regions.

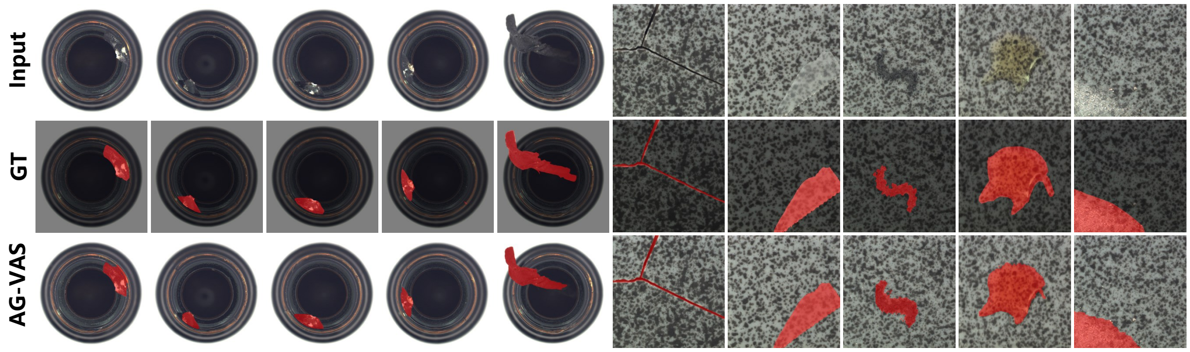

Figure A.14. Visualization of segmentation results for the **bottle** and **tile** classes on **MVTecAD**. The top row shows the original input images, the middle row presents the ground-truth masks, and the bottom row displays the binarized segmentation results using the default threshold of 0.5..

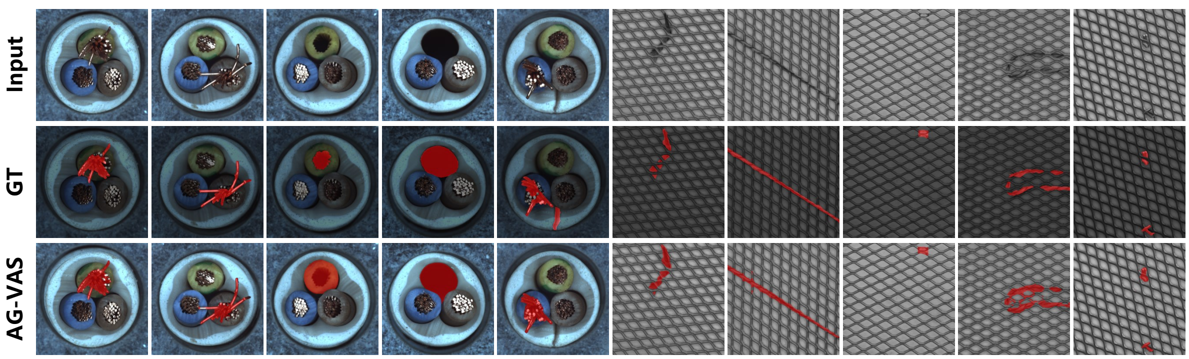

Figure A.15. Visualization of segmentation results for the **cable** and **grid** classes on **MVTecAD**. The top row shows the original input images, the middle row presents the ground-truth masks, and the bottom row displays the binarized segmentation results using the default threshold of 0.5.

Figure A.16. Visualization of segmentation results for the **capsule** and **carpet** classes on **MVTecAD**. The top row shows the original input images, the middle row presents the ground-truth masks, and the bottom row displays the binarized segmentation results using the default threshold of 0.5.

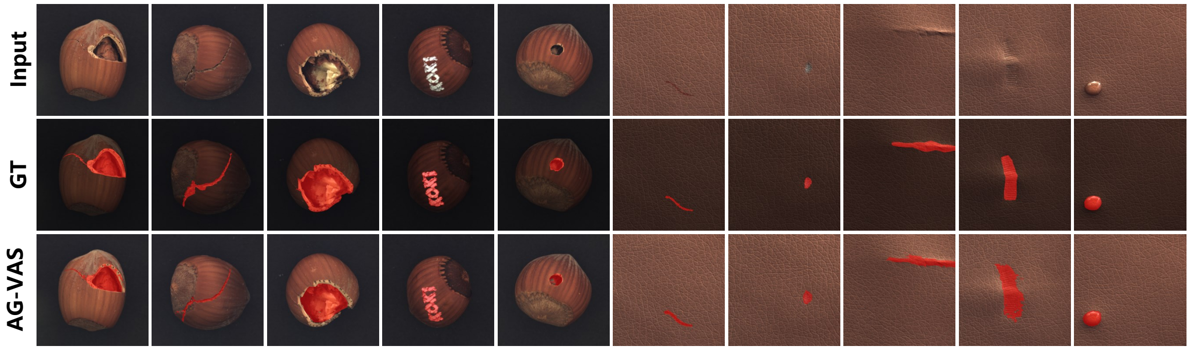

Figure A.17. Visualization of segmentation results for the **hazelnut** and **leather** classes on **MVTecAD**. The top row shows the original input images, the middle row presents the ground-truth masks, and the bottom row displays the binarized segmentation results using the default threshold of 0.5.

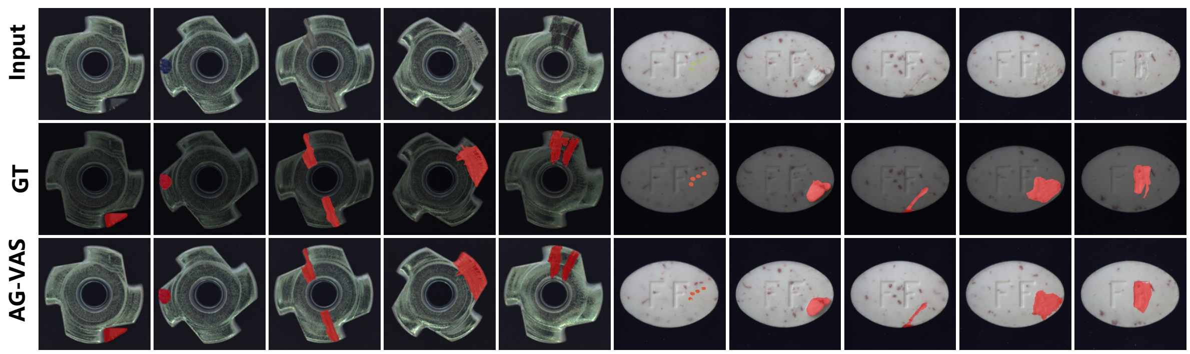

Figure A.18. Visualization of segmentation results for the **metal nut** and **pill** classes on **MVTecAD**. The top row shows the original input images, the middle row presents the ground-truth masks, and the bottom row displays the binarized segmentation results using the default threshold of 0.5.

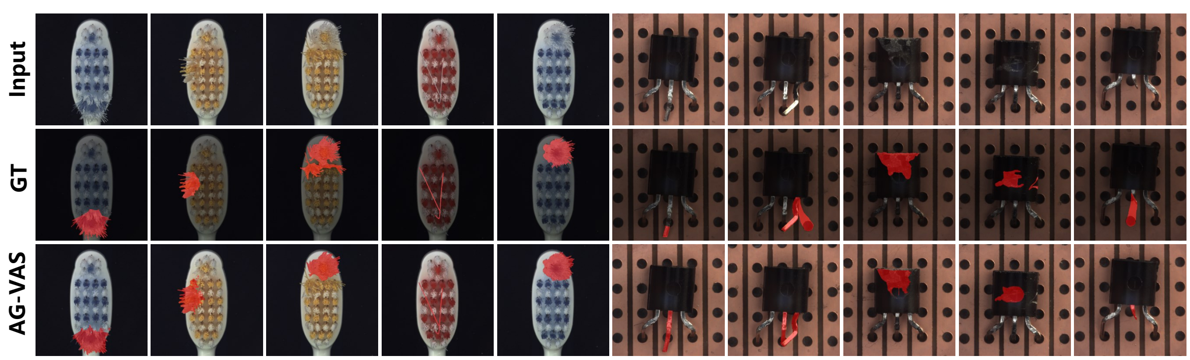

Figure A.19. Visualization of segmentation results for the **toothbrush** and **transistor** classes on **MVTecAD**. The top row shows the original input images, the middle row presents the ground-truth masks, and the bottom row displays the binarized segmentation results using the default threshold of 0.5.

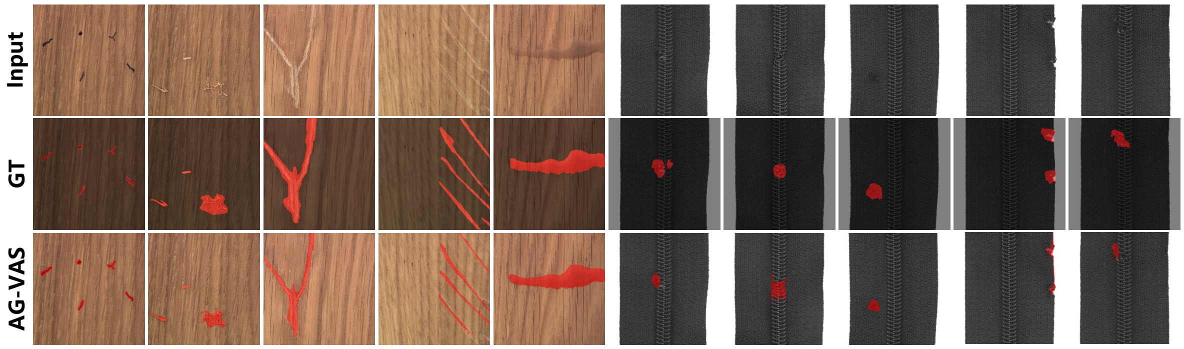

Figure A.20. Visualization of segmentation results for the **wood** and **zipper** classes on **MVTecAD**. The top row shows the original input images, the middle row presents the ground-truth masks, and the bottom row displays the binarized segmentation results using the default threshold of 0.5.

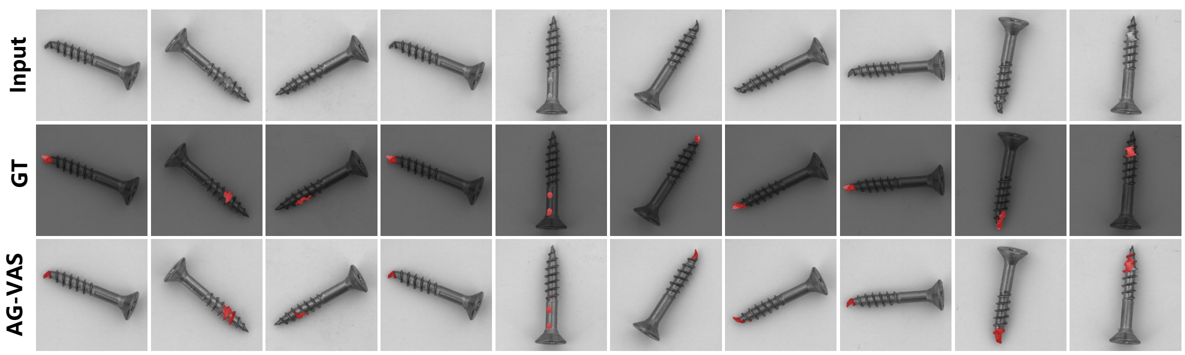

Figure A.21. Visualization of segmentation results for the **screw** class on **MVTecAD**. The top row shows the original input images, the middle row presents the ground-truth masks, and the bottom row displays the binarized segmentation results using the default threshold of 0.5.

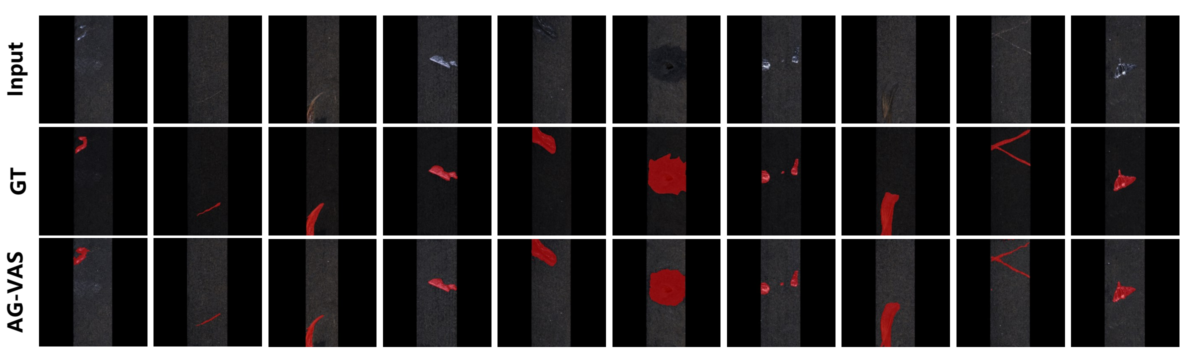

Figure A.22. Visualization of segmentation results on **KSDD2**. The top row shows the original input images, the middle row presents the ground-truth masks, and the bottom row displays the binarized segmentation results using the default threshold of 0.5.

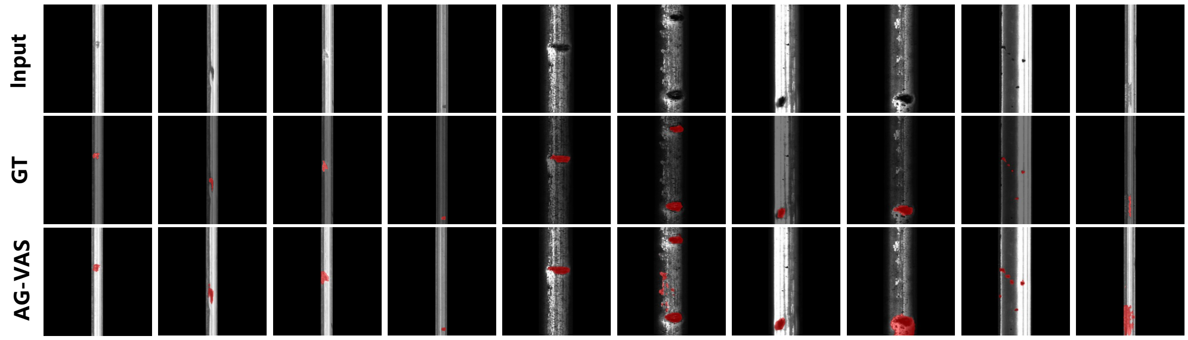

Figure A.23. Visualization of segmentation results on **RSDD**. The top row shows the original input images, the middle row presents the ground-truth masks, and the bottom row displays the binarized segmentation results using the default threshold of 0.5.

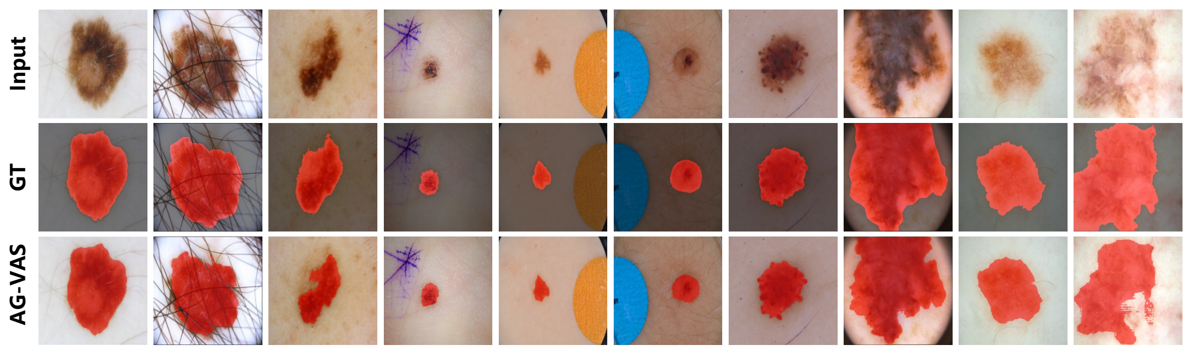

Figure A.24. Visualization of segmentation results for the **skin** class on **ISIC**. The top row shows the original input images, the middle row presents the ground-truth masks, and the bottom row displays the binarized segmentation results using the default threshold of 0.5.

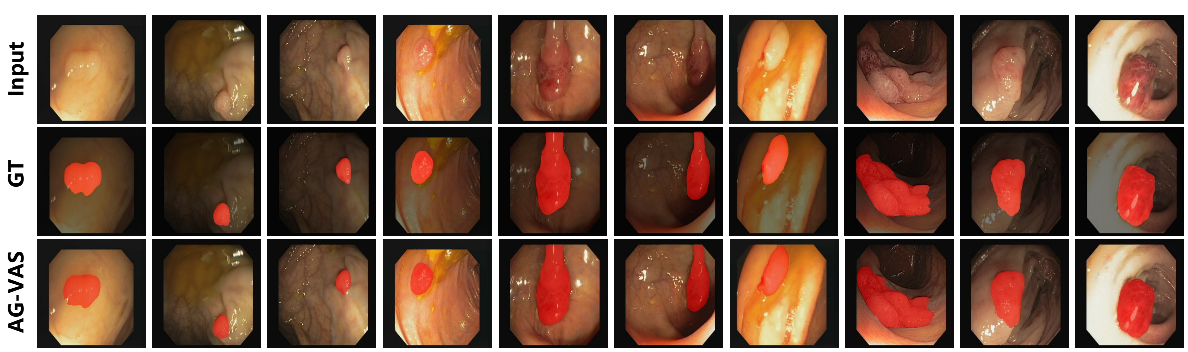

Figure A.25. Visualization of segmentation results for the **colon** class on **ClinicDB**. The top row shows the original input images, the middle row presents the ground-truth masks, and the bottom row displays the binarized segmentation results using the default threshold of 0.5.

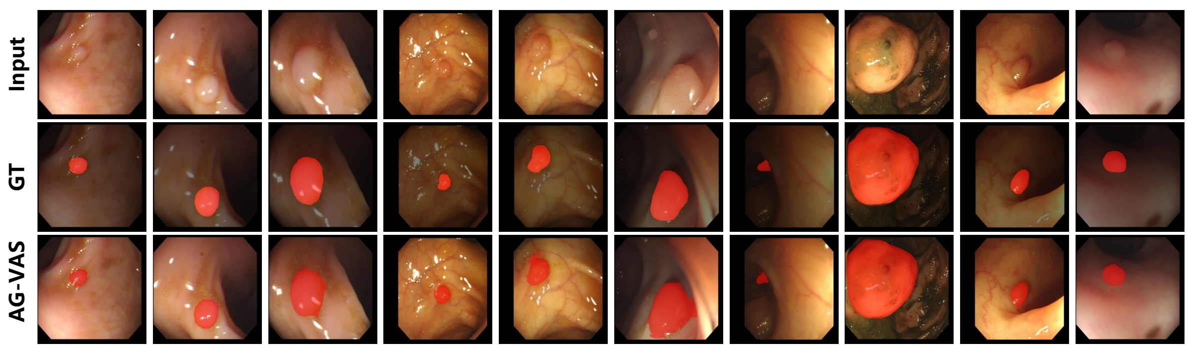

Figure A.26. Visualization of segmentation results for the **colon** class on **ColonDB**. The top row shows the original input images, the middle row presents the ground-truth masks, and the bottom row displays the binarized segmentation results using the default threshold of 0.5.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Toshimichi Aota, Lloyd Teh Tzer Tong, and Takayuki Okatani. Zero-shot versus many-shot: Unsupervised texture anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5564–5572, 2023. 1, 3

[3] Lei Bai, Zhongrui Cai, Yuhang Cao, Maosong Cao, Weihan Cao, Chiyu Chen, Haojiong Chen, Kai Chen, Pengcheng Chen, Ying Chen, et al. Intern-s1: A scientific multimodal foundation model. *arXiv preprint arXiv:2508.15763*, 2025. 2

[4] Tianpeng Bao, Jiadong Chen, Wei Li, Xiang Wang, Jingjing Fei, Liwei Wu, Rui Zhao, and Ye Zheng. Miad: A maintenance inspection dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 993–1002, 2023. 3

[5] 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. 4

[6] Jorge Bernal, F Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodríguez, and Fernando Vilariño. 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

[7] Anil Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distribution. *Bulletin of the Calcutta Mathematical Society*, 35:99–110, 1943. 2

[8] Jakob Božič, Domen Tabernik, and Danijel Skočaj. Mixed supervision for surface-defect detection: From weakly to fully supervised learning. *Computers in Industry*, 129:103459, 2021. 4

[9] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 3

[10] Yunkang Cao, Jiangning Zhang, Luca Frittoli, Yuqi Cheng, Weiming Shen, and Giacomo Boracchi. Adaclip: Adapting clip with hybrid learnable prompts for zero-shot anomaly detection. In *European Conference on Computer Vision*, pages 55–72. Springer, 2025. 4

[11] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014. 3

[12] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 168–172. IEEE, 2018. 4

[13] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, pages 128–145. Springer, 2022. 3

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 4

[15] Xi Jiang, Jian Li, Hanqiu Deng, Yong Liu, Bin-Bin Gao, Yifeng Zhou, Jialin Li, Chengjie Wang, and Feng Zheng. Mmad: The comprehensive benchmark for multimodal large language models in industrial anomaly detection. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[16] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014. 3

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 4, 5

[18] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024. 3, 11, 12

[19] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 4, 5

[20] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 3, 4

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[22] Xiaoming Lv, Fajie Duan, Jia-jia Jiang, Xiao Fu, and Lin Gan. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors*, 20(6):1562, 2020. 3

[23] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016. 3

[24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 5

[25] Zhen Qu, Xian Tao, Xinyi Gong, Shichen Qu, Qiyu Chen, Zhengtao Zhang, Xingang Wang, and Guiguang Ding. Bayesian prompt flow learning for zero-shot anomaly detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30398–30408, 2025. 4

[26] 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. 5

[27] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7151, 2023. 3

[28] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506, 2020. 4

[29] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26374–26383, 2024. 2

[30] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016. 3

[31] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 5

[32] Nima Tajbakhsh, Suryakanth R Gurudu, and Jianming Liang. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE transactions on medical imaging*, 35(2):630–644, 2015. 4

[33] Qwen Team et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3), 2024. 5

[34] Chengjie Wang, Wenbing Zhu, Bin-Bin Gao, Zhenye Gan, Jiangning Zhang, Zhihao Gu, Shuguang Qian, Mingang Chen, and Lizhuang Ma. Real-iad: A real-world multi-view dataset for benchmarking versatile industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22883–22892, 2024. 1, 3

[35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 2

[36] Jiacong Xu, Shao-Yuan Lo, Bardia Safaei, Vishal M Patel, and Isht Dwivedi. Towards zero-shot anomaly detection and reasoning with multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20370–20382, 2025. 3, 4

[37] Haomin Yu, Qingyong Li, Yunqiang Tan, Jinrui Gan, Jianzhu Wang, Yangli-ao Geng, and Lei Jia. A coarse-to-fine model for rail surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, 68(3):656–666, 2018. 4

[38] Chenkai Zhang, Shaozhe Feng, Xulongqi Wang, and Yueming Wang. Zju-leaper: A benchmark dataset for fabric defect detection and a comparative study. *IEEE Transactions on Artificial Intelligence*, 1(3):219–232, 2021. 3

[39] Jian Zhang, Runwei Ding, Miaoju Ban, and Linhui Dai. Pku-goodsad: A supermarket goods dataset for unsupervised anomaly detection and segmentation. *IEEE Robotics and Automation Letters*, 9(3):2008–2015, 2024. 1, 3

[40] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 3

[41] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. In *The Twelfth International Conference on Learning Representations*, 2023. 4