

# Do LLMs Understand Visual Anomalies?

## Uncovering LLM’s Capabilities in Zero-shot Anomaly Detection

[Supplementary Materials]

Anonymous Author(s)

### A IMPLEMENTATION DETAILS

#### A.1 Data pre-processing

We employ the data pre-processing pipeline outlined in the OpenCLIP implementation [7] for both the MVTec [1] and VisA [15] benchmarks. This process entails channel-wise standardization using the pre-computed mean [0.48145466, 0.4578275, 0.40821073] and standard deviation [0.26862954, 0.26130258, 0.27577711] after normalizing each RGB image to [0, 1]. Subsequently, we apply bicubic resizing based on the Pillow implementation (PIL). We note that CLIP requires a square-shaped resolution for input image queries. To adapt CLIP to non-square images in the VisA benchmark, we employ an “image tiling” scheme, as suggested by [8]. The resizing policy is also adopted for other baseline models to ensure fair comparisons, while the remaining parts of their original data pre-processing pipelines are retained.

#### A.2 Prompt generation via unified templates

Recall that we formulate a unified template to generate contrastive-state prompts as: “A {*domain*} image of a {*state*} {*class*} [with {*specific details*}]”, where “class” in the template is replaced by the class name in MVTec and VisA benchmarks. Along this line, we perform prompt enhancement by replacing some fixed words in the template such as “image” with “photo”. Additionally, given the potential ambiguity of certain class names, such as “bottle” in MVTec and “pcb1” and “pcb2” in VisA, we opted to substitute these class names in the original dataset with more specific object names or concise descriptions. We present a sample list of templates utilized in this study, denoted as: templates = [ “an image of a { $\Omega$ ”, “a close-up image of a { $\Omega$ ”, “an industrial image of a { $\Omega$ ”, “a manufacturing image of a { $\Omega$ ”, “a production image of a { $\Omega$ ”, “a textural image of a { $\Omega$ ”, “a surface image of a { $\Omega$ ”, “a cross-section image of a { $\Omega$ ” ]. Here, “{ $\Omega$ ” represents a description of the contrastive state of a specific class.

Specifically, we formulate “{ $\Omega$ ” as: normal state = [“ $\zeta$ ”, “normal  $\zeta$ ”, “undamaged  $\zeta$ ”, “flawless  $\zeta$ ”, “perfect  $\zeta$ ”, “unblemished  $\zeta$ ”, “ $\zeta$  without flaw”, “ $\zeta$  without defect”, “ $\zeta$  without damage”, ] and abnormal state = [“abnormal  $\zeta$ ”, “damaged  $\zeta$ ”, “flawed  $\zeta$ ”, “imperfect  $\zeta$ ”, “impaired  $\zeta$ ”, “blemished  $\zeta$ ”, “ $\zeta$  with flaw”, “ $\zeta$  with defect”, “ $\zeta$  with damage”], where  $\zeta$  represents the class name in the benchmarks.

#### A.3 Prompt generation via an LLM

The LLM-based prompt generation is constrained to a maximum of 50 tokens. The temperature is set to 0.9 to generate diverse and creative prompts while avoiding duplicate prompts. The input query fed to the LLM typically follows the format: “Describe what the image will look like if there is an anomaly in the image of  $\zeta$ ”. We aim for syntactic alignment between LLM-generated prompts and template-generated prompts by incorporating the instruction,

such as: “Please state the description beginning with: An abnormal image of  $\zeta$ .” into the input query for the LLM. By conducting ablation experiments, we have discovered that maintaining consistent sentence patterns significantly enhances the model’s semantic comprehension, in particular, aligning with the sentence structure employed by [10], yields improvements in anomaly detection accuracy. For the class name, we use the same operation as for the template-generated prompts to replace the ambiguous class name with a more comprehensible one for the LLM.

#### A.4 Baseline

In our evaluation, ALFA is compared to various state-of-the-art methods in zero-shot, few-shot, and full-shot regimes, spanning image-level and pixel-level anomaly detection, including CLIP-AC [10], Trans-MM [3], WinCLIP [8], AnoVL [5], PatchCore [11], PaDiM [4], JNLD [13], UniAD [12], AnomalyGPT [6], AnomalyCLIP [14], SAA+ [2], and MuSc [9], underlining its competitiveness across diverse benchmark scenarios.

Our code is available at <https://anonymous.4open.science/r/ALFA-41D6/>. A brief introduction and reproduction details of the baselines are given as follows.

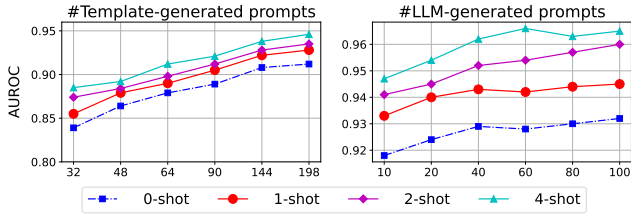
CLIP-AC [10], WinCLIP [8], AnoVL [5], AnomalyGPT [6], and AnomalyCLIP [14] are variants built upon the CLIP designed for the unified training paradigm. CLIP-AC [10] refers to the original CLIP zero-shot classification extended with prompts of the form “normal class” and “anomalous class” using a prompt ensemble. WinCLIP [8] is a window-based CLIP variant featuring a compositional ensemble of state words and prompt templates and using sliding windows for dense visual feature extraction. AnoVL [5] introduces a training-free adaptation through value-to-value attention for local patch features and a test-time adaptation using a learnable residual-like adapter to enhance anomaly localization performance. AnomalyGPT [6] generates training data by simulating anomalous images, providing fine-grained semantics through an image decoder, and fine-tuning the LVM with prompt embeddings using a prompt learner. AnomalyCLIP [14] generates learnable prompt templates for normality and abnormality and then utilizes image-level and pixel-level loss to capture generic normality and abnormality in an image regardless of its foreground objects. CLIP-AC and WinCLIP results are reported from [8], while the results for AnoVL<sup>1</sup>, AnomalyGPT<sup>2</sup>, and AnomalyCLIP<sup>3</sup> are executed using publicly available implementations.

There are five additional unified visual anomaly detection approaches. Trans-MM [3] is a model interpretation method for Transformers, offering pixel-level masks for anomaly localization. PatchCore [11] introduces a maximally representative memory bank

<sup>1</sup><https://github.com/hq-deng/AnoVL>

<sup>2</sup><https://github.com/CASIA-IVA-Lab/AnomalyGPT>

<sup>3</sup><https://github.com/zqhang/AnomalyCLIP>



**Figure 1: Effect of varying the number of anomaly prompts generated by templates and LLM on MVTec.**

of neighborhood-aware patch features obtained from an encoder pre-trained on ImageNet. UniAD [12] proposes a feature reconstruction framework consisting of a neighbor-masked encoder and a layer-wise query decoder to facilitate the model escape from the identity shortcut. SAA+ [2] introduces hybrid prompts regularization to incorporate domain expert knowledge and integrates models like GroundingDINO and SAM, leveraging their strong zero-shot generalization, to harness diverse multi-modal prior knowledge for anomaly localization. MuSc [9] performs local neighborhood aggregation with multiple degrees to obtain the patch features and then designs a mutual score of the unlabeled images as patch-level anomaly score. In the unified case, Trans-MM<sup>4</sup>, PatchCore<sup>5</sup>, UniAD<sup>6</sup>, SAA+<sup>7</sup>, and MuSc<sup>8</sup> are executed using publicly available implementations.

Additionally, PaDiM [4] and JNLD [13] are both “one-class-one-model” methods, employing a separate training scheme. The former utilizes a pre-trained convolutional neural network (CNN) for patch embedding and employs multivariate Gaussian distributions to model the probabilistic representation of the normal class. The latter comprises an anomaly reconstruction sub-network and a segmentation sub-network, both structured as encoder-decoder architectures, aiming to localize anomalies through the generated multi-scale anomalies. For fairness, we obtain the results of PaDiM and JNLD from [6], following the unified paradigm of one model across all classes in the benchmark.

## A.5 Experimental environment

In this work, all the experiments are conducted in a server with Xeon(R) Silver 4214R CPU @ 2.40GHz (12 cores), 128G memory, and GeForce RTX 3090. All the models are implemented in PyTorch 2.0.0 with CUDA 11.8.

## B ADDITIONAL RESULTS ON ABLATION STUDY

### B.1 Effect of varying prompt quantities

In Figure 1, we assess the impact of varying the number of prompts generated by both templates and LLM. First, we evaluate the performance solely based on the anomaly prompts generated by templates. Our findings reveal a notable improvement in performance in both

<sup>4</sup><https://github.com/hila-chefer/Transformer-Explainability>

<sup>5</sup><https://github.com/amazon-science/patchcore-inspection>

<sup>6</sup><https://github.com/zhiyuanyou/UniAD>

<sup>7</sup><https://github.com/caoyunkang/Segment-Any-Anomaly>

<sup>8</sup><https://github.com/xrli-U/MuSc>

**Table 1: Scalability analysis of ALFA on MVTec.**

Model	Backbone		#shot (AUROC)			
	Image Size	Layers	0	1	2	4
ViT-B/16	224 × 224	12	89.7	90.8	92.1	93.0
ViT-B/16+	240 × 240	12	93.2	94.5	95.9	96.5
ViT-L/14	224 × 224	24	92.0	93.2	94.0	94.9

zero-shot and few-shot scenarios with an increasing number of template-generated anomaly prompts, as illustrated in the right subfigure of Figure 1.

Building upon this, we introduce the anomaly prompts generated by an LLM and evaluate their impact with varying quantities. We observed that the LLM-generated anomaly prompts can further improve the performance, even when it has already reached advanced stages. Moreover, with an increasing number of LLM-generated anomaly prompts, performance gradually improves. However, the rate of improvement progressively slows down and may exhibit slight oscillations. Therefore, we adopt a combination of 198 template-generated anomaly prompts and 100 LLM-generated anomaly prompts as an initial setup in our study.

## B.2 Scalability

We evaluate the effect of various LAION-400M-based CLIP pre-trained models available at OpenCLIP<sup>9</sup>, as detailed in Table 1. Given that different backbones are associated with distinct input image sizes, we implement a unified data pre-processing procedure, employing bicubic resizing based on PIL, as described in Section A. We note that larger models or resolutions contribute to improved performance, thus we choose ViT-B/16+ with increased resolution as our default backbone.

## C ADDITIONAL QUALITATIVE RESULTS

In Figure 2-5, we present further qualitative results obtained from ALFA in the zero-shot and few-shot regime, showcasing diverse anomalies such as broken, cracked, structural changes, and bent instances of varying sizes and quantities. These results span various image classes from the MVTec and VisA benchmarks. In Figure 3 and 5, we provide normal images for each class in benchmarks as reference points in order to facilitate a clearer identification of anomalies, which were not included as inputs in the model during the zero-shot experiments.

In the zero-shot regime, relying solely on language-driven ALFA demonstrates strong capabilities in generalizing to new and unseen anomalies, excelling in detecting common anomalies. However, in certain cases, assessing anomalies without a normal image reference proves challenging, such as anomalies that are very similar to normal samples. ALFA exhibits improved detection of finer, more domain-specific anomalies in the few-shot regime. The combination of zero-shot capabilities for broad applicability and few-shot capabilities for domain-specific adaptation highlights the importance of striking a balance between generalization and specificity for effective anomaly detection.

<sup>9</sup>[https://github.com/mlfoundations/open\\_clip](https://github.com/mlfoundations/open_clip)

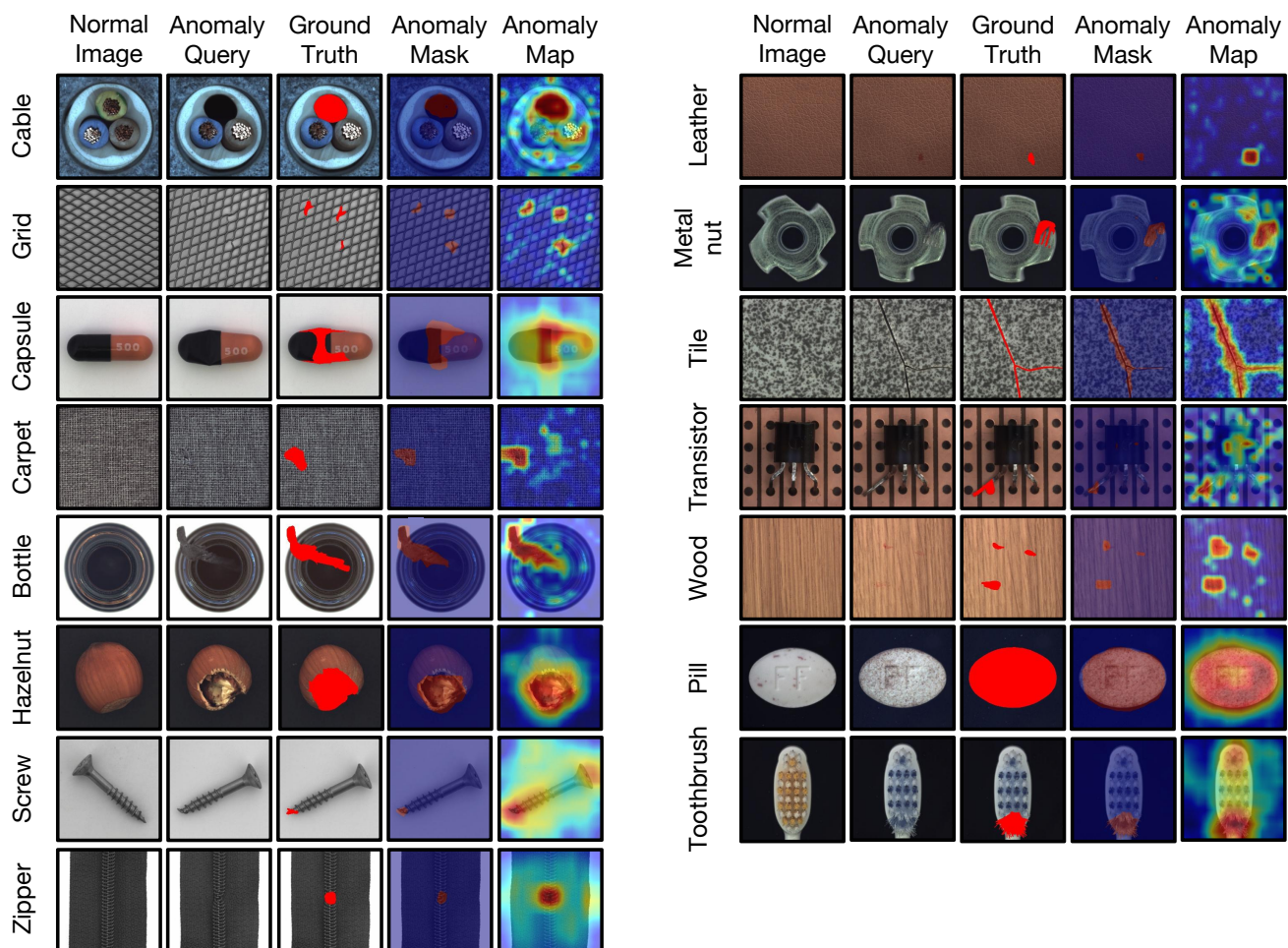


Figure 2: Detailed qualitative results of 0-shot visual anomaly detection on MVTec. The anomaly mask represents the binarized pixel-level anomaly map, where the annotated orange regions indicate detected anomalies.



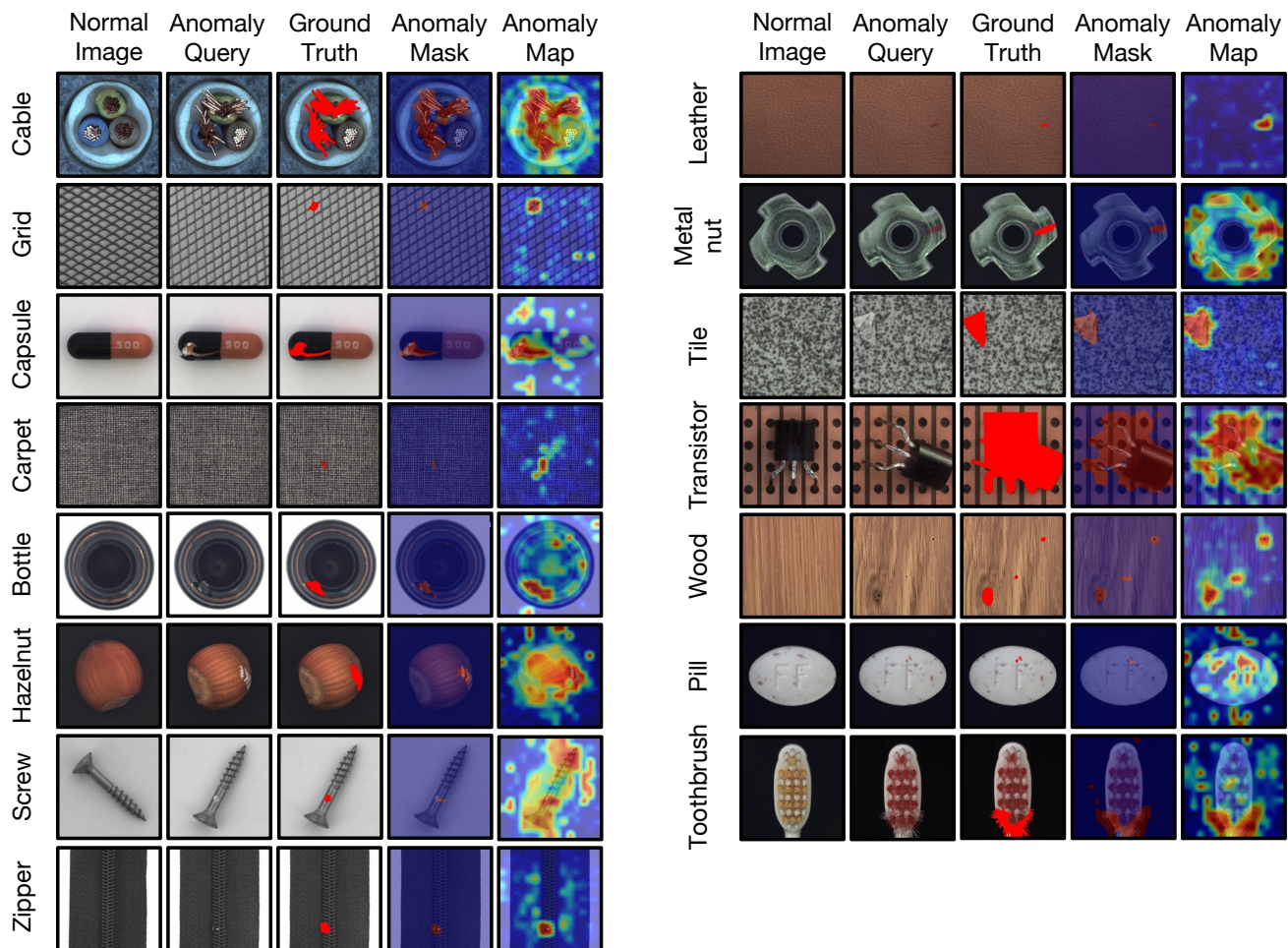


Figure 3: Detailed qualitative results of 4-shot visual anomaly detection on MVTec. The anomaly mask represents the binarized pixel-level anomaly map, where the annotated orange regions indicate detected anomalies.

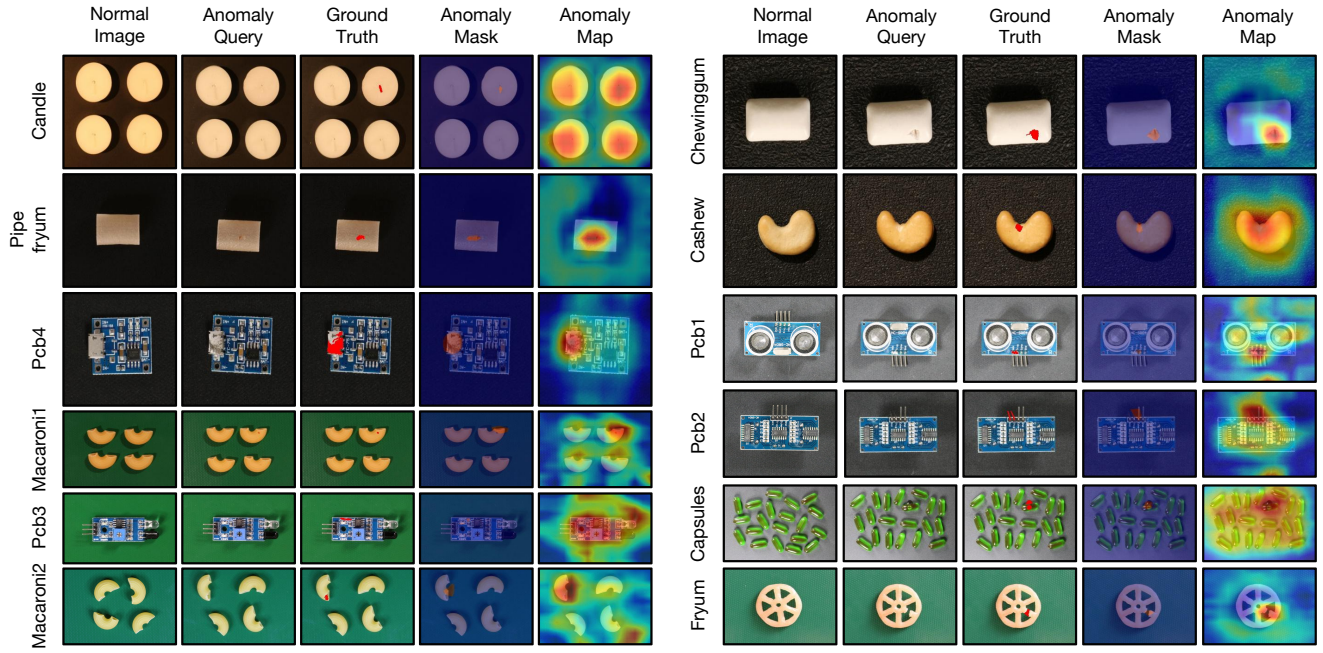


Figure 4: Detailed qualitative results of 0-shot visual anomaly detection on VisA. The anomaly mask represents the binarized pixel-level anomaly map, where the annotated orange regions indicate detected anomalies.

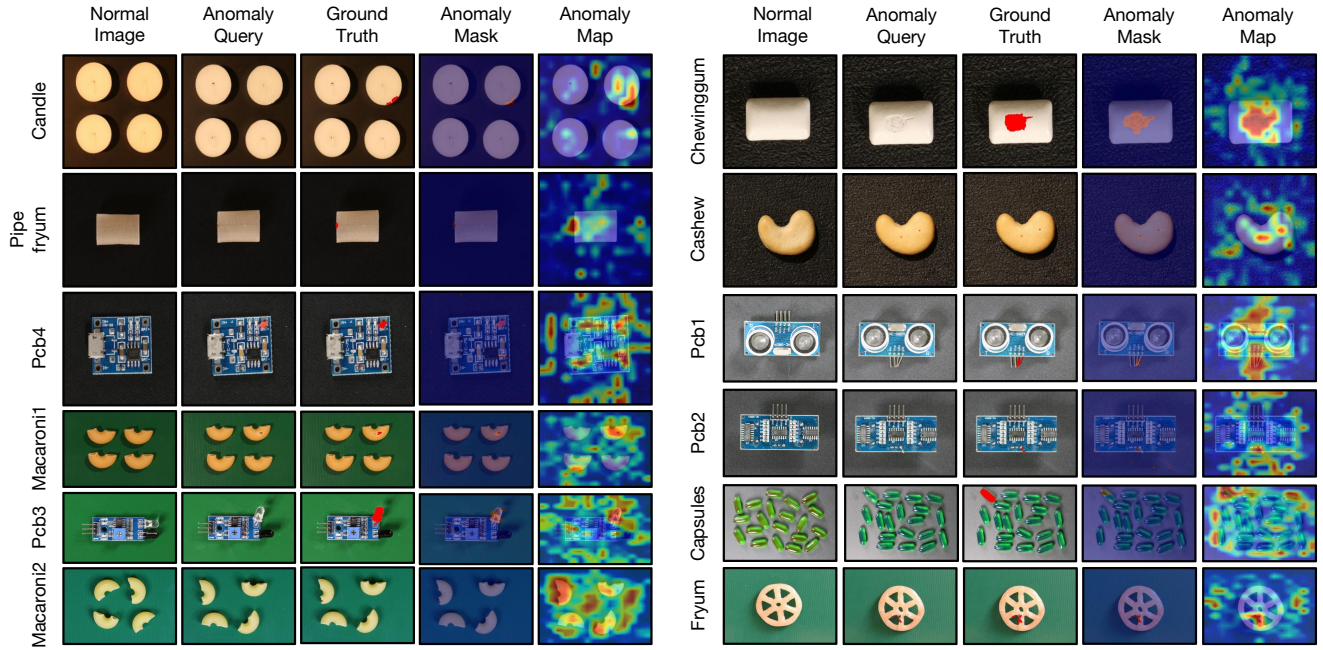


Figure 5: Detailed qualitative results of 4-shot visual anomaly detection on VisA. The anomaly mask represents the binarized pixel-level anomaly map, where the annotated orange regions indicate detected anomalies.

## REFERENCES

- [1] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. 2019. MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9592–9600.
- [2] Yunkang Cao, Xiaohao Xu, Chen Sun, Yuqi Cheng, Zongwei Du, Liang Gao, and Weiming Shen. 2023. Segment Any Anomaly without Training via Hybrid Prompt Regularization. *arXiv preprint arXiv:2305.10724* (2023).
- [3] Hila Chefer, Shir Gur, and Lior Wolf. 2021. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 397–406.
- [4] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. 2021. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*. Springer, 475–489.
- [5] Hanqiu Deng, Zhaoxiang Zhang, Jinan Bao, and Xingyu Li. 2023. AnoVL: Adapting Vision-Language Models for Unified Zero-shot Anomaly Localization. *arXiv preprint arXiv:2308.15939* (2023).
- [6] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. 2023. AnomalyGPT: Detecting Industrial Anomalies using Large Vision-Language Models. *arXiv preprint arXiv:2308.15366* (2023).
- [7] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. *OpenCLIP*. <https://doi.org/10.5281/zenodo.5143773> If you use this software, please cite it as below..
- [8] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. 2023. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19606–19616.
- [9] Xurui Li, Ziming Huang, Feng Xue, and Yu Zhou. 2024. MuSc: Zero-Shot Industrial Anomaly Classification and Segmentation with Mutual Scoring of the Unlabeled Images. *arXiv preprint arXiv:2401.16753* (2024).
- [10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.
- [11] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. 2022. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14318–14328.
- [12] Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. 2022. A unified model for multi-class anomaly detection. *Advances in Neural Information Processing Systems* 35 (2022), 4571–4584.
- [13] Ying Zhao. 2022. Just noticeable learning for unsupervised anomaly localization and detection. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 01–06.
- [14] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. 2023. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. *arXiv preprint arXiv:2310.18961* (2023).
- [15] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. 2022. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*. Springer, 392–408.