

Article

DCS: A Zero-Shot Anomaly Detection Framework with DINO-CLIP-SAM Integration

Yan Wan, Yingqi Lang * and Li Yao

College of Information and Intelligent Sciences, Donghua University, Shanghai 201620, China; winniewan@dhu.edu.cn (Y.W.); yaoli@dhu.edu.cn (L.Y.)

* Correspondence: 2232828@mail.dhu.edu.cn

Abstract

Recently, the progress of foundation models such as CLIP and SAM has shown the great potential of zero-shot anomaly detection tasks. However, existing methods usually rely on general descriptions such as “abnormal”, and the semantic coverage is insufficient, making it difficult to express fine-grained anomaly semantics. In addition, CLIP primarily performs global-level alignment, and it is difficult to accurately locate minor defects, while the segmentation quality of SAM is highly dependent on prompt constraints. In order to solve these problems, we proposed DCS, a unified framework that integrates Grounding DINO, CLIP and SAM through three key innovations. First of all, we introduced FinePrompt for adaptive learning, which significantly enhanced the modeling ability of exception semantics by building a fine-grained exception description library and adopting learnable text embeddings. Secondly, we have designed an Adaptive Dual-path Cross-modal Interaction (ADCI) module to achieve more effective cross-modal information exchange through dual-path fusion. Finally, we proposed a Box-Point Prompt Combiner (BPPC), which combines box prior information provided by DINO with the point prompt generated by CLIP, so as to guide SAM to generate finer and more complete segmentation results. A large number of experiments have proved the effectiveness of our method. On the MVTec-AD and VisA datasets, DCS has achieved the most state-of-the-art zero-shot anomaly detection results.

Keywords: anomaly detection; zero-shot learning; prompt learning

1. Introduction

Anomaly detection aims to identify whether there are anomalies or defects in industrial products and locate anomalous regions in the sample, and it plays a crucial role in product quality control and safety monitoring. Traditional anomaly detection methods [1–4] usually rely on normal samples or labeled anomaly samples for training. However, anomaly data are often limited in real-world scenarios, while factors such as privacy protection further restrict data accessibility. This issue is addressed by the emerging zero-shot anomaly detection paradigm [5], which is designed to perform anomaly detection tasks without relying on target-class anomaly samples or anomaly annotations, and is typically trained using only normal samples or general prior knowledge, thus achieving the ability to detect unknown anomalies.

Benefiting from the open-vocabulary representation capability of the large-scale vision-language model CLIP [6], many methods [5,7,8] use textual prompts like “normal”



Academic Editor: Andrea Prati

Received: 14 January 2026

Revised: 3 February 2026

Accepted: 9 February 2026

Published: 12 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

and “abnormal” to compare image features in order to achieve zero-shot anomaly detection. Recently, several approaches [9] have also attempted to leverage the segmentation foundation model SAM [10] to convert anomaly heatmaps into pixel-level segmentation results. However, existing zero-shot anomaly detection and segmentation methods still face three fundamental challenges. (1) Weak and ambiguous prompt semantics: fixed templates with a limited number of tokens are insufficient to cover fine-grained anomaly semantics, leading to semantic confusion and unstable predictions. (2) Insufficient cross-modal interaction: simply using CLIP-derived anomaly cues to guide segmentation often neglects spatial structure and multi-scale context, resulting in inaccurate boundaries and noisy masks. (3) Limited SAM prompting strategies: point-based prompts tend to concentrate on local extrema and lack stable spatial priors, making the segmentation highly sensitive to noise and threshold selection.

To overcome the aforementioned issues, we propose DCS, a novel framework for zero-shot anomaly detection that synergistically integrates CLIP [6], Grounding DINO [11], and SAM [10]. Specifically, we first employ a large language model (LLM) to generate fine-grained anomaly descriptions and combine them with the localization information produced by Grounding DINO to construct learnable prompt templates. In order to make CLIP detection more accurate, we have achieved more effective cross-modal information interaction by adopting dual-path fusion and multi-scale interaction, and weighted fusion based on the output quality of different stages. At the same time, we combine the bounding boxes of Grounding DINO with the point prompts generated by CLIP to guide SAM to produce more accurate and complete segmentation.

Our contributions can be summarized as follows:

- To enhance the anomaly semantic modeling capability of CLIP, we propose adaptive learned fine-grained text prompts (FinePrompt), which replace generic “abnormal” descriptions by constructing a library of fine-grained anomaly descriptions. In addition, we have introduced learnable text embeddings and adaptive prompt weighting, which effectively alleviate semantic ambiguity and improve accuracy.
- In order to enhance the cross-modal interaction in CLIP and improve the segmentation quality, we have proposed an Adaptive Dual-path Cross-modal Interaction (ADCI) module. By integrating dual paths at the patch level and multi-scale level, ADCI promotes the effective interaction between semantic features and visual features, so that it can learn the local and global semantics of anomaly regions. Therefore, the accuracy and stability of coarse segmentation based on CLIP have been significantly improved.
- In order to make the SAM segmentation more fine-grained and accurate, we introduced Box-Point Prompt Combiner (BPPC), which uses Grounding DINO’s box prior and CLIP-guided positive/negative point prompts to provide SAM with a more reliable prompt combination.
- Experiments conducted on multiple datasets show that our method has achieved the state-of-the-art performance in zero-shot anomaly detection. In particular, on the MVTec-AD dataset [12], our methods are 3.5%, 9.3% and 13.7% higher than the state-of-the-art CLIP-based methods in AUROC, F1-max and AP respectively, and surpass prior state-of-the-art SAM-based methods by 21.4%, 10.6% and 19.4% on the same metrics.

2. Related Works

2.1. Foundation Models

CLIP [6] achieves cross-modal feature sharing by aligning visual and textual representations. With its strong semantic understanding ability and visual-semantic expression ability, CLIP has unprecedented generalization ability. In the field of image segmentation,

the emergence of SAM [10] offers an excellent solution, leveraging boundary boxes and rough masks to generate high-quality target segmentation masks in open world scenarios, so as to carry out segmentation reasoning and accurately outline target boundaries. Grounding DINO [11] further expands the ability of the foundation model in open vocabulary target localization tasks. It can generate candidate target boundary boxes based on natural language descriptions, thus providing reliable spatial prior information for subsequent segmentation or anomaly positioning.

2.2. Zero-Shot Anomaly Detection

At present, there are three mainstream methods for zero-shot anomaly detection (ZSAD). The first method is based on CLIP [6]. WinCLIP [5] uses handcrafted text prompts and uses sliding windows to extract multi-scale features. APRIL-GAN [7] uses a learnable linear projection layer to further refine feature alignment. However, the above methods require manual text templates. AnomalyCLIP [13] replaces artificial text templates with learnable text vectors that are not related to objects. However, AnomalyCLIP uniformly uses the word “damage” to describe anomalies, which is obviously not enough to cover all types of anomalies. The above CLIP-based methods are relatively successful in solving the problem of zero-shot anomaly classification. However, in anomaly segmentation tasks, many existing methods employ coarse spatial resolution to generate anomaly maps, subsequently upscaling them via block-based bilinear interpolation. This inherently limits the precision of boundary localization. The second type of method is based on the construction of SAM [10]. Its representative methods, such as SAA [14], use Grounding DINO [11] to generate bounding-box prompts to guide SAM to perform anomaly segmentation. Although the strategy can use a powerful basic model, its effectiveness is limited by the positioning ability of Grounding DINO—the model is not specifically designed to detect abnormal patterns, so it may not be able to accurately identify abnormal areas. In addition, SAM essentially focuses on object-centered segmentation, which may lead to the division of the whole object instead of precisely isolating the fine-grained anomaly area, ultimately limiting the segmentation accuracy. The third type of method combines CLIP and SAM, such as CLIP-SAM [9], which achieves zero-shot anomaly segmentation by using CLIP to locate and provide prompts to SAM, but also uses a hand-designed text template, and needs to rely on the anomaly response of CLIP. The anomaly response greatly limits the boundary perception ability and overall anomaly segmentation ability of SAM.

2.3. Cross-Modal Interaction

In recent years, because cross-modal interaction can effectively use complementary information of different modalities, it has attracted more and more attention in the field of multimodal learning. Hu et al. [15] integrate multimodal information in the unified characterization space by serializing different modal features and using convolutional operations. STEP [16] further strengthens the cross-modal integration ability by explicitly modeling the correspondence between the prominent area of the image and the keywords related to the semantics of the text, thus improving the accuracy of fine semantic alignment. BRINet [17] improves the image segmentation performance by exchanging cross-modal information between different encoder modules, so that multi-layer features can benefit from complementary modal clues. Meanwhile, Kumar et al. [18] proposed a hybrid deep-learning anomaly detection framework (CNN-BiGRU), and emphasised the importance of automated hyper-parameter tuning to cope with multiple types of data and to improve the stability of detection. This viewpoint further illustrates the necessity of ‘hybrid structure + adaptive mechanism’ in complex scenarios from the perspective of anomaly detection. Therefore, we have introduced an adaptive dual-path cross-modal interaction module to

learn the local and global semantics of the anomaly part by integrating strip-level and multi-scale dual paths and interacting semantic features with visual features.

3. Methods

3.1. Overall Architecture

The overall architecture of the proposed model is illustrated in Figure 1. Given an input image, we first employ a large language model to generate detailed anomaly descriptions. The image and the anomaly text are then fed into Grounding DINO to obtain anomaly localization bounding boxes and corresponding positional information. Meanwhile, the fine-grained anomaly descriptions are combined with learnable text embeddings and positional information to construct prompt templates. Subsequently, the image and the text are input into the CLIP encoder to extract multi-level image features and text features. These text features, together with the image features, are then processed by the ADCI module to perform cross-modal attention-based interaction, producing a coarse anomaly map. Next, the anomaly map is passed to the BPPC module to generate point prompts, which are fused with the bounding box priors produced by Grounding DINO. The resulting combined box and point prompts are provided to SAM, thereby guiding it to segment more precise and complete anomalous regions.

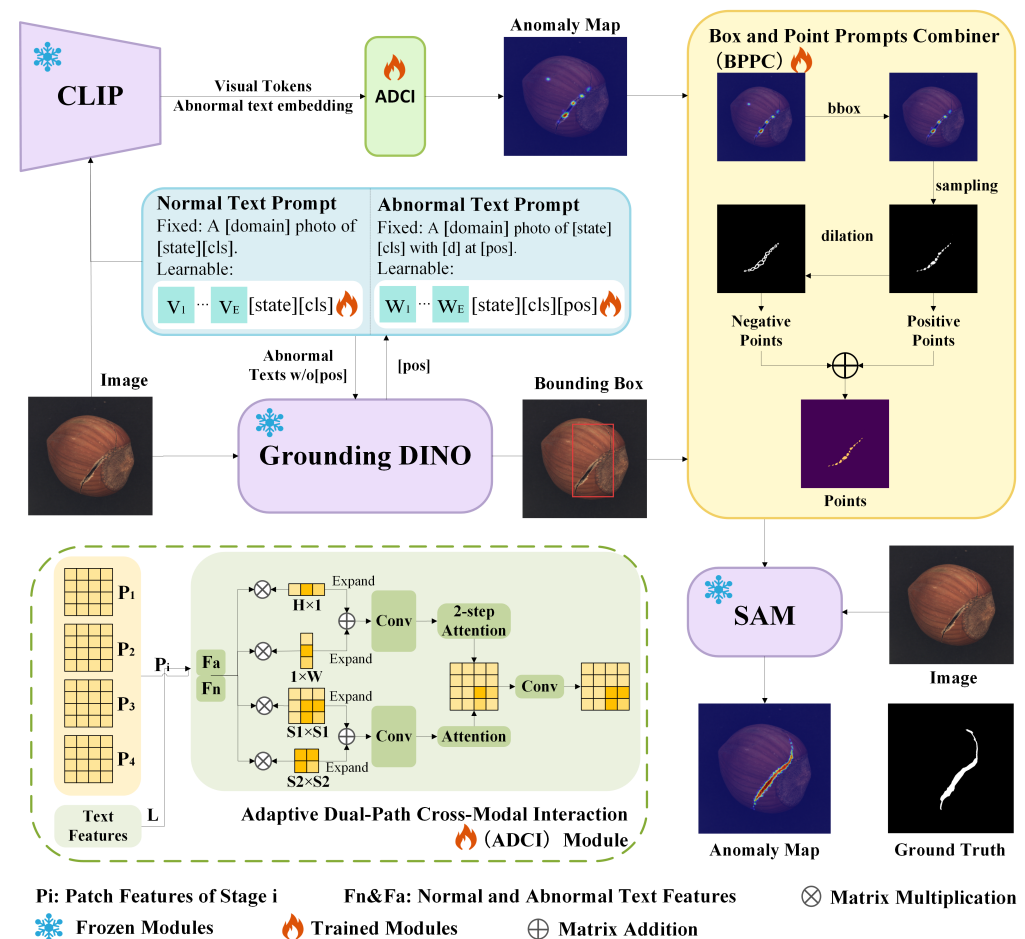


Figure 1. Overview of the proposed DCS framework. Given an input image, the LLM generates fine-grained anomaly types. Then the text is input into Grounding DINO to get the bounding box and position information, and the position information is added to the prompt template and input into CLIP. CLIP goes through ADCI cross-modal interaction to get the anomaly map, and the anomaly map goes through BPPC to get the point prompts, then it combines with the box prior and is fed into SAM, and finally, the fine-grained anomaly segmentation map is obtained.

3.2. Fine-Grained Text Templates for Adaptive Learning

Many CLIP-based methods [5,7,19] have demonstrated that the quality of text prompts has a significant impact on detection results. However, previous methods [5,7,8] usually rely on fixed text templates, such as “abnormal/damaged/defect”, or use static anomaly descriptions to describe anomaly categories. In the face of complex and diverse anomaly patterns, this practice suffers from insufficient semantic coverage, and it is difficult to express fine-grained anomaly semantics. In addition, due to the semantic overlap between different anomaly descriptions, it leads to cross-semantic confusion and unstable anomaly responses. To this end, we have dynamically enhanced text prompts by building a fine-grained anomaly description library, learning text embeddings and applying adaptive prompt weighting, which has significantly improved the ability of anomaly semantic modeling.

3.2.1. Fine-Grained Anomaly Description Library

We use a Large Language Model (LLM), such as the GPT-4o model, to prompt the LLM using the following template: ‘Based on your knowledge, enumerate the diverse types of defects that may occur in the category [CLS]’, constructing a collection of fine-grained exception descriptions for each category:

$$D_n^{cls} = \{d_1^{cls}, d_2^{cls}, \dots, d_n^{cls}\}$$

These detailed descriptions provide richer semantic coverage of the exceptions, which can more accurately match the test image and help improve detection accuracy.

It should be emphasised that LLM is only used to construct the anomaly description library in the offline phase, and does not participate in the training or inference process of the model, thus avoiding the risk of potential data leakage. To ensure the reproducibility of the anomaly description libraries, we fix the cue templates and generation configurations, and save the generated anomaly description libraries as static text libraries, which remain unchanged in both the training and inference phases.

In order to avoid a lot of semantic duplications or insufficient coverage in the anomaly descriptions generated by LLM, we introduce a diversity checking strategy when constructing the fine-grained anomaly descriptions library: firstly, removing completely duplicated descriptions; secondly, screening out descriptions that are highly semantically similar to each other to reduce the redundancy of tautology; and finally, making sure that the library is able to cover different anomaly morphologies, degrees and manifestations (e.g., scratches, scratches and scratches, etc.). Finally, we ensure that the description library can cover different anomaly forms, degrees and expressions (e.g., scratches, breaks, defects, etc.), so as to ensure the diversity and generalisation of anomaly semantics.

3.2.2. Learning Text Embeddings

Previous methods used the template: A photo of [cls]. This only includes the category of prospective objects. We introduced coarse-grained localization information and obtained the location information $pos \in P$ through the bounding box output by Grounding DINO, where P represents a 3×3 grid region (such as top-left, center, and bottom-right). Then, we combine class, state, anomaly type and position on the basis of the predefined anomaly description library:

$$A \text{ [domain] photo of [state] [cls], with [d] at [pos].}$$

where [domain] represents the domain of the object, [state] represents the normal or abnormal state, [cls] represents the category name of the object, [d] represents the detailed anomaly description, and [pos] represents the anomaly location information.

Compared with the original template, our template has significantly enhanced its performance and interpretability. Manual templates rely on manual experience design, and they are difficult to adapt to different data distributions, scene changes or anomaly types. Therefore, FinePrompt introduces learnable text embeddings. We do not share global learnable tokens between all categories, but use lightweight conditional functions $f_\theta(\cdot)$ to generate category/domain-aware tokens:

$$V = f_\theta(e_{cls}, e_{dom}), \quad W = f_\theta(e_{cls}, e_{dom}) \quad (1)$$

where e_{cls} and e_{dom} represent learnable class embedding and domain embedding, respectively. We add the above conditional tokens to the prompt sequence to form normal and abnormal prompts:

$$T_n = [V_1][V_2] \cdots [V_n][state][cls]$$

$$T_a = [W_1][W_2] \cdots [W_n][state][cls] \text{ with } [d] \text{ at } [pos]$$

where T_n and T_a represent normal and abnormal text templates, and $[V_i]$ and $[W_i]$ are learnable text embeddings.

To ensure the training stability of the learnable text vectors, we adopt a parameter-efficient optimisation: only the learnable text vectors $[V_i]$, $[W_i]$, and the class/domain embeddings e_{cls} , e_{dom} with the lightweight conditional function $f_\theta(-)$ are updated, while the image of the CLIP encoder remains frozen with the text encoder, thus ensuring semantic consistency in the image-text alignment space. We use segmentation supervision to optimise the network end-to-end, and the overall loss function is weighted and summed by pixel-level cross-entropy loss and Dice loss:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{dice} \quad (2)$$

This training method can adaptively align the learnable text vectors to the anomaly detection task while maintaining the semantic consistency of the pre-trained model, thus effectively avoiding the collapsing and overfitting problems in the text vector learning process.

3.2.3. Adaptive Prompt Weighting

In real scenarios, there is often a cross-semantic ambiguity [20] problem, i.e., there is an overlapping distance between normal text features and anomaly text features, which leads to unstable detection. To address this issue, we do not choose to directly filter and delete overlapping prompts, but assign weights to anomaly prompts according to the degree of image-text matching. Therefore, vague prompts will not be deleted, but will be naturally down-weighted, thus improving the robustness to subtle anomalies, effectively reducing cross-semantic confusion.

Specifically, let the image feature representation be $g \in R^{C_t}$, while constructing the corresponding anomaly text prompt for each anomaly description and obtaining the text feature $t_i \in R^{C_t}$. We first calculate the match score of each anomaly description with the image:

$$s_i = \frac{g^\top t_i}{\|g\| \|t_i\|} \quad (3)$$

Subsequently, all anomaly descriptions are normalised to obtain the adaptive weights:

$$w_i = \frac{\exp(s_i/\tau)}{\sum_{j=1}^n \exp(s_j/\tau)}, \quad \sum_{i=1}^n w_i = 1 \quad (4)$$

where τ is the temperature coefficient to control the sharpness of the weight distribution. Eventually, we weighted fusion of the anomalously described textual features to obtain the anomalous semantic representation:

$$t_a = \sum_{i=1}^n w_i t_i \quad (5)$$

Under this mechanism, prompts that are semantically ambiguous or do not match the current image typically exhibit lower similarity s_i , and thus receive smaller weights w_i after softmax normalization, achieving “natural weight reduction”. Compared with the hard filtering strategy, our approach preserves potentially valid fine-grained descriptions and mitigates the impact of noise through adaptive weight adjustment, making anomaly detection more stable in complex cross-semantic scenarios.

3.3. Adaptive Dual-Path Cross-Modal Interaction

The existing CLIP-based anomaly detection methods directly construct an anomaly map by calculating the similarity. This lacks spatial structure modeling and multi-scale interaction, resulting in significant anomaly localization noise. To address this issue, we propose ADCI. The core idea of ADCI is a dual-path cross-modal interaction mechanism: Strip Path strengthens the localisation ability of strip and boundary structures through row/column direction attentional aggregation; Scale Path strengthens the robustness to anomalies of different sizes through multi-scale context fusion; and ultimately, the outputs of the two paths are adaptively fused through the gating coefficients. The final result is a more stable and accurate anomaly segmentation prediction through the adaptive fusion of the outputs of the two paths by gating coefficients.

Given an image img and the prompts constructed by FinePrompt, the CLIP encoder generates image patch tokens $P_i \in \mathbb{R}^{H \times W \times C}$ and text feature vectors $L \in \mathbb{R}^{C_t \times 2}$. Instead of performing post-processing on the CLIP outputs, we establish image–text interactions at the token level and directly produce an anomaly response map, thereby enabling more effective cross-modal information fusion.

3.3.1. Strip Path Based on Attention-Weighted Pooling

The Strip Path enhances localization capability by incorporating row-wise and column-wise feature modeling. We first project the patch tokens into the same embedding space as the text features:

$$\hat{P} = \phi(P) \in \mathbb{R}^{H \times W \times C_t} \quad (6)$$

To extract features from $\hat{P}$, average pooling is commonly adopted. However, average pooling treats all positions within a strip equally and may overlook subtle anomaly cues. Therefore, we replace average pooling with attention-weighted pooling. Specifically, for each row h , attention weights along the width dimension are computed as:

$$a_{h,w}^{row} = \frac{\exp(\psi_{row}(\hat{P}_{h,w}))}{\sum_{w'=1}^W \exp(\psi_{row}(\hat{P}_{h,w'}))} \quad (7)$$

and the corresponding row-level feature is obtained as:

$$v_{\text{row}}(h) = \sum_{w=1}^W a_{h,w}^{\text{row}} \hat{p}_{h,w} \quad (8)$$

Similarly, for each column w , we compute:

$$a_{h,w}^{\text{col}} = \frac{\exp(\psi_{\text{col}}(\hat{p}_{h,w}))}{\sum_{h'=1}^H \exp(\psi_{\text{col}}(\hat{p}_{h',w}))}, \quad v_{\text{col}}(w) = \sum_{h=1}^H a_{h,w}^{\text{col}} \hat{p}_{h,w} \quad (9)$$

Subsequently, 1D convolutions are applied to model local continuity:

$$v_{\text{row}} = \text{conv}_{1 \times 3}(v_{\text{row}}), \quad v_{\text{col}} = \text{conv}_{3 \times 1}(v_{\text{col}}) \quad (10)$$

where $v_{\text{row}} \in \mathbb{R}^{H \times c_h}$ and $v_{\text{col}} \in \mathbb{R}^{W \times c_h}$ denote the row-level and column-level features, respectively, and c_h is the hidden dimension. Next, we describe the internal procedure of the Strip Path. Taking v_{row} as an example, two convolutional layers are applied to the text features L to obtain:

$$t_{\text{row}}^1, t_{\text{row}}^2 \in \mathbb{R}^{c_h \times 2} \quad (11)$$

A two-step attention mechanism is then employed to predict the language-aware strip representation:

$$M_{\text{row}} \in \mathbb{R}^{H \times c_h} \quad (12)$$

where v_{row} serves as the query, and $t_{\text{row}}^1, t_{\text{row}}^2$ act as the key and value, respectively. In the same manner, $M_{\text{col}} \in \mathbb{R}^{W \times c_h}$ can be obtained.

Finally, M_{row} and M_{col} are upsampled to the original spatial resolution via bilinear interpolation and fused as follows:

$$M_{\text{row,col}} = \text{conv}_{3 \times 3}(B(M_{\text{row}}) + B(M_{\text{col}})) \quad (13)$$

where $B(\cdot)$ denotes the bilinear interpolation layer. The final output is $M_{\text{row,col}} \in \mathbb{R}^{H \times W \times c_h}$.

3.3.2. Scale Path

Scale Path enhances semantic understanding through multi-scale global context. In the scale path, the projected image feature $\hat{P} \in \mathbb{R}^{H \times W \times c_h}$ is processed at multiple scales to capture global semantic information. We use two types of pooled cores, s_1 and s_2 :

$$v_{g1} = \text{conv}_{3 \times 3}^{g1}(\text{Avg_Pool}_{s1 \times s1}(\hat{P})), \quad (14)$$

$$v_{g2} = \text{conv}_{3 \times 3}^{g2}(\text{Avg_Pool}_{s2 \times s2}(\hat{P})) \quad (15)$$

where $v_{g1} \in \mathbb{R}^{h_{g1} \times w_{g1} \times c_h}$ and $v_{g2} \in \mathbb{R}^{h_{g2} \times w_{g2} \times c_h}$ denote visual features at different scales.

To illustrate the internal process of the Scale Path, we take v_{g1} as an example. The text features are first convolved to obtain: $t_{g1}^1, t_{g1}^2 \in \mathbb{R}^{c_h \times 2}$. Then, the language perception map is obtained through the attention mechanism:

$$M_{g1} \in \mathbb{R}^{h_{g1} \times w_{g1} \times c_h} \quad (16)$$

where v_{g1} is the query, and (t_{g1}^1, t_{g1}^2) is the key/value. Similarly, M_{g2} can be obtained. Then, the feature maps are resized via bilinear interpolation and fused as follows:

$$M_{g1,g2} = \text{conv}_{3 \times 3}^{g1,g2}(B(M_{g1}) + B(M_{g2})) \quad (17)$$

Finally, the fused feature $M_{g1,g2} \in \mathbb{R}^{H \times W \times c_h}$ is obtained.

3.3.3. Gated Dual Path Fusion

Through Strip Path and Scale Path, we get pixel-level predictions $M_{row,col}$ and $M_{g1,g2}$, which provide localization and semantic information respectively. However, different anomalies depend on the two paths to different degrees. Therefore, we have introduced a gate-controlled integration mechanism and adaptively balanced the contribution of the two paths. We first calculate the gate control coefficient based on the projected patch tokens $\hat{P}$:

$$\eta = \sigma(MLP(GAP(\hat{P}))), \quad \eta \in [0, 1] \quad (18)$$

where $GAP(\cdot)$ is a global average pooling operation, and σ is a sigmoid function. The fused cross-modal prediction is then formulated as:

$$M_{fuse} = \eta \cdot M_{row,col} + (1 - \eta) \cdot M_{g1,g2} \quad (19)$$

In order to preserve the visual details, we introduce residual branches from $\hat{P}$:

$$v_{ori} = conv_{3 \times 3}^{ori}(\hat{P}) \quad (20)$$

and integrate it with the cross-modal prediction:

$$M_{all} = conv_{3 \times 3}^{all}\left(concat\left(v_{ori}, M_{fuse}\right)\right) \quad (21)$$

Finally, we use an MLP as the segmentation head to get the coarse segmentation result:

$$O = MLP(ReLU(M_{all} + \hat{P})) \quad (22)$$

where $O \in \mathbb{R}^{H \times W \times 2}$ is the output of the ADCI module, and dimension 2 corresponds to the anomaly foreground and background.

3.3.4. Multi-Stage Weighted Integration of Quality Perception

Assuming that the encoder has a total of n stages, the split output of the i th stage is O_i . Different stages have different feature scales, and simple average fusion can easily introduce noise. Therefore, we use quality-aware weighted fusion to weight it according to the output quality of each stage, so as to suppress the error propagation of low-quality stages and achieve more reliable aggregation of multi-stage results:

$$\alpha = \text{softmax}(q), \quad q \in \mathbb{R}^n, \quad O = \sum_{i=1}^n \alpha_i \cdot O_i \quad (23)$$

Our adaptive aggregation mechanism can emphasize more effective stages, thus improving the stability and accuracy of segmentation.

3.4. Box-Point Prompt Combiner

The segmentation quality of SAM is highly dependent on the spatial priors and discriminative capability of the prompts. Using only the point prompts selected from the anomaly map tends to concentrate on local extrema and lacks spatial constraints; using only box prompts tends to be too rough and makes it difficult to delineate anomaly boundaries. For this reason, we propose a Box-Point Prompt Combiner (BPPC), which complements and integrates the coarse-grained box prior generated by Grounding DINO with the positive and negative point prompts guided by CLIP, providing SAM with combined box prompts and point prompts. Box prompts provide coarse-grained spatial constraints

to avoid over-segmentation, while point prompts provide fine-grained boundary guidance to refine anomaly edges, resulting in a finer and more complete anomaly region for SAM segmentation.

3.4.1. Box Prompt Positioning

We use Grounding DINO to generate candidate bounding boxes $\{B_i\}$ and their confidence scores, which can be used as rough object-level spatial priors rather than anomaly detectors. Although Grounding DINO is not specifically trained for anomaly detection, it provides reliable foreground anomaly localisation and effectively suppresses background interference. However, the raw DINO box typically covers the entire object, which may be too coarse for accurate anomaly segmentation. We therefore use the CLIP anomaly map Map_a to filter and calibrate the boxes. Calculate the anomaly energy inside each candidate box:

$$\text{Score}(B_i) = \frac{1}{|B_i|} \sum_{(x,y) \in B_i} \text{Map}_a(x,y) \quad (24)$$

and the Top- K_b boxes with the highest anomaly energy are selected:

$$B = \text{Top}_{K_b}(\{\text{Score}(B_i)\}) \quad (25)$$

To further align the bounding boxes with anomalous regions, we extract high-response anomaly areas within each selected box and compute the tight enclosing rectangle to obtain the refined box:

$$B^* = \text{BBox}(\{(x,y) \in B \mid \text{Map}_a(x,y) \geq \rho\}) \quad (26)$$

where ρ denotes a percentile-based threshold (e.g., selecting the top $\rho\%$ of anomaly responses). The calibrated box B^* provides more informative spatial constraints for SAM, improving boundary accuracy and reducing redundant segmentation.

3.4.2. Positive-Point Positioning

After processing by ADCI, CLIP produces an anomaly map Map_a . Meanwhile, extreme anomalous regions S_a are identified via thresholding. The anomaly map is then intersected with the extreme anomalous regions to obtain the anomaly map of extreme regions R_a :

$$R_a = \text{Map}_a \otimes S_a \quad (27)$$

where $\otimes$ denotes element-wise multiplication. The number of positive points k is adaptively determined based on the anomaly scale:

$$k = \text{clip}\left(\alpha \cdot \frac{|S_a|}{HW}, k_{\min}, k_{\max}\right) \quad (28)$$

Next, directly selecting the Top- k anomalous points may lead to excessive spatial concentration. Therefore, we adopt a representative sampling strategy. Specifically, the Top- k candidate anomalous points are first selected from R_a , and then farthest point sampling is applied to obtain a more spatially uniform set of positive points P_h , ensuring that the positive points adequately cover the entire anomalous region.

$$P_h = \text{RepSample}_k(R_a) \quad (29)$$

3.4.3. Negative Point Positioning

In order to better refine the anomaly boundary, we mine negative points in the ring-shaped neighborhood of the extreme anomaly area. If the negative points are selected only according to the global anomaly score, the negative points often fall into background regions away from the anomaly area, which will cause SAM to tend to segment the whole object instead of focusing on the anomaly area, resulting in the ineffectiveness of the negative point prompt. Therefore, we construct negative samples of boundary perception around the extreme anomaly region. First, we perform a dilation operation on the anomaly area to obtain the ring area N_a around the extreme anomaly area:

$$N_a = \text{dilate}(S_a) - S_a \quad (30)$$

In order to make the negative points more conducive to boundary refinement, we further limit the negative point candidate area to the intersection of the surrounding area N_a and the calibration box B :

$$N_a^* = N_a \cap B^* \quad (31)$$

Next, we extract global features F using the image encoder:

$$F = \text{Enc}_I(\text{img}) \quad (32)$$

where Enc_I denotes the SAM image encoder. Subsequently, local features of the extreme anomaly region F_a and its surrounding region F_n are obtained via spatial-wise multiplication:

$$F_a = F \otimes S_a, \quad F_n = F \otimes N_a \quad (33)$$

Then, we compare F_a and F_n by cosine similarity. The lower the score, the more likely these points are to be outside the anomaly area. We use this method to choose k pixels as negative points:

$$\text{Map}_s = \text{Similarity}(F_a, F_n), \quad P_l = \text{Lowest}_k(\text{Map}_s) \quad (34)$$

where k is adaptively determined according to the anomaly scale.

3.4.4. Box-Point Prompt Combination

Finally, we combine the obtained box prompt B^* , positive points P_h , and negative points P_l into a unified prompt set P , and feed the composed prompt P together with the image features F into the SAM decoder:

$$P = \text{Compose}(B^*, P_h, P_l), \quad M_1, \text{logit}_1 = \text{Dec}_m(F, P) \quad (35)$$

We then select the segmentation result with the highest confidence as the final mask:

$$M = \arg \max_j \text{logit}_j \quad (36)$$

By using both box priors and fine-grained point prompts, BPPC can guide SAM to segment anomaly areas more steadily and significantly improve boundary accuracy.

4. Experiments

4.1. Datasets

We used two publicly available datasets to evaluate the performance: MVTec-AD [12] and VisA [21]. MVTec-AD [12] is one of the most widely used industrial anomaly detection datasets, containing 5354 normal and abnormal sample images from 15 different object categories, with resolutions varying from 700×700 to 1024×1024 pixels. VisA [21] is an emerging industrial anomaly detection dataset, containing 10,821 normal and abnormal sample images, covering 12 image categories, with a resolution of about 1500×1000 pixels. We follow the same training settings as existing zero-shot anomaly segmentation studies [5,7] to evaluate the performance of our method. Specifically, the model is first trained on the MVTec-AD dataset and then tested on the VisA dataset, and vice versa.

4.2. Evaluation Metrics

Like previous anomaly detection research [5], we use widely used indicators, namely pixel-level AUROC, F1-max and AP as evaluation metrics, and make a fair and comprehensive comparison with existing methods. Specifically, pixel-level AUROC reflects the ability of the model to distinguish anomaly and normal regions at different threshold levels. Pixel-level F1-max represents the harmonic mean of precision and recall under the optimal threshold. Pixel-level AP quantifies the accuracy of the model at different levels of recall. The higher the values of these indicators, the better the performance of the evaluation method.

4.3. Implementation Details

In the experiment, the CLIP encoder adopted a public pre-trained ViT-L-14-336 model, which consists of 24 Transformer layers and remains frozen during the training process. We extract patch-level features from layers 6, 12, 18 and 24 of the CLIP image encoder and input them into the ADCI module as multi-stage visual representation. For anomaly localization, we adopt the pre-trained Grounding DINO model and keep its parameters unchanged in all experiments. For anomaly segmentation, we use the pre-trained ViT-H model, and the parameters of the model are also fixed. In the experiment, only FinePrompt, ADCI and BPPC are updated. We use the AdamW optimizer [22] to optimize the trainable components. The learning rate is set to 1×10^{-4} , the batch size is 8, and a total of 6 epochs are trained.

4.4. Main Results

In order to verify the effectiveness of our method, we evaluated and compared it with the most advanced zero-shot anomaly detection methods, including WinCLIP [5], APRIL-GAN [7], SDP [23], SDP+ [23], AnomalyCLIP [13], AdaCLIP [24], SAA [14], SAA+ [14] and CLIP-SAM [9]. These comparison methods cover CLIP-based text cueing methods, SAM-based segmentation cueing methods, and representative routes combining CLIP and SAM. Table 1 lists the experimental results on the VisA and MVTec-AD datasets. The results show that our method is superior to the existing most advanced methods in all three indicators, verifying the FinePrompt, ADCI and BPPC modules.

Figure 2 shows the visualization results of our method, among which ten sample pictures are selected from the MVTec-AD and VisA datasets. At the same time, we also showed the visual comparison results of APRIL-GAN [7] and AnomalyCLIP [13]. It can be seen that compared with these methods, our method achieves better anomaly region localization and segmentation, showing stronger performance.

Table 1. Performance comparison of different kinds of ZSAD approaches on the MVTec-AD and VisA datasets. Evaluation metrics include AUROC, F1-max and AP. Bold indicates the best results, and underline indicates the second-best results.

Base Model	Method	Anomaly Description	VisA			MVTec-AD		
			AUROC	F1-Max	AP	AUROC	F1-Max	AP
CLIP-based Approaches	WinCLIP	state ensemble	79.6	14.8	–	85.1	31.7	–
	APRIL-GAN	state ensemble	94.2	32.3	25.7	87.6	43.3	40.8
	SDP	normal/anomalous	84.1	16.0	9.6	88.7	35.3	28.5
	SDP+	normal/anomalous	94.8	26.5	20.3	91.2	41.9	39.4
	AnomalyCLIP	normal/damaged	95.5	28.3	21.3	91.1	39.1	34.5
	AdaCLIP	normal/damaged	95.5	32.9	25.8	88.7	45.2	42.7
SAM-based Approaches	SAA	normal/defective	83.7	12.8	5.5	67.7	23.8	15.2
	SAA+	normal/defective	74.0	27.1	22.4	73.2	37.8	28.8
CLIP&SAM	ClipSAM	normal/anomalous	<u>95.6</u>	<u>33.1</u>	<u>26.0</u>	<u>92.3</u>	<u>47.8</u>	<u>45.9</u>
CLIP&DINO&SAM	Ours	fine-grained description	97.2	35.6	29.1	94.6	48.4	48.2

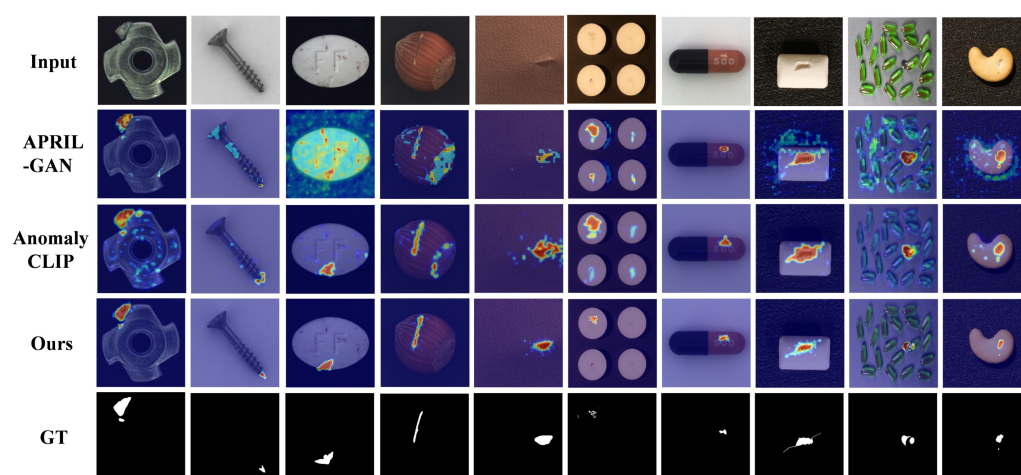


Figure 2. Comparison of visualization results among DCS, APRIL-GAN and AnomalyCLIP on the MVTec-AD dataset.

In order to evaluate the additional complexity brought by multi-model collaboration, we compared the reasoning speed (FPS) of DCS, WinCLIP [5] and SAA [14] under the same hardware environment and input settings, and also decomposed and analysed the overhead of the DCS inference phase, and the results are shown in Table 2. Table 2A gives a comparison of the inference efficiency at the method level. WinCLIP relies only on CLIP for graphical matching and has the most efficient inference speed; SAA is based on SAM for segmentation inference, which has a larger computational volume and significantly lower inference speed. In contrast, DCS combines the three models of CLIP, Grounding DINO and SAM to achieve finer anomaly localisation and segmentation, so the overall inference speed decreases further, but still remains within the acceptable range.

Additionally, Table 2B further demonstrates the module-level inference overhead distribution of DCS. It can be observed that the main inference overhead of DCS comes from the candidate frame generation of Grounding DINO and the mask decoding process of SAM (ViT-H), which take up the vast majority of the computation time; in contrast, the online overhead of FinePrompt is small due to the fact that the anomaly description library with the set of discrete locations $pos \in P$ is kept fixed in the inference phase, and we pre-cache the corresponding text feature representations, which reduces the computational

burden of text construction and encoding. ADCI, as a cross-modal interaction module, introduces an additional computational overhead, but its time consumption is relatively controllable, which means that it does not impose a high inference burden while providing a significant performance improvement. Overall, the results show that DCS strikes a good balance between accuracy improvement and computational overhead.

Table 2. Comparison of method-level reasoning efficiency and DCS internal overhead percentage. (A) Comparison of method-level reasoning efficiency; (B) DCS reasoning phase overhead decomposition. For FPS, higher values indicate better performance ($\uparrow$), while for latency, lower values are better ($\downarrow$).

(A) Method-level reasoning efficiency comparison		
Method	FPS $\uparrow$	Latency (ms/img) $\downarrow$
WinCLIP	40.0	25.0
SAA	4.0	250.0
DCS (Ours)	2.2	455.0

(B) DCS reasoning phase overhead decomposition (Latency)		
Component (Inference)	Latency (ms/img)	Share (%)
Grounding DINO (box + pos)	140.0	30.8
FinePrompt (pos-aware prompt assembly)	3.0	0.7
CLIP image encoding (ViT-L/14-336)	25.0	5.5
CLIP text (cached lookup + similarity)	10.0	2.2
ADCI (cross-modal interaction head)	40.0	8.8
BPPC (point sampling + prompt compose)	12.0	2.6
SAM (ViT-H prompt encoder + mask decoder)	225.0	49.5
Total (DCS)	455.0	100

Note: The table counts the overhead of the CLIP image/text encoder separately as the backbone feature extraction time; ADCI counts only the additional overhead of the cross-modal interaction head (Equations (2)–(18)) to more clearly reflect the incremental computational cost of the proposed module.

4.5. Ablation Study

In order to evaluate the effectiveness of the modules we proposed and their internal designs, we conducted extensive ablation experiments. Among them, we use the VisA dataset as the training set and the MVTec-AD dataset as the test set. In addition, all ablation experiments are carried out under the same training and inference settings.

4.5.1. Main Ablation Study

Table 3 presents the ablation results of the FinePrompt, ADCI, and BPPC modules. As can be observed, each module yields a certain performance improvement over the baseline. Notably, combinations of any two modules generally outperform the use of a single module. When all three modules are jointly applied, the model achieves the highest performance, indicating a positive synergistic effect among semantic modeling, feature interaction, and prompt strategy, and thereby validating the rationality and effectiveness of our model design.

Table 3. Main ablation study of different modules in the DCS framework. The best performing results are shown in bold.

FinePrompt	ADCI	BPPC	AUROC	F1-Max	AP
$\checkmark$			93.6	44.5	44.2
	$\checkmark$		92.8	45.9	44.6
		$\checkmark$	92.7	46.4	45.4
$\checkmark$	$\checkmark$		94.1	46.9	46.0

Table 3. *Cont.*

FinePrompt	ADCI	BPPC	AUROC	F1-Max	AP
✓		✓	94.2	47.6	47.0
	✓	✓	93.9	47.8	47.1
✓	✓	✓	94.6	48.4	48.2

Note: ✓ indicates that the corresponding module is enabled in the model.

4.5.2. Ablation Study of the FinePrompt Module

To verify the effectiveness of the FinePrompt design, we conduct an ablation study on the internal components of the FinePrompt module in Table 4, while keeping ADCI and BPPC fixed. As shown by the results, removing the fine-grained anomaly descriptions leads to the most significant performance degradation, indicating that fine-grained descriptions provide more discriminative semantic information. The position tokens, learnable text embeddings, and adaptive weighting mechanisms also contribute to performance improvements. Using the complete FinePrompt achieves the best overall performance, demonstrating the effectiveness and rationality of each component.

Table 4. Ablation study on FinePrompt. We keep all other modules fixed and only vary the components inside FinePrompt. The best performing results are shown in bold.

FG-desc	Pos	LearnTok	ReWeight	AUROC	F1-Max	AP
	✓	✓	✓	93.4	46.3	46.0
✓		✓	✓	93.9	47.1	46.8
✓	✓		✓	94.1	47.4	47.1
✓	✓	✓		94.3	47.8	47.6
✓	✓	✓	✓	94.6	48.4	48.2

Note: ✓ indicates that the corresponding module is enabled in the model.

4.5.3. Ablation Study of ADCI Modules

Next, we fix FinePrompt and BPPC and conduct an ablation study on each component within ADCI. The results are reported in Table 5. According to the results, replacing the attention-weighted Strip Pooling with standard average pooling leads to decreases in both F1-max and AP, indicating that attention-weighted Strip Pooling can effectively preserve local anomaly responses. We then remove the dual-path structure by retaining only the Strip Path and discarding the Scale Path, which results in degraded performance. This demonstrates that the Strip Path and Scale Path exhibit strong complementarity in terms of localization information and multi-scale semantic modeling. Furthermore, replacing the gated fusion mechanism with simple summation also causes a performance drop, suggesting that gated dual-path fusion is beneficial for adaptively balancing the contributions of the two paths. When the quality-aware multi-stage weighted fusion is removed, the performance further deteriorates, indicating that this strategy helps suppress noise interference from low-quality stages. Overall, the complete ADCI achieves the best performance across all metrics, confirming that its internal components cooperate effectively without mutual interference. These results validate the rationality of the proposed method.

Table 5. Ablation study on ADCI. All other modules are fixed, and we only vary the internal designs of ADCI. The best performing results are shown in bold.

SoftPool	DualPath	Gate	StageW	AUROC	F1-Max	AP
	✓	✓	✓	94.0	47.2	47.0
✓		✓	✓	94.1	47.4	47.2
✓	✓		✓	94.3	47.8	47.6
✓	✓	✓		94.4	48.0	47.8
✓	✓	✓	✓	94.6	48.4	48.2

Note: ✓ indicates that the corresponding module is enabled in the model.

4.5.4. An Ablation Study of the BPPC Module

As shown in Table 6, we further conduct an ablation study on different prompting strategies in BPPC. The results indicate that using only DINO box prompts or only CLIP point prompts fails to achieve optimal performance. When box prompts are combined with positive point prompts, the performance improves significantly. The best results are obtained by further introducing boundary-aware negative point prompts, demonstrating that representative positive point sampling enhances the coverage of anomalous regions, while boundary-aware hard negative points facilitate boundary refinement. The combination of positive and negative point prompts with box prompts is crucial for fine-grained anomaly segmentation with SAM.

Table 6. Ablation study on BPPC. All other modules are fixed, and we only vary the prompt composition strategy for SAM. The best performing results are shown in bold.

Box (B^*)	PosPts (P_h)	NegPts (P_l)	AUROC	F1-Max	AP
✓			94.0	47.1	46.7
	✓	✓	93.8	47.0	46.6
✓	✓		94.3	47.8	47.6
✓	✓	✓	94.6	48.4	48.2

Note: B^* denotes the refined bounding boxes after calibration. ✓ indicates that the corresponding module is enabled in the model.

5. Conclusions

In this article, we propose DCS, a zero-shot anomaly detection framework that integrates Grounding DINO, CLIP, and SAM, which effectively solves the dual challenges of precise localization and precise segmentation. Specifically, we have introduced three modules. First of all, the FinePrompt module combines the knowledge advantages of an LLM with learnable text prompts to effectively adapt to various types of anomalies. This method significantly improves the accuracy and interpretability of anomaly detection. In addition, the ADCI module we designed significantly improves the accuracy and stability of CLIP coarse segmentation through the dual-path cross-modal interaction. Finally, the BPPC module provides SAM with joint box and point prompts, guiding SAM to achieve more robust segmentation with more precise boundaries. Extensive experimental results validate the superiority of DCS, confirming its effectiveness and practicality in zero-shot anomaly detection tasks.

Author Contributions: Conceptualization, Y.W., Y.L. and L.Y.; methodology, Y.W. and Y.L.; software, Y.L.; validation, Y.W. and Y.L.; formal analysis, Y.L.; investigation, Y.W. and Y.L.; resources, Y.W.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, Y.W. and L.Y.; visualization, Y.L.; supervision, Y.W.; and project administration, Y.W. and L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Defard, T.; Setkov, A.; Loesch, A.; Audigier, R. Padim: A patch distribution modeling framework for anomaly detection and localization. In *Proceedings of the International Conference on Pattern Recognition, Virtual Event, 10–15 January 2021*; Springer International Publishing: Cham, Switzerland, 2021; pp. 475–489.
- Deng, H.; Li, X. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022*; pp. 9737–9746.
- 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, New Orleans, LA, USA, 18–24 June 2022*; pp. 14318–14328.
- You, Z.; Cui, L.; Shen, Y.; Yang, K.; Lu, X.; Zheng, Y.; Le, X. A unified model for multi-class anomaly detection. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 4571–4584.
- 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, Vancouver, BC, Canada, 17–24 June 2023*; pp. 19606–19616.
- 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 *Proceedings of the 38th International Conference on Machine Learning, Pmlr, Virtual, 18–24 July 2021*; pp. 8748–8763.
- 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* **2023**, arXiv:2305.17382.
- 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, Vancouver, BC, Canada, 20–27 February 2024*; AAAI Press: Menlo Park, CA, USA, 2024; Volume 38, pp. 1932–1940.
- Li, S.; Cao, J.; Ye, P.; Ding, Y.; Tu, C.; Chen, T. ClipSAM: CLIP and SAM collaboration for zero-shot anomaly segmentation. *Neurocomputing* **2025**, *618*, 129122. [[CrossRef](#)]
- Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.-Y.; et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023*; pp. 4015–4026.
- Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024*; Springer Nature: Cham, Switzerland, 2024; pp. 38–55.
- Bergmann, P.; Fauser, M.; Sattlegger, D.; Steger, C. MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019*; pp. 9592–9600.
- Zhou, Q.; Pang, G.; Tian, Y.; He, S.; Chen, J. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. *arXiv* **2023**, arXiv:2310.18961.
- Cao, Y.; Xu, X.; Sun, C.; Cheng, Y.; Du, Z.; Gao, L.; Shen, W. Segment any anomaly without training via hybrid prompt regularization. *arXiv* **2023**, arXiv:2305.10724. [[CrossRef](#)]
- Hu, R.; Rohrbach, M.; Darrell, T. Segmentation from natural language expressions. In *Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016*; Springer International Publishing: Cham, Switzerland, 2016; pp. 108–124.
- Chen, D.J.; Jia, S.; Lo, Y.C.; Chen, H.T.; Liu, T.L. See-through-text grouping for referring image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019*; pp. 7454–7463.
- Feng, G.; Hu, Z.; Zhang, L.; Sun, J.; Lu, H. Bidirectional relationship inferring network for referring image localization and segmentation. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *34*, 2246–2258. [[CrossRef](#)]
- Kumar, D.; Pawar, P.P.; Addula, S.R.; Meesala, M.K.; Oni, O.; Cheema, Q.N.; Haq, A.U.; Sajja, G.S. AI-Powered security for IoT ecosystems: A hybrid deep learning approach to anomaly detection. *J. Cybersecur. Priv.* **2025**, *5*, 90. [[CrossRef](#)]
- Deng, H.; Zhang, Z.; Bao, J.; Li, X.A. Adapting Vision-Language Models for Unified Zero-Shot Anomaly Localization. *arXiv* **2023**, arXiv:2308.15939.

20. Zhu, J.; Cai, S.; Deng, F.; Ooi, B.C.; Wu, J. Do LLMs Understand Visual Anomalies? Uncovering LLM's Capabilities in Zero-shot Anomaly Detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, Melbourne, VIC, Australia, 28 October–1 November 2024; pp. 48–57.
21. Zou, Y.; Jeong, J.; Pemula, L.; Zhang, D.; Dabeer, O. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *Proceedings of the European Conference on Computer Vision*, Tel Aviv, Israel, 23–27 October 2022; Springer Nature: Cham, Switzerland, 2022; pp. 392–408.
22. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2024**, arXiv:1711.05101.
23. Chen, X.; Zhang, J.; Tian, G.; He, H.; Zhang, W.; Wang, Y.; Wang, C.; Liu, Y. Clip-ad: A language-guided staged dual-path model for zero-shot anomaly detection. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Jeju, Republic of Korea, 2–8 August 2024; Springer Nature: Singapore, 2024; pp. 17–33.
24. Cao, Y.; Zhang, J.; Frittoli, L.; Cheng, Y.; Shen, W.; Boracchi, G. Adaclip: Adapting clip with hybrid learnable prompts for zero-shot anomaly detection. In *Proceedings of the European Conference on Computer Vision*, Milan, Italy, 29 September–4 October 2024; Springer Nature: Cham, Switzerland, 2024; pp. 55–72.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.