Logical Anomaly Detection with Text-based Logic via Component-Aware Contrastive Language-Image Training

Seung-eon Lee*†
tmddjs7531@gmail.com
DGIST
School of Undergraduate
Daegu, Republic of Korea
ENERZAi
Seoul, Republic of Korea

Soopil Kim†
soopilkim@dgist.ac.kr
DGIST
Robots and Mechatronics Engineering
Daegu, Republic of Korea
DGIST
Division of Intelligent Robot
Daegu, Republic of Korea

Sion An
sion_an@dgist.ac.kr
DGIST
Robots and Mechatronics Engineering
Daegu, Republic of Korea

Sang-Chul Lee‡
sangchul.lee@dgist.ac.kr
DGIST
Division of Nanotechnology
Daegu, Republic of Korea

Sang Hyun Park‡
shpark13135@dgist.ac.kr
DGIST
Robots and Mechatronics Engineering
Daegu, Republic of Korea
DGIST
The Interdisciplinary Studies of
Artificial Intelligence
Daegu, Republic of Korea

Abstract

AI-based automatic visual inspection systems have been extensively researched to streamline various industrial products' labor-intensive anomaly detection processes. Despite significant advancements, detecting logical anomalies remains challenging due to the multitude of rules governing the assembly of multiple components to create a normal product. Existing methods have relied solely on image information for anomaly detection, resulting in limited accuracy as they fail to account for these diverse complex rules. Instead, humans detect anomalies by comparing the image with pre-defined logic which can be clearly expressed with natural language. Inspired by the human decision process, we propose a logical anomaly detection model that leverages text-based logic like human reasoning. With user-defined rules (*i.e.,* positive rules) and logically distinct negative rules, we train the model using component-aware contrastive learning that increases the similarity between images and positive rules while decreasing the similarity with negative rules. However, accurately comparing textual and visual features is challenging due to multiple components, each governed by different rules, within a single image. To address this, we developed a zero-shot related region detection technique, which guides the model's focus on components relevant to each rule. We evaluated the proposed model on three public datasets and achieved state-of-the-art results in a few-shot logical anomaly detection task. Our findings highlight the potential of integrating vision-language models to enhance logical anomaly detection and utilizing text-based logic in complex industrial settings.

CCS Concepts

• **Computing methodologies** → **Visual inspection**.

Keywords

Anomaly Detection, Logical Anomaly, Few Shot Learning, Vision Language Model, Contrastive Learning

ACM Reference Format:
Seung-eon Lee, Soopil Kim, Sion An, Sang-Chul Lee, and Sang Hyun Park. 2025. Logical Anomaly Detection with Text-based Logic via Component-Aware Contrastive Language-Image Training. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737032

1 Introduction

Deep learning-based defect detection models have been actively researched to address labor-intensive anomaly detection of products across various industries [30]. The mainstream of this field is unsupervised anomaly detection [6] that learns the distribution of normal industrial images and identifies outliers as anomalies by comparing test images to the learned distribution. Recent methods, such as PatchCore [39] and student-teacher [2] models, have

*The author completed this work as an undergraduate at DGIST and is currently affiliated with ENERZAi Inc.
†co-first authors
‡co-corresponding authors

KDD '25, August 3–7, 2025, Toronto, ON, Canada.

ACM ISBN 979-8-4007-1454-2/25/08
https://doi.org/10.1145/3711896.3737032

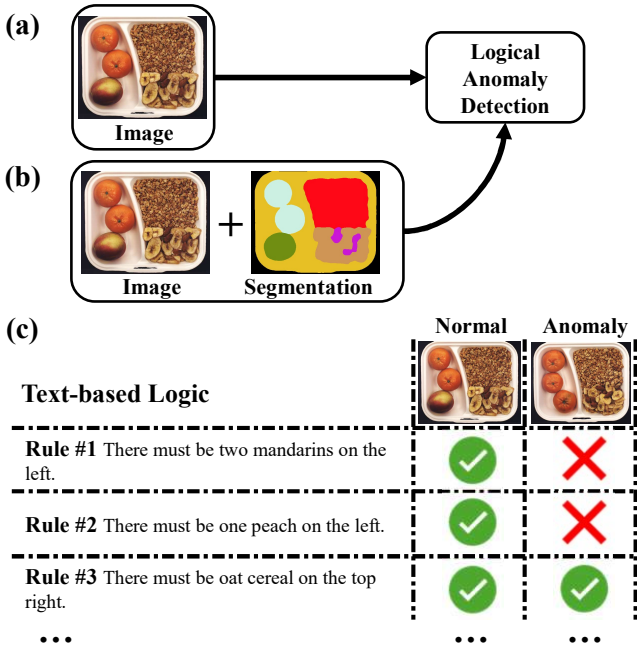

Figure 1: Conceptual comparison of our proposed methods and existing approaches. (a) The anomaly detection (AD) model is trained directly on images [3, 31, 45]. (b) The model uses pixel-level annotation to train a segmentation model, which is then used for AD [22]. (c) Our proposed method guides logical anomaly detection based on text-based logic.

achieved over 99% AUROC performance on datasets like MVTec AD [4] and MTD [18].

However, Bergmann *et al.* [3] pointed out that these datasets predominantly consist of a certain type of anomalies (*structural*) and existing models are inadequate for detecting the other type of anomalies (*logical*). Structural anomalies are localized defects (*e.g.,* impurities, cracks, or bending), whereas logical anomalies refer to the data that deviate from user-defined rules regarding the number, arrangement, or combination of specific components, requiring consideration of long-range dependencies. In comparison to structural anomalies, relatively fewer methods have been developed for logical AD. Most of them extract image features using a pre-trained encoder and use them for AD as shown in Figure 1(a) [3, 31, 45]. However, their accuracy is limited as they are unable to differentiate between various components. A recent study [22] integrates few-shot segmentation with AD to differentiate components and examine their composition, as shown in Figure 1(b). Despite the efforts in prior research, image-only approaches have struggled to leverage the detailed information encapsulated in pre-defined rules effectively. On the other hand, humans perform logical AD by verifying each given rule one by one when presented with an image, as illustrated in Figure 1(c). Utilizing text-based rules in anomaly detection (AD) could be a promising alternative, as users can conveniently and clearly express rules through natural language.

With recent advancement of vision-language models (VLMs) [11, 38, 59, 60] that learn the correlation between images and language, there is growing interest in text-based few-shot AD [12, 20]. However, existing VLMs often exhibit limited accuracy in industrial AD, as shown in Table 6, since industrial data is typically proprietary and not publicly accessible, whereas VLMs are primarily trained on open web datasets. This necessitates effective strategies for rapidly adapting VLMs to industrial tasks. Current methods [5, 20, 28] are primarily designed to detect structural anomalies since they rely on simple text descriptions such as "anomaly" or "damaged", and compare their textual features with local visual features. However, these models often fail to detect logical anomalies as they are unable to consider the relationship between various components. To this end, we propose a VLM-based logical AD method for industrial images that mimics the human anomaly detection process, utilizing language-expressed logic with a few images.

Our model verifies each user-defined rule (*i.e.,* positive rule) by computing the similarity between the image feature and positive rules. Specifically, after a large language model generates logically different rules (*i.e.,* negative rules) from each positive rule, the model is trained to increase the similarity between the features of images and positive rules while decreasing the similarity with negative rules. However, considering distinct rules for the different components within an image is challenging as reported in recent literature [33, 34, 41]. To address this, we perform zero-shot object detection inspired by insights from a recent work [22] that improves anomaly detection performance by discriminating various components. With the detection outcomes, we enhance visual features to help the model focus on the components governed by each rule.

We evaluate our proposed method on three public datasets and achieve state-of-the-art few-shot logical anomaly detection performance. Our contributions are summarized as follows:

- We introduce a novel problem of logical anomaly detection with text-based logic for the first time, to the best of our knowledge. Unlike previous logical anomaly detection methods that relied solely on images, our approach detects logical anomalies by utilizing user-defined rules expressed in natural language.
- We develop a text-based zero-shot related region detection technique specifically tailored for industrial images. We empirically demonstrate that existing models underperform in the industrial domain, whereas our method accurately identifies relevant objects.
- We propose a component-aware contrastive image-language training approach for few-shot logical anomaly detection. Leveraging the related region detection results, our model focuses on components pertinent to the specified logic. Additionally, we incorporate a large language model to automatically generate negative rules from positive ones.
- Our model achieves state-of-the-art performance in few-shot logical anomaly detection with large margins.

2 Related Works

2.1 Industrial Logical Anomaly Detection

Industrial anomaly detection (AD) models are designed to identify unexpected conditions in products at manufacturing sites. Deep

learning-based AD methods have recently achieved remarkable performance due to their ability to learn diverse patterns from normal data. Broadly, they are categorized into three branches: (i) reconstruction [26, 32], (ii) self-supervision [27, 57], and (iii) density estimation-based methods [19, 21, 39]. However, most of them are primarily designed to detect structural anomalies and they fail to detect logical anomalies.

To detect logical anomalies, the model needs to consider a broader picture, such as component composition and arrangement. While detecting structural anomalies has been extensively studied as mentioned above, only a few methods have been recently proposed for logical AD. For example, GCAD [3] proposed a logical AD dataset and hybrid reconstruction-based logical AD model using knowledge distillation loss. SINBAD [8] uses histogram projection to match the component composition. SLSG [52] proposed the self-supervision-based method using a graph convolutional network to consider the dense and sparse relationships among components.

To detect the logical anomalies effectively, they tried to use global features, including information on all components in the data, through model structure or anomaly score computation method. Moreover, some logical AD models leveraged the segmentation masks of the components to consider both local and global information. ComAD [31] segments multiple components using the KMeans clustering with a pre-trained model. PSAD [22] incorporates the few-shot segmentation model to the density-estimation-based method. Similar to Industrial AD, they require large amounts of train data or pixel-level annotation of images, limiting their application on diverse manufacturing sites. In this paper, we propose a novel logical AD method that can detect anomalies based on a few images and text-based logic.

2.2 Few-Shot Anomaly Detection

While industrial AD models have achieved high accuracy [39], their applicability across diverse manufacturing sites is limited by the required training data. Several few-shot AD models using a small amount of normal data have been proposed [9, 16, 20, 51]. For example, RegAD [16] proposed a registration model-based AD model to align test data to a few training data points and computed the Mahalanobis distance for the anomaly score. GraphCore [51] extracts and stores rotation-invariant features in a memory bank, predicting the anomaly score by comparing test data features with the memory bank. FastRecon [9] trains a transformation function to reconstruct test data features using a few training data points, using these reconstructions to predict the anomaly score.

2.3 Vision-Language Model

Vision-Language Models (VLMs) demonstrate promising applications across various fields [58]. Notably, CLIP [38] was the first to introduce contrastive language-image training for cross-modality representation matching. Their ability to learn from diverse and extensive datasets has allowed them to excel in numerous tasks. VLMs have been particularly influential in zero-/few-shot learning for computer vision [11, 59, 60]. This versatility has also provided new insights for their application in industrial AD, which typically relies only on normal image data. For example, WinCLIP [20], PromptAD [28], and APRIL-GAN [7] use CLIP for text-guided anomaly detection, utilizing the similarity between text and image features as part of the anomaly score. AnomalyGPT [12] incorporates an image encoder and a large language model (LLM) and follows the self-supervision-based AD method to compute the anomaly score. These models have outperformed traditional image-based few-shot AD models but have primarily focused on structural anomalies. In this paper, we propose a VLM for logical anomaly detection.

However, utilizing VLMs for industrial images, which often contain various components, is challenging because VLMs struggle to understand compositional relationships [33]. Although several methods [14, 56] have been proposed to enhance the compositional reasoning abilities of VLMs through hard negative sample generation, a significant domain gap still exists between their training data and industrial images. In this paper, we propose a component-aware contrastive learning approach to validate industrial images with complex logic involving various components. To the best of our knowledge, contrastive language-image training has not yet been explored in the context of industrial images.

2.4 Referring Image Segmentation (RIS)

Referring image segmentation (RIS) models aim to segment objects in an image as specified by a given text. These models are primarily trained in a fully supervised manner, combining multi-modality features [50, 53, 54]. Recent approaches use VLM or LLM for segmentation [24, 29, 48], demonstrating good performance but requiring dense mask annotations and detailed textual descriptions for training. To overcome these limitations, weakly-supervised methods using point annotations [25, 43] and zero-shot approaches trained without paired data [35, 44, 55] have been developed, enhancing applicability across various fields. Segmenting regions of interest using RIS can be beneficial for logical anomaly detection (AD) by precisely interpreting texts that describe logical constraints. However, we empirically found that existing methods often fail on industrial images, as they highly rely on mask generators like FreeSolo [47] and Segment Anything [23], which may not perform well in such contexts due to the domain shift. Instead, we propose a patch-text similarity-based related region detection method that does not rely on mask generators.

3 Method

3.1 Problem Setting & Overview

Our anomaly detection (AD) task aims to train a model capable of identifying abnormal data within a dataset using a few normal images $I_1, ..., I_{n_{tr}}$, where n_{tr} represents the number of data points, all labeled 0 (normal). In addition, several user-defined sentences $P_1, ..., P_{n_P}$ (*i.e.*, positive rules) expressing the logic used to ensure normality for each category are provided, where n_P is the number of positive rules. The model is trained to differentiate between normal and abnormal test data by comparing the image with positive rules, assigning labels of 0 for normal and 1 for abnormal.

Our proposed model detects logical anomalies by comparing image features with textual features of user-defined logic in a systematic manner. When we have multiple rules for a single product, one can use the rules as connected sentences, but focusing on individual rules and their relevant components within images is

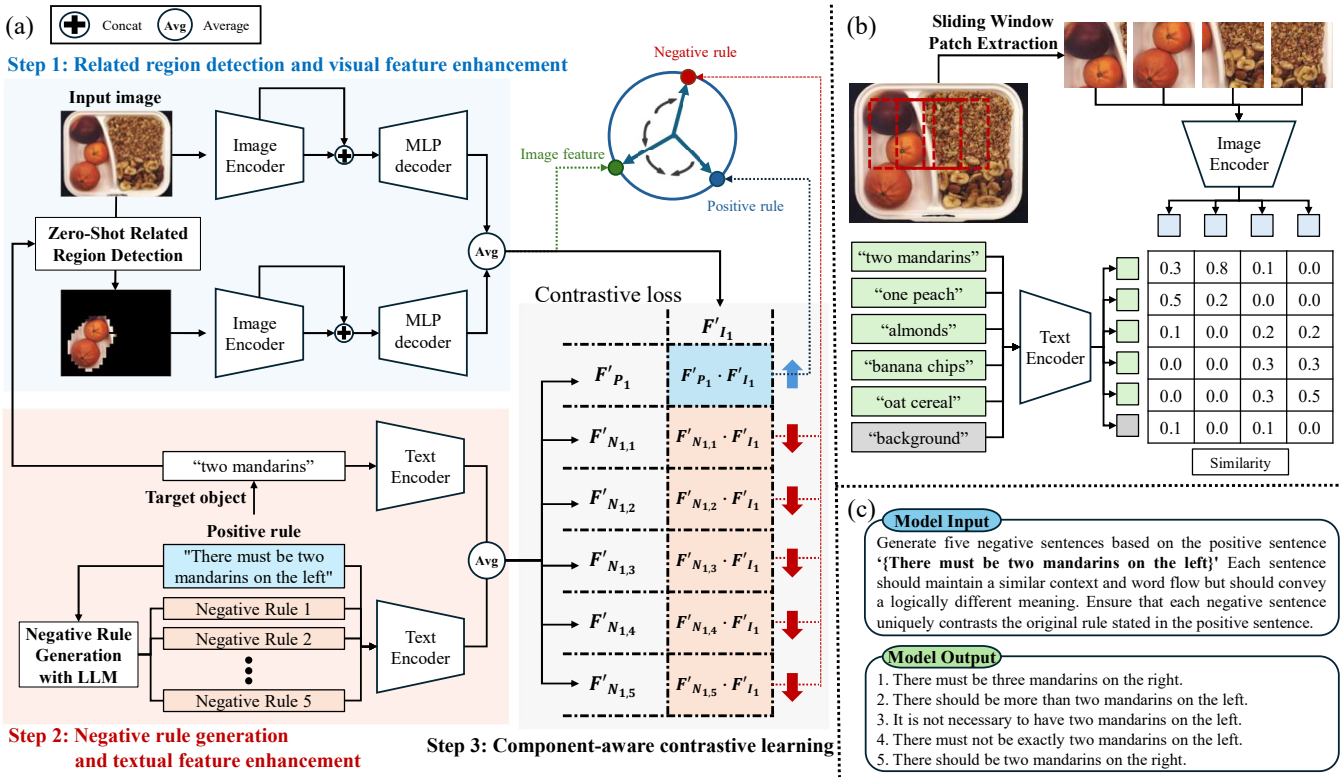

Figure 2: (a) Overview of the proposed method. (Step1): the model extracts visual features from an input image. Applying zero-shot related region detection lets the visual feature be target object-centric. (Step2): the model extracts textual features from positive and negative rules. When given a positive rule, the model automatically extracts the target object and generates corresponding negative rules using LLM. (Step3) learn from component-aware contrastive loss between visual and textual features. The details of 'Zero-Shot Related Region Detection' and 'Negative Rule Generation with LLM' are described in (b) and (c). (b) The process of zero-shot related region detection. Our model detects related regions based on the similarity between patches extracted by a sliding window and target object nouns. (c) Examples of generated negative rules. When a positive rule is given, negative rules are automatically generated using a large language model.

challenging. Therefore, we encode positive rules individually and compare them with image features.

This method comprises three steps, as illustrated in Figure 2(a): (i) visual feature enhancement with zero-shot related region detection, (ii) negative rule generation and textual feature enhancement, and (iii) component-aware contrastive language-image training.

In Step 1, we obtain image features focused on the target object using zero-shot-related region detection. Since the image includes various objects, we can enhance the visual features by utilizing only the parts corresponding to the target object of the positive rule. The image feature activated by the target object is added to the original image feature. In Step 2, we automatically generate negative rules using a large language model and enhance the textual features using the target noun phrase of the positive rule. Adding features related to the target noun phrase to the textual features ensures a better match with the target object image detected in Step 1. In Step 3, we perform component-aware contrastive language-image training using the discriminative image and text features obtained in Steps 1 and 2. In this process, our model increases the similarity between the image and the positive rule while decreasing the similarity with the negative rules. At inference time, the similarities between image features and each positive rule are used to predict the anomaly score. When a logical anomaly is determined, the anomaly scores are compared to identify which rule is the cause of the anomaly.

3.2 Zero-Shot Related Region Detection

When a positive rule P_i pertains to multiple objects $O_i^1, ..., O_i^{k_i}$, where k_i is the number of objects related to P_i. The model must focus on the key features of these objects within the image I. Although several methods such as LISA [24] and Global-Local CLIP [55] have been proposed to segment relevant regions in an image based on a given text, their performance is limited in industrial images due to domain shift. To address this, we introduce a method that effectively operates on industrial images using a sliding patch representation without requiring additional training of the VLMs, as shown in Figure 2(b). Algorithm 1 provides the detailed process.

Algorithm 1 Zero-Shot Related Region Detection

Require: image_encoder, text_encoder, object_names, raw_image, τ (threshold), k (kernel_size), s (stride)

```
text_features ← text_encoder(object_names)
width W, height H ← raw_image.size
patch_result ← [ [] for _ in range(len(object_names)) ]
related_region_images ← []
for w ∈ [0, W − k, s] do
    for h ∈ [0, H − k, stride] do
        image_patch = raw_image.crop((w, h, w + k, h + k))
        image_patch_feature = image_encoder(image_patch)
        prob = image_patch_feature · text_features.T
        patch_result[arg max(prob)].append((w, h))
    end for
end for
for patch in patch_result do
    mask ← zeros_like(raw_image)
    for (w, h) in patch do
        mask[w : w + k, h : h + k] += 1
    end for
    mask[mask > τ] ← True
    related_region ← raw_image[mask]
    related_region_images.append(related_region)
end for
return related_region_images
```

The VLMs like CLIP [38] are trained with contrastive learning using global features. It means that inputting the entire image into the image encoder can result in features of diverse objects appearing in the image, leading to ineffective feature extraction for the object of interest. To address this limitation, we use a sliding window approach to extract patches, which serve as inputs to the VLM. This method enables focused feature extraction on the limited objects in each patch, ensuring accurate similarity calculations.

Given an image I_j, patches are extracted using a sliding window and encoded by the VLM's image encoder. Simultaneously, the names of all objects in I_j and "background" texts are encoded by the VLM's text encoder. The similarity between the encoded patches and text features is calculated, and each patch is assigned to the most similar text. After processing all patches, each object O and the "background" will have a defined area of patches. If the number of overlapping patches exceeds a threshold τ, the region corresponding to O is considered a related region. When an image I has multiple objects related to a rule (*i.e.* $k_i > 1$), we simultaneously detect multiple regions for multiple objects. The regions for different objects are aggregated and returned as a masked image M_j^i

3.3 Automatic Negative Rule Generation

Since we do not have access to anomalous features during the training stage, we only know the positive rules that must be followed. Therefore, given the positive rule P_i that should align with an input image I, we utilize LLM to automatically generate negative rules $N_{i,1}, ..., N_{i,n_N}$ that should not match the image, where n_N is the number of negative rules. Figure 2(c) illustrates the example of automatic negative rule generation. We used the prompt shown in the figure 2 (c) model input. For instance, if the positive rule P_i is "There must be two mandarins on the left side," the LLM might generate negative rules like "There must be *three* mandarins on the *right*." These rules, P_i and $N_{i,1}, ..., N_{i,n_N}$, are subsequently used in contrastive learning.

Although incorrect rules may be generated due to the hallucination issue of LLMs [17], we could obtain correct rules with our automatic negative rule generation. Incorporating reasoning-based methods [10, 49] may also improve the quality of negative rules.

3.4 Visual And Textual Feature Enhancement

Inspired by recent work [55], we enhance both visual and textual features by utilizing masked images and the noun phrase of the target object. It enables the model to focus better on the target object. Specifically, our model extracts visual features F_{I_j} from the input image I_j and $F_{M_j^i}$ from the masked image M_j^i. The extracted visual features F_{I_j} and $F_{M_j^i}$ are concatenated with the low-level features of the image encoder. And then they passed through the MLP decoder (D). Before calculating the contrastive loss, we average the two features that have passed through the MLP decoder. Formally, we compute $F'_{I_{j,i}} = \beta D(F_{I_j}) + (1-\beta)D(F_{M_j^i})$, where β is set to 0.5.

Similarly to our approach for visual features, we employ a textual feature enhancement process to focus on the target object. Given a positive (or negative) rule P_i, we extract the textual features $F_{NP(P_i)}$ from the target noun phrase $NP(P_i)$ [13] and combine them with F_{P_i}, the features of positive (or negative) rules. Formally, we compute $F'_{P_i} = \beta F_{P_i} + (1-\beta)F_{NP(P_i)}$. The textual features of negative rules $F'_{N_{i,1}}, ..., F'_{N_{i,n_N}}$ are computed in the same way.

3.5 Contrastive Language-Image Training

We perform contrastive language-image training to ensure that the model's visual features of the image become closer to the textual features of the positive rules and farther from the features of the negative rules. In traditional contrastive learning [38], there are positive and negative text pairs for images and positive and negative image pairs for texts, with separate losses for each. Our problem, however, is that there are no negative images. Therefore, our model is trained using cross-entropy on pairs of positive and negative texts for each image. We first calculate the similarity between the visual and textual features using the dot product. Specifically, we compute the similarity between one visual feature $F'_{I_{j,i}}$ and $(1+n_N)$ textual features (i.e., $F'_{P_i}, F'_{N_{i,1}}, ..., F'_{N_{i,n_N}}$) and use these similarities to define the cross-entropy loss. Formally, when an input image I_j is given, our contrastive learning loss (L_{CL}) is defined as:

$$L_{CL} = -\frac{1}{n_P}\sum_{i=1}^{n_P}\log\left(\frac{\exp\left(F'_{I_{j,i}}\cdot F'_{P_i}\right)}{\exp\left(F'_{I_{j,i}}\cdot F'_{P_i}\right)+\sum_{l=1}^{n_N}\exp\left(F'_{I_{j,i}}\cdot F'_{N_{i,l}}\right)}\right) \tag{1}$$

Our model calculates the anomaly score using only the similarity between the positive rules and the input image. After applying softmax while calculating L_{CL}, we obtain scores for each positive rule. The anomaly score (S_{anomaly}) is then calculated by subtracting the mean of these scores for the positive rules from the maximum anomaly score of 1. Our anomaly score (S_{anomaly}) is defined as:

$$S_{\text{anomaly}} = 1 - \frac{1}{n_p} \sum_{i=1}^{n_P} \frac{\exp\left(F'_{I_{j,i}} \cdot F'_{P_i}\right)}{\exp\left(F'_{I_{j,i}} \cdot F'_{P_i}\right) + \sum_{l=1}^{n_N} \exp\left(F'_{I_{j,i}} \cdot F'_{N_{i,l}}\right)} \quad (2)$$

3.6 Multi-Product Type Anomaly Detection

Products such as a 'juice bottle' and 'splicing connectors' in the MVTec LOCO AD dataset may consist of multiple product types according to the kind of juice and type of connector. Since different rules may be required for each product type, we adopt a product type classifier before performing anomaly detection. We utilize a pre-trained vision-language model to perform zero-shot classification. It follows the general flow of using VLM for zero-shot classification [38]. For various product types, we extract features $F_{T_1}, ..., F_{T_i}$ from the prompt "a photo of [product type]"[61] using the text encoder, where F_{T_i} is the text feature for i-th type. After encoding an input image with the image encoder, we compute cosine similarities between the visual features F_I and textual features. The image is then classified into the product type with the highest similarity score and passed through the same three-step anomaly detection process described in Section 3.1.

4 Experiments

4.1 Experimental Setting

4.1.1 ***Dataset.*** We tested our method on the MVTec LOCO AD dataset [3], which is, to our knowledge, the most challenging benchmark for detecting logical anomalies. This dataset comprises five categories: breakfast box, juice bottle, pushpins, screw bag, and splicing connectors. This dataset contains hundreds of normal images for each category, but we evaluate our model's logical anomaly detection performance in a scenario with limited training data. We measure the accuracy of 1-shot and 5-shot logical anomaly detection performance (*i.e.* $n_{tr} = 1$ and $n_{tr} = 5$, respectively). We report the average performance across 30 random seeds.

Among the categories in the dataset, 'juice bottle' and 'splicing connectors' contain images with three different product types, while the others have a single product type. As introduced in Section 3.6, we first classify the type of the test image using zero-shot classification and then evaluate anomaly detection performance for the classified images. We confirmed that the classification accuracy on normal samples is 100%. For evaluation, we measure the anomaly detection performance using the samples classified into the same type, and the scores are averaged to obtain the final results.

To check whether the model generalizes well to other datasets, we also conducted further experiments on subsets of MVTec-AD [4] and VisA [62] that include data potentially considered as logical anomalies referring to [3].

4.1.2 ***Text-based Logic.*** Generally, users are familiar with the logic required to assemble components and can express it in text. We created positive rules for the MVTec LOCO AD dataset by referring to the description. Table 1 and 9 show the composition of the whole positive rules created for each category. For each positive rule, we could obtain five reasonable negative rules with ChatGPT [36]. (*i.e.* $n_N = 5$).

Table 1: Composition of positive rules

Breakfast box
1. There must be two mandarins on the left.
2. There must be a peach on the left.
3. There must be oat cereal on the top right.
4. There must be almonds down the right.
5. There must be banana chips down the right.
6. Each content should not overflow.
7. The amount of banana chips and almonds should be the same.
Juice bottle
1. Juice bottle must have {product_fruit} picture label.
2. Juice bottle must have {product_color} colored juice.
3. Picture label must be at the center.
4. Text label must be at the bottom.
5. The bottle should be full of juice.
Pushpins
1. There must be fifteen pushpins.
2. Each pushpin must be separated by plastic case.
3. There must be only one pushpin in each part.
4. There must be no blank spaces.
Screw bag
1. There must be two bolts.
2. There must be a long bolt.
3. There must be a short bolt
4. There must be two hexagonal nuts.
5. There must be two round washers.
Splicing connectors
1. There must be two splicing connectors.
2. The heights of the two connectors should be the same.
3. Only one cable must be connected.
4. The color of the cable must be {product_color}.
5. Each connector must have three blocks.
6. The cable must be connected to the same level of block.

4.1.3 ***Evaluation Metric.*** We used the AUROC (Area Under the Receiver Operating Characteristic) score as main for evaluation, which is widely used for anomaly detection performance measurement. The AUROC represents the degree to which a model can distinguish between normal and anomalous data. Additionally, to enable comparisons from more diverse perspectives, we also use AUPR (Area Under the Precision-Recall curve) score and F_1-max (F_1-score at optimal threshold).

4.1.4 ***Comparison Methods.*** To validate our method, we compared its performance with three types of baselines: (1) the state-of-the-art (SOTA) logical anomaly detection model (PSAD[22]), (2) models from [22] that rank among the top 5 in logical anomaly detection performance and have official code available (AST[40], SINBAD[45], ComAD[31]), and (3) a structural anomaly detection model that uses text-based logic (AnomalyGPT[12]) and pre-trained CLIP (PromptAD[28]). PromptAD achieves the highest performance among various CLIP-based methods. We implemented these models for 1-shot and 5-shot logical anomaly detection tasks using their official code releases. Since PSAD's segmentation model operates in a semi-supervised manner with additional segmentation labels

Table 2: Few-shot logical anomaly detection AUROC, AUPR and F1-max score of our proposed method against state-of-the-art anomaly detection methods and their standard deviation on MVTec LOCO AD dataset. Bold indicates the best performance.

	AUROC		AUPR		$F_1 - max$	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
AST	60.6 ± 3.8	68.0 ± 2.9	63.7 ± 3.3	70.0 ± 2.8	67.6 ± 1.2	69.5 ± 1.2
AnomalyGPT	58.5 ± 3.6	62.2 ± 4.1	60.1 ± 3.7	62.8 ± 4.2	66.6 ± 0.9	67.0 ± 1.1
PromptAD	61.3 ± 4.2	66.4 ± 4.2	62.2 ± 4.2	66.0 ± 4.4	67.9 ± 1.3	68.9 ± 1.5
SINBAD	50.0 ± 0.0	63.7 ± 3.9	**74.5 ± 0.1**	65.4 ± 3.9	65.5 ± 0.1	69.1 ± 0.9
PSAD	62.5 ± 4.7	72.1 ± 3.6	64.9 ± 4.2	70.6 ± 3.4	70.4 ± 1.7	**77.8 ± 2.3**
ComAD	64.1 ± 7.5	73.4 ± 6.0	67.1 ± 7.8	**75.2 ± 6.4**	68.4 ± 2.2	72.3 ± 3.4
Ours	**72.7 ± 4.4**	**77.0 ± 2.6**	71.5 ± 4.8	**75.1 ± 3.8**	**76.6 ± 1.8**	**78.8 ± 1.7**

Table 3: Logical Anomaly Detection AUROC performance compared to Zero-shot Related Region Detection method. 'Base' denotes a baseline model without any feature enhancement. CLIP indicates the Global-Local CLIP.

	1-shot	5-shot
Base	69.0 (+0.0%)	73.1 (+0.0%)
Base+LISA	67.2 (-2.6%)	70.5 (-3.5%)
Base+CLIP	66.4 (-3.8%)	70.6 (-3.5%)
Base+Ours	**72.3 (+4.8%)**	**75.8 (+3.7%)**

Table 4: Few-shot logical anomaly detection AUROC, AUPR and F1-max score of our proposed method and their standard deviation against Top-3 baselines on the subset of MVTec AD and VisA. Bold indicates the best performance.

		MVTec-AD			VisA		
		AUROC	AUPC	$F_1 - max$	AUROC	AUPC	$F_1 - max$
1-shot	PSAD	70.4 ± 8.3	70.4 ± 6.5	68.7 ± 5.1	63.8 ± 14.4	70.8 ± 10.3	73.0 ± 4.6
	ComAD	62.6 ± 13.1	58.5 ± 11.8	62.7 ± 6.2	62.2 ± 12.5	64.0 ± 11.0	70.8 ± 4.5
	Ours	**78.2 ± 6.2**	**82.2 ± 6.0**	**82.8 ± 2.0**	**74.9 ± 7.2**	**74.5 ± 8.1**	**75.6 ± 4.0**
5-shot	PSAD	82.5 ± 4.5	74.9 ± 4.5	80.1 ± 4.5	67.0 ± 8.0	66.2 ± 6.8	72.7 ± 3.7
	ComAD	67.6 ± 11.3	61.5 ± 11.5	65.1 ± 6.5	64.5 ± 8.8	65.6 ± 8.5	70.3 ± 3.6
	Ours	**83.6 ± 4.6**	**86.7 ± 4.1**	**84.8 ± 2.8**	**79.0 ± 5.1**	**79.4 ± 5.8**	**76.9 ± 3.6**

Table 5: Effect of visual and textual feature enhancement. 'Base' denotes a baseline model without any feature enhancement. 'V' and 'T' indicate visual feature enhancement and textual feature enhancement, respectively.

	1-shot	5-shot
Base	69.0 (+0.0%)	73.1 (+0.0%)
Base+V	72.3 (+4.8%)	75.8 (+3.7%)
Base+T	69.5 (+0.7%)	73.7 (+0.8%)
Base+V+T	**72.7 (+5.4%)**	**77.0 (+5.3%)**

and numerous unlabeled images, we trained a segmentation model with 1 labeled image and used the segmentation results in PSAD.

4.1.5 ***Implementation Details.*** We initialized the image encoder and text encoder with the pre-trained CLIP ViT-B/32, trained with the LAION-2B English subset of LAION-5B [42]. To effectively leverage the representations from the pre-trained CLIP model and learn industrial features, we incorporated LORA [15] and trained a multi-layer perceptron (MLP) decoder. The MLP decoder comprises seven 3-layer MLP models. For training, we used an Adam optimizer with a learning rate of 1e-4 and a batch size of 6 for 50 epochs. For zero-shot related region detection, the general input image size is larger than 1600x800 and we used a kernel size of 250 and a stride of 50. The threshold was set as a value between 8 to 12. Table 8 in Appendix B compares the performance of our method according to the hyperparameters and shows its robustness.

4.2 Few-shot Logical Anomaly Detection

In Table 2 and 4 we show the average few-shot logical anomaly detection performance of state-of-the-art methods. AST, AnomalyGPT and PromptAD were proposed for structural anomaly detection. AnomalyGPT utilizes a language model and PromptAD utilizes CLIP for few-shot scenarios. We observed that these three models performed poorly, failing to exceed 70% AUROC, AUPR, and F1-max scores in both the 1-shot and 5-shot settings. Despite using our custom positive rules as text inputs in AnomalyGPT and PromptAD, their performance was low, indicating that the model fails to consider the various components within the image and struggles with logical anomaly detection using conventional methods.

On the other hand, SINBAD, PSAD, and ComAD were proposed for logical anomaly detection but have not been evaluated in few-shot settings until now. The performance of SINBAD was lower than those of AST and PromptAD, which were designed for structural anomaly detection. PSAD and ComAD also failed to achieve an average AUROC of 70% in the 1-shot setting. Although literature reports that PSAD achieves an average AUROC of 98% using the entire dataset, its performance in the few-shot setting was significantly lower. Meanwhile, our proposed model achieved the best score compared to other methods in both 1-shot and 5-shot settings, with an improvement of +8.9% in 1-shot and +3.6% AUROC scores in 5-shot settings over the next best method.

We further evaluated the top-performing methods (PSAD, ComAD, and our proposed method) on subsets of MVTec-AD and VisA. Our method consistently outperformed the others by a large margin on both datasets. Please refer to Appendix C for the details.

4.3 Zero-Shot Related Region Detection

As the MVTec LOCO AD dataset lacks segmentation ground truth, we indirectly compare the segmentation results of our proposed method with state-of-the-art zero-shot referring image segmentation methods. Figure 3 presents qualitative examples from the comparison methods, including LISA [24] and Global-Local CLIP [55], applied to the 'breakfast box', 'splicing connectors' and 'screw bag'. Although both LISA and Global-Local CLIP perform object detection based on text, similar to our approach, they fail in industrial image contexts primarily due to domain shift.

For instance, in the breakfast box case, normal images contain two mandarins, one peach, banana chips, almonds, and oat cereal,

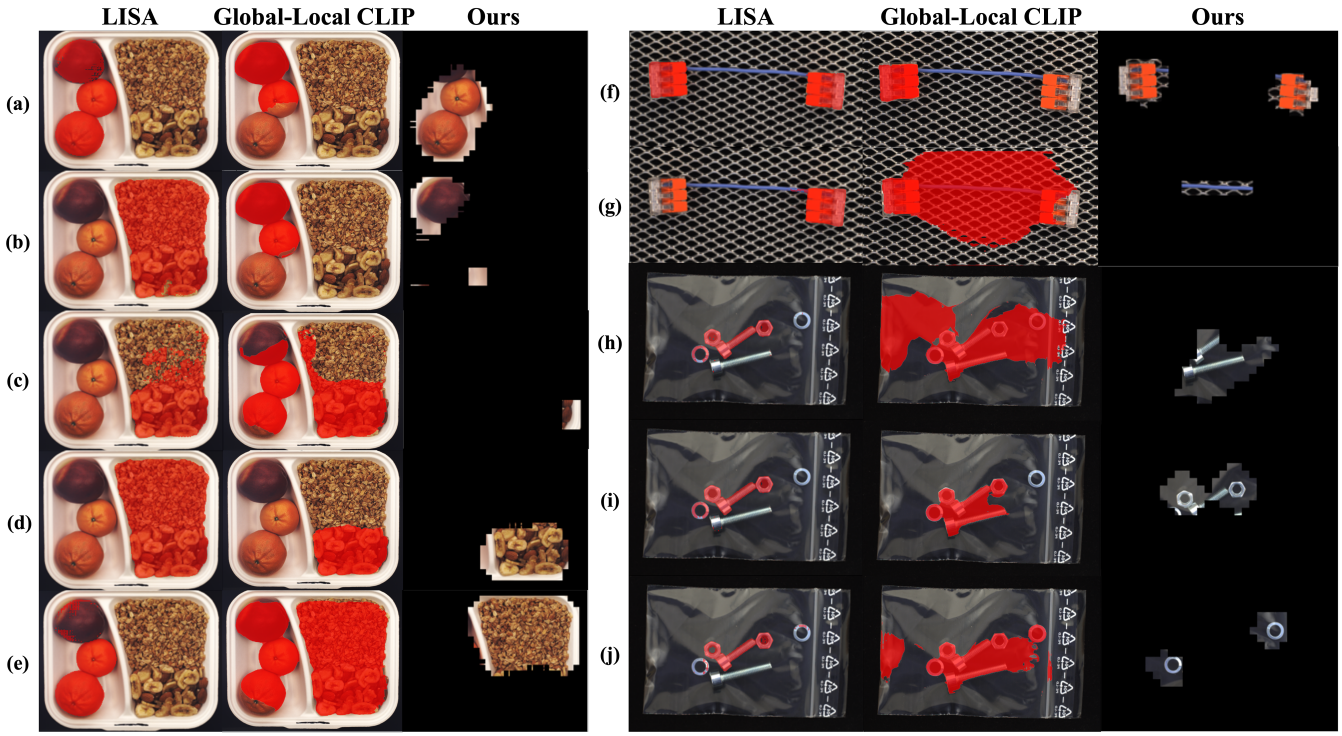

Figure 3: Zero-shot Related Region Detection Results of breakfast box and splicing connectors. Detection of LISA and Global-Local CLIP are marked with a red mask. The text information used for referring region detection follows (a) "two mandarins on the left", (b) "one peach on the left", (c) "almonds on the right", (d) "banana chips on the right", (e) "oat cereal on the right", (f) "two connectors on each side", (g) "blue cable", (h) "two bolts", (i) "two hexagonal nuts", (j) "two round washers".

which must be distinguished. However, both methods struggled to differentiate between mandarin and peach or banana chips and oat cereal. In contrast, our model accurately detected the target objects related to each rule. Even if it does not precisely distinguish between banana chips and almonds, it accurately identifies mandarins and peaches and correctly isolates oat cereal.The splicing connectors and screw bag cases showed similar patterns. Even if the comparison methods occasionally achieved decent localization, they often only identified parts of the target object or included unintended objects and background. In contrast, our method more accurately distinguished connectors from a cable and bolts from a plastic bag while providing more precise localization.

We also measured the anomaly detection performance of our proposed method using these segmentation maps as shown in Table 3. When LISA and Global-Local CLIP were applied, the average performance decreased by up to 3.8% compared to the baseline. In contrast, our method increased performance by an average of 4.3%. The average inference times for LISA and Global-Local CLIP were 0.195s and 1.635s, respectively, while our method took 0.965s.

4.4 Effect of Feature Enhancement

Table 5 shows the result of the experiment on visual and textual feature enhancement. We observed an approximately 4% improvement when using visual feature enhancement, demonstrating the effectiveness of utilizing a component-aware feature in logical anomaly detection. In the case of textual feature enhancement, even if there was little to no performance improvement when used alone, it achieved the best performance with approximately 5.4%. improvement when used in conjunction with visual feature enhancement. This indicates that our proposed feature enhancement technique helps focus on the corresponding rules of the target objects in the image, thereby improving logical anomaly detection performance.

Table 6: Logical anomaly detection Accuracy (%) of GPT-4o (OpenAI) and our proposed Method on MVTec LOCO AD.

	ChatGPT	Ours 1-shot	Ours 5-shot
Breakfast box	78.9	80.2	85.2
Juice bottle	68.2	80.9	83.7
Pushpins	40.6	63.3	64.4
Screw bag	61.0	51.7	55.8
Splicing connenctors	55.0	73.8	75.4
Average	60.7	70.0	72.9

4.5 Comparison with ChatGPT

We investigate the performance of ChatGPT (GPT-4o) as a state-of-the-art VLM with approximately 1.8 trillion parameters in Table 6. Despite its significantly larger model size and extensive training data of ChatGPT, It exhibited a notably low average accuracy of

60.7%, which is approximately 10% lower than the 1-shot performance and about 12% lower than the 5-shot performance of the proposed method. It may be attributed to the domain gap between web-based data used during training and industrial data, showing the need for adapting VLMs with few-shot industrial data. For more details, please see Appendix A.

4.6 Explainable Logical Anomaly Detection

While existing logical anomaly detection methods cannot explain why an anomaly is detected, our method provides clear explanations based on the anomaly score associated with each positive rule. Figure 4 illustrates anomalous examples along with their anomaly scores for different positive rules. For instance, in the case of the 'breakfast box,' the image contains only almonds, whereas normal samples include both banana chips and almonds. The highest anomaly score was associated with the rule stating, "The amount of banana chips and almonds should be the same." (Rule 7 in Table 1). Similarly, when assessing a partially empty 'juice bottle', the model assigned the highest anomaly score to the rule, "The bottle must be full of cherry juice" (Rule 5 in Table 1). These results demonstrate that our method effectively detects component-related anomalies and provides evidence based on the defined logic, thereby offering clear explanations for the detected anomalies.

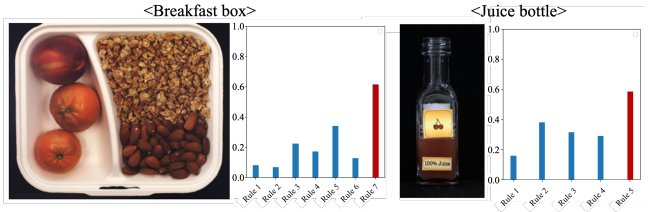

Figure 4: Examples of the anomaly scores (y-axis) for each positive rule (x-axis) predicted by our model. The breakfast box lacks banana chips, and the juice bottle is empty.

4.7 Ablation Study: Number of Positive Rule

In Table 7, we evaluated the performance of our model by varying the number of positive rules. In most cases, we observed a proportional relationship between n_P and performance, with an approximate 13. 4% increase in performance when n_P was at its maximum. This indicates that providing precise positive rules leads to better logical anomaly detection performance.

5 Discussion

While the proposed model significantly enhances performance compared to state-of-the-art anomaly detection methods, some areas need improvement. Firstly, the method relies on the capabilities of pre-trained Vision-Language Models (VLMs). Limited anomaly detection performance may be attributed to the existing VLMs' limitations in their counting capabilities, as reported in [37], and in comparing the lengths and sizes of different objects. For example, a normal 'pushpins' sample contains 15 pushpins, but CLIP [38] cannot accurately distinguish whether there are 14, 15, or 16 pushpins.

Table 7: Ablation study on the number of positive rules for training. This experiment is conducted at the 1-shot scenario with visual and textual feature enhancements

Data	n_P=1	n_P=3	n_P=max	max n_p
Breakfast box	65.8	76.6	84.8	7
Juice bottle	77.1	83.6	83.1	5
Pushpins	54.7	55.7	57.2	4
Screw bag	52.8	56.4	56.6	5
Splicing connectors	70.1	72.4	76.4	6
Average	64.1	70.1 (+9.3%)	72.7 (+13.4%)	5.4

We believe this is the fundamental reason for our model's lower performance on the 'pushpins' task. In the case of the 'screw bag', a normal sample contains two bolts of similar texture but different lengths, randomly rotated and positioned. Since CLIP cannot compare the lengths and sizes of these objects, anomaly detection performance may be degraded even if component-relevant regions are successfully detected. We expect that anomaly detection performance will improve as better VLMs are developed in the future.

Secondly, our method relies on text-based logic, which requires users to have prior knowledge of the data set. As shown in Table 7, performance varies with the composition of positive rules, highlighting the importance of defining them effectively. However, in industrial settings, these rules are typically predefined by manufacturers, which makes this requirement less burdensome and aligns with real-world industrial practices. For a fully automated visual system, caption generation methods [1, 46] could be used to generate positive rules from normal samples, streamlining the process and improving anomaly detection efficiency.

6 Conclusion

In this paper, we introduced a novel approach for logical anomaly detection in industrial images using text-based logic. Our model mimics human behavior by verifying user-defined rules expressed in natural language, enhancing the detection process. We developed a text-based zero-shot related region detection technique specifically for industrial images, which outperforms existing models. Our component-aware contrastive image-language training method, along with the automatic generation of negative rules via a large language model, significantly improves the detection of logical anomalies. Experimental results demonstrate that our model achieves state-of-the-art performance in few-shot logical anomaly detection. Further improvements are anticipated with advancements in vision-language models.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government (MSIT) (No.RS-2025-02219277), AI Star Fellowship Support Project (DGIST), Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea government (MOTIE) (No.RS-2025-02633871), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.RS-2024-00345398, No.2340012392).

References

[1] Shuang Bai and Shan An. 2018. A survey on automatic image caption generation. *Neurocomputing* 311 (2018), 291–304.

[2] Kilian Batzner, Lars Heckler, and Rebecca König. 2024. Efficientad: Accurate visual anomaly detection at millisecond-level latencies. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 128–138.

[3] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. 2022. Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. *International Journal of Computer Vision* 130, 4 (2022), 947–969.

[4] 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.

[5] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. 2020. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4183–4192.

[6] Yunkang Cao, Xiaohao Xu, Jiangning Zhang, Yuqi Cheng, Xiaonan Huang, Guansong Pang, and Weiming Shen. 2024. A survey on visual anomaly detection: Challenge, approach, and prospect. *arXiv preprint arXiv:2401.16402* (2024).

[7] Xuhai Chen, Yue Han, and Jiangning Zhang. 2023. A zero-/few-shot anomaly classification and segmentation method for cvpr 2023 vand workshop challenge tracks 1&2: 1st place on zero-shot ad and 4th place on few-shot ad. *arXiv preprint arXiv:2305.17382* (2023).

[8] Niv Cohen, Issar Tzachor, and Yedid Hoshen. 2023. Set features for fine-grained anomaly detection. *arXiv preprint arXiv:2302.12245* (2023).

[9] Zheng Fang, Xiaoyang Wang, Haocheng Li, Jiejie Liu, Qiugui Hu, and Jimin Xiao. 2023. Fastrecon: Few-shot industrial anomaly detection via fast feature reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 17481–17490.

[10] Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720* (2022).

[11] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. 2024. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision* 132, 2 (2024), 581–595.

[12] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. 2024. Anomalygpt: Detecting industrial anomalies using large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 1932–1940.

[13] Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 conference on empirical methods in natural language processing*. 1373–1378.

[14] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. 2024. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *Advances in Neural Information Processing Systems* 36 (2024).

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[16] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. 2022. Registration based few-shot anomaly detection. In *European Conference on Computer Vision*. Springer, 303–319.

[17] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* (2024).

[18] Yibin Huang et al. 2020. Surface defect saliency of magnetic tile. *The Visual Computer* 36 (2020), 85–96.

[19] Jeeho Hyun, Sangyun Kim, Giyoung Jeon, Seung Hwan Kim, Kyunghoon Bae, and Byung Jun Kang. 2023. ReConPatch: Contrastive Patch Representation Learning for Industrial Anomaly Detection. *arXiv preprint arXiv:2305.16713* (2023).

[20] 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.

[21] Xi Jiang, Jianlin Liu, Jinbao Wang, Qiang Nie, Kai Wu, Yong Liu, Chengjie Wang, and Feng Zheng. 2022. Softpatch: Unsupervised anomaly detection with noisy data. *Advances in Neural Information Processing Systems* 35 (2022), 15433–15445.

[22] Soopil Kim, Sion An, Philip Chikontwe, Myeongkyun Kang, Ehsan Adeli, Kilian M Pohl, and Sang Hyun Park. 2024. Few shot part segmentation reveals compositional logic for industrial anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8591–8599.

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643* (2023).

[24] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. 2023. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692* (2023).

[25] Jungbeom Lee, Sungjin Lee, Jinseok Nam, Seunghak Yu, Jaeyoung Do, and Tara Taghavi. 2023. Weakly supervised referring image segmentation with intra-chunk and inter-chunk consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 21870–21881.

[26] Sungwook Lee, Seunghyun Lee, and Byung Cheol Song. 2022. Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *IEEE Access* 10 (2022), 78446–78454.

[27] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. 2021. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9664–9674.

[28] Xiaofan Li, Zhizhong Zhang, Xin Tan, Chengwei Chen, Yanyun Qu, Yuan Xie, and Lizhuang Ma. 2024. Promptad: Learning prompts with only normal samples for few-shot anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16838–16848.

[29] Chang Liu, Henghui Ding, and Xudong Jiang. 2023. Gres: Generalized referring expression segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 23592–23601.

[30] Jiaqi Liu, Guoyang Xie, Jinbao Wang, Shangnian Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. 2024. Deep industrial image anomaly detection: A survey. *Machine Intelligence Research* 21, 1 (2024), 104–135.

[31] Tongkun Liu, Bing Li, Xiao Du, Bingke Jiang, Xiao Jin, Liuyi Jin, and Zhuo Zhao. 2023. Component-aware anomaly detection framework for adjustable and logical industrial visual inspection. *Advanced Engineering Informatics* 58 (2023), 102161.

[32] Wenrui Liu, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. 2023. Diversity-measurable anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12147–12156.

[33] Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. 2023. Crepe: Can vision-language foundation models reason compositionally?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10910–10921.

[34] Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. 2024. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14420–14431.

[35] Minheng Ni, Yabo Zhang, Kailai Feng, Xiaoming Li, Yiwen Guo, and Wangmeng Zuo. 2023. Ref-diff: Zero-shot referring image segmentation with generative models. *arXiv preprint arXiv:2308.16777* (2023).

[36] OpenAI. 2024. ChatGPT: OpenAI Language Model. https://chat.openai.com/. Version 4.0, Available at https://chat.openai.com/.

[37] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. 2023. Teaching clip to count to ten. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3170–3180.

[38] 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.

[39] 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.

[40] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. 2023. Asymmetric student-teacher networks for industrial anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2592–2602.

[41] Ugur Sahin, Hang Li, Qadeer Khan, Daniel Cremers, and Volker Tresp. 2024. Enhancing multimodal compositional reasoning of visual language models with generative negative mining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 5563–5573.

[42] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems* 35 (2022), 25278–25294.

[43] Robin Strudel, Ivan Laptev, and Cordelia Schmid. 2022. Weakly-supervised segmentation of referring expressions. *arXiv preprint arXiv:2205.04725* (2022).

[44] Yucheng Suo, Linchao Zhu, and Yi Yang. 2023. Text Augmented Spatial Aware Zero-shot Referring Image Segmentation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

[45] Niv Cohen Tzachor, Yedid Hoshen, et al. 2023. Set features for fine-grained anomaly detection. *arXiv preprint arXiv:2302.12245* (2023).

[46] Haoran Wang, Yue Zhang, and Xiaosheng Yu. 2020. An overview of image caption generation methods. *Computational intelligence and neuroscience* 2020 (2020).

[47] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. 2022. Freesolo: Learning to segment objects without annotations. In *Proceedings of the IEEE/CVF conference on computer vision*

and pattern recognition. 14176–14186.
[48] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. 2022. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11686–11695.
[49] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.
[50] Jianzong Wu, Xiangtai Li, Xia Li, Henghui Ding, Yunhai Tong, and Dacheng Tao. 2024. Towards robust referring image segmentation. *IEEE Transactions on Image Processing* (2024).
[51] Guoyang Xie, Jinbao Wang, Jiaqi Liu, Yaochu Jin, and Feng Zheng. 2022. Pushing the Limits of Fewshot Anomaly Detection in Industry Vision: Graphcore. In *The Eleventh International Conference on Learning Representations*.
[52] Minghui Yang, Jing Liu, Zhiwei Yang, and Zhaoyang Wu. 2023. SLSG: Industrial Image Anomaly Detection by Learning Better Feature Embeddings and One-Class Classification. *arXiv preprint arXiv:2305.00398* (2023).
[53] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. 2022. Lavt: Language-aware vision transformer for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18155–18165.
[54] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. 2019. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10502–10511.
[55] Seonghoon Yu, Paul Hongsuck Seo, and Jeany Son. 2023. Zero-shot referring image segmentation with global-local context features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19456–19465.
[56] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2022. When and why vision-language models behave like bags-of-words, and what to do about it?. In *The Eleventh International Conference on Learning Representations*.
[57] Vitjan Zavrtanik et al. 2021. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8330–8339.
[58] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).
[59] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. 2023. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15211–15222.
[60] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2022. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*. Springer, 493–510.
[61] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to prompt for vision-language models. *International Journal of Computer Vision* 130, 9 (2022), 2337–2348.
[62] 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.

A Motivation: ChatGPT fails to detect logical anomaly in industrial data

This section outlines the motivation of our research. We hypothesized that leveraging large VLM models, such as GPT, could potentially address this issue through text-based reasoning. However, when we posed questions using our proposed positive rules to the latest version of ChatGPT, the model failed to produce correct results in many cases. To further investigate, we used OpenAI's GPT-4o API to evaluate its accuracy on the MVTec Loco AD dataset.

As it was not possible to compute the anomaly scores for GPT-4o, accuracy was used as the evaluation metric. Algorithm 2 presents the pseudocode for performing logical anomaly detection using GPT-4o. GPT-4o achieved an average accuracy of 60.76%, which is approximately 10% lower than the 1-shot performance and about 12% lower than the 5-shot performance of the proposed method. Although the exact size of GPT-4o's model parameters has not been publicly disclosed, estimates suggest it ranges from at least 200 billion to over 1 trillion parameters. Its performance remains subpar despite employing such a significantly large model with extensive computational resources.

We attribute this limitation to the domain gap arising from the composition of web-based data used in GPT's training, where industrial images are either underrepresented or entirely absent. The images are often proprietary assets of manufacturing companies and are typically not open. As a result, large models trained solely on web-based data encounter challenges when addressing such domain-specific tasks. This result shows that adapting VLMs for industrial data is crucial to bridge the domain gap and improve performance.

Algorithm 2 Pseduo code: Testing GPT-4o

Require: GPT-4o, Positive rule_list, query_image

```
for each positive_rule in Positive do
    prompt ← convert_to_question(positive_rule) + "Answer yes or no"
    response ← ask_gpt(query_image, prompt)
    if "no" ∈ response then
        return Anomaly
    end if
end for
return Normal
```

B Hyper parameter Robustness

Table 8 shows the few-shot logical anomaly detection AUROC scores with different combinations of key hyper-parameters (kernel size and threshold) for zero-shot related region detection. The results demonstrate that our model is not heavily dependent on these hyperparameters. We selected a kernel size close to the smallest object to limit the number of objects per patch. Consistent results indicate that accuracy is not highly sensitive to kernel size.

Table 8: Few-shot Logical Anomalyt Detection AUROC score on MVTec LOCO AD breakfast box dataset.

Kernel size	threshold	1-shot	5-shot
200	8	87.8 ± 2.1	89.6 ± 2.6
	10	86.5 ± 3.2	90.4 ± 2.1
	12	81.6 ± 5.6	88.3 ± 3.5
250	8	84.5 ± 4.7	89.1 ± 2.2
	10	84.6 ± 3.5	88.3 ± 2.7
	12	83.3 ± 4.2	87.1 ± 2.4
300	8	84.3 ± 1.9	87.9 ± 2.1
	10	80.6 ± 4.2	87.7 ± 1.5
	12	81.8 ± 3.3	86.4 ± 3.4
Standard deviation		2.3	1.2

C Evaluation on MVTec AD and VisA dataset

MVTec AD and VisA datasets are commonly utilized for structural AD. However, despite their limited number and closer alignment

Table 9: Comosition of positive rules for the subsets of MVTec-AD and VisA.

MVTec-AD: Cable
1. cable must not have bent wire.
2. cable must not have missing part.
3. cable must not have missing wire.
4. cable must not be cutted.
5. cable must not be poked.
MVTec-AD: Capsule
1. capsule must not be scratched.
2. capsule must not be discolored.
3. capsule must not be misshaped.
4. capsule must not have leak.
5. capsule must not have bubble.
MVTec-AD: Transistor
1. transistor must not have bent lead.
2. transistor must not have cut lead.
3. transistor must not be damage.
4. transistor must not have misplaced transistor.
VisA: pcb1, pcb2, pcb3, pcb4
1. PCB must not be bent.
2. PCB must not be scratched.
3. PCB must not be missing.
4. PCB must not be melted.

Table 10: MVTec AD defect type composition and selected logical anomaly defect type. The defect types in "()" are selected.

	Defect types
Cable	**(cable swap, combined, missing cable , missing wire)**, bent wire, poke insulation, cut inner insulation cut outer insulation
Capsule	**(faulty imprint)**, crack, poke, scratch, squeeze
Transistor	**(bent lead, misplaced)**, cut lead, damaged case

with structural anomalies, subsets of them can also be employed for evaluating logical AD.

Table 9 shows the composition of positive rules of MVTec AD and VisA. We referred to the prompts designed for structural anomaly detection in PromptAD to construct positive rules.

Table 10 presents the composition of the defect types for three data classes within the MVTec AD dataset, which include logical anomalies as a subset and a detailed enumeration of the defect types selected as logical anomalies. For the VisA dataset, even though the pcb1, pcb2, pcb3, and pcb4 classes may include logical anomalies, an explicit split by defect type is not provided. Consequently, anomaly detection performance was evaluated without additional splits.

Experimental results demonstrate that our proposed method significantly outperforms PSAD and ComAD across both VisA and MVTec AD datasets. In all cases of 1-shot and 5-shot, there is an average improvement of more than 12%.