LogicAD: Explainable Anomaly Detection via VLM-based Text Feature Extraction

Er Jin[1,*], Qihui Feng[2,*], Yongli Mou[2], Gerhard Lakemeyer[2], Stefan Decker[2,3], Oliver Simons[4,†], Johannes Stegmaier[1,†]

[1]Institute of Imaging and Computer Vision, RWTH Aachen University, Aachen, Germany
[2]Department of Computer Science, RWTH Aachen University, Aachen, Germany
[3] Fraunhofer Institute for Applied Information Technology FIT, Sankt Augustin, Germany
[4] Independent Researcher
{er.jin, johannes.stegmaier}@lfb.rwth-aachen.de, {feng, gerhard}@kbsg.rwth-aachen.de,
{mou, decker}@dbis.rwth-aachen.de, science.osimons@posteo.de

Abstract

Logical image understanding involves interpreting and reasoning about the relationships and consistency within an image's visual content. This capability is essential in applications such as industrial inspection, where logical anomaly detection (AD) is critical for maintaining high-quality standards and minimizing costly recalls. Previous research in AD has relied on prior knowledge for designing algorithms, which often requires extensive manual annotation effort, significant computing power, and large amounts of data for training. Autoregressive, multimodal Vision Language Models (AVLMs) offer a promising alternative due to their exceptional performance in visual reasoning across various domains. Despite this, their application in logical AD remains unexplored. In this work, we investigate using AVLMs for logical AD and demonstrate that they are well-suited to the task. Combining AVLMs with format embedding and a logic reasoner, we achieve state-of-the-art (SOTA) AD performance on public benchmarks, MVTec LOCO AD, with an AUROC of **86.0%** and an F_1-max of **83.7%** along with explanations of the anomalies. This significantly outperforms the existing SOTA method by **18.1%** in AUROC and **4.6%** in F_1-max score.

The dataset, code and supplementary materials — https://jasonjin34.github.io/logicad.github.io/

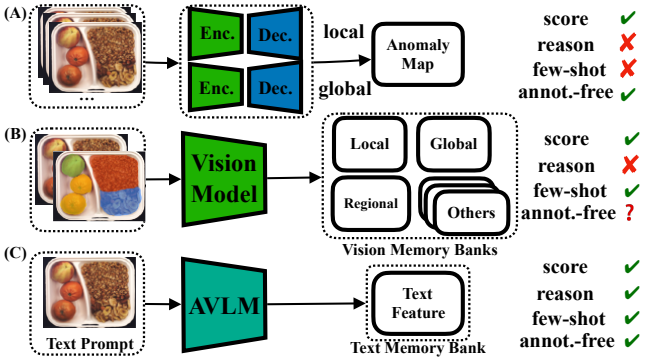


Figure 1: Overview of Anomaly Detection Approaches: (A) AD models trained from scratch require large-scale datasets and are capable of AD but lack reasoning capabilities. (B) Memory-based AD methods leverage a pre-trained vision model to extract features from normal images, enabling few-shot AD. However, they often require additional visual annotations and lack reasoning. (C) Our method uses pre-trained AVLMs as a text feature extractors and uses it for logical anomaly (LA) detection and reasoning with only text prompts, eliminating the need for visual annotations.

Introduction

Anomalies in industrial image data can be broadly classified into two distinct categories: structural anomalys (SAs) and LAs (Bergmann et al. 2022). The structures of SA observed in industrial images are often referred to as localized regional features, such as broken parts, color contamination, and minor deformations. These anomalies are typically observable only in the abnormal object. In contrast, logical anomalies, such as missing objects, misplacements, and incorrect object color combinations, are not confined to a specific area. Generally, detecting them requires a more comprehensive, abstract understanding of the normal and abnormal states. Moreover, some LA features even appear in both abnormal and normal objects (Bergmann et al. 2019; Zou et al. 2022). Current methods face challenges in effectively detecting LA. The full-shot method, which is trained on a full dataset in the AD domain, captures all the error-free localized features as a memory bank and uses them to detect anomalies, which can be very effective at SA detection while facing difficulties in detecting LAs (Roth et al. 2022; Kim et al. 2024). As shown in Figure 1(A), some approaches attempt to solve this issue by training multiple global and local autoencoder (AE) networks to capture both LA and SA related features (Bergmann et al. 2022). Others rely on additional manual visual annotations and multiple memory banks to achieve remarkable LA detection (Kim et al. 2024; Zhang et al. 2024). However, the inability to perform

*Corresponding authors with equal contributions.
†These authors contributed equally.

few-shot learning and the requirement for additional manual visual annotation is undesirable. Recently, many methods based on vision-language models, such as contrastive language-image pretraining (CLIP) (Radford et al. 2021), have shown remarkable performance in few-shot SA detection (Jeong et al. 2023; Chen, Han, and Zhang 2023). In LA detection, particularly under few-shot learning scenarios, the challenge of understanding long-range global features remains unresolved. Recently, many methods, as shown in Figure 1(B), aim to capture global features by using multiple memory banks using Mixture of Experts (MoE) (Gu et al. 2024b). However, compared with other full-shot methods, which are trained with the whole dataset, the performance is noticeably inferior (Kim et al. 2024). Figure 1 provides an overview of current LA detection methods (A) and (B), which are predominantly based on visual features. Recent advancements in AVLMs such as GPT-4o and LLaVA 1.6 (Achiam et al. 2023; Liu et al. 2024a) have demonstrated significant capabilities in image understanding and text generation. Despite these advancements, using text features for AD, particularly logical anomalies, remains underexplored. In this paper, we introduce our AD algorithm, LogicAD, which primarily utilizes text features extracted from AVLMs rather than relying solely on visual features and without relying on additional visual annotations. Our evaluations on multiple public datasets reveal that LogicAD surpasses the current SOTA method by a large margin. Our contributions are as follows: **(1)** We introduce LogicAD, a novel one-shot algorithm for LA detection that leverages text feature memory banks, employing AVLMs and large language models (LLMs) to achieve the SOTA performance in one-shot logical AD. **(2)** We design a text feature extraction pipeline that enables AVLMs to generate logical, robust, and reliable text features for detailed logical descriptions. **(3)** We introduce a *logic reasoner*, which leverages automated theorem prover (ATP) for LA detection and generates descriptive explanations for the identified anomaly without using manual or dynamic thresholding.

Related Work

LA in Industrial Images. Following the release of the logical anomaly dataset MVTec LOCO AD (Bergmann et al. 2022), numerous unsupervised methods have been proposed (Liu et al. 2023b; Rudolph et al. 2023). These methods can be categorized into vision memory bank-based methods and reconstruction networks. As illustrated in Figure 1(A), GCAD employs multiple high-capacity AE networks to capture the global context through its latent space. For SA, GCAD requires an additional local branch. Furthermore, each branch demands a significant amount of data to learn and capture the latent features, posing a limitation in few-shot scenarios (Bergmann et al. 2022). Inspired by PatchCore (Roth et al. 2022), vision memory-based methods have gained popularity in the few-shot domain due to their simplicity and effectiveness. Visual features are often extracted via pre-trained networks, commonly using CNN-based architectures such as ResNet (Rippel, Mertens, and Merhof 2021; Rippel et al. 2021; Liu et al. 2023b; Zhou et al. 2024; He et al. 2016). To understand logical-related long-range contexts, ComAD (Liu et al. 2023b) utilizes DINO (Caron et al. 2021) for segmenting images into region of interests (ROIs) and extracting region-based features as a memory bank. PSTD (Kim et al. 2024) proposes using a few fully-annotated segmentation masks to help models capture long-range contexts. However, PSTD still requires three vision-based memory banks. ViperGPT utilizes the AVLM to generate task-related code for handling downstream tasks such as AD (Surís, Menon, and Vondrick 2023). However, framing AD as a vision-centric task is challenging due to the high level of semantic understanding required. Additionally, the generated code can introduce bias and increase computational cost. AnomalyGPT proposed an approach that fine-tunes open source LLMs (Touvron et al. 2023) and a vision encoder for effective SA detection and even achieve anomaly localization without the need to use a manual threshold. Another similar method, VisionLLM, can be adapted for AD (Wang et al. 2024). However, these methods require synthesized anomalous datasets and demand substantial computational resources for fine-tuning, making the process resource-intensive and computationally demanding. In contrast, our few-shot method (one-shot) can achieve AD without any fine-tuning. It is also the first method to integrate AVLM capabilities with a purely logical reasoning framework, which is further supported by an automated theorem prover, significantly improving the explainability of anomaly detection results.

Autoregressive, multimodal Vision Language Models (AVLMs). Recently, many AVLMs, such as GPT-4o and LLaVA1.6 (Achiam et al. 2023; Liu et al. 2024a), have achieved remarkable results across multiple benchmarks, such as VQA-v2 (Goyal et al. 2017) and ScienceQA (Lu et al. 2022). These models perform exceptionally well in tasks involving naive logical scenarios, such as object localization and size comparisons. However, in real-world scenarios, researchers have noted that AVLMs often struggle with hallucinations, where the models fail to accurately ground both the provided text and visual context (Gunjal, Yin, and Bas 2024). For instance, the ability of AVLMs to understand quantitative and logical information decreases as the complexity of the data increases, as demonstrated in Figure 5a. Research suggests that using Chain-of-Thought (CoT) can partially mitigate the hallucination issue in AVLMs (Chen et al. 2024). However, the effectiveness of CoT in industrial AD has not yet been fully explored.

Logic Reasoning. The development of LLMs also boosts the investigation of neuro-symbolic approaches, which combine language models with formal methods by parsing natural language statements into formal languages such as first-order logic (Enderton 2001). Previous work such as LINC (Olausson et al. 2023), Logic-LM (Pan et al. 2023) and SatLM (Ye et al. 2024) utilize LLMs as a semantic translator and convert natural language problems into formal specifications. Then, ATP such as Prover9[1] performs inference to solve the queries. For tasks with complex logical relations, these approaches significantly outperform in-context reasoning methods such as CoT, inspiring us to combine our

[1]https://www.cs.unm.edu/ mccune/prover9/

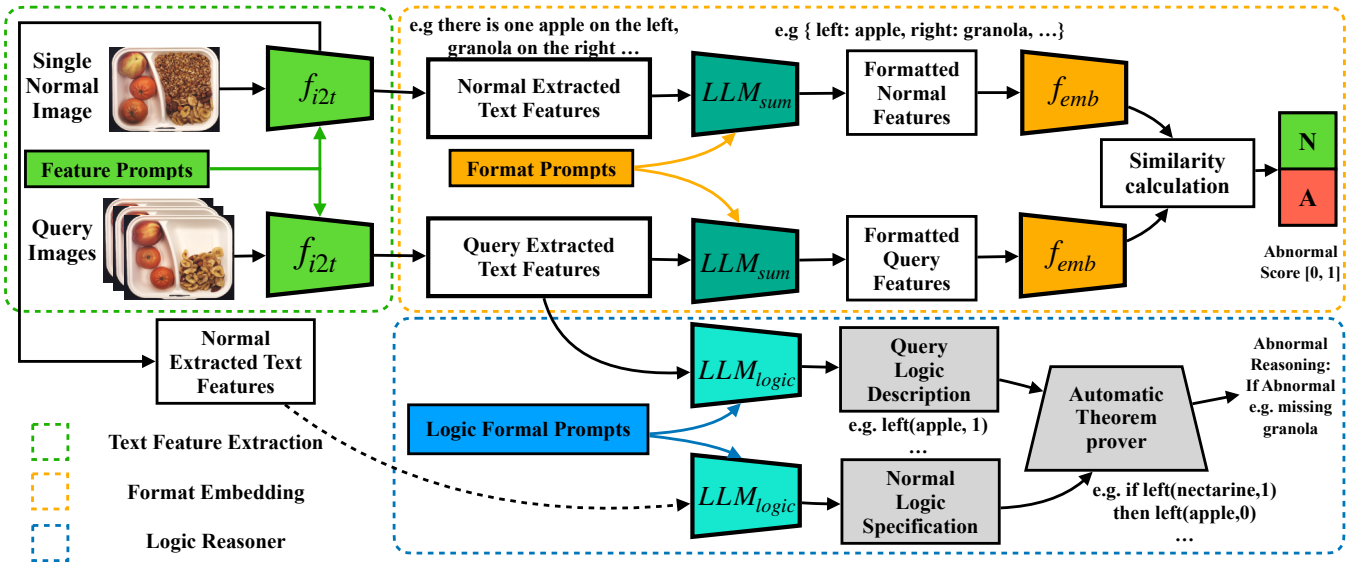


Figure 2: Pipeline overview of LogicAD. The green box represents *text feature extraction*, f_{i2t}, which extracts features via pre-trained AVLMs from the input image, the detailed process is depicted in Figure 3. These features are then processed by two separate modules: *format embedding* (orange box) and *logic reasoner* (blue box). The *format embedding* module computes an anomaly score based on the similarity between embeddings of formatted normal and query features. The *logic reasoner* module utilizes logical rules derived from normal data to classify inputs as normal or abnormal while providing reasoning.

system with formal methods to handle complicated relations among objects and to explain detection results.

LogicAD

Problem Definition: AD is a task that focuses on identifying abnormal features by learning from normal features denoted as F_{normal} extracted from a set of training images $\{X_1, X_2, ..., X_N\}$, where N denotes the total number of anomaly-free training images. Zero-shot (ZS) AD leverages pre-trained vision-language model (VLM) with provided text prompts for AD without any training images. Few-shot uses few images or even one training image (One-shot) for AD. Compared to the ZS approach, which relies solely on reference text descriptions, the few-shot method uses AVLMs to generate reference descriptions, simplifying prompt creation while making the process considerably more AVLM-agnostic.

Overview: We propose the LogicAD algorithm, which consists of three primary components: *text feature extraction*, *format embedding* and *logic reasoner*, as shown in Figure 2. The *text feature extraction*, f_{i2t}, which converts image to text, generates consistent and reliable logical text descriptions. The *format embedding* component calculates an anomaly score to identify deviations from normal patterns. Meanwhile, the *logic reasoner* generates textual reasonings for the observed anomalies. Together, these components enable not only AD but also the explanation of the underlying reasons.

Text Feature Extraction

Vision-language models such as CLIP (Radford et al. 2021), ImageBind (Girdhar et al. 2023), and EVA-CLIP (Sun et al. 2023) are extensively utilized in zero-shot and AD algorithms (Jeong et al. 2023; Gu et al. 2024a). However, in the domain of LA detection, CLIP-based VLMs exhibit significant limitations, yielding markedly poorer performance on tasks involving logical inconsistencies compared to their effectiveness in detecting SA. Table 6 (categories shaded in grey) shows that WinCLIP performs significantly worse in categories containing primarily naive logic-related anomalies, such as missing or mislocated objects (Jeong et al. 2023).

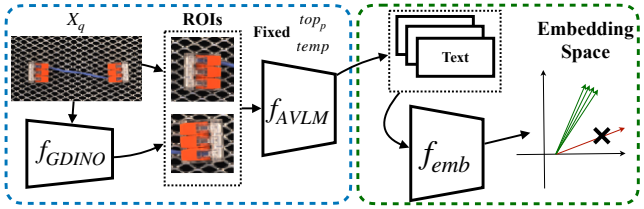


Figure 3: Text feature extraction, f_{i2t} involves ROI extraction (blue box) and text embedding filtering (green box). Patches and the original image are processed by an AVLM to generate K text descriptions. The green box uses text-embedding-3-large for output stabilization.

AVLMs, such as BLIPv2, LLaVA and GPT-4o (Li et al. 2023; Liu et al. 2024a; Achiam et al. 2023) have demonstrated remarkable potential in image understanding but also face challenges in handling logic-related tasks such as counting objects, localizing objects, understanding chains of logic (Pan et al. 2023) along with inconsistent results (Pan et al. 2023; Achiam et al. 2023). To alleviate these issues,

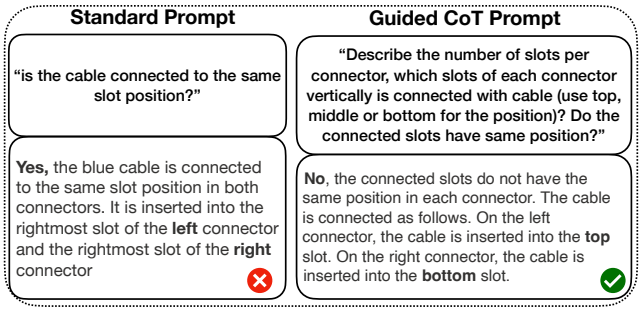
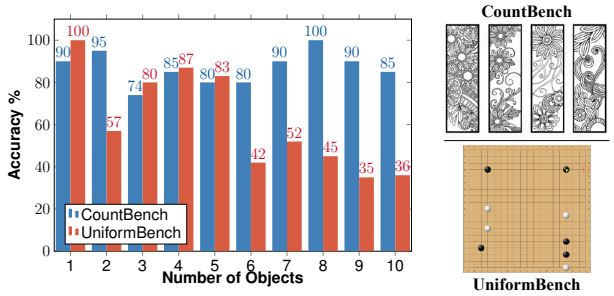


Figure 4: Illustration of a standard prompt versus a Guided Chain-of-Thought (Guided CoT) prompt. We use image X_q from Figure 3 as the input. The ground truth description specifies that two cables are not connected to the same slot position. Using prompts based on Guided CoT, AVLMs can generate more accurate descriptions of the input image.



(a) The accuracy of GPT-4o in counting. (b) Sample Images

Figure 5: Comparison of GPT-4o counting accuracy and additional visual examples. Accuracy of GPT-4o drops significantly with an increasing number of homogeneous objects. Figure 5b shows two samples, one from CountBench (top) and one from UniformBench (bottom).

we designed our *text feature extraction* module as shown in Figure 3 with the following components: Guided CoT, ROI segmentation, and text embedding filtering.

Guided CoT. WinCLIP proposes the Compositional Prompt Ensemble (CPE), which generates all predefined text prompts containing words that describe the state of the object, such as "flawless, damaged, or defect" (Jeong et al. 2023). Many other competing methods (Zhou et al. 2024; Gu et al. 2024a) are based on WinCLIP. CPE is suitable for SA but is insufficient for handling LA, as shown in Table 6. Inspired by (Wei et al. 2022), we propose to use a Guided CoT for logical description. We guided the AVLMs model to inspect the images based on Guided CoT text prompts, as shown in Figure 4. Without providing specific location-related logic guides, such as describing the position only "vertically", or numerical logic guides, such as "specifying the number of slots per connector", AVLMs such as GPT-4o fail to comprehend the semantic meaning of localization, leading to logical hallucinations. Appendix A.1 demonstrates more Guided CoT examples. Although our method requires manual prompt creation, providing detailed and reusable text prompts is significantly easier in practical applications than other methods requiring visual annotations (Kim et al. 2024; Zhang and Wang 2024) (Figure 1).

ROI Segmentation and Text Embedding Filtering. Current autoregressive Vision-Language Models have limitations in logical reasoning tasks, such as counting, object-size estimation, localization, and basic calculations (Achiam et al. 2023; Zhang and Wang 2024; Lee 2023). To further investigate these limitations, we evaluated GPT-4o using a subset of the CountBench dataset (Paiss et al. 2023). We randomly selected 250 images from this dataset. Additionally, we created a custom dataset, UniformBench composed of 150 images characterized by homogeneous features such as varying numbers of pawns on a chessboard, beer bottles in a basket, and Go boards with random distributions of white and black stones. More sample images from CountBench and UniformBench are shown in Appendix A.2. As illustrated in Figure 5a, GPT-4o demonstrates high accuracy in object counting tasks within CountBench, where images often present heterogeneous features. However, the model's performance significantly declines when tested on the UniformBench dataset, which lacks these additional heterogeneous features. This reduction in accuracy correlates with increasing task complexity in object counting, demonstrating a critical limitation in the model's ability to handle essential tasks for industrial logical AD.

To alleviate this issue, we propose reducing the complexity by selecting ROIs from the image, using GroundingDINO as our ROI extraction model (Liu et al. 2023a). All the prompts used for GroundingDINO are keywords directly extracted from AVLMs without any additional effort. The detailed prompts are provided in Appendix A.1. Most AVLMs demonstrate a noticeable level of inconsistency even when using the same seed or tuning the hyperparameters of top_p and *temperature* (Song et al. 2024). We suggest generating the extracted text multiple times and then using an embedding model as a filter to eliminate outlier text by using the Local Outlier Factor (LOF) method (Breunig et al. 2000). The detailed steps of text extraction can be summarized as follows: **(1)** Generating a set of regions $\mathbf{w}_i$, where $i \in [1, N]$ and N represents the total number of ROIs, using the function f_{GDINO} with feature prompts. Each region, along with the original image, is then processed by the function f_{AVLM}, $K = 3$ times, yielding a collection of textual descriptions $\mathcal{T} = \{t_1, t_2, t_3\}$. **(2)** Constructing the text embedding space $\mathcal{M}$ by applying the text embedding model, text-embedding-3-large from OpenAI (Achiam et al. 2023), f_{emb} to $\mathcal{T}$, resulting in $\mathcal{M} = f_{emb}(\mathcal{T}) = \{\mathbf{e}_i\}_{i=1}^{k}$, where $\mathbf{e}_i$ is the embedding of the extracted text. Subsequently, these text embeddings are fed into an outlier detection model, specifically the Local Outlier Factor (LOF) function f_{LOF} , to generate the filtered text embedding space, which is denoted as $\mathcal{T}_{filter}$. We then randomly select corresponding text from the filtered embedding space $\mathcal{T}_{filter}$.

Format Embedding

After extracting the text features, we use an LLM to summarize the text into *JSON* format. Both the normal/reference image X_n and the query image X_q are then fed into

the embedding function f_{emb} to generate the respective embedding features $\hat{\mathbf{e}}_n$ and $\hat{\mathbf{e}}_q$. We then calculate the anomaly score based on the cosine similarity between these normalized embeddings as follows: $ascore = 1 - \langle \hat{\mathbf{e}}_n, \hat{\mathbf{e}}_q \rangle$.

Logic Reasoner

We aim not only to achieve high performance in AD, but also to make the detection results explainable. To achieve the latter, we convert the text summary into a formal specification and use an external automated theorem prover (ATP) to perform logical reasoning. To obtain formal specifications of an image, we use an LLM and a two-shot prompt. Consider the category "breakfast box" from MVTec LOCO AD (Bergmann et al. 2022) for example, the hypothetical text features of an image include the following statement:

"On the left side of the box, there is a nectarine and an apple, and on the right, there are some nuts..."
The formal specification (denoted as Σ_0) generated via LLM is as follows:

$$left(nectarine, 1); left(apple, 1); right(nut, irrel) \dots$$

Here $irrel$ means that the number (of the nuts) is irrelevant. Identifying anomalies via logical reasoning requires a rigorous specification of what is normal. Consider a description of normality as follows (It may not necessarily be the definition of normality in the original dataset. Our approach can be easily adopted for different definitions of "anomaly"):

"On the left side of the plate, either there is an apple and no nectarine, or there is a nectarine but no apples. Besides, there should be two tangerines on the left..."
Based on that, we derive the normal specification Σ_{norm}:

$$\begin{aligned}&\big((left(apple, 1) \wedge left(nectarine, 0)) \vee \\ &\qquad (left(nectarine, 1) \wedge left(apple, 0))\big) \\ &left(tangerine, 2) \dots\end{aligned}$$

Σ_{norm} can include any relational formulae. We exclude function symbols since empirically, they reduce the stability and precision of parsing. Due to the ambiguity of natural languages, some features may not be covered here: When one says "there is **an** apple on the left", implicitly, we rule out other numbers. For things that do not occur in the image, one may not explicitly specify their numbers. Hence, we use the following formulae to complete the logic program:

- Σ_{na}: Inspired by the *unique name assumption*, we use a set of (in-)equalities to explicitly specify whether two constants denote the same object or not, e.g.

  $$tangerine = mandarin\ ,\ tangerine \neq apple\ ,\ \dots$$

  The distinguishability of constants is obtained by the response to LLM queries such as "*Answer Yes or No: Are* $object1$ *and* $object2$ *synonymous or similar?*"
- Σ_{fa}: We can specify if a predicate has functional behaviour. For instance, the predicate $right$ denotes the number of instances of an object on the right, and each object should be assigned a unique number Thus Σ_{fa} will include:

  $$\forall x \exists y. right(x, y) \wedge (\forall y'. right(x, y') \rightarrow (y' = y))$$

  One can manually specify if a predicate is functional, or it can be automatically decided by an LLM query. Appendix A.3 shows some detailed examples for multiple categories.
- Σ_{dca}: To identify out-of-domain anomalies such as "there's a bug on the plate", we introduce axioms which serve as the *domain closure* (Reiter 1980):

  $$\forall x. (\neg left(x, 0) \rightarrow (x = apple \,|\, x = tangerine \,|\, \cdots))$$

  On the right-hand side, this is a disjunction of all constants mentioned in the normal specification Σ_{norm}. With this axiom, unmentioned objects should not exist on the left, i.e. $left(bug, 1)$ will be verified as an anomaly.
- Default: Conversely, to identify the missing item, we complete the formal description by adding default values: for example, if $left(tangerine, \_\_)$ (or anything synonymous with tangerine) is not mentioned in the image description Σ_0, then $left(tangerine, 0)$ is added to Σ_0.

Among the aforementioned formulae, Σ_{norm} and default values are essential and need to be provided for each class of AD task ("breakfast box", "juice bottle", etc.). The rest can be automatically generated or manually specified.

Let $\Gamma = \Sigma_{norm} \cup \Sigma_{na} \cup \Sigma_{fa} \cup \Sigma_{dca}$ be the union of all hypotheses. The AD is then converted to a task of theorem proving:

- If $\Gamma \models \neg\Sigma_0$, then we label the image as abnormal.
- If $\Gamma \not\models \neg\Sigma_0$, then we label the image as normal

We use Prover9 for theorem proving. Here $\Gamma \models \neg\Sigma_0$ means Γ logically entails $\neg\Sigma_0$, i.e. every logical model satisfying Γ will also satisfy the negation of Σ_0. Since Σ_0 is the formal description of the image, it shows that the image description contradicts the normal cases and hence the image is abnormal. To identify the actual anomaly, we look for a minimal subset of Σ_0 which causes the anomaly, i.e. for $\Sigma_a \subseteq \Sigma_0$, if $\Gamma \models \neg\Sigma_a$ and for any $\Sigma' \subsetneq \Sigma_a$, we have $\Gamma \not\models \neg\Sigma'$, then Σ_a forms an explanation: Consider Σ_0 and Γ defined as above with all the mentioned formulae, then $\Gamma \models \neg\Sigma_0$ since both an apple and a nectarine are on the left. Then $\Sigma_a = \{left(nectarine, 1), left(apple, 1)\}$ is a formal explanation of the anomaly since $\Sigma_a \subseteq \Sigma_0$, and

$$\begin{aligned}&\Gamma \models \neg(left(nectarine, 1) \wedge left(apple, 1)) \\ &\Gamma \not\models \neg left(nectarine, 1) \\ &\Gamma \not\models \neg left(apple, 1)\end{aligned}$$

Experiments and Results

We conduct comprehensive experiments to evaluate the effectiveness of our algorithm using three SOTA AVLMs: GPT-4o (Achiam et al. 2023), LLaVA 1.6 (Liu et al. 2024b), LLaVA 1.5 (Liu et al. 2024a). Detailed information on the deployment and versions of AVLMs is provided in Appendix A.4. All experiments are training-free. Our evaluations are performed on two datasets, MVTec AD and MVTec LOCO AD (Bergmann et al. 2019, 2022). We perform one-shot experiments using a single training image and compare

MVTec LOCO AD (only LA)	**LogicAD** (Ours)		**AnomalyMoE**† (CVPR VAND 24)		**WinCLIP** (CVPR 23)		**GCAD** (IJCV 23)	PatchCore (CVPR 22)	ComAD (AEI 22)	AST (ICCV 23)
Category	AUROC	F_1-max	AUROC	F_1-max	AUROC	F_1-max	AUROC	AUROC	AUROC	AUROC
Breakfast Box	93.1±2.1	82.7±1.4	-	-	57.6	63.3	87.0	74.8	94.5	80.0
Juice Bottle	81.6±3.5	83.2±4.3	-	-	75.1	58.2	100.0	93.9	90.9	91.6
Pushpins	98.1±0.1	98.5±0.1	-	-	54.9	57.3	97.5	63.6	89.0	65.1
Screw Bag	83.8±5.2	77.9±4.5	-	-	69.5	58.8	56.0	57.8	79.7	90.1
Splicing Connector	73.4±3.2	76.1±2.1	-	-	64.5	59.9	89.7	79.2	84.4	81.8
Average	**86.0** (**18.1**% ↑)	**83.7** (**4.6**% ↑)	67.9	79.1	64.3	59.5	86.0	74.0	87.7	79.7

Table 1: Logical Anomaly detection (classification) performance on MVTec LOCO AD **(one-shot)**. AnomalyMoE†is the SOTA few-short logical AD algorithm (CVPR 2024 VAND Challenge Winner). PatchCore, GCAD, ComAD, and AST are all full-shot unsupervised methods trained on all images. GCAD and AnomalyMoE are designed to handle logical AD. For each category, we conducted five experiments and calculated the average and standard deviation. Values highlighted in red indicate increased scores compared to AnomalyMoE†. The evaluation results for WinCLIP were generated using Anomalib.

our results with competing methods, including full-shot approaches trained on all available images.

Dataset and Metrics: The MVTec LOCO AD dataset (Bergmann et al. 2022) is a benchmark for detecting logical anomalies in industrial settings. It comprises five categories, each featuring a variety of LAs, including missing objects, extra objects, mismatches between colors and objects, and other logical inconsistencies. Additionally, we evaluate our model using MVTec AD (Bergmann et al. 2019) and focus on some categories, such as screw, pill, toothbrush, capsule, and transistor. These categories are often considered difficult and perform significantly less than others due to their logic-related anomaly characteristics (Jeong et al. 2023; Zhou et al. 2024). We use F_1-max and Area Under the Receiver Operating Characteristic (AUROC) as evaluation metrics, consistent with with SOTA and competing methods.

Results

Can LogicAD detect naive logical anomalies? Current vision features-based algorithms, including WinCLIP and AnomalyCLIP, have shown remarkable performance in SA AD while facing challenges in several categories, e.g. , capsule, transistor and toothbrush, from MVTec AD (Jeong et al. 2023; Zhou et al. 2024; Bergmann et al. 2019). Although MVTec AD is not designed for evaluating the performance of LA AD, these categories share some naive logical-related characteristics, such as missing objects and mislocation (Jeong et al. 2023). Table 2 and Table 6 indicate that the evaluated algorithms exhibit suboptimal performance in these categories. Since LogicAD is explicitly designed to address logical inconsistencies, Table 2 shows that LogicAD surpasses WinCLIP by an impressive 5.6% in detecting LA related categories. Moreover, when compared with other SOTA AD methods, such as AnomalyCLIP (Zhou et al. 2024) and VAND (Chen, Han, and Zhang 2023) (all of which are fine-tuned with domain-specific datasets), the training-free LogicAD significantly outperforms the VAND algorithm by 15% and is comparable to AnomalyCLIP (Zhou et al. 2024).

Can LogicAD detect sophisticated logical anomalies? Compared with MVTec AD (Bergmann et al. 2019), MVTec LOCO AD (Bergmann et al. 2022) is specifically designed to evaluate the performance of logical AD with more sophisticated LA. Table 1 presents the performance evaluation of LogicAD on the MVTec LOCO AD dataset. Compared to the SOTA few-shot VLM-based algorithm, AnomalyMoE (Gu et al. 2024b), LogicAD demonstrates superior performance across all metrics, achieving an increase of **18.1**% in AUROC and **4.6**% in F_1-max score. Even when compared to full-shot methods, such as PatchCore (Roth et al. 2022) and AST (Paiss et al. 2023), our method outperforms in many categories. Additionally, when compared to other full-shot algorithms with additional global features such as GCAD and ComAD (Bergmann et al. 2022; Liu et al. 2023b), LogicAD exhibits highly competitive results. These findings underscore our model's effectiveness and

MVTec AD	SA	LA	**w/o** Training		**w/** Training	
			LogicAD	WinCLIP	Anomaly-CLIP	VAND
Texture	✓	✗	96.9	98.1	98.7	96.9
Object*	✓	✓	86.5	80.9	89.1	71.5

Table 2: One-shot Logical AD performance comparison (AUROC). The term Object* includes four categories related to LA in the MVTec AD dataset: capsule, pill, transistor, and toothbrush. Carpet, grid, leather, tile, and wood are categorized as texture and mainly contain SA.

$\mathcal{M}_{GCoT}$	$\mathcal{M}_{ROI}$	$\mathcal{M}_{FEmbe}$	$\mathcal{M}_{LR}$	AUROC	F_1-max
✗	✗	✗	✗	23.4	9.5
✓	✗	✗	✗	48.5	51.3
✓	✓	✗	✗	60.4	65.3
✓	✓	✓	✗	86.0	**83.7**
✓	✓	✗	✓	N/A	**83.3**

Table 3: Ablation of different modules in LogicAD model on MVTec LOCO AD. $\mathcal{M}_{GCoT}$, $\mathcal{M}_{ROI}$, $\mathcal{M}_{FEmbe}$, $\mathcal{M}_{LR}$ denote as Guided CoT, Region of Interest, *format embedding*, and *logic reasoner*. With *logic reasoner*, our model can predict abnormal scores of either 0 or 1 based on ATP, consequently not applicable for AUROC.

MVTec LOCO AD (only SA)	AUROC
LogicAD (ours)	81.5
WinCLIP	64.6
GCAD	80.7
AST	87.7
PatchCore	89.3

Table 4: LogicAD performance on SA in MVTec LOCO AD dataset.

VLMs	AUROC	F_1-max
GPT-4o	**86.0**	**83.2**
LLaVA1.5	73.3	71.0
LLaVA1.6	76.2	78.1

Table 5: LogicAD performance with different AVLMs backbones on MVTec LOCO AD.

demonstrate that for non-parametric tasks such as logical understanding, LLMs with AVLMs have a notable advantage over parametric visual memory bank methods such as PatchCore and AST. Using the *logic reasoner* as shown in Table 3, we achieve an impressive score of 83.3%, which is only 0.4% lower than the score achieved with *format embedding* and only in very few cases, *logic reasoner* has different predictions compared to using *format embedding*, as shown Appendix A.5. However, the advantages of using the *logic reasoner* are substantial, namely, it enhances the model's explainability and eliminates the need for manual or dynamic thresholding, which can cause significant issues in real-world scenarios (Gu et al. 2024a). Furthermore, we conduct experiments using different AVLMs, specifically LLaVA 1.5 and LLaVA 1.6 (Liu et al. 2024a,b). Although LLaVA 1.5 peforms worse in F_1-max score than the SOTA method, LLaVA 1.6 achieves results comparable to the SOTA. However, regarding AUROC, both LLaVA 1.5 and LLaVA 1.6 outperform the SOTA significantly. This indicates that our method will continue to benefit from future advances in AVLMs research.

Can LogicAD detect structural anomalies? LogicAD employs a Guided CoT-based methodology primarily for detecting logical anomalies. However, by utilizing carefully curated prompts, our algorithm is also capable of identifying structural anomalies. We evaluated our model on two benchmark datasets: MVTec AD and MVTec LOCO AD. On the MVTec LOCO AD dataset, our model outperforms the one-shot method, WinCLIP and the full-shot method GCAD, but is slightly inferior compared to PatchCore and AST based on Table 4. When evaluated on the MVTec AD benchmark, we compared our approach with WinCLIP, a baseline model that does not require fine-tuning with domain-specific datasets and can be applied directly. As shown in Table 6 and Appendix A.6, LogicAD achieves highly competitive scores compared to WinCLIP-based methods, demonstrating its effectiveness in detecting SA. As illustrated in Table 2, for texture categories, LogicAD attains an AUROC of 96.9%, which is marginally lower than the 98.1% achieved by WinCLIP and the 98.7% achieved by AnomalyCLIP, but comparable to VAND (Chen, Han, and Zhang 2023). These results suggest that while LogicAD is highly effective in general AD, particularly those related to logical inconsistencies, there is a minor performance gap when compared to leading train-free vision feature-based methods.

MVTec (AD)		LogicAD		WinCLIP	
	Category	AUROC	F_1-max	AUROC	F_1-max
Texture	Average	96.9	**97.3**	**98.1**	96.7
Object	Capsule	**84.7**	**92.2**	77.3	91.5
	Pill	**78.4**	**91.5**	78.1	91.2
	Transistor	**84.4**	**81.3**	81.1	62.6
	Toothbrush	**90.0**	**89.9**	87.1	88.1
	Zipper	**93.1**	**92.5**	84.3	89.8
	Screw	**89.1**	81.8	74.3	**87.5**
	Hazelnut	93.5	**95.1**	92.2	89.7
	Bottle	79.5	81.5	**98.7**	**96.8**
	Cable	79.4	81.2	**85.9**	**85.1**
	Metal Nut	89.6	90.1	**92.2**	**93.2**
	Objects	86.2	87.7	85.1	88.7
	Average	**89.7**	90.9	88.9	**91.4**

Table 6: Anomaly classification performance comparison on MVTec AD between LogicAD and WinCLIP. Object categories (the grey-out sections), such as capsule, pill, transistor and toothbrush, contain LAs, and LogicAD performs better in these categories. Bold indicates the best score.

Limitations

Although our method brings a new perspective and achieves remarkable results, it still has some limitations, such as inconsistent results obtained with different AVLMs and relatively long inference time with an average of a few seconds per image. While methods such as BitNet or AVLM pruning can accelerate inference time, such optimizations are beyond the scope of this work (Shang et al. 2024; Wang et al. 2023). Additionally, we observe some of our failing cases, mainly caused by logical inconsistency in Appendix A.3. Lastly, with Guided CoT, our prompts still require minor manual text prompt input, but we note that curated prompts can be reused and need to be defined only once per AD task.

Conclusion

In this paper, we propose a novel framework for AD utilizing extracted text from AVLMs. By incorporating Guided CoT, ROI, and text formatting, our approach leverages the robust logical understanding capabilities of AVLMs, achieving remarkable one-shot performance in logical AD and surpassing SOTA by a significant margin on the latest logical AD benchmarks. LogicAD also integrates a theorem prover to predict logical anomalies with corresponding explanations, thereby enhancing the explainability of the model. Our work explores a novel direction in AD, demonstrating that using text features can be highly effective, particularly in logical AD. In the future, we plan to extend our work by developing a fully automated prompting process by fine-tuning and distilling AVLMs with logic-related data to reduce inference time while enhancing logical understanding.

Acknowledgments

This paper is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 2236/2 and

the EU ICT-48 2020 project TAILOR (No. 952215), the German Federal Ministry of Education and Research (BMBF) under the project WestAI (Grant no. 01IS22094D) and Bio4Monitoring (Grant no. 031B1155).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Bergmann, P.; Batzner, K.; Fauser, M.; Sattlegger, D.; and Steger, C. 2022. Beyond Dents and Scratches: Logical Constraints in Unsupervised Anomaly Detection and Localization. *International Journal of Computer Vision*, 130(4): 947–969.

Bergmann, P.; Fauser, M.; Sattlegger, D.; and Steger, C. 2019. MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *Conference on Computer Vision and Pattern Recognition*, 9592–9600.

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000. LOF: Identifying Density-based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 93–104.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9650–9660.

Chen, X.; Han, Y.; and Zhang, J. 2023. APRIL-GAN: 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*.

Chen, Z.; Zhou, Q.; Shen, Y.; Hong, Y.; Sun, Z.; Gutfreund, D.; and Gan, C. 2024. Visual Chain-of-Thought Prompting for Knowledge-Based Visual Reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 1254–1262.

Enderton, H. B. 2001. *A Mathematical Introduction to Logic*.

Girdhar, R.; El-Nouby, A.; Liu, Z.; Singh, M.; Alwala, K. V.; Joulin, A.; and Misra, I. 2023. ImageBind: One Embedding Space To Bind Them All. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15180–15190.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6904–6913.

Gu, Z.; Zhu, B.; Zhu, G.; Chen, Y.; Tang, M.; and Wang, J. 2024a. AnomalyGPT: Detecting industrial anomalies using large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 1932–1940.

Gu, Z.; Zhu, B.; Zhu, G.; Chen, Y.; and Wang, J. 2024b. CVPR, Visual Anomaly and Novelty Detection 2.0 Winner 2024, https://www.hackster.io/contests/openvino2024, Accessed: 2024-08-01.

Gunjal, A.; Yin, J.; and Bas, E. 2024. Detecting and Preventing Hallucinations in Large Vision Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 18135–18143.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Jeong, J.; Zou, Y.; Kim, T.; Zhang, D.; Ravichandran, A.; and Dabeer, O. 2023. WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 19606–19616.

Kim, S.; An, S.; Chikontwe, P.; Kang, M.; Adeli, E.; Pohl, K. M.; and Park, S. H. 2024. Few Shot Part Segmentation Reveals Compositional Logic for Industrial Anomaly Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 8591–8599.

Lee, M. 2023. A Mathematical Investigation of Hallucination and Creativity in GPT Models. *Mathematics*, 11(10): 2320.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *International Conference on Machine Learning*, 19730–19742.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024a. Improved Baselines with Visual Instruction Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26296–26306.

Liu, H.; Li, C.; Li, Y.; Li, B.; Zhang, Y.; Shen, S.; and Lee, Y. J. 2024b. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge.

Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Li, C.; Yang, J.; Su, H.; Zhu, J.; et al. 2023a. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*.

Liu, T.; Li, B.; Du, X.; Jiang, B.; Jin, X.; Jin, L.; and Zhao, Z. 2023b. Component-aware anomaly detection framework for adjustable and logical industrial visual inspection. *Advanced Engineering Informatics*, 58: 102161.

Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.-W.; Zhu, S.-C.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. *Advances in Neural Information Processing Systems*, 35: 2507–2521.

Olausson, T. X.; Gu, A.; Lipkin, B.; Zhang, C. E.; Solar-Lezama, A.; Tenenbaum, J. B.; and Levy, R. P. 2023. LINC: A Neurosymbolic Approach for Logical Reasoning by Combining Language Models with First-Order Logic Provers. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Paiss, R.; Ephrat, A.; Tov, O.; Zada, S.; Mosseri, I.; Irani, M.; and Dekel, T. 2023. Teaching CLIP to Count to Ten. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, 3170–3180.

Pan, L.; Albalak, A.; Wang, X.; and Wang, W. 2023. Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 3806–3824. Singapore: Association for Computational Linguistics.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*, 8748–8763.

Reiter, R. 1980. Equality and domain closure in first-order databases. *Journal of the ACM (JACM)*, 27(2): 235–249.

Rippel, O.; Mertens, P.; König, E.; and Merhof, D. 2021. Modeling the Distribution of Normal Data in Pre-Trained Deep Features for Anomaly Detection. *IEEE Transactions on Instrumentation and Measurement*, 70: 1–13.

Rippel, O.; Mertens, P.; and Merhof, D. 2021. Modeling the Distribution of Normal Data in Pre-Trained Deep Features for Anomaly Detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 6726–6733.

Roth, K.; Pemula, L.; Zepeda, J.; Schölkopf, B.; Brox, T.; and Gehler, P. 2022. Towards Total Recall in Industrial Anomaly Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14318–14328.

Rudolph, M.; Wehrbein, T.; Rosenhahn, B.; and Wandt, B. 2023. Asymmetric Student-Teacher Networks for Industrial Anomaly Detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2592–2602.

Shang, Y.; Cai, M.; Xu, B.; Lee, Y. J.; and Yan, Y. 2024. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*.

Song, Y.; Wang, G.; Li, S.; and Lin, B. Y. 2024. The Good, The Bad, and The Greedy: Evaluation of LLMs Should Not Ignore Non-Determinism. *arXiv preprint arXiv:2407.10457*.

Sun, Q.; Fang, Y.; Wu, L.; Wang, X.; and Cao, Y. 2023. EVA-CLIP: Improved Training Techniques for CLIP at Scale. *arXiv preprint arXiv:2303.15389*.

Surís, D.; Menon, S.; and Vondrick, C. 2023. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11888–11898.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

Wang, H.; Ma, S.; Dong, L.; Huang, S.; Wang, H.; Ma, L.; Yang, F.; Wang, R.; Wu, Y.; and Wei, F. 2023. Bitnet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*.

Wang, W.; Chen, Z.; Chen, X.; Wu, J.; Zhu, X.; Zeng, G.; Luo, P.; Lu, T.; Zhou, J.; Qiao, Y.; et al. 2024. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *Advances in Neural Information Processing Systems*, 36.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35: 24824–24837.

Ye, X.; Chen, Q.; Dillig, I.; and Durrett, G. 2024. SatLM: Satisfiability-Aided Language Models Using Declarative Prompting. *Advances in Neural Information Processing Systems*, 36.

Zhang, C.; and Wang, S. 2024. Good at Captioning, Bad at Counting: Benchmarking GPT-4v on Earth Observation data. *arXiv preprint arXiv:2401.17600*.

Zhang, Y.; Cao, Y.; Xu, X.; and Shen, W. 2024. LogiCode: an LLM-Driven Framework for Logical Anomaly Detection. *arXiv preprint arXiv:2406.04687*.

Zhou, Q.; Pang, G.; Tian, Y.; He, S.; and Chen, J. 2024. AnomalyCLIP: Object-agnostic Prompt Learning for Zero-shot Anomaly Detection. In *The Twelfth International Conference on Learning Representations*.

Zou, Y.; Jeong, J.; Pemula, L.; Zhang, D.; and Dabeer, O. 2022. SPot-the-Difference Self-Supervised Pre-training for Anomaly Detection and Segmentation. In *European Conference on Computer Vision*, 392–408.