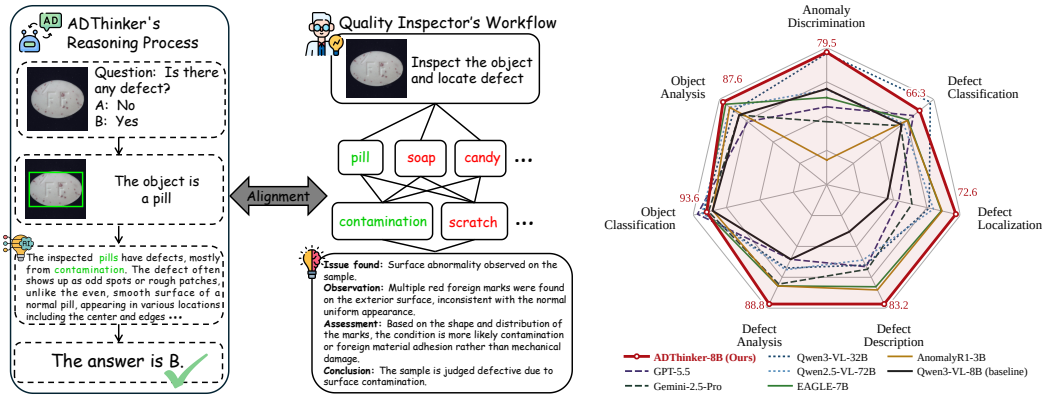
TEACHING MLLMS TO THINK LIKE QUALITY INSPECTOR VIA HIERARCHICAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Driven by the rapid evolution of foundation models, the paradigm of industrial anomaly detection is shifting from traditional deep learning approaches to Multimodal Large Language Models (MLLMs). While MLLM-based methods have demonstrated impressive capabilities, existing works often fail to satisfy the requirements of real-world industrial production. Current research predominantly focuses on determining merely whether a defect exists, while neglecting that industrial quality inspection and subsequent process adjustments rely heavily on diagnostic details, i.e., which object is involved, why a defect is suspected, and what judgment is made. Inspired by the diagnostic process of human inspectors, we abstract object and defect morphologies into generalized structured knowledge and propose Hierarchical Knowledge Preference Optimization (HKPO). Through hierarchical branch-level and segment-level preference optimization, the model learns to identify objects, reference structured knowledge, and judge defects, thereby ensuring a rational and well-grounded quality inspection process. Extensive experiments on public benchmarks show that our framework lifts a lightweight 8B model 7.15% above its own backbone and 2.71% above a 4× larger 32B counterpart, surpasses the strong proprietary model by 5.46% in average accuracy, and outperforms dedicated anomaly-detection MLLMs by at least 7.95% on anomaly discrimination.



(a) Motivation. (b) Performance.

Figure 1: Design motivation and empirical performance of ADThinker. (a) Design motivation: the hierarchical reasoning process of ADThinker is aligned with the workflow of practical quality inspection. (b) Empirical performance on seven IAD tasks from MMAD, where ADThinker achieves strong and balanced results.

1 INTRODUCTION

Driven by the rapid evolution of large models, the paradigm of industrial anomaly detection has gradually shifted from traditional deep learning to Multimodal Large Language Models (MLLMs). Leveraging an interactive question-answering format, these generative models facilitate anomaly

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

detection by autoregressively generating categorical responses conditioned on natural language instructions and visual inputs. Benefiting from the superior generalization capabilities of large models, MLLM-based approaches enable rapid adaptation to new production lines with minimal samples. This significantly alleviates the repetitive and labor-intensive burden of creating customized models for every specific defect type.

Recent studies on MLLM-based industrial anomaly detection have largely followed two lines of research. The first focuses on improving anomaly perception through enhanced visual modeling, anomaly-aware data construction, and finer alignment between image regions and defect semantics (Gu et al., 2024; Xu et al., 2025; Zhang et al., 2025). The second seeks to strengthen reasoning capabilities through instruction tuning, multi-step inference, and reinforcement-style optimization, enabling models to produce more reliable judgments for industrial inspection (Chao et al., 2025; Guan et al., 2025; Li et al., 2025; Zeng et al., 2025; Zhao et al., 2025). Together, these efforts have advanced the integration of visual perception and language reasoning in industrial anomaly detection.

Despite the impressive performance achieved by MLLM-based anomaly detection methods, existing works suffer from a critical limitation: merely providing a binary conclusion of defective or non-defective is far from sufficient for industrial quality inspection. As shown in Figure 1a, a professional quality inspector is required to provide detailed analysis and evidence based on quality standards. Hence, MLLM methods need to be highly aligned with human inspector workflows to facilitate subsequent process optimization.

To address the above issue, we propose Hierarchical Knowledge Preference Optimization (HKPO), a framework that bridges the cognitive gap between MLLMs and professional quality inspectors. Our contributions are threefold:

- **General, Transferable and Hierarchical Knowledge Base**: Instead of relying on the memorization of concrete knowledge cases, our goal is to empower the model with genuine reasoning abilities by abstracting the essential forms of objects, knowledge and defects. Given that common defect morphologies are relatively centralized, our constructed knowledge base can be generalized to unseen defects based on common morphological characteristics.
- **Hierarchical Knowledge Preference Optimization Framework for Anomaly Detection**: Inspired by the cognitive process of human quality inspectors, we propose a hierarchical preference optimization framework that mimics the step-by-step diagnostic process: first identifying the object, retrieving structured knowledge, and finally performing anomaly detection. This framework proposes branch-level and segment-level preference optimization to jointly enforce a transparent and interpretable reasoning chain that aligns with human quality inspectors.
- **Extensive Experiments and Systematic Analysis on Public Benchmarks**: We conduct extensive experiments on MMAD and SSGD, demonstrating that a lightweight 8B model, when aligned with hierarchical knowledge and guided by our reasoning framework, can surpass the performance of a significantly larger 32B model by approximately 3%. Furthermore, this architecture exhibits strong generalization capabilities, enabling the detection of previously unseen defect types through logical inference rather than mere pattern matching.

2 Related Work

Industrial Anomaly Detection (IAD). Industrial anomaly detection aims to identify defective components to support production quality control, encompassing anomaly detection, localization, and diagnostic analysis. Most traditional methods operate under unsupervised settings, learning normal patterns and detecting deviations at inference time. Representative approaches include local region similarity analysis (Li et al., 2024), memory bank methods (Gao, 2024; Jiang et al., 2022; Wang et al., 2025), and synthetic anomaly generation (Zavrtanik et al., 2021). Although effective in conventional settings, these methods are typically tied to predefined anomaly patterns, limiting generalization to diverse and unseen anomaly types.

MLLM-based Anomaly Detection. With the rapid progress of multimodal large models, recent studies have explored their application to IAD. MMAD (Jiang et al., 2024) provides a systematic benchmark defining seven representative subtasks. Prior work can be broadly categorized into two groups. The first aims to improve anomaly perception through data construction or architectural

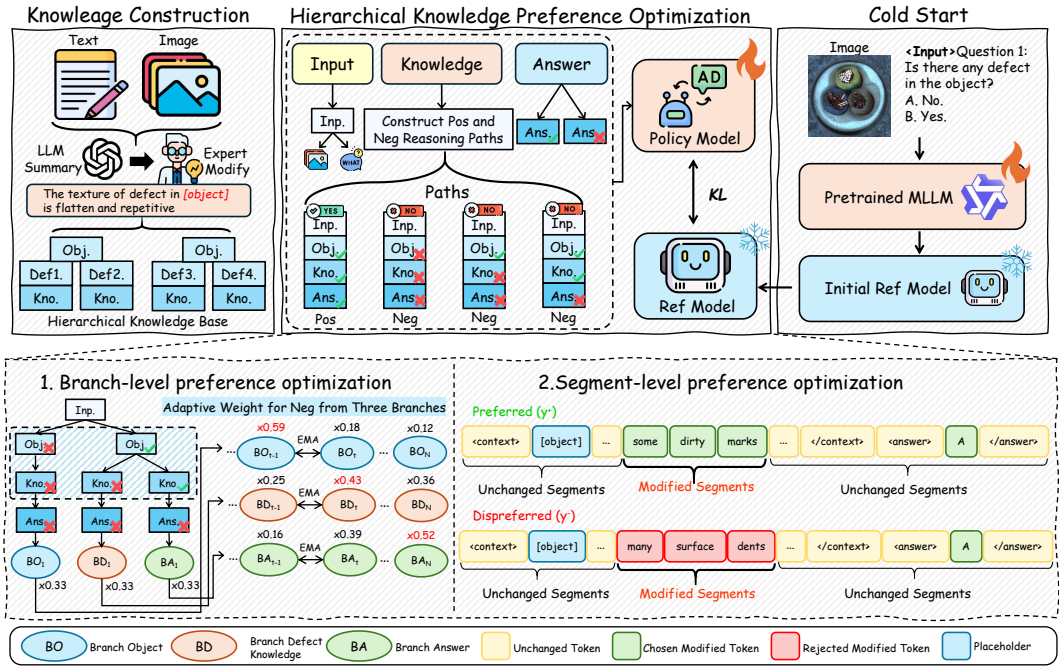


Figure 2: Overview of ADThinker. The proposed framework integrates a hierarchical knowledge base with the HKPO training module. The knowledge base is structured according to an *Object–Defect–Knowledge* schema. Within the HKPO module, we propose branch-level and segment-level preference optimization that force the model to reason the correct answer along the precise trajectory of object, defect knowledge, and answer branches.

adaptation: AnomalyGPT (Gu et al., 2024) synthesizes anomalous images with textual descriptions to fine-tune LLMs for anomaly region identification; Anomaly-OneVision (Xu et al., 2025) introduces a "look-twice" mechanism to emphasize anomalous visual tokens; LogSAD (Zhang et al., 2025) presents a training-free "thought matching" framework that calibrates anomaly scores from multiple LLM detectors. The second group focuses on enhancing reasoning: methods such as AnomalyR1 (Chao et al., 2025), EMIT (Guan et al., 2025), IAD-R1 (Li et al., 2025), LR-IAD (Zeng et al., 2025), and OmniAD (Zhao et al., 2025) strengthen multi-step reasoning through GRPO-based training. However, current methods mainly focus on generating high-quality CoT reasoning, while grounding such reasoning in reliable visual evidence and robust anomaly semantics remains insufficiently addressed.

3 Methodology

As shown in Figure 2, **ADThinker** framework consists of two core components: (1) *hierarchical knowledge modeling*, which decomposes industrial anomaly diagnosis into a sequence of dependent stages, including object identification, defect knowledge reasoning, and final anomaly prediction, and structures these stages hierarchically; and (2) *hierarchical knowledge preference optimization (HKPO)*, which enables fine-grained hierarchical diagnosis by performing preference optimization at both hierarchical branch-level and segment-level for multimodal industrial anomaly detection.

3.1 Hierarchical Knowledge Modeling

In industrial anomaly detection, annotating fine-grained domain knowledge for each scenario is expensive and often impractical. Rather than requiring exhaustive case-by-case annotation, we use expert-informed general knowledge to guide the MLLM through step-by-step diagnostic reasoning that follows industrial practice. To this end, we construct hierarchical general knowledge that decomposes diagnosis into successive reasoning stages, enabling the model to infer the final anomaly result through hierarchical reasoning grounded in the given scenario.This hierarchical general knowledge is used solely to construct preference pairs during preference alignment and is not retrieved as external knowledge during inference.



Knowledge **Input:[Object, Defect]**

<Scratch>
Description: A scratch is a linear surface discontinuity on the *[object]* that appears as a narrow groove, score line, or abrasion, with a noticeable change in gloss, color, or reflectivity relative to adjacent areas.
- It is typically straight or gently curved, with a fairly consistent width, and may exhibit sharp edges, light scattering, or localized material removal.
- Scratches are commonly found on exposed, high-contact regions, such as edges, corners, raised features, or along likely handling and contact paths.
- Under directional lighting, the mark often becomes more prominent due to specular highlights or shadowing along the groove.
- Plausible causes include contact with hard surfaces, sliding friction, debris trapped between surfaces, or incidental tool contact during handling.
- Shallow scratches are mainly cosmetic, while deeper scratches can reduce protective layer thickness, initiate cracking, or compromise barrier, insulation, or sealing performance, depending on the application.

Figure 3: The extracted knowledge details the morphological characteristics of each defect category within a specific object type.

Knowledge Extraction. Knowledge extraction of anomaly detection primarily involves three core steps. First, we establish a knowledge template encompassing objects, defects, and morphological descriptions. Second, we organize the training data of Real-IAD (Wang et al., 2024) into object-defect pairs. For each pair, 10 images are sampled, and then GPT-5 is utilized to describe descriptive details such as material and texture. Finally, we also employ GPT-5 to summarize general characteristics of defects and formulate defect knowledge according to the template, get 67 items in total. An example of knowledge is provided in Figure 3. Notably, these generic knowledge items are manually curated in strict isolation from the evaluation dataset, ensuring that no data leakage occurs.

Hierarchical Knowledge Base Construction. We construct a hierarchical knowledge base by systematically associating objects, defects, and diagnostic criteria. Formally, for each object-defect pair (c, d), we define a specific knowledge item $k_{c,d}$ that encapsulates the diagnostic evidence. The knowledge base is structured as:

$$\mathcal{K} \;:=\; \{(c, d, k_{c,d}) \,|\, c \in \mathcal{C},\; d \in \mathcal{D}_c\}\,, \tag{1}$$

This formulation establishes a hierarchical diagnostic space that mirrors the human inspection workflow: identifying the object, thinking with defect knowledge, and making an anomaly decision. Crucially, to decouple semantic knowledge from specific object categories, we replace explicit object names within the text descriptions with a generic placeholder, *[object]*. This design ensures that the knowledge items focus exclusively on the morphological characteristics of the defect, such as texture or shape, without encoding category-specific information.

3.2 Hierarchical Knowledge Preference Optimization

In this section, we propose HKPO, which simulates the systematic diagnostic process of a human quality inspector through two complementary levels of granularity, i.e., branch-level and segment-level preference optimization. Here, the branch-level preference optimization adaptively guides the model to derive correct answers along the correct hierarchical reasoning path, while segment-level preference optimization guides the model to focus on correcting error-prone details within the reasoning process.

Vanilla Direct Preference Optimization (DPO). Direct Preference Optimization (DPO) Rafailov et al. (2023) aligns a policy model $\pi_\theta(y \mid x)$ with preference data relative to a reference model $\pi_{\text{ref}}(y \mid x)$. Given an input x and a preference pair (y^+, y^-), where y^+ is preferred over y^-, DPO defines the preference margin as:

$$\Delta_\theta(x, y^+, y^-) = \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\text{ref}}(y^+ \mid x)} - \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\text{ref}}(y^- \mid x)}. \tag{2}$$

The corresponding training objective is

$$\ell(\theta;\, x, y^+, y^-) = -\log \sigma\big(\beta\, \Delta_\theta(x, y^+, y^-)\big), \tag{3}$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\beta > 0$ is a temperature parameter.

Preferred and dispreferred hierarchical reasoning paths. In this work, our preferred and dispreferred reasoning trajectories are constructed along the hierarchical structure of knowledge base.

Let $\mathbf{r}(y) := (c, d, a)$ and $\mathbf{r}^* := (c^*, d^*, a^*)$ denote the predicted and ground-truth diagnostic trajectories, respectively. We consider three stages, $\mathcal{S} := \{\text{object}, \text{def knowledge}, \text{answer}\}$, corresponding to object identification, defect knowledge usage, and anomaly prediction. The hierarchical correctness indicator is defined as:

$$v_s(y) := \mathbb{I}\big[\mathbf{r}_{\le s}(y) = \mathbf{r}^*_{\le s}\big], \qquad s \in \mathcal{S}, \tag{4}$$

where $\mathbf{r}_{\le \text{obj}} = (c)$, $\mathbf{r}_{\le \text{def}} = (c, d)$, and $\mathbf{r}_{\le \text{ans}} = (c, d, a)$. Thus, a preferred reasoning path y^+ is fully correct if and only if $v_{\text{obj}}(y) = v_{\text{def}}(y) = v_{\text{ans}}(y) = 1$.

As for dispreferred paths, our goal is to learn to point out the failure pivot within the hierarchy. Therefore, dispreferred paths are constructed by sampling incorrect trajectories that deviate from the correct object-defect knowledge-answer path along the structured knowledge. For each dispreferred path, ADThinker assigns it a branch label $z \in \mathcal{Z}$, where $\mathcal{Z} = \{O, D, A\}$ is the set of branch labels, indicating the stage where it first departs from the correct trajectory.

Branch-level preference optimization. We construct preferred y^+ and dispreferred paths y_z^- for branch-level preference optimization, and the weight of each pair is modulated by the specific branch at which the dispreferred paths begins to deviate:

$$\mathcal{L}_{\text{Branch}}(\theta) \;:=\; \mathbb{E}_{(x,\, y^+,\, y_z^-)}\big[w_z^{(t)}\, \ell(\theta;\, x, y^+, y_z^-)\big]. \tag{5}$$

Here $w_z^{(t)}$ denotes the weight associated with branch z at training step t. Rather than using fixed branch weights, ADThinker updates them adaptively based on the optimization status of each branch. To mitigate training instability and catastrophic forgetting during joint optimization of hierarchical branches, we apply exponential moving averages (EMA) to smooth the branch-level statistics, leading to more stable convergence. Let $\hat{m}_z^{(t)}$ and $\hat{\ell}_z^{(t)}$ be the empirical average preference margin and average pairwise loss, respectively, for branch z over the current update interval at step t. Their EMA are maintained as

$$\bar{m}_z^{(t)} \;=\; \rho_m\, \bar{m}_z^{(t-1)} + (1-\rho_m)\, \hat{m}_z^{(t)}, \quad \bar{\ell}_z^{(t)} \;=\; \rho_\ell\, \bar{\ell}_z^{(t-1)} + (1-\rho_\ell)\, \hat{\ell}_z^{(t)}, \tag{6}$$

where $\rho_m, \rho_\ell \in [0, 1)$ are the momentum coefficients. The adaptive branch weights are computed from the smoothed branch statistics as

$$\tilde{w}_z^{(t)} \;=\; \lambda\big(1 - \bar{m}_z^{(t)}\big) + (1-\lambda)\bar{\ell}_z^{(t)}, \tag{7}$$

where $\lambda \in [0, 1]$ balances the contribution of the preference margin and the pairwise loss. Branches with smaller margins or larger losses receive larger weights, encouraging the model to focus on branches that are less well learned. The normalized branch weights are obtained as

$$w_z^{(t)} = N \cdot \frac{\text{clip}\Big(\tilde{w}_z^{(t)}\Big)}{\text{clip}\Big(\tilde{w}_O^{(t)}\Big) + \text{clip}\Big(\tilde{w}_D^{(t)}\Big) + \text{clip}\Big(\tilde{w}_A^{(t)}\Big)}, \tag{8}$$

where $N = |\mathcal{Z}|$ denotes the number of branches, ensuring unit-mean normalized weights.

Segment-level preference optimization. Although the branch-level objective in equation 5 reduces imbalance across hierarchical error branches, it still operates at the whole-response level. In many preference pairs, however, the difference between the chosen response y^+ and the rejected response y^- is limited to a few modified segments, while most other tokens are shared or less relevant to the preference decision (Fu et al., 2025). ADThinker therefore introduces a segment-level preference objective to emphasize these modified segments. Let $\mathcal{S}_c(y)$ denote the set of token positions in the modified segments, i.e., the spans where y^+ and y^- differ. We define the segment-aware sequence score as

$$\log \pi_\theta^{\text{seg}}(y \mid x) = \frac{1}{C(y)} \left(\sum_{i=1}^{|y|} \log p_\theta(y_i \mid x, y_{<i}) + \lambda_{\text{seg}} \sum_{i \in \mathcal{S}_c(y)} \log p_\theta(y_i \mid x, y_{<i}) \right), \tag{9}$$

where $\lambda_{\text{seg}} \ge 0$ controls the additional emphasis on modified segments, and

$$C(y) = |y| + \lambda_{\text{seg}} |\mathcal{S}_c(y)| \tag{10}$$

is the effective token weight used for normalization. This normalization prevents responses with different lengths or different numbers of modified tokens from receiving systematically different scores due only to scale. Based on equation 9, we define the segment-level preference margin as

$$\Delta_\theta^{\text{seg}}(x, y^+, y^-) \;:=\; \log \frac{\pi_\theta^{\text{seg}}(y^+ \mid x)}{\pi_{\text{ref}}^{\text{seg}}(y^+ \mid x)} \;-\; \log \frac{\pi_\theta^{\text{seg}}(y^- \mid x)}{\pi_{\text{ref}}^{\text{seg}}(y^- \mid x)}, \tag{11}$$

where $\pi_{\text{ref}}^{\text{seg}}$ is defined analogously using the reference model. The corresponding segment-level pairwise loss is

$$\ell^{\text{seg}}(\theta;\, x, y^+, y^-) \;:=\; -\log \sigma\big(\beta\, \Delta_\theta^{\text{seg}}(x, y^+, y^-)\big). \tag{12}$$

Overall HKPO objective. The above two objectives jointly facilitate the preference learning. By applying adaptive branch weights to the segment-level pairwise loss, HKPO achieves more fine-grained credit assignment that points out where to optimize:

$$\mathcal{L}_{\text{HKPO}}(\theta) \;:=\; \mathbb{E}_{(x,\, y^+,\, y_z^-)}\big[w_z^{(t)}\, \ell^{\text{seg}}(\theta;\, x,\, y^+,\, y_z^-)\big]. \tag{13}$$

In this formulation, $w_z^{(t)}$ dynamically adjusts the contribution of each error branch on the basis of its EMA margin and loss statistics, while ℓ^{seg} amplifies the preference signal on the modified segments within each preference pair.

4 Experiments

4.1 Experimental Setup

Training Data. We use Real-IAD (Wang et al., 2024) as the primary data source and construct **ADThinker-SFT-10K**, an instruction-tuning dataset for cold-start supervised fine-tuning. At this stage, each response contains only the final answer option, without generic knowledge or intermediate reasoning. We further curate a high-quality preference dataset, **ADThinker-Preference-1K**, for training HKPO. Specifically, high-quality samples are first selected from the SFT data through rule-based review and manual filtering, after which generic knowledge is injected into the corresponding samples to construct preference pairs for alignment. Additional details are provided in Appendix A.1 and Appendix A.2.

Evaluation Benchmark. We evaluate on MMAD (Jiang et al., 2024), an industrial anomaly detection benchmark containing 8,366 samples and 39,673 multiple-choice questions across seven subtasks: anomaly discrimination, defect classification, defect localization, defect description, defect analysis, object classification, and object analysis. Following the official protocol, we report results under both 0-shot and 1-shot settings. In the 1-shot setting, an additional randomly sampled normal image is provided as a reference. To ensure a strict separation between training and evaluation, ADThinker-SFT-10K and ADThinker-Preference-1K are constructed entirely independently of MMAD, with no overlap in samples, images, questions, or annotations.

4.2 Implementation Details

We instantiate ADThinker at a single model scale, denoted as ADThinker-8B, based on Qwen3-VL-8B. Training proceeds in two stages. In the first stage, we conduct cold-start supervised fine-tuning on ADThinker-SFT-10K for one epoch with a learning rate of 1×10^{-5}. In the second stage, we conduct HKPO training on ADThinker-Preference-1K for one epoch with a learning rate of 5×10^{-6} and an inverse temperature of $\beta = 0.1$. Across both stages, we use the AdamW optimizer, cosine learning rate decay, and a warmup ratio of 5%. The batch size is set to 4 in both stages, and training is performed on $4\times$ NVIDIA A100 (80GB) GPUs.

4.3 Main Results

On MMAD under the standard 1-shot setting, Table 1 organizes all baselines into three families: proprietary MLLMs up to GPT-5.5 (Anthropic, 2024; Achiam et al., 2023; Reid et al., 2024; OpenAI, 2026), general-purpose open-source VLMs from 7B to 76B (Liu et al., 2024; Chen et al., 2023; Bai et al., 2025b;a; Team et al., 2025; Zhu et al., 2025; Team, 2025), and anomaly-specialized open-source models (Gu et al., 2024; Chao et al., 2025; Zhao et al., 2025; Zhang et al., 2026; Peng et al., 2026).

Table 1: Performance comparison on MMAD under the standard 1-shot setting. Darker and lighter purple mark the best and second-best results, computed within the proprietary group and across all open-source rows. "-" denotes a metric that is not reported in the original paper. †: augmented with an external knowledge retrieval module.

Model	Scale	F1	Anomaly	Defect				Object		Average
			Dis.	Cls.	Loc.	Des.	Ana.	Cls.	Ana.	
Random Chance	-	-	50.00	25.00	25.00	25.00	25.00	25.00	25.00	28.57
Human (expert)	-	-	95.24	75.00	92.31	83.33	94.20	86.11	80.37	86.65
Human (ordinary)	-	-	86.90	66.25	85.58	71.25	81.52	89.58	69.72	78.69
Proprietary Models										
Claude-3.5-Sonnet (Anthropic, 2024)	-	41.92	60.14	60.14	48.81	67.13	79.11	85.19	79.83	68.36
Gemini-1.5-Pro (Reid et al., 2024)	-	57.60	68.63	60.12	58.56	70.38	82.46	89.20	82.25	73.09
GPT-4o (Achiam et al., 2023)	-	71.04	68.63	**65.80**	55.62	73.21	83.41	94.98	82.80	74.92
Gemini-2.5-Pro (Reid et al., 2024)	-	**79.84**	67.33	62.07	**64.73**	**76.45**	**84.91**	93.25	**83.96**	76.16
GPT-5.5 (OpenAI, 2026)	-	-	**69.95**	64.91	62.34	75.95	80.10	**95.23**	82.11	**76.20**
General-Purpose Open-Source Models (no anomaly-specific training)										
InternVL3 (Zhu et al., 2025)	8B	41.23	68.69	55.09	59.98	70.83	80.58	87.24	82.92	72.19
Keye-VL-Preview (Team, 2025)	8B	58.41	65.25	45.34	43.53	56.56	40.57	81.59	77.81	58.66
Qwen2.5-VL-Instruct (Bai et al., 2025b)	7B	72.49	71.10	56.02	60.69	64.13	78.26	91.49	83.67	72.19
MiMo-VL-SFT (Team et al., 2025)	7B	72.77	58.26	62.17	59.40	72.64	84.50	90.02	83.19	72.88
MiMo-VL-RL (Team et al., 2025)	7B	73.85	56.60	63.03	61.89	72.60	83.68	90.35	82.86	73.00
Qwen2.5-VL-Instruct (Bai et al., 2025b)	32B	68.19	70.88	58.29	62.40	65.38	83.16	76.89	84.91	71.70
LLaVA-NeXT (Liu et al., 2024)	34B	54.44	57.92	48.79	52.87	71.34	80.28	81.12	77.80	67.16
InternVL2 (Chen et al., 2023)	76B	64.40	68.25	54.22	56.66	66.30	80.47	86.40	82.92	70.75
Qwen2.5-VL-Instruct (Bai et al., 2025b)	72B	74.08	72.96	62.71	68.58	74.65	82.12	94.41	86.12	77.36
Qwen3-VL-Instruct (Bai et al., 2025a)	32B	78.67	79.29	68.71	67.83	75.64	81.79	94.23	85.18	78.95
Anomaly-Specialized Open-Source Models										
AnomalyGPT (Gu et al., 2024)	7B	76.68	65.57	27.49	27.97	36.86	32.11	29.84	35.82	36.52
AnomalyR1 (Chao et al., 2025)	3B	-	60.62	63.56	70.14	80.47	85.28	92.48	86.15	76.96
OmniAD (Zhao et al., 2025)	7B	-	68.80	**78.80**	**75.50**	67.20	86.40	**96.00**	86.40	79.90
ADSeeker† (Zhang et al., 2026)	7B	-	68.62	53.82	55.57	60.51	77.40	89.91	83.48	69.90
EAGLE (Peng et al., 2026)	7B	-	71.56	63.71	70.02	79.85	85.26	93.75	87.02	78.73
Qwen3-VL-Instruct (Baseline) (Bai et al., 2025a)	8B	74.11	73.08	62.33	60.34	69.15	80.09	92.53	84.01	74.51
ADThinker (Ours)	8B	**83.35**	**79.51**	66.31	72.62	**83.21**	**88.79**	93.57	**87.62**	**81.66**

Table 2: Generalization ability with domain knowledge (DK) on MMAD.

Model	w/o DK	w/ DK
Qwen3-VL-8B-Instruct	74.51	78.85
GraphRAG	–	74.49
ADThinker-8B (Ours)	**81.66**	**85.05**

Table 3: Zero-shot generalization results on SSGD.

Model	Anomaly Dis.	F1
Qwen3-VL-8B-Instruct	74.96	88.55
ADThinker-8B (Ours)	**84.95**	**90.90**

Gains come from training, not scale. With 8B parameters, ADThinker reaches 81.66% average and 83.35% F1, the best of all open-source entries. Under an identical decoding configuration it improves its own Qwen3-VL-8B backbone by 7.15% average and 9.24% F1, with the largest per-dimension gains on defect description (+14.06%) and defect localization (+12.28%); it further exceeds Qwen3-VL-32B by 2.71%/4.68% and Qwen2.5-VL-72B by 4.30%/9.27% while using 4–9× fewer parameters. Scale alone therefore does not explain the improvement: the same backbone family at 4× the size remains behind.

Closing the gap to proprietary models and humans. ADThinker-8B exceeds the strongest proprietary model, GPT-5.5 (76.20% average), by 5.46%, tops the best proprietary F1 (Gemini-2.5-Pro, 79.84%) by 3.51%, and outperforms ordinary humans (78.69%) by 2.97%, leaving a 4.99% gap to expert humans (86.65%). The only dimension where proprietary models still lead is object classification (GPT-5.5 95.23% vs. 93.57%), which probes generic recognition breadth rather than anomaly reasoning.

Comparison with anomaly-specialized models. ADThinker attains the best average score among anomaly-specialized models, ahead of OmniAD by 1.76%, EAGLE by 2.93%, AnomalyR1 by 4.70% and ADSeeker by 11.76%, and leads all of them by at least 7.95% on anomaly discrimination, the only detection-oriented subtask. They stay ahead only on region- and option-level subtasks (OmniAD: +12.49% defect classification, +2.88% localization, +2.43% object classification; EAGLE: +0.18% object classification), so these models are complementary to ours rather than strictly ordered.

Table 4: Comparison of different fine-tuning strategies on MMAD under the standard 1-shot setting. ADThinker-8B is obtained by applying SFT+HKPO on the 8B backbone.

Method	Size	Anomaly	Defect				Object		Average
		Dis.	Cls.	Loc.	Des.	Ana.	Cls.	Ana.	
Base	8B	73.08	62.33	60.34	69.15	80.09	92.53	84.01	74.51
+SFT	8B	74.51	65.05	70.49	82.62	87.72	92.16	86.47	79.86
+SFT+KTO	8B	76.52	64.59	70.89	82.66	86.36	92.49	86.45	79.93
+SFT+GRPO	8B	76.45	63.31	70.90	82.52	87.61	92.41	86.39	79.94
+SFT+DPO	8B	77.58	62.71	71.60	82.41	88.12	92.91	87.21	80.36
+SFT+HKPO (Ours)	8B	**79.51**	**66.31**	**72.62**	**83.21**	**88.79**	**93.57**	**87.62**	**81.66**

Table 5: Ablation of preference optimization branch designs on Qwen3-VL-8B-Instruct under the standard 1-shot setting.

SFT	DPO	HKPO (two-branches)	HKPO (three-branches)	Anomaly	Defect				Object		Average
				Dis.	Cls.	Loc.	Des.	Ana.	Cls.	Ana.	
✓				74.51	65.05	70.49	82.62	87.72	92.16	86.47	79.86
✓	✓			77.58	62.71	71.60	82.41	88.12	92.91	87.21	80.36
✓		✓		77.78	62.74	71.70	82.51	**88.81**	93.18	87.25	80.57
✓			✓	**79.51**	**66.31**	**72.62**	**83.21**	88.79	**93.57**	**87.62**	**81.66**

Table 6: Ablation of adaptive weighting strategies on Qwen3-VL-8B-Instruct under the standard 1-shot setting.

Branch Weight	Segment-level	Anomaly	Defect				Object		Average
		Dis.	Cls.	Loc.	Des.	Ana.	Cls.	Ana.	
None		77.58	62.71	71.60	82.41	88.12	92.91	87.21	80.36
None	✓	79.50	63.86	72.58	82.83	88.41	93.29	87.35	81.12
EMA		79.48	64.93	72.56	82.94	88.49	93.27	87.31	81.28
EMA	✓	**79.51**	**66.31**	**72.62**	**83.21**	**88.79**	**93.57**	**87.62**	**81.66**

4.4 Generalization to Domain Knowledge

We evaluate whether ADThinker learns a transferable capability for expert-knowledge utilization on MMAD under the standard 1-shot setting (Table 2). ADThinker is trained with dataset-independent *general knowledge* and tested with benchmark-specific *domain knowledge*. Without domain knowledge, ADThinker-8B reaches 81.66%, outperforming Qwen3-VL-8B-Instruct by 7.15%, Qwen3-VL-8B-Instruct+DK by 2.81%, and GraphRAG by 7.17%. With MMAD domain knowledge, ADThinker-8B+DK rises to 85.05%, improving by 3.39% over ADThinker-8B and 6.20% over Qwen3-VL-8B-Instruct+DK. This gain from domain knowledge indicates that ADThinker enables flexible knowledge reasoning by inferring correct answers from provided knowledge.

4.5 Zero-Shot Generalization

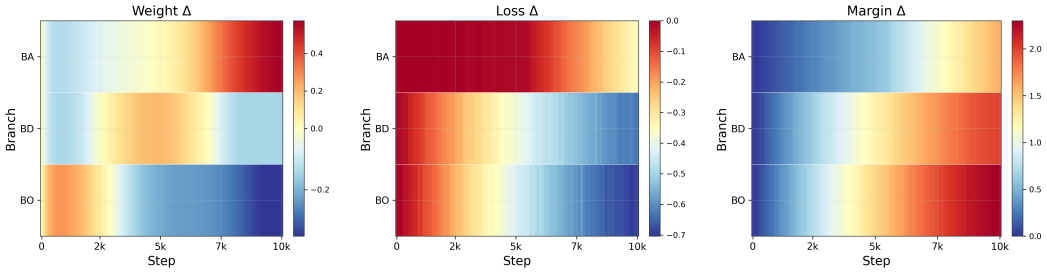
We further evaluate generalization on the entirely unseen SSGD dataset, whose object categories overlap with neither the training set nor the knowledge base. This provides a strict test of transfer to novel categories without object-specific memorization. As shown in Table 3, ADThinker-8B achieves 84.95% accuracy and 90.90% F1, outperforming Qwen3-VL-8B-Instruct by 9.99% and 2.35%, respectively. We attribute this result to the object-agnostic design of ADThinker, which learns general defect patterns and anomaly cues rather than category-specific priors.

4.6 Comparison of Different Fine-Tuning Strategies

On the 8B backbone under the standard 1-shot setting, we compare HKPO against SFT, KTO (Ethayarajh et al., 2024), GRPO (Shao et al., 2024) and DPO (Rafailov et al., 2023) (Table 4). HKPO achieves the best average score of 81.66%, exceeding them by 1.80%, 1.73%, 1.72% and 1.30%, respectively, and remains best or competitive across individual task categories.

4.7 Ablation & Analysis

Effect of Preference Optimization Branch Design. Table 5 ablates the branch design. Preference optimization already helps beyond SFT (79.86%→80.36% for vanilla DPO). Within HKPO, the two-branch variant yields only a modest gain (80.57%), whereas the three-branch variant is best



(a) Weight variations. (b) Loss variations. (c) Margin variations.

Figure 4: Branch-wise variations in weights, losses, and margins over training steps. (a) Variations in the weights assigned to samples from different branches throughout training. (b) Corresponding variations in the losses of samples from different branches. (c) Variations in the branch-wise preference margins over training.

(81.66%) on most subtasks, indicating that a finer-grained branch design aligns better with the diverse objectives of MMAD.

Effect of Adaptive Weighting Strategy. Table 6 evaluates the effect of EMA-based branch weighting and segment-level weighting. Both modules individually outperform the unweighted baseline, achieving average scores of 81.28% and 81.12%, respectively, compared to 80.36%. Combining the two further improves the average to 81.66%. The gains are most pronounced on challenging tasks such as anomaly discrimination (+1.93%), defect classification (+3.60%), and defect localization (+1.02%), where ambiguous patterns and fine-grained distinctions make optimization more difficult. Consistent improvements are also observed on easier tasks with stronger baselines (e.g., defect description, object classification), demonstrating the complementarity and robustness of the two modules.

Emergent Curriculum Across Branches. Figure 4 shows that HKPO induces a clear curriculum across branches. In Figure 4(a), BO receives higher weights early, BD is emphasized in the middle stage, and BA becomes dominant later in training. This shift is better explained by the margin dynamics in Figure 4(c) than by the loss trends in Figure 4(b): BO quickly attains a large preference margin, while BA remains the least separated for most of training. As a result, adaptive weighting shifts the optimization focus from BO to BA, with BD as an intermediate case. This suggests that HKPO prioritizes branches with weaker preference separation rather than simply larger losses, yielding a branch-level curriculum without explicit scheduling.

Effect of segment-level alignment. As shown in Figure 5, the log-probability gap on differential segments widens steadily throughout HKPO training. The corresponding statistics on non-differential segments, reported in Appendix B.4, stay almost flat over the same schedule, so the separation is not a global shift of the sequence likelihood. This contrast indicates that segment weighting sharpens discrimination precisely on the spans that carry branch-specific supervision, rather than uniformly across the response, thereby improving preference separation without distorting the shared prefix.

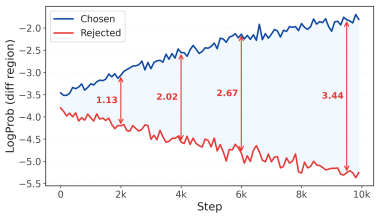


Figure 5: Log-probability gap on *differential* segments.

5 Conclusion

In this work, we bridge the gap between the capabilities of MLLMs and the rigorous demands of real-world industrial quality inspection. Departing from the binary defect detection paradigm prevalent in existing literature, we propose Hierarchical Knowledge Preference Optimization (HKPO), which emulates the cognitive logic of human inspectors. By abstracting object and defect morphologies into generalized structured knowledge, HKPO enables multi-granularity reasoning: identifying target objects, analyzing defect causality, and rendering precise judgments. Extensive experiments validate its efficacy, showing that HKPO significantly boosts the diagnostic ability of MLLMs.

ETHICS STATEMENT

This work studies visual quality inspection on publicly available industrial anomaly detection benchmarks (Real-IAD, MMAD and SSGD). No human subjects are involved, and none of the data contains personally identifiable or otherwise sensitive information. All datasets and pre-trained models are used in accordance with their original licenses and intended research use. The intended positive impact of this work is to improve inspection efficiency, support defect analysis, and reduce the burden of manual inspection in resource-constrained manufacturing settings. At the same time, model behavior may vary across products, environments, and imaging conditions, and occasional prediction errors or reasoning inconsistencies may still arise in challenging or previously unseen cases. As discussed in Appendix C and Appendix D, the system should therefore be deployed as an assistive tool that supports human inspectors under appropriate oversight, rather than as a fully autonomous decision-maker in safety-critical scenarios.

REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure that our results can be reproduced. The hierarchical knowledge modeling procedure and both levels of the preference optimization objective are specified in Section 3, including all loss formulations. The construction pipelines of the two datasets used for training, ADThinker-SFT-10K and ADThinker-Preference-1K, are described in Appendix A.1 and Appendix A.2, covering the source data, the generation and randomization procedure, the mask-based verification step, and the boundary-sample selection threshold. Backbones, training schedules, optimizer settings, and all remaining hyper-parameters are reported in the implementation details of Section 4. Evaluation follows the standard protocols of the public MMAD and SSGD benchmarks without modification, so that the reported numbers are directly comparable to previously published results. The source code, the constructed knowledge base, and the two datasets will be released to support full reproduction of the experiments.

AI USE STATEMENT

In this work, generative AI tools were used in two distinct roles. First, as an explicit component of our research methodology: GPT-5 was used to generate the templated question–answer annotations of ADThinker-SFT-10K and to draft the entries of the hierarchical knowledge base. These uses are described in Section 3 and Appendix A.1; all generated annotations were subsequently filtered by an automatic verification step against pixel-accurate segmentation masks and reviewed by the authors. Second, as a general-purpose writing aid: large language models were used for grammar correction and for polishing the phrasing of the manuscript. Generative AI tools were not used to generate research ideas, to design the method, to produce experimental results, or to retrieve or summarize related work. All AI-assisted content was reviewed by the authors, and we take full responsibility for the final content of this work, including all text, claims, and artifacts.

REFERENCES

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

Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2024. URL `https://www.anthropic.com/news/claude-3-family`. Claude-3 Model Card.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9592–9600, 2019.

Yuhao Chao, Jie Liu, Jie Tang, and Gangshan Wu. Anomalyr1: A grpo-based end-to-end mllm for industrial anomaly detection, 2025. URL https://arxiv.org/abs/2504.11914.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In *Forty-first International Conference on Machine Learning*, 2024.

Jinlan Fu, Shenzhen Huangfu, Hao Fei, Xiaoyu Shen, Bryan Hooi, Xipeng Qiu, and See-Kiong Ng. Chip: Cross-modal hierarchical direct preference optimization for multimodal llms. 2025.

Bin-Bin Gao. Learning to detect multi-class anomalies with just one normal image prompt. In *European Conference on Computer Vision*, pp. 454–470. Springer, 2024.

Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. Anomalygpt: Detecting industrial anomalies using large vision-language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 1932–1940, 2024.

Wei Guan, Jun Lan, Jian Cao, Hao Tan, Huijia Zhu, and Weiqiang Wang. Emit: Enhancing mllms for industrial anomaly detection via difficulty-aware grpo, 2025. URL https://arxiv.org/abs/2507.21619.

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

Xi Jiang, Jian Li, Hanqiu Deng, Yong Liu, Bin-Bin Gao, Yifeng Zhou, Jialin Li, Chengjie Wang, and Feng Zheng. Mmad: A comprehensive benchmark for multimodal large language models in industrial anomaly detection. *arXiv preprint arXiv:2410.09453*, 2024.

Xurui Li, Ziming Huang, Feng Xue, and Yu Zhou. Musc: Zero-shot industrial anomaly classification and segmentation with mutual scoring of the unlabeled images. In *The Twelfth International Conference on Learning Representations*, 2024.

Yanhui Li, Yunkang Cao, Chengliang Liu, Yuan Xiong, Xinghui Dong, and Chao Huang. Iad-r1: Reinforcing consistent reasoning in industrial anomaly detection, 2025. URL https://arxiv.org/abs/2508.09178.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

OpenAI. GPT-5.5 system card. https://deploymentsafety.openai.com/gpt-5-5/gpt-5-5.pdf, 2026. Accessed: 2026-09-25.

Xiaomeng Peng, Xilang Huang, and Seon Han Choi. EAGLE: Expert-augmented attention guidance for tuning-free industrial anomaly detection in multimodal large language models, 2026. URL https://arxiv.org/abs/2602.17419.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Core Team, Zihao Yue, Zhenru Lin, Yifan Song, Weikun Wang, Shuhuai Ren, Shuhao Gu, Shicheng Li, Peidian Li, Liang Zhao, Lei Li, Kainan Bao, Hao Tian, Hailin Zhang, Gang Wang, Dawei Zhu, Cici, Chenhong He, Bowen Ye, Bowen Shen, Zihan Zhang, Zihan Jiang, Zhixian Zheng, Zhichao Song, Zhenbo Luo, Yue Yu, Yudong Wang, Yuanyuan Tian, Yu Tu, Yihan Yan, Yi Huang, Xu Wang, Xinzhe Xu, Xingchen Song, Xing Zhang, Xing Yong, Xin Zhang, Xiangwei Deng, Wenyu Yang, Wenhan Ma, Weiwei Lv, Weiji Zhuang, Wei Liu, Sirui Deng, Shuo Liu, Shimao Chen, Shihua Yu, Shaohui Liu, Shande Wang, Rui Ma, Qiantong Wang, Peng Wang, Nuo Chen, Menghang Zhu, Kangyang Zhou, Kang Zhou, Kai Fang, Jun Shi, Jinhao Dong, Jiebao Xiao, Jiaming Xu, Huaqiu Liu, Hongshen Xu, Heng Qu, Haochen Zhao, Hanglong Lv, Guoan Wang, Duo Zhang, Dong Zhang, Di Zhang, Chong Ma, Chang Liu, Can Cai, and Bingquan Xia. Mimo-vl technical report, 2025. URL https://arxiv.org/abs/2506.03569.

Kwai Keye Team. Kwai keye-vl technical report, 2025. URL https://arxiv.org/abs/2507.01949.

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

Chengjie Wang, Xi Jiang, Bin-Bin Gao, Zhenye Gan, Yong Liu, Feng Zheng, and Lizhuang Ma. Softpatch+: Fully unsupervised anomaly classification and segmentation. *Pattern Recognition*, 161:111295, 2025.

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

Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8330–8339, 2021.

Peijian Zeng, Feiyan Pang, Zhanbo Wang, and Aimin Yang. Lr-iad:mask-free industrial anomaly detection with logical reasoning, 2025. URL https://arxiv.org/abs/2504.19524.

Jinjin Zhang, Guodong Wang, Yizhou Jin, and Di Huang. Towards training-free anomaly detection with vision and language foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15204–15213, 2025.

Kai Zhang, Zekai Zhang, Xihe Sun, Anpeng Wang, Jingmeng Nie, Qinghui Chen, Han Hao, Jianyuan Guo, and Jinglin Zhang. Adseeker: A knowledge-grounded reasoning framework for industry anomaly detection and reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21379–21388, 2026.

Shifang Zhao, Yiheng Lin, Lu Han, Yao Zhao, and Yunchao Wei. Omniad: Detect and understand industrial anomaly via multimodal reasoning, 2025. URL https://arxiv.org/abs/2505.22039.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Dataset Construction Details

A.1 ADThinker-SFT-10K

Existing industrial anomaly detection datasets (Bergmann et al., 2019; Wang et al., 2024) primarily focus on object class labels, visual mask annotations, and defect class labels. However, these datasets often lack contextual annotations, which limits their usefulness for multimodal understanding.

To address this limitation, we introduce ADThinker-SFT-10K, a carefully curated instruction-tuning dataset designed for industrial visual anomaly detection. ADThinker-SFT-10K preserves the authentic characteristics of industrial anomalies while further providing precisely localized and domain-specific contextual annotations that capture subtle differences among manufacturing defects across diverse industrial sectors. ADThinker-SFT-10K is built through a novel multi-stage generation pipeline that systematically transforms and enriches real industrial defect images from the Real-IAD dataset (Wang et al., 2024). For the initial generation of QA pairs, we adopt a templated question-answer generation framework and use GPT-5 with carefully designed prompts containing explicit instructions. We then apply a randomization process to the answer options by shuffling both the answer identifiers (A, B, C, D) and their corresponding textual content. This step ensures that models trained on the dataset learn the semantic content of anomaly descriptions instead of memorizing positional patterns of correct answers. To ensure annotation quality, we further introduce a secondary verification process based on the pixel-accurate segmentation masks provided in the Real-IAD dataset. This process is used to validate the spatial accuracy of the defect localization descriptions generated by GPT-5, thereby improving the consistency between the annotations and the true anomaly boundaries.

In total, five distinct types of question-answer pairs are generated for each image, closely following the established paradigm of MMAD. ADThinker-SFT-10K therefore provides a comprehensive and extensible resource for advancing the state of the art of visual anomaly detection in industrial inspection and quality control.

A.2 ADThinker-Preference-1K

Building on ADThinker-SFT-10K, we developed a systematic method to extract challenging samples for Preference Optimization fine-tuning. We first used the model previously fine-tuned on ADThinker-SFT-10K to generate predictions for the entire dataset. We then applied a strict filtering criterion and retained only samples for which the probability assigned to the ground-truth answer showed only a small deviation from random guessing. Specifically, we kept two types of samples: those with probabilities slightly above the level of random guessing, which indicate correct but uncertain predictions, and those with probabilities slightly below that level, which correspond to incorrect but potentially recoverable predictions. To implement this selection strategy, we set a threshold of 0.1 to define "marginal deviation." This filtering process produced 1K selected instruction data samples, which form a concentrated set of boundary cases well suited for preference-based optimization.

B Additional Experiments

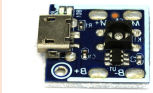
B.1 Zero-shot Performance

As shown in Table 7, our **ADThinker-8B** delivers the strongest zero-shot performance on MMAD, achieving the best average score of **78.26%**. It surpasses the strongest baseline, Qwen2.5-VL-72B, by **3.86%**, despite using a much smaller model scale. Moreover, ADThinker-8B obtains the best results on six of the seven task dimensions, covering both anomaly-level and defect-level understanding, and reaches the second-best score on object analysis. In particular, its clear advantages in defect localization, description, and analysis highlight the effectiveness of our framework in zero-shot industrial anomaly understanding.

B.2 Impact of Domain Knowledge

Domain knowledge in MMAD provides an effective way to further improve MLLM performance in the 1-shot setting. As shown in Table 8, our **ADThinker-8B** achieves the best overall average score of **85.05%**, which is competitive with expert human performance (**86.65%**) and significantly

Object Branch



Q: You are a helpful assistant. You are an industrial inspector who checks products by images. You should judge whether there is a defect in the query image and answer the questions about it. Answer with the option's letter from the given choices directly.
Following is the query image: <image>
Following is the question:
Question 1: Is there any defect in the object?
A. No. B. Yes.

A+:<context> <Missing Parts> Description: One or more required features on the [object] are absent, leaving a clear void where a component or structural segment should be present. Visual cues include empty footprints or pads, unpopulated holes/slots, missing pins or terminals, absent bridges or links, and abrupt discontinuities in an otherwise repetitive or symmetric layout. The affected area often shows sharp outline boundaries, exposed base material, or unused attachment points that contrast with neighboring populated regions. Missing elements may occur at interfaces, along edges, or at designated mounting/connection locations where engagement or continuity is expected. Plausible causes include assembly omission, break-off during handling, or incomplete attachment resulting in detachment. This condition can lead to open circuits, loss of mechanical retention, poor mating/engagement, and reduced reliability or complete functional failure.</context>
<answer>B</answer>

A-:<context> <Missing Parts> Description: This defect is defined by the absence of one or more required components or segments on the [object], producing an obvious void where a feature is expected. Visual cues include missing geometry, open pads/holes/slots, interrupted outlines, and asymmetry relative to the nominal layout, often accompanied by exposed underlying surfaces or attachment points. The anomaly commonly appears at interfaces and functional regions such as connectors, fastener locations, edges, or repeated feature arrays where uniformity is expected. The missing area typically has clean boundaries or fracture-like edges rather than the continuous contours seen in conforming parts. Plausible causes include incomplete assembly, component detachment, breakage during handling, or material loss. Functional risk is high, as the absence can prevent proper engagement, compromise structural integrity, create open circuits, or lead to unstable operation and downstream failures.</context>
<answer>A</answer>

Figure 6: Illustration of the object branch in ADThinker-Preference-1K. Given an image and a multiple-choice question asking whether the queried object is defective, the preferred and rejected responses are grounded in contextual knowledge describing different object categories, which leads to different final decisions.

Table 7: Performance comparison in MMAD under the 0-shot setting. **Bold** type indicates the best performance, underlined type indicates the second best performance.

Model	Scale	Anomaly	Defect				Object		Average
		Dis.	Cls.	Loc.	Des.	Ana.	Cls.	Ana.	
Gemini-1.5-flash	-	58.43	49.93	53.11	63.07	82.83	-	-	68.58
LLaVA-NEXT-Interleave	7B	58.39	36.98	48.98	51.51	66.64	-	-	60.04
InternLM-XComposer2-VL	7B	58.33	43.10	54.56	57.84	75.30	-	-	62.78
Cambrian	8B	55.60	32.53	35.39	43.46	49.14	78.15	67.22	51.64
InternVL3	8B	64.67	50.91	60.06	65.49	77.16	82.01	83.58	69.13
Keye-VL-Preview	8B	61.85	51.37	52.94	63.79	43.99	85.05	80.10	62.73
Deepseek-vl2-tiny	1B	50.60	43.21	54.97	65.52	75.66	88.64	76.88	65.07
Deepseek-vl2-small	2.8B	62.33	49.69	55.16	67.84	79.14	93.01	83.09	70.04
Qwen2.5-VL-Instruct	7B	60.45	50.16	57.80	60.73	75.31	93.24	84.96	68.95
MiMo-VL-SFT	7B	64.80	55.90	54.52	68.29	83.39	91.29	84.87	71.87
MiMo-VL-RL	7B	66.06	57.78	56.95	69.48	83.72	91.63	84.97	72.94
LLaVA-NeXT	34B	60.25	51.57	55.49	71.62	80.43	-	-	68.45
Qwen2.5-VL-Instruct	72B	66.22	57.99	62.95	72.21	81.01	93.92	**86.51**	74.40
InternVL2	76B	64.30	51.19	54.20	63.46	79.92	89.34	83.48	69.41
Qwen3-VL-Instruct	8B	69.33	58.38	62.95	66.60	75.76	92.52	85.39	72.99
ADThinker-8B (Ours)	8B	**72.47**	**60.99**	**71.54**	**76.89**	**85.43**	**94.10**	86.43	**78.26**

surpasses all other MLLM baselines. Moreover, ADThinker-8B consistently ranks first on all seven task dimensions, demonstrating the strong capability of our framework to leverage domain knowledge

Defect Branch



Q: You are a helpful assistant. You are an industrial inspector who checks products by images. You should judge whether there is a defect in the query image and answer the questions about it. Answer with the option's letter from the given choices directly.
Following is the query image: <image>
Following is the question:
Question 1: Is there any defect in the object?
A. Yes. B. No.

A+:<context> <Contamination / Residue> Description: This defect is characterized by foreign material on the surface that changes the local color, texture, or reflectivity compared with the surrounding finish. Typical visual cues include hazy or cloudy patches, glossy smears, dark streaks, fingerprint-like patterns, ring-shaped stains, or fine dust films; in some cases, discrete particles may be embedded or loosely attached. It is most often observed on exposed outer surfaces, edges, recesses, and around interfaces where material can accumulate. Plausible causes include handling transfer (skin oils), airborne dust, packaging contact, or residual films from prior cleaning or processing steps. Contamination can interfere with downstream bonding or coating, reduce contact performance, obscure markings or inspection features, and increase the risk of corrosion or long-term reliability degradation.</context>
<answer>A</answer>

A-:<context> <Pit / Void> Description: Localized depressions or pinhole-like cavities are present on the surface, interrupting an otherwise smooth, continuous finish. The affected areas appear as dark, shadowed indentations or small holes with sharp to irregular rims, and may reveal a different tone or texture from underlying material. Features are often isolated but can occur in small clusters, commonly on flat faces, edges, or around geometric transitions where surface continuity is most apparent. Plausible causes include entrapped air or contamination, localized material shrinkage, or minor mechanical impact after forming. This condition degrades cosmetic appearance and, when located on functional interfaces, can reduce sealing effectiveness, mechanical robustness, or insulation clearance. Severity increases with pit depth, density, and proximity to critical contact or sealing regions..</context>
<answer>B</answer>

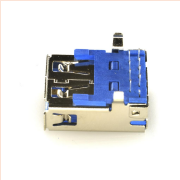
Figure 7: Illustration of the defect branch in ADThinker-Preference-1K. The task requires identifying the defect category of the queried object, and the response pair contrasts a correct defect-specific explanation against an incorrect alternative.

Table 8: Performance comparison in MMAD with domain knowledge under the 1-shot setting. **Bold** type indicates the best performance, underlined type indicates the second best performance.

Model	Scale	Anomaly	Defect				Object		Average
		Dis.	Cls.	Loc.	Des.	Ana.	Cls.	Ana.	
Human (expert)	-	95.24	75.00	92.31	83.33	94.20	86.11	80.37	86.65
Human (ordinary)	-	86.90	66.25	85.58	71.25	81.52	89.58	69.72	78.69
Qwen2.5-VL-Instruct	7B	70.74	68.11	60.58	72.64	82.59	97.09	84.79	76.65
MiMo-VL-SFT	7B	69.10	71.27	63.78	77.93	85.08	96.64	85.39	78.46
MiMo-VL-RL	7B	66.87	71.93	65.65	77.19	85.91	96.26	84.30	78.30
InternVL2	26B	68.64	67.32	53.81	70.84	82.18	93.81	83.31	74.27
LLaVA-NeXT	34B	56.72	68.22	57.36	73.12	82.24	91.78	81.35	72.97
InternVL2	40B	70.01	70.09	56.89	73.29	83.26	96.50	84.41	76.35
Qwen2.5-VL-Instruct	72B	71.63	73.85	69.11	79.18	83.87	96.81	87.04	80.21
InternVL2	76B	66.68	70.95	60.57	75.32	82.71	91.71	85.29	76.18
Qwen3-VL-Instruct	8B	75.13	74.99	62.96	76.13	81.72	96.33	84.68	78.85
ADThinker-8B (Ours)	8B	**81.12**	**78.42**	**72.57**	**86.67**	**90.69**	**97.82**	**88.06**	**85.05**

for comprehensive anomaly understanding. We also observe that the benefit of domain knowledge is model-dependent. While smaller and medium-scale models often obtain larger improvements, stronger large-scale models generally show smaller gains, implying that they already possess richer internal prior knowledge and are therefore less dependent on externally provided domain expertise.

Answer Branch



Q: You are a helpful assistant. You are an industrial inspector who checks products by images. You should judge whether there is a defect in the query image and answer the questions about it. Answer with the option's letter from the given choices directly.
Following is the query image: <image>
Following is the question:
Question 1: Is there any defect in the object?
A. Yes. B. No.

A+:<context> <Missing Parts> Description: This defect is defined by the absence of one or more required components or features on the [object], producing a clear void where material or structure is expected. Visual cues include missing terminals or pins, absent plastic ribs/bridges, missing links, or missing structural segments, often seen as empty slots, unpopulated pads, or abrupt discontinuities in an otherwise regular pattern. The affected region commonly appears asymmetric relative to the nominal geometry, with exposed cavities, incomplete outlines, or misaligned neighboring features due to the missing element. Missing parts may occur at interfaces, along edges, or at repeated feature arrays where uniformity is expected. Plausible causes include incomplete assembly, component loss during handling, or break-off from mechanical stress. The condition typically degrades fit and engagement, can create open circuits or intermittent contact, and increases the likelihood of early mechanical failure or unreliable operation.</context>
<answer>B</answer>
A-:<context> <Missing Parts> Description: This defect is defined by the absence of one or more required components or features on the [object], producing a clear void where material or structure is expected. Visual cues include missing terminals or pins, absent plastic ribs/bridges, missing links, or missing structural segments, often seen as empty slots, unpopulated pads, or abrupt discontinuities in an otherwise regular pattern. The affected region commonly appears asymmetric relative to the nominal geometry, with exposed cavities, incomplete outlines, or misaligned neighboring features due to the missing element. Missing parts may occur at interfaces, along edges, or at repeated feature arrays where uniformity is expected. Plausible causes include incomplete assembly, component loss during handling, or break-off from mechanical stress. The condition typically degrades fit and engagement, can create open circuits or intermittent contact, and increases the likelihood of early mechanical failure or unreliable operation..</context>
<answer>A</answer>

Figure 8: Illustration of the answer branch in ADThinker-Preference-1K. Both responses share the same defect-related context, but differ in the final answer option, isolating preference learning at the answer selection level.

B.3 Comparison with State-of-the-Art Closed-Source Models

As shown in Table 9, our **ADThinker-8B** consistently achieves the best average performance across all three settings, including 0-shot, 1-shot, and 1-shot with domain knowledge. Specifically, it surpasses Gemini-2.5-pro by **3.40%** in the 0-shot setting, by **5.50%** in the 1-shot setting, and by **3.96%** in the 1-shot with domain knowledge setting, achieving the best overall result of **85.05%**. Moreover, ADThinker-8B shows clear advantages on most fine-grained subtasks. In particular, it demonstrates outstanding performance on defect-related tasks, including defect classification, localization, description, and analysis. It also achieves the best results on anomaly discrimination and object analysis across all three settings. These results verify the strong competitiveness of our method against state-of-the-art closed-source MLLMs.

B.4 Segment-Level Log-Probability Statistics

Figure 9 complements Figure 5 in Section 4.7: it reports the segment-level log-probability statistics on *non-differential* segments collected during HKPO training. While the differential segments exhibit a steadily widening log-probability gap, the non-differential segments remain weakly separated throughout training, confirming that the segment-level weighting concentrates the preference signal on branch-specific spans instead of shifting the likelihood of the entire response.

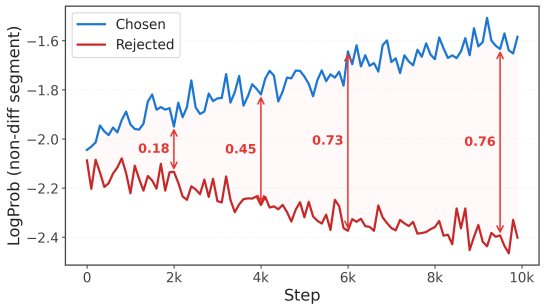


Figure 9: Segment-level log-probability statistics on non-differential segments during HKPO training. Unlike the differential segments in Figure 5, the separation here remains limited throughout training.

Table 9: Comparison with state-of-the-art closed-source models. **Bold** type indicates the best performance.

Setting	Model	Anomaly	Defect				Object		Average
		Dis.	Cls.	Loc.	Des.	Ana.	Cls.	Ana.	
0-shot	GPT-4o	63.50	52.97	53.62	68.69	77.24	**94.95**	86.34	71.04
	Gemini-2.5-pro	68.96	59.20	61.19	74.91	83.92	93.55	83.40	75.02
	ADThinker-8B (Ours)	**72.47**	**60.99**	**71.54**	**76.89**	**85.43**	94.10	**86.43**	**78.26**
1-shot	GPT-4o	68.63	65.80	55.62	73.21	83.41	**94.98**	82.80	74.92
	Gemini-2.5-pro	67.73	62.07	64.73	76.45	84.91	93.25	83.96	76.16
	ADThinker-8B (Ours)	**79.51**	**66.31**	**72.62**	**83.21**	**88.79**	93.57	**87.62**	**81.66**
1-shot with domain knowledge	GPT-4o	69.36	65.41	55.26	78.78	78.92	95.40	86.59	75.68
	Gemini-2.5-pro	74.81	73.55	67.52	82.39	87.49	96.33	85.54	81.09
	ADThinker-8B (Ours)	**81.12**	**78.42**	**72.57**	**86.67**	**90.69**	**97.82**	**88.06**	**85.05**

C Limitation and Future Work

Although ADThinker-8B achieves strong performance and establishes a new state of the art among both open-source and closed-source MLLMs on MMAD, it still does not fully match the performance of human experts. As shown in Table 8, under the 1-shot with knowledge setting, the average score of ADThinker-8B is 85.05%, which remains below the expert score of 86.65%. This gap indicates that the model is still limited in handling challenging industrial anomaly cases that require subtle visual discrimination, rich domain-specific knowledge, and expert-level reasoning derived from long-term experience. Therefore, at the current stage, ADThinker should be regarded as a strong assistant for industrial anomaly inspection rather than a complete substitute for human experts, particularly in safety-critical scenarios. In future work, we will focus on narrowing the gap to expert performance through several directions. We will explore larger-scale and higher-quality domain-specific supervision to improve the understanding of fine-grained defect patterns. We will also develop more effective preference optimization strategies and reasoning supervision signals to improve decision making on ambiguous and difficult cases. In addition, we will investigate deeper integration of external industrial knowledge and expert feedback so that the model can generalize better to diverse real-world inspection settings. We believe that these directions are important for developing more reliable industrial anomaly understanding systems with stronger expert-level capability.

D Broader Impacts

This work advances industrial anomaly understanding by teaching MLLMs to perform hierarchical reasoning in a manner more consistent with quality inspectors, enabling more accurate and interpretable decision-making in complex inspection scenarios. Positive impacts include improving inspection efficiency, supporting defect analysis, and reducing the burden of manual inspection in resource-constrained settings. Such capabilities may benefit a range of applications, including industrial quality control, infrastructure maintenance, and safety monitoring. At the same time, as with other learning-based systems, model performance may vary across different products, environments,

and imaging conditions. In practical deployment, occasional prediction errors or reasoning inconsistencies may still arise, particularly in challenging or previously unseen cases. These limitations suggest that the system is best used as an assistive tool to support human inspectors rather than as a fully autonomous decision-maker. We therefore encourage careful evaluation in real-world settings and responsible deployment with appropriate human oversight.