

What Should Be Fixed? Rule-Grounded Corrective Actions for Reworkable Logical Anomalies in Industrial Inspection

Haitao Zhang¹, Bosheng Chen¹, Haote Xu^{2,3}, Xiaotong Tu¹, Yue Huang¹, Xinghao Ding¹

¹Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, Xiamen University

²Zhejiang Expressway Co., Ltd.

³School of Transportation Science and Engineering, Beihang University

haitaozhang0829@gmail.com, chenbosheng@stu.xmu.edu.cn

hotxu2025@163.com, xttu@xmu.edu.cn, yhuang2010@xmu.edu.cn, dxh@xmu.edu.cn

Abstract

Logical anomalies in industrial inspection are rule violations: an object count is wrong, an item is misplaced, a component type does not match, or a required relation is broken. Existing anomaly understanding systems usually stop at diagnosis, answering whether an image is anomalous, where the anomaly is, or why it violates a rule. Reworkable inspection workflows require the next question: *What should be fixed?* We introduce an industry-oriented benchmark and training pipeline for rule-grounded corrective actions. The task asks a model to produce an operator-facing action that supports one of three downstream decisions: no action, rework, or reject/manual review. This setting exposes a *Diagnosis-Action Gap*: identifying an abnormal object does not imply knowing the valid, minimal action that restores normality. We construct LOCO-Correct from MVTec LOCO and VID-AD-Correct from VID-AD, adding correction-oriented annotations that connect normal rules, visual evidence, reparability, corrective actions, and verification targets. The datasets support multiple-choice correction selection, structured JSON action prediction, and open-ended correction generation. We further train **ReworkVLM**, a Qwen3-VL-based model optimized with RACO-GRPO, which combines semantic correctness, trace-answer alignment, and minimal-edit rewards. Experiments on both benchmarks show that correction-oriented evaluation reveals failures hidden by diagnosis alone, including wrong action direction, unnecessary repair, and non-minimal edits, while ReworkVLM improves structured corrective success and open-ended correction quality over general and anomaly-specific VLM baselines.

1 Introduction

Logical anomalies in industrial inspection are not arbitrary visual defects. They are violations of operating rules: a product contains the wrong number

of objects, an item appears in the wrong position, a required type is missing, a length is inconsistent, or a relation among parts is broken. This rule structure makes logical anomaly inspection a language and reasoning problem as much as a visual recognition problem. In deployed inspection workflows, however, diagnosis is only an intermediate step. Operators also need to know whether the sample can be reworked and what minimal correction would restore the normal rule.

As shown in Figure 1, consider a breakfast box whose normal rule requires a fixed fruit composition, while the observed state contains an extra orange and is missing a peach. A detector may flag the image as anomalous, and an explanation model may say that the fruit composition rule is violated. The useful correction is more specific: *replace one extra orange with one peach*. Removing an orange is incomplete; adding a peach may overfill the box; replacing a peach with an orange has the right action type but the wrong direction. The example shows that recognizing the abnormal object is not the same as knowing how to fix it.

We call this mismatch the **Diagnosis-Action Gap**. In logical anomaly correction, a valid output must connect visual evidence, a normal rule, and an executable edit. The correction must restore the violated rule while changing as little as possible. This requirement separates four levels of understanding: anomaly detection, anomaly explanation, correction recommendation, and rule-restoring verification. The central question is therefore not only “what is wrong?” but “what should be fixed?”

The practical setting we target is a reworkable inspection workflow rather than a fully deployed autonomous repair system. A model output is useful only if it supports one of three downstream decisions: *no action*, *rework*, or *reject/manual review*. We therefore evaluate not only anomaly diagnosis, but whether the model produces an operator-facing correction that is rule-grounded, minimal, and veri-

fiable. Minimality is economically meaningful: it can reduce false rework, avoid unnecessary product rejection, and give operators a smaller action to inspect.

We introduce Actionable Visual Anomaly Understanding, a multimodal task for rule-grounded correction in industrial logical anomaly inspection. Given an image and an inspection question, a model predicts whether the image is normal or anomalous and, when anomalous, recommends a minimal rule-restoring action. We build LOCO-Correct from MVTec LOCO (Bergmann et al., 2022) and VID-AD-Correct from VID-AD (Nakata et al., 2026). Both datasets are used as correction-oriented training and evaluation corpora, supporting multiple-choice correction selection, structured JSON action prediction, and open-ended correction generation.

We develop ReworkVLM with correction-aware post-training through RACO-GRPO, short for Rule-grounded Actionable Correction Optimization with GRPO. Starting from a Qwen3-VL-4B backbone (Qwen Team, 2025), supervised fine-tuning teaches the correction format, action vocabulary, and anomaly-to-correction mapping. RACO-GRPO then optimizes the model with correction-aware rewards that evaluate semantic correctness, trace-answer consistency, and minimal rule-restoring edits. This 4B setting is deliberate: in inspection environments, latency, serving cost, and maintainability make a smaller task-tuned model more attractive than repeatedly querying much larger general VLMs.

Our contributions are threefold:

- We formulate rule-grounded corrective action generation for reworkable logical anomalies and identify the Diagnosis-Action Gap between anomaly diagnosis and usable repair actions.
- We construct LOCO-Correct and VID-AD-Correct, two correction-oriented industrial anomaly corpora with multiple-choice, structured JSON, and open-ended evaluation protocols.
- We train ReworkVLM with correction-aware post-training and provide an industry-oriented evaluation that connects correction errors to rework cost, manual review, and operator trust.

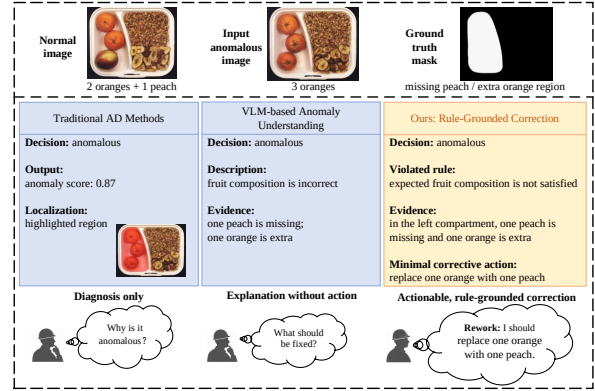


Figure 1: From anomaly diagnosis to actionable correction. Given a logical anomaly in a breakfast box, traditional anomaly detection methods provide anomaly scores and localization masks, while VLM-based methods can further describe the violated composition rule. However, neither directly specifies what should be fixed. Our setting requires a minimal rule-grounded corrective action, e.g., replacing one extra orange with one peach.

2 Related Work

2.1 Logical Anomaly Detection in Industrial Inspection

Industrial anomaly detection often targets visual deviations with image- or pixel-level labels (Bergmann et al., 2019), using patch distributions, reconstruction residuals, or student-teacher discrepancies (Defard et al., 2021; Zavrtanik et al., 2021; Wang et al., 2021; Roth et al., 2022; Batzner et al., 2024). Logical anomaly detection instead concerns global rule violations: count, type, placement, matching, length, or relation can be wrong even when local appearance is plausible. MVTec LOCO (Bergmann et al., 2022) and VID-AD (Nakata et al., 2026) study such rule-based anomalies. We change the target output from detecting a violated rule to identifying the minimal action that restores it.

2.2 Vision-Language Anomaly Understanding and Benchmarks

Vision-language models support anomaly VQA, explanation, and conversational inspection (Antol et al., 2015; Jeong et al., 2023; Zhou et al., 2024; Gu et al., 2024; Li et al., 2025; Jiang et al., 2024; Zhao et al., 2025; Bai et al., 2025; Qwen Team, 2025), yet industrial MLLM benchmarks still report failures on anomaly reasoning (Jiang et al., 2024). More importantly, fluent descriptions do not ensure actionability: a response may name the abnormal component but omit the repair, or re-

verse the correction direction. LOCO-Correct and VID-AD-Correct therefore evaluate correction in multiple-choice, structured JSON, and open-ended settings with action-preserving metrics.

2.3 Verifiable Post-Training and Actionable Evaluation

Post-training and evaluation methods such as RLHF, DPO, RLAIF, GRPO, LLM-as-judge scoring, and consistency-oriented reasoning objectives optimize or assess behavior beyond imitation (Ouyang et al., 2022; Rafailov et al., 2023; Bai et al., 2022; Shao et al., 2024; Liu et al., 2023; Zheng et al., 2023; Zhang et al., 2026). For logical anomaly correction, rewards must be verifiable: paraphrases can be correct, traces can contradict answers, and over-repair can satisfy a rule while being unusable. RACO-GRPO aligns rewards with action type, direction, quantity, trace consistency, and rule restoration.

3 Task and Workflow Framing

Actionable Visual Anomaly Understanding takes an inspection image I , an optional question q , and category-level normal rules $\mathcal{R}$, and returns an inspection decision plus an operator-facing correction when rework is possible. The output distinguishes anomaly detection, anomaly explanation, correction recommendation, and rule-restoring verification. In structured settings, the correction is represented by an action label such as *add*, *remove*, *replace*, *move*, *reject*, *manual_review*, or *none*, together with source/target objects, quantity, and location fields. In open-ended settings, the same information is expressed as a natural-language instruction.

Let S be the current visual state, $T(S, a)$ the state after applying action a , $R(\cdot)$ a rule verifier, and $C(a)$ the edit cost. The desired correction is the lowest-cost action that restores the normal rule:

$$a^* = \arg \min_{a \in \mathcal{A}} C(a) \quad \text{s.t.} \quad R(T(S, a)) = 1. \quad (1)$$

Thus, discarding a whole product when one object should be replaced is not a valid minimal correction; modifying a normal product is also an action failure. We evaluate this task in three settings: multiple-choice correction selection, structured JSON action prediction, and open-ended correction generation.

Field	Meaning	Example
Rule	Normal condition	two oranges + one peach
Evidence	Visual support	extra orange; missing peach
Action	Edit operation	replace
Correction	Operator instruction	replace one extra orange with one peach
Verify	Rule restored?	yes

Table 1: Compact correction annotation schema.

Dataset	Img.	Norm.	Anom.	Train QA/Img.	Eval QA/Img.
LOCO-Correct	3651	2658	993	12196 / 2539	5387 / 1112
VID-AD-Correct	10395	5000	5395	29200 / 8300	7380 / 2095

Table 2: Dataset statistics used in correction-oriented training and evaluation.

4 Datasets

Our construction goal is to convert each logical violation into an operator-facing correction record, rather than adding generic captions. Each record links a normal rule, visible evidence, repairability, action type, minimal correction, and a verification target. Figure 2 summarizes the pipeline: raw images, masks, normal references, and category rules are converted into correction annotations, then instantiated into multiple-choice, structured JSON, open-ended, and rationale-style VQA examples.

LOCO-Correct is built from MVTec LOCO (Bergmann et al., 2022). Each image receives a *.loco_correct.json annotation with anomaly, evidence, correction, QA seed, and quality fields. Anomaly status is inherited from the original benchmark labels and masks; Gemini 3 Pro (Google DeepMind, 2026) and human reviewers annotate the violated rule and minimal correction rather than the anomaly label itself. The construction pipeline scans the dataset, builds normal references, generates category rules and correction annotations with Gemini 3 Pro, integrates available image text, and then applies human review, adjudication, and final quality control. All images and model-generated annotation text are checked before entering the released correction corpus. On a 1,000-record subset independently reviewed by three annotators, exact three-way agreement on the key structured fields is 94.6% (Fleiss’ $\kappa = 0.8895$; 96.4% pairwise). In a blind-to-Gemini audit, three annotators independently re-annotated 300 records without seeing the generated draft. Separately, in a 1,000-record revision audit, human review revised 23.3% of generated drafts in at least one key structured field.

VID-AD-Correct is built from VID-AD (Nakata et al., 2026) as a second correction corpus for log-

ical anomalies under visual distraction. It is used for VID-AD-specific correction training and evaluation. This separation lets us measure whether the same actionable-correction protocol works across different logical inspection scenes while keeping the table labels aligned with the tuned models.

5 Method

ReworkVLM is trained in two stages: supervised fine-tuning teaches the correction format, action space, and structured rationale, while RACO-GRPO further optimizes semantic correctness, trace-answer alignment, and minimal rule-restoring edits. The training pipeline in Figure 2 uses Qwen3-VL-4B (Qwen Team, 2025) as the backbone and turns it into an operator-facing correction model.

5.1 SFT for Operator-Facing Correction

For image-question pair $x = (I, q)$, target y^* contains a short trace and a final answer: an option, JSON action, or correction. SFT teaches the evidence–rule–action mapping:

$$\mathcal{L}_{\text{sft}}(\theta) = -\mathbb{E}_{(x, y^*)} \sum_t \log \pi_\theta(y_t^* | x, y_{<t}^*). \quad (2)$$

This imitates corrections but does not directly penalize wrong objects, reversed direction, or over-repair.

5.2 Correction-Aware GRPO

RACO-GRPO scores sampled output $y = (z, \hat{a})$ with four rewards: parseable format, semantic action correctness, trace-answer alignment, and minimal rule restoration:

$$r_{\text{fmt}} = \mathbf{1}[\text{parse}(y)], \quad (3)$$

$$r_{\text{open}} = \mathbf{1}[\text{SemEq}(\hat{a}, a^*)] \mathbf{1}[\text{DirEq}(\hat{a}, a^*)], \quad (4)$$

$$r_{\text{trace}} = \mathbf{1}[\text{Supp}(z, \hat{a})] \mathbf{1}[\text{Ent}(z) \subseteq E] \cdot \mathbf{1}[\neg \text{Contr}(z, \hat{a})], \quad (5)$$

$$r_{\text{min}} = \mathbf{1}[R(T(S, \hat{a})) = 1] \exp(-\beta \Delta C_+). \quad (6)$$

Here E is the set of image-supported entities and rules, $\Delta C_+ = \max(0, C(\hat{a}) - C(a^*))$, and normal images receive minimal-edit credit only for none. The edit cost $C(a)$ is a fixed action-space table ordered as $\text{none} < \text{add/remove} < \text{replace/move} < \text{reject/manual_review}$ and scaled by the number of affected object instances, so ΔC_+ penalizes only edits that exceed the reference scope. For the breakfast box in Figure 1, “replace one orange with one peach” restores the

rule at minimal cost; “discard the box” is an over-scoped handling action and therefore incurs a large ΔC_+ , while an unrelated edit fails $R(T(S, \hat{a})) = 1$. TraceAlign discourages spurious rationales that accidentally lead to correct-looking answers.

The final reward is a weighted sum:

$$r = \lambda_f r_{\text{fmt}} + \lambda_o r_{\text{open}} + \lambda_t r_{\text{trace}} + \lambda_m r_{\text{min}}. \quad (7)$$

For each prompt, GRPO samples K outputs and computes group-normalized advantages,

$$A_i = \frac{r_i - \mu_G}{\sigma_G + \epsilon}, \quad \mu_G = \frac{1}{K} \sum_j r_j, \quad (8)$$

and optimizes a clipped objective with a KL penalty to the SFT model:

$$\rho_i = \frac{\pi_\theta(y_i | x)}{\pi_{\text{old}}(y_i | x)}, \quad (9)$$

$$g_i = \min(\rho_i A_i, \text{clip}(\rho_i, 1 - \epsilon_c, 1 + \epsilon_c) A_i),$$

$$\mathcal{J}(\theta) = \mathbb{E}_x \frac{1}{K} \sum_i g_i - \eta \text{KL}(\pi_\theta \| \pi_{\text{sft}}). \quad (10)$$

This shifts training from imitation toward semantically correct, rule-consistent, minimal, trace-aligned actions.

5.3 Inference and Fallback

At inference time, structured outputs are parsed into action fields. Invalid, contradictory, unsupported, or unnecessary edits receive the `manual_review` action label and are routed to manual review, preventing low-confidence corrections from becoming operator instructions.

6 Experiments

6.1 Settings, Metrics, and Baselines

Experiments evaluate correction behavior on LOCO-Correct and VID-AD-Correct with dataset-specific training and held-out evaluation; for LOCO-Correct reporting, the validation split is merged into the test split but is not used for training or tuning.

We evaluate three operator-facing formats: multiple-choice correction selection, structured JSON action prediction, and open-ended correction generation. General VLM baselines include Qwen3-VL variants (Qwen Team, 2025), Qwen3.5 (Qwen Team, 2026), ERNIE-4.5-VL (Baidu ERNIE Team, 2025), Gemma-4 (Gemma Team, 2026), AnomalyGPT (Gu et al., 2024), and

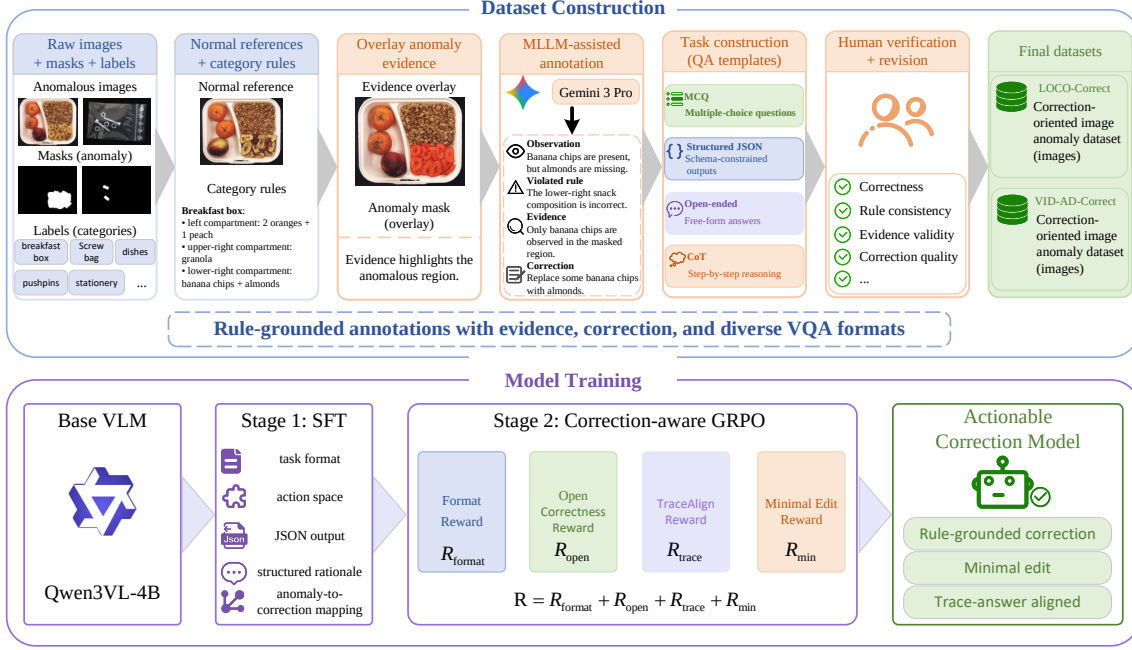


Figure 2: Overview of the dataset construction and model training pipeline. The upper panel shows how raw industrial anomaly images, masks, normal references, and category rules are converted into correction-oriented VQA data through MLLM-assisted annotation, task construction, and human verification. The lower panel illustrates the two-stage training process: SFT teaches the model the correction format and action space, while correction-aware GRPO optimizes rule-grounded, minimal, and trace-aligned corrective actions.

Triad (Li et al., 2025) where corresponding outputs are available. For prompting baselines, the prompt provides standard output-format guidance for each evaluation setting, instructing models to return an option, structured action fields, or an open-ended correction as appropriate. ReworkVLM uses a Qwen3-VL-4B backbone (Qwen Team, 2025) and is trained on the corresponding correction corpus. This comparison is intended to test whether a smaller task-tuned model can produce more actionable inspection outputs than larger general VLMs under structured constraints.

MC Acc is the overall multiple-choice accuracy. Anom Det measures whether the model correctly detects normal versus anomalous samples. MC Corr measures whether it selects the correct correction option. Action Acc checks whether the structured action type is correct. CSR is stricter: it requires the anomaly judgment, action type, key objects, direction, quantity when applicable, and rule-restoring effect to be correct. These multiple-choice and structured metrics are reported as percentages. Open-ended VQA uses GPT-5.2-judged scores with a unified, fixed judge prompt and a 1–5 rubric for Overall, Reasoning, Correction, and Minimal Edit. For ablation, hallucination is judged using the same fixed rubric as open-ended evalua-

tion.

6.2 Main Results

Table 3 shows that correction quality cannot be inferred from diagnosis alone. Larger general VLMs can be competitive on multiple-choice questions, but ReworkVLM has the strongest structured Action Acc and CSR on LOCO-Correct and improves both constrained metrics and GPT-5.2-judged open-ended correction on VID-AD-Correct. This supports the industrial motivation for training a smaller model to produce parseable, minimal, operator-facing actions.

To separate in-domain fine-tuning from model scale, we also fine-tune Qwen3-VL-32B with the same Stage-1 SFT. SFT improves the 32B substantially over its zero-shot result, yet the full 4B pipeline remains higher on CSR (LOCO 75.86 vs. 71.20; VID-AD 80.38 vs. 76.50). Model scale plus the same supervision does not close the gap to the full correction-aware pipeline; a two-stage 32B run is left to future work.

Open-ended scores measure semantic helpfulness, while CSR checks whether the action restores the rule; a fluent correction can still fail CSR by changing the wrong object, reversing direction, or over-repairing. We frame AnomalyGPT and Triad

Model	Params	Multiple Choice			Structured JSON		Open-ended VQA			
		MC Acc	Anom Det	MC Corr	Action Acc	CSR	Overall	Reasoning	Correction	Min Edit
Panel A: LOCO-Correct										
Qwen3-VL	4B	61.06	59.15	51.33	62.66	61.87	3.15	3.36	3.32	2.84
AnomalyGPT	7B	2.56	0.45	2.00	3.41	0.00	1.07	1.14	1.34	1.00
Triad	7B	64.40	63.61	70.00	0.00	0.00	1.95	1.46	1.65	1.60
Qwen3-VL	8B	73.85	66.24	73.33	59.71	58.98	3.22	3.39	3.36	2.99
Qwen3.5	27B	65.29	50.82	56.67	68.50	67.96	3.37	3.57	3.54	3.15
ERNIE-4.5-VL	28B	41.68	35.66	35.33	58.73	52.09	2.68	2.05	1.91	2.14
Gemma-4	31B	40.70	28.77	72.00	69.68	68.96	2.63	2.20	2.20	2.51
Qwen3-VL	32B	69.29	58.62	73.33	71.87	70.14	3.33	3.54	3.48	3.05
Qwen3-VL + SFT	32B	74.50	68.10	80.00	76.20	71.20	4.42	4.48	4.25	3.95
ReworkVLM (Ours)	4B	75.70	70.96	83.33	78.10	75.86	4.54	4.59	4.34	4.38
Panel B: VID-AD-Correct										
Qwen3-VL	4B	38.81	49.02	31.50	47.59	45.59	2.93	2.93	2.93	2.93
AnomalyGPT	7B	19.53	17.09	23.29	26.04	0.62	1.58	1.14	1.12	1.13
Triad	7B	36.23	50.45	29.26	43.48	42.43	3.01	3.12	3.30	3.64
Qwen3-VL	8B	37.12	46.25	29.55	47.30	47.30	2.86	2.85	2.86	2.83
Qwen3.5	27B	28.70	29.36	30.17	24.56	40.14	3.30	3.62	2.90	3.73
ERNIE-4.5-VL	28B	40.27	45.22	29.87	51.91	48.75	3.99	3.87	4.09	4.43
Qwen3-VL	32B	37.94	38.71	35.18	46.21	46.16	2.25	2.26	2.21	2.18
Qwen3-VL + SFT	32B	93.10	85.20	96.50	78.40	76.50	4.81	4.85	4.88	4.65
ReworkVLM (Ours)	4B	95.27	88.40	100.00	80.67	80.38	4.93	4.94	4.96	4.93

Table 3: Main results on LOCO-Correct and VID-AD-Correct. MC Acc, Anom Det, MC Corr, Action Acc, and CSR are reported as percentages. Open-ended VQA uses GPT-5.2-judged scores on a 1–5 scale with a unified prompt.

Model Variant	MCQ	Structured JSON		Open-ended		Halluc. ↓
	MC Acc	Action Acc	CSR	Corr.	Min Edit	
SFT	72.10	73.40	69.80	4.05	3.92	12.80
SFT + Format/Open GRPO	74.00	75.10	72.30	4.20	4.06	10.10
+ TraceAlign Reward	74.80	76.80	73.90	4.27	4.18	7.60
Full ReworkVLM	75.70	78.10	75.86	4.34	4.38	6.20

Table 4: Ablation of post-training on LOCO-Correct. MC Acc, Action Acc, CSR, and Hallucination are reported as percentages. Corr. and Min Edit are GPT-5.2-judged scores on a 1–5 scale.

as transferability probes rather than matched capability baselines: their outputs remain in anomaly-description formats under correction prompts, so their low structured scores indicate that diagnosis-oriented models do not transfer directly to operator-facing correction.

7 Analysis and Practical Discussion

Ablation. Table 4 ablates the correction-aware rewards used in ReworkVLM. SFT already provides a strong starting point because it teaches the output format, action vocabulary, and anomaly-to-correction mapping. Adding format and open-correctness rewards improves MC Acc and open-ended correction quality, while TraceAlign reduces unsupported reasoning by encouraging the rationale to support the final action. The full model achieves the best CSR and Minimal Edit scores, indicating that explicit minimal-edit rewards mat-

ter for operator-facing corrections. Across three independent training runs, CSR is 69.8 ± 0.8 for SFT and 75.9 ± 0.4 for the full model, so the improvement exceeds run-to-run variation.

Diagnosis-Action Gap. The central failure mode is not simply missing an anomaly. A model can answer anomaly detection correctly yet fail to produce a repairable action, or choose the right multiple-choice correction but fail in structured JSON because it must generate the action type, objects, direction, and quantity itself. This gap matters operationally: a wrong action can send a normal product to rework, make an anomalous product worse, or force an operator to ignore the system.

Why strong general VLMs fail MCQ. On VID-AD-Correct, Qwen3-VL 8B/32B score far below their diagnosis capability on multiple-choice correction even though the rule and image are provided. This is not mainly an output-format artifact; inspection of the parsed outputs shows that the models often produce the wrong option rather than an unparseable response. They typically detect the anomaly yet choose the wrong action direction or an over-scoped edit, showing the Diagnosis-Action Gap in its clearest form.

Table 5 summarizes the main correction-specific failure categories and their operational risks. These categories help separate ordinary diagnosis errors

Failure type	Industrial risk	Example
Wrong direction	Makes defect worse	replace orange with peach vs. reverse
Non-minimal edit	Unnecessary rework cost	discard whole box
Unnecessary repair	False rework	modify normal product
Hallucinated object	Untrustworthy instruction	replace banana when no banana exists

Table 5: Correction failures and inspection risks.

from action-level failures that directly affect rework decisions.

Open-ended quality is not executability. Open-ended judge scores are useful because operators read natural-language corrections, but they are not a substitute for structured checks. A fluent correction may receive a high score yet fail CSR if it changes the wrong object or direction, so Table 3 reports both GPT-5.2-judged scores and CSR.

Why tune a smaller model. Industry-facing inspection systems must consider latency, serving cost, maintenance, and predictable output formats. On a single A100 (BF16, batch 1, 1,000 samples), ReworkVLM-4B uses 8.9 GB model GPU memory versus 66.7 GB for Qwen3-VL-32B, with mean end-to-end inference latency 4.20 s versus 10.49 s (P95 6.76 s versus 13.90 s) and about 2× higher decoding throughput. The goal is not to replace high-capacity VLMs everywhere, but to distill the correction behavior needed for a narrow workflow into a smaller model with substantially lower serving cost.

Fallback and human workflow. The intended workflow is decision support, not autonomous repair. Parsed outputs enter a decision layer: none maps to no action, valid minimal edits map to rework, and invalid or ambiguous corrections map to manual_review or reject. This keeps human review for uncertain rules, evidence, or edits.

8 Conclusion

We introduced Actionable Visual Anomaly Understanding, asking inspection models to answer “What should be fixed?” rather than stopping at diagnosis. By formalizing correction as a minimal rule-restoring edit, the task exposes a Diagnosis-Action Gap between recognizing an anomaly and producing a usable action. LOCO-Correct, VID-AD-Correct, ReworkVLM, and RACO-GRPO move logical anomaly understanding toward verifiable, rule-grounded, operator-facing correction.

Limitations

The task focuses on logical anomalies expressible as discrete object, quantity, location, size, and relation rules; it is less direct for surface defects or multi-step physical repairs. The datasets derive from existing benchmarks, so residual ambiguity may remain despite normal-reference construction, rule generation, review, and adjudication. GPT-5.2-judged open-ended scores should be calibrated against expert judgments before operational use, and deployment would still require production-line latency, interface, confidence, and escalation tests.

Ethics Statement

The datasets are derived from industrial anomaly benchmarks and describe object-level corrections, not people or sensitive attributes. In real workflows, correction suggestions should remain operator-verified decision support, especially when repair actions have safety or economic consequences.

References

- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. [VQA: Visual question answering](#). In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao 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, and 8 others. 2025. [Qwen2.5-VL technical report](#). *arXiv preprint arXiv:2502.13923*.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022. [Constitutional AI: Harmlessness from AI feedback](#). *arXiv preprint arXiv:2212.08073*.
- Baidu ERNIE Team. 2025. [ERNIE 4.5 technical report](#).
- Kilian Batzner, Lars Heckler, and Rebecca König. 2024. [EfficientAD: Accurate visual anomaly detection at millisecond-level latencies](#). In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 128–138.
- Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. 2022. [Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization](#). *International Journal of Computer Vision*, 130(4):947–969.

- Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. 2019. [MVTec AD: A comprehensive real-world dataset for unsupervised anomaly detection](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9592–9600.
- Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. 2021. [PaDiM: A patch distribution modeling framework for anomaly detection and localization](#). In *Pattern Recognition. ICPR International Workshops and Challenges*, volume 12664 of *Lecture Notes in Computer Science*, pages 475–489. Springer.
- Gemma Team. 2026. [Gemma 4 technical report](#). Preprint, arXiv:2607.02770.
- Google DeepMind. 2026. [Gemini 3 pro model card](#). Last updated May 2026.
- Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. 2024. [AnomalyGPT: Detecting industrial anomalies using large vision-language models](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1932–1940.
- Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. 2023. [WinCLIP: Zero-/few-shot anomaly classification and segmentation](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19606–19616.
- Xi Jiang, Jian Li, Hanqiu Deng, Yong Liu, Bin-Bin Gao, Yifeng Zhou, Jialin Li, Chengjie Wang, and Feng Zheng. 2024. [MMAD: The first-ever comprehensive benchmark for multimodal large language models in industrial anomaly detection](#). *arXiv preprint arXiv:2410.09453*.
- Yuanze Li, Shihao Yuan, Haolin Wang, Qizhang Li, Ming Liu, Chen Xu, Guangming Shi, and Wangmeng Zuo. 2025. [Triad: Empowering LMM-based anomaly detection with vision expert-guided visual tokenizer and manufacturing process](#). *arXiv preprint arXiv:2503.13184*.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. [G-Eval: NLG evaluation using GPT-4 with better human alignment](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522. Association for Computational Linguistics.
- Hiroto Nakata, Yawen Zou, Shunsuke Sakai, Shun Maeda, Chunzhi Gu, Yijin Wei, Shange Gao, and Chao Zhang. 2026. [VID-AD: A dataset for image-level logical anomaly detection under vision-induced distraction](#). *arXiv preprint arXiv:2603.13964*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.
- Qwen Team. 2025. [Qwen3-VL technical report](#). *arXiv preprint arXiv:2511.21631*.
- Qwen Team. 2026. [Qwen3.5: Towards native multi-modal agents](#).
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Advances in Neural Information Processing Systems*, volume 36.
- Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. 2022. [Towards total recall in industrial anomaly detection](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yang Wu, and Daya Guo. 2024. [DeepSeekMath: Pushing the limits of mathematical reasoning in open language models](#). *arXiv preprint arXiv:2402.03300*.
- Guodong Wang, Shumin Han, Errui Ding, and Di Huang. 2021. [Student-teacher feature pyramid matching for anomaly detection](#). In *Proceedings of the British Machine Vision Conference*.
- Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. 2021. [DRAEM: A discriminatively trained reconstruction embedding for surface anomaly detection](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8330–8339.
- Haitao Zhang, Yingying Wang, Jiaxiang Wang, Haote Xu, Hongyang Zhang, Yirong Chen, Yue Huang, and Xinghao Ding. 2026. [MedAD-R1: Eliciting consistent reasoning in interpretable medical anomaly detection via consistency-reinforced policy optimization](#). *arXiv preprint arXiv:2602.01081*.
- Shifang Zhao, Yiheng Lin, Lu Han, Yao Zhao, and Yunchao Wei. 2025. [OmniAD: Detect and understand industrial anomaly via multimodal reasoning](#). *arXiv preprint arXiv:2505.22039*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. [Judging LLM-as-a-judge with MT-Bench and chatbot arena](#). In *Advances in Neural Information Processing Systems*, volume 36.
- Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. 2024. [AnomalyCLIP: Object-agnostic prompt learning for zero-shot anomaly detection](#). In *Proceedings of the Twelfth International Conference on Learning Representations*.