RILO: FEEDING IMPLICIT PROCESS REWARDS BACK INTO VISUAL REASONING FOR INDUSTRIAL ANOMALY UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Industrial anomaly understanding requires reasoning over subtle, localized, and category-dependent visual evidence whose relevance may change as the response evolves. However, standard multimodal large language models (MLLMs) typically encode the input image only once and keep visual-token representations fixed throughout decoding. This creates a mismatch between the evolving visual demands of anomaly reasoning and the static visual representation used for generation. To bridge this gap, we propose RILO (Reward-guided Inspection Loop with Online Visual Modulation), a parameter-efficient post-training framework that uses implicit process rewards not only for policy learning but also as online feedback for visual adaptation. RILO learns state-aligned token-level rewards from answer correctness, evidence compatibility, and structured-output consistency, and periodically uses them to adapt query image representations during response generation. This generate–evaluate–revise loop redirects visual inspection toward evidence relevant to the current reasoning state, without requiring annotated reasoning traces or additional spatial supervision. Extensive experiments demonstrate that RILO achieves state-of-the-art accuracy on the MMAD benchmark under both one-shot and zero-shot settings, reaching 81.26 and 79.52, respectively. The code will be publicly available.

1 INTRODUCTION

"You see, but you do not observe. The distinction is clear."

—Sherlock Holmes

Industrial anomaly question answering extends conventional anomaly detection beyond image-level classification and pixel-level localization to reasoning about whether a defect is present, what type of defect it is, where it is located, how it appears, and the overall condition of the object. Recent MLLM-based efforts include AnomalyGPT are supposed for conversational anomaly detection and localization, while MMAD provides systematic evaluation of anomaly-related QA capabilities Gu et al. (2024); Jiang et al. (2025). Industrial anomaly QA remains challenging because defects are often small, localized, category-dependent, and rare in general-purpose pretraining data Gu et al. (2024); Chen et al. (2026). The substantial gap between object-oriented and defect-oriented questions on MMAD further indicates that object recognition alone is insufficient for anomaly reasoning Jiang et al. (2025). Moreover, different reasoning objectives also require visual evidence at different levels of detail: global discrepancies may reveal an anomaly, whereas identifying its type, location, or appearance requires more specific local cues. As the reasoning objective evolves, the model must therefore repeatedly redirect visual inspection toward the evidence relevant to the current decision.

However, current MLLMs provide limited support for such adaptive visual inspection. Images are typically encoded only once, leaving visual-token representations fixed throughout decoding Liu et al. (2023); Bai et al. (2025b). As a result, the model can only rely on the same static visual representation as evidence demands evolve, creating a mismatch between anomaly reasoning and visual processing.

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.

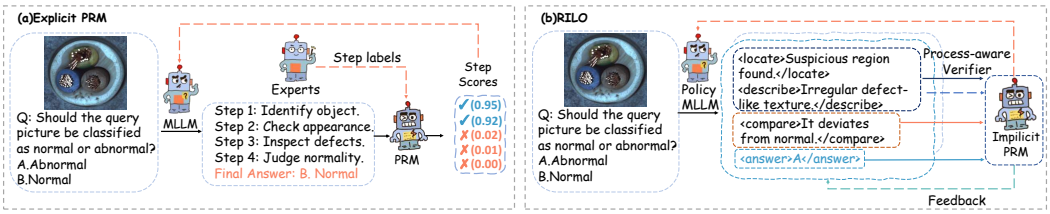


Figure 1: Comparison between existing Explicit PRM method and the proposed RILO. Explicit PRMs add process supervision with step-level annotations, whereas RILO derives implicit process rewards from structured QA supervision and feeds them back for within-response visual adaptation.

To address this mismatch, the model should adapt its visual inspection to the evolving reasoning state. This raises a natural question: how can the model assess whether the current visual evidence remains sufficient as reasoning progresses? Recent industrial anomaly reasoning methods have primarily approached this challenge from the policy-learning side through reinforcement learning. Methods such as AnomalyR1, OmniAD, and AD-FM use reward-based optimization to encourage policies that produce more accurate anomaly reasoning and predictions Chao et al. (2025); Zhao et al. (2025); Liao et al. (2026). Such supervision can improve how the policy uses anomaly evidence across training episodes. However, because the reward is obtained from completed responses, it cannot indicate during the current response when the visual evidence has become insufficient or when visual inspection should be redirected. This motivates us to consider process-level supervision developed in broader MLLM reasoning. Process Reward Models (PRMs) evaluate intermediate reasoning states, providing the type of intermediate assessment required before a response is completed. Explicit PRMs, however, typically rely on separately constructed step-level supervision Lightman et al. (2024); Wang et al. (2025), which is particularly costly for industrial anomaly reasoning where intermediate supervision must capture fine-grained visual evidence. Implicit process rewards avoid such step-level annotations by deriving intermediate feedback from response-level supervision Yuan et al. (2025); Cui et al. (2026). This makes them well suited to industrial anomaly reasoning. However, existing implicit process-reward methods still convert these intermediate signals mainly into training returns for policy optimization after rollout. As a result, they provide no direct learning signal for the model to dynamically adjust its use of visual information as reasoning unfolds.

To bridge the gap between process evaluation and within-response visual adaptation, We introduce RILO, a parameter-efficient **R**eward-guided **I**nspection **L**oop with **O**nline Visual Modulation. RILO learns token-level implicit rewards from a structured response-level target combining answer correctness, evidence compatibility, and output consistency. During RL rollout generation, these rewards are periodically fed back to modulate query-image representations and influence subsequent reasoning. This generate–evaluate–revise–continue loop turns process evaluation into within-rollout feedback for closed-loop policy learning, without requiring annotated reasoning traces or additional spatial supervision.

Our main contributions are:

- We introduce a closed-loop RL paradigm that feeds implicit process rewards back into rollout generation, allowing visual representations to adapt with the evolving reasoning process and training the policy to make reasoning-dependent use of visual evidence for industrial anomaly reasoning.
- We propose RILO, which learns an implicit process reward from deterministic outcome supervision and reuses the same learned signal for both policy optimization and reasoning-conditioned visual modulation, without requiring additional process-level or spatial annotations.
- RILO achieves state-of-the-art one-shot and zero-shot performance on MMAD, with particularly strong improvements on defect-oriented tasks that require fine-grained visual reasoning.

2 Related Work

2.1 Multimodal Large Language Models for Industrial Anomaly Question Answering

Traditional industrial anomaly detection focuses mainly on anomaly classification and defect localization Bergmann et al. (2019). MLLMs extend this setting toward language-based inspection: AnomalyGPT adapts vision-language models to anomaly judgment, while MMAD evaluates seven subtasks spanning anomaly discrimination, defect understanding, and object analysis Gu et al. (2024); Jiang et al. (2025). Recent methods improve industrial adaptation through task-specific RL, domain rewards, knowledge routing, difference grounding, reflection, retrieval, and latent reasoning Chao et al. (2025); Guan et al. (2025); Li et al. (2026); Liao et al. (2026); Kang et al. (2026); Zhao et al. (2025); Anonymous (2026); Wang et al. (2026); Huang et al. (2026); Miao et al. (2025); Chen et al. (2026).These advances improve answer accuracy and evidence acquisition but leave the mismatch between evolving evidence demands and static visual representations unresolved. Without state-aligned process feedback, models may still miss subtle, localized anomalies that require adaptive visual inspection.

2.2 Visual State Evolution

In broader multimodal reasoning, recent studies have shown that a correct final answer does not necessarily imply faithful use of visual evidence. Models may rely on language priors or insufficiently grounded visual cuesAgrawal et al. (2018); Cadene et al. (2019); Li et al. (2023); Wu et al. (2025); Kang et al. (2025). This issue is particularly relevant to industrial anomaly QA, where subtle and localized defects require reasoning to remain grounded in fine-grained visual evidence. A growing line of work therefore allows visual information to evolve during reasoning. Mirage introduces latent visual tokens, H-GIVR repeatedly re-observes the image, SIEVE injects salient regions, and ReasonIAD dynamically updates visual evidence Yang et al. (2026b;a); Shi et al. (2026); Chen et al. (2026). These methods show the value of evolving visual information during reasoning, but their updates are typically driven by indirect signals rather than reasoning-state evaluation. RILO instead uses state-aligned implicit process feedback to couple policy learning with visual adaptation, forming a closed loop between process evaluation and visual-state evolution. This is particularly suitable for industrial anomaly reasoning with subtle defects and limited process or spatial supervision.

2.3 Implicit Process Rewards

In broader language and multimodal reasoning, Process Reward Models (PRMs) have been developed to provide denser credit by evaluating intermediate reasoning states Uesato et al. (2022); Lightman et al. (2024). Explicit PRMs, such as VisualPRM, rely on separately constructed process supervision and are commonly used to evaluate or rank completed reasoning trajectories Wang et al. (2025). More recent implicit approaches avoid step-level annotations by deriving intermediate signals from response-level outcomes. Free Process Rewards obtains implicit token-level rewards through policy-to-reference likelihood ratios, while PRIME uses such signals for online policy optimization Yuan et al. (2025); Cui et al. (2026). These methods provide intermediate reasoning feedback but mainly use it for policy learning. RILO extends this feedback to online visual adaptation, allowing the visual state to evolve with changing evidence demands. This better supports industrial anomaly reasoning, where subtle and localized defects require fine-grained visual inspection.

3 Methodology

3.1 Problem Formulation

We study an industrial anomaly question-answering task, where the model answers anomaly-related questions based on a given industrial image. Given a query image I_i^q, a question Q_i, and a set of answer options $\mathcal{O}_i$, the model aims to predict the correct answer.

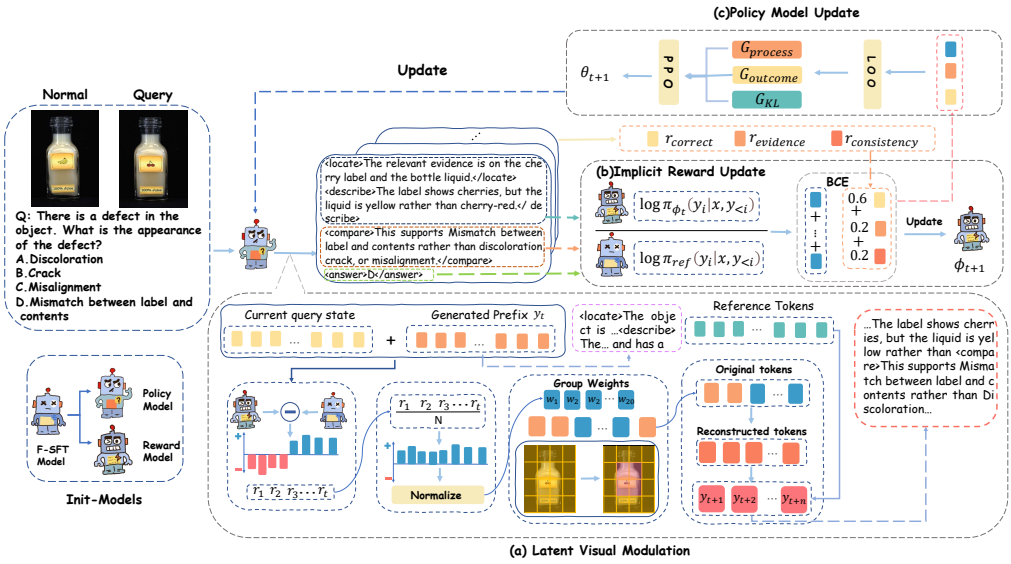


Figure 2: Overview of RILO. A frozen policy snapshot generates structured responses in token chunks. At each nonterminal boundary, the implicit reward branch is compared with the frozen SFT reference to derive trajectory-conditioned visual modulation coefficients. RILO reconstructs the query-image representation and re-prefills the multimodal prefix before generation continues.

We consider two settings. In the *one-shot* setting, the model additionally receives a normal reference image I_i^n from the same category as the query image. The model input is

$$x_i^{\text{1shot}} = (I_i^n, I_i^q, Q_i, \mathcal{O}_i, \tau), \tag{1}$$

where τ specifies the required output structure. In the *zero-shot* setting, no reference image is provided, and the input becomes

$$x_i^{\text{0shot}} = (I_i^q, Q_i, \mathcal{O}_i, \tau). \tag{2}$$

Conditioned on x_i, the policy generates a structured response

$$z_i = (z_i^{\text{loc}}, z_i^{\text{desc}}, z_i^{\text{cmp}}, \hat{a}_i), \tag{3}$$

where the first three terms denote structured intermediate outputs and $\hat{a}_i$ is the predicted answer.

3.2 RILO Framework

RILO couples process evaluation with visual-state adaptation within a single generation trajectory. It consists of a policy π_θ, an autoregressive implicit reward branch π_ϕ, and a frozen SFT reference π_{SFT}. Figure 2 provides an overview of RILO.

Let $H_i^{q,(k)}$ denote the query-image representation after the k-th visual update and t_k the corresponding generation boundary. RILO models generation as the iterative transition.

$$\left(y_{i,\le t_k}, H_i^{q,(k)}\right) \xrightarrow{\mathcal{E}_\phi} \mathbf{q}_{i,k}^{\text{vis}} \xrightarrow{\mathcal{C}} H_i^{q,(k+1)} \xrightarrow{\pi_\theta} y_{i,t_k+1:t_{k+1}}, \tag{4}$$

where $\mathcal{E}_\phi$ evaluates the current generation state and $\mathcal{C}$ converts the resulting process signal into an updated visual state. When a normal reference is available, its representation remains fixed while only the query representation is adapted.

The same process signal serves two roles in RILO: it provides dense credit for policy learning and guides visual adaptation before subsequent generation. We describe these two roles below. Algorithm in Appendix B summarizes the complete RILO procedure.

3.3 State-Aligned Implicit Process Reward

Answer correctness alone does not indicate whether a prediction is supported by appropriate anomaly evidenc. Given the structured response z_i, ground-truth answer $a_i^\star$, and reference evidence $E_i^\star$, the deterministic verifier produces a response-level soft process target:

$$s_{\mathrm{proc},i} = v_{\mathrm{fmt},i}\left(0.6s_{\mathrm{ans},i} + 0.2s_{\mathrm{evi},i} + 0.2s_{\mathrm{con},i}\right). \tag{5}$$

Here, s_{ans} measures answer correctness, s_{evi} measures compatibility between the generated and reference evidence, and s_{con} measures consistency within the structured response. The weighting prioritizes final-answer correctness while retaining graded evidence and consistency supervision; its local sensitivity is evaluated in Appendix C.4.

Since s_{proc} is available only at the response level, RILO uses an autoregressive reward branch to distribute this supervision over the trajectory. For a token $y_{i,t}$ generated under visual state $H_i^{q,(k_i(t))}$, its context is

$$c_{i,t} = \left(x_i, y_{i,<t}; H_i^{q,(k_i(t))}\right), \tag{6}$$

where $k_i(t)$ denotes the visual-state index used to generate token $y_{i,t}$. With the normal-reference state additionally included when available. The implicit token reward is defined relative to the frozen SFT reference: Yuan et al. (2025); Cui et al. (2026):

$$q_{i,t}^{\mathrm{RM}} = \log \pi_\phi(y_{i,t} \mid c_{i,t}) - \log \pi_{\mathrm{SFT}}(y_{i,t} \mid c_{i,t}). \tag{7}$$

The accumulated token scores predict the process target,

$$\mathcal{L}_{\mathrm{RM}} = \mathrm{BCE}\left(\sigma\left(\beta_{\mathrm{RM}} \sum_{t=1}^{T_i} q_{i,t}^{\mathrm{RM}}\right), s_{\mathrm{proc},i}\right). \tag{8}$$

Because each token is evaluated under the visual state that produced it, $q_{i,t}^{\mathrm{RM}}$ provides state-aligned process credit without requiring step-level reward annotations.

3.4 Reward-Guided Latent Visual Modulation

RILO further uses process evaluation to update the visual state during generation. At each nonterminal boundary t_k, the accumulated prefix is rescored by the reward branch and the frozen SFT reference under the current visual state:

$$q_{i,u,k}^{\mathrm{vis}} = \beta_{\mathrm{vis}}\left[\log \pi_\phi(y_{i,u} \mid c_{i,u}^{(k)}) - \log \pi_{\mathrm{SFT}}(y_{i,u} \mid c_{i,u}^{(k)})\right], \qquad u \le t_k. \tag{9}$$

This prefix-level profile describes the relative process support for the current trajectory with respect to a fixed reference model.

We decompose the profile into an overall modulation magnitude

$$P_i^{(k)} = \frac{1}{t_k} \sum_{u=1}^{t_k} q_{i,u,k}^{\mathrm{vis}}, \tag{10}$$

Negative token-level scores are converted into a risk trajectory, linearly interpolated to G ordered visual-token groups, and normalized to obtain $\mathbf{w}_i^{(k)}$; full details are given in Appendix D.

$$h_{i,j}^{q,(k+1)} = h_{i,j}^{q,(0)} + \alpha P_i^{(k)} w_{i,g(j)}^{(k)} \mathbf{1}_D. \tag{11}$$

Thus, $P_i^{(k)}$ provides a global signed modulation scale, while $w_{i,g(j)}^{(k)}$ distributes this update across visual-token groups. Each update is defined relative to the original query representation $H_i^{q,(0)}$. The revised visual state is then used to reconstruct the multimodal context before generation continues.

Together, Eqs. equation 9–equation 11 turn intermediate process evaluation into a visual-state transition within the same response, completing the generate–evaluate–revise–continue loop.

3.5 Online Policy Optimization

Given the state-annotated rollouts generated by the closed-loop procedure above, RILO updates the actor using three complementary reward sources: state-aligned implicit process rewards, terminal verifier rewards, and KL regularization toward the frozen SFT policy:

$$r_{i,t}^{(\mathrm{imp})} = q_{i,t}^{\mathrm{RM}}, \qquad r_{i,t}^{(\mathrm{ver})} = s_{\mathrm{proc},i}\mathbf{1}\{t = T_i\}, \tag{12}$$

$$r_{i,t}^{(\mathrm{kl})} = -\lambda_{\mathrm{KL}}\left(\ell_{\mathrm{old},i,t} - \ell_{\mathrm{SFT},i,t}\right). \tag{13}$$

Because these reward sources differ in density and scale, RILO constructs their advantages independently using source-wise leave-one-out normalization and then combines them into $A_{i,t}$. The complete advantage construction is given in Appendix B.

For policy optimization, both the current and rollout policies are evaluated under the visual state used to generate each token. The importance ratio is

$$\rho_{i,t}(\theta) = \exp\left(\ell_{\theta,i,t} - \ell_{\mathrm{old},i,t}\right), \tag{14}$$

and the actor minimizes the clipped objective

$$\mathcal{L}_{\mathrm{pol}} = -\mathbb{E}_{i,t}\left[\min\left(\rho_{i,t}A_{i,t}, \mathrm{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)A_{i,t}\right)\right]. \tag{15}$$

RILO therefore uses a common state-aligned process-evaluation mechanism to provide dense policy credit and guide visual-state adaptation within the same generation trajectory.

4 Experiments

4.1 Experimental Setup

Benchmarks and metrics. We evaluate RILO on MMAD Jiang et al. (2025), which integrates DS-MVTec Bergmann et al. (2019), GoodsAD Zhang et al. (2024), VisA Zou et al. (2022), and MVTec-LOCO Bergmann et al. (2022). It covers seven subtasks: anomaly discrimination, defect classification, defect localization, defect description, defect analysis, object classification and object analysis.

Compared baselines. We compare RILO with a broad set of representative baselines. The proprietary MLLMs include Gemini2.5-Pro, Gemini-2.5-Flash, GPT-5-mini. General-purpose open-source MLLMs include Qwen2.5-VL Bai et al. (2025b) Qwen3-VL Bai et al. (2025a) LLaVA-OneVision Li et al. (2024) Kimi-VL Kimi Team et al. (2025) InternVL3 Zhu et al. (2025) GLM-4.1V-Thinking GLM-V Team et al. (2025), covering both standard instruction-tuned models and reasoning-oriented variants. We further compare with task-specific industrial anomaly reasoning methods. These include GRPO-based post-training approaches, such as AnomalyR1 Chao et al. (2025) OmniAD Zhao et al. (2025) JUDO Kang et al. (2026) EMIT Guan et al. (2025), as well as knowledge- and latent-reasoning approaches, including Reason-IAD Chen et al. (2026) and DCR^2-AD Anonymous (2026). We also report human-expert performance Jiang et al. (2025) as an additional reference.

Implementation details. RILO-3B and RILO-4B use Qwen2.5-VL-3B-Instruct Bai et al. (2025b) and Qwen3-VL-4B Bai et al. (2025a). SFT and online RL use disjoint QA sets, and their query/reference images are isolated from official evaluation images. Training uses only structured QA, without spatial or human-authored process labels. Experiments use two NVIDIA A800 80GB GPU. More details are provided in Appendix B.

Tables 1 and 2 report one-shot and zero-shot results; Avg follows the official seven-task protocol.

One-shot results. RILO performs strongly at both model scales. RILO-4B achieves the highest average MMAD score of 81.26, surpassing larger task-specific models including DCR^2-AD (+DK)-7B (80.83), JUDO-7B (80.32), and EMIT-8B (79.97), as well as Reason-IAD-4B (79.10). RILO-3B also improves over the matched-scale AnomalyR1-3B baseline from 74.64 to 76.09, indicating that the gain is not simply due to model capacity.

Table 1: One-shot results on full MMAD; Avg follows the official protocol. For compactness, we report the three strongest proprietary baselines by reported Avg.; the complete comparison is provided in the Appendix. †: official MMAD Jiang et al. (2025); ‡: Reason-IAD uniform evaluation Chen et al. (2026); *: EMIT native inference, including pre-generated domain knowledge Guan et al. (2025). Bold subtask values are best among non-human methods; underlined Avg is best within each displayed block.

Method	Scale	Anomaly	Defect				Object		Avg
		Discrimination	Classification	Localization	Description	Analysis	Classification	Analysis	
Human (expert)†	–	95.24	75.00	92.31	83.33	94.20	86.11	80.37	86.65
Gemini-2.5-Flash‡	–	83.38	69.88	63.30	76.41	81.57	94.04	82.00	78.65
GPT-5-mini‡	–	64.10	67.35	69.07	79.02	86.72	93.96	83.37	77.65
Gemini-2.5-Pro‡	–	83.07	73.86	67.20	79.97	86.27	94.88	83.08	81.19
Qwen2.5-VL	3B	60.47	44.20	55.25	62.81	79.22	85.38	81.48	66.97
AnomalyR1	3B	53.47	69.95	64.99	77.27	81.89	89.71	85.19	74.64
Qwen2.5-VL	3B	60.47	44.20	55.25	62.81	79.22	85.38	81.48	66.97
AnomalyR1	3B	53.47	69.95	64.99	77.27	81.89	89.71	85.19	74.64
Qwen3-VL‡	4B	74.02	59.25	62.74	70.09	79.22	92.47	83.59	74.48
Reason-IAD‡	4B	75.05	73.72	66.48	76.83	82.33	94.90	84.41	79.10
LLaVA-OneVision†	7B	55.53	71.18	55.21	74.24	80.53	90.53	82.77	72.86
Qwen2.5-VL‡	7B	70.74	67.97	60.82	72.78	82.62	**97.17**	84.99	76.73
DCR^2-AD	7B	70.00	62.06	69.60	79.34	84.75	93.52	86.79	78.01
OmniAD	7B	68.80	78.80	75.50	67.20	86.40	96.00	86.40	79.87
DCR^2-AD (+DK)	7B	70.46	74.28	68.07	**83.69**	86.01	96.90	86.40	80.83
JUDO	7B	71.55	73.05	69.88	82.06	**88.41**	90.47	86.79	80.32
InternVL3 (GRPO)	8B	71.44	75.07	74.51	78.72	84.73	86.32	83.35	79.16
Qwen3-VL‡	8B	75.24	75.07	62.78	76.38	85.58	96.31	84.62	79.43
EMIT*	8B	73.20	**80.05**	**76.37**	82.05	86.41	85.96	75.75	79.97
Kimi-VL‡	16B	66.57	71.72	61.88	79.78	82.93	92.18	86.51	77.37
GLM-4.1V-Thinking‡	9B	72.31	72.01	66.47	79.98	84.61	93.79	83.26	78.92
DCR^2-AD	72B	74.37	64.60	71.72	82.23	87.17	93.24	**89.28**	80.37
RILO (ours)	3B	55.35	70.95	68.88	83.25	85.73	85.90	82.56	**76.09**
RILO (ours)	4B	**75.70**	76.36	75.78	80.11	84.33	92.25	84.29	**81.26**

Table 2: Zero-shot results on full MMAD. Avg follows the official protocol Guan et al. (2025); Chen et al. (2026); Zhao et al. (2025); Kang et al. (2026); bold denotes the best result in each column.

Method	Scale	Anomaly	Defect				Object		Avg
		Discrimination	Classification	Localization	Description	Analysis	Classification	Analysis	
Qwen3-VL (Base)	4B	65.83	53.97	58.16	66.35	77.82	91.98	83.38	71.07
EMIT	8B	66.71	63.04	73.36	77.87	85.74	86.94	77.58	75.89
Reason-IAD	4B	66.58	70.62	62.82	73.28	80.88	93.80	84.88	76.12
OmniAD	7B	61.40	67.80	**73.50**	**81.70**	87.50	**95.20**	**86.90**	79.10
JUDO	7B	**68.85**	70.54	72.68	80.91	**88.09**	88.74	86.08	79.41
RILO (ours)	4B	67.45	**73.71**	73.16	80.22	84.75	92.16	85.17	**79.52**

Compared with GRPO-based methods such as AnomalyR1, OmniAD, JUDO, and EMIT Chao et al. (2025); Zhao et al. (2025); Kang et al. (2026); Guan et al. (2025), RILO additionally feeds intermediate process signals back into the ongoing generation to adapt visual inspection. Its gains over matched-scale AnomalyR1 and larger GRPO-based models suggest that policy optimization alone may not fully address the changing visual-evidence demands of anomaly reasoning. RILO-4B also outperforms methods using additional domain knowledge or specialized reasoning mechanisms Anonymous (2026); Chen et al. (2026).

The advantage is most pronounced on defect-oriented tasks. At 3B, Def. Avg increases from 60.37 to 77.20 (+16.83), with Defect Classification rising from 44.20 to 70.95 and Description from 62.81 to 83.25. At 4B, Def. Avg improves from 67.83 to 79.15, including gains in Classification (59.25→76.36) and Localization (62.74→75.78). The concentration of improvements on these fine-grained tasks supports the role of RILO in anomaly inspection rather than coarse object recognition

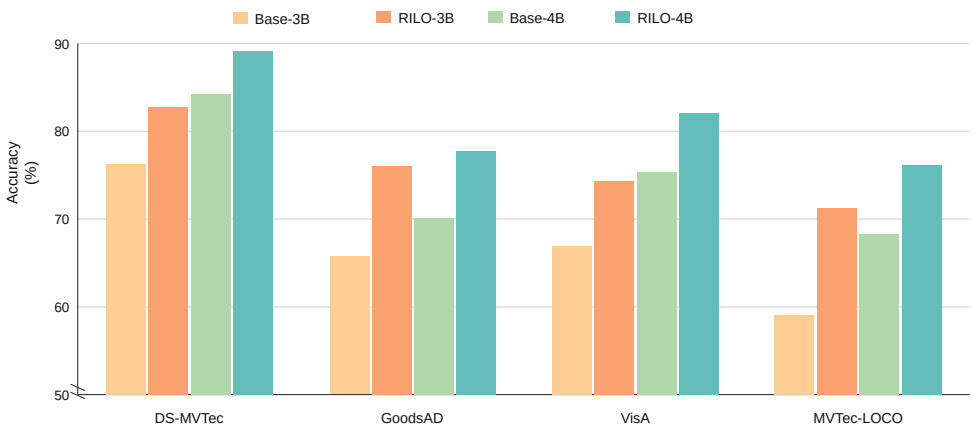


Figure 3: Per-dataset MMAD accuracy of the base and RILO models at the 3B and 4B scales under the official task-balanced protocol.

Table 3: Component ablations on MMAD using Qwen2.5-VL-3B and Qwen3-VL-4B. Δ Avg. is measured from the corresponding actor-only baseline in Panels B–C.

Method	Term.	iPRM	VM_{tr}	$\text{Control}_{\text{te}}$	Def. Avg	Avg. ($\mu \pm \sigma$)	Δ Avg.
A. 3B: Reward-learning ablations							
Base actor	–	–	–	Off	60.37	66.97	–
F-SFT	–	–	–	Off	65.15	67.49 ± 0.29	–
Answer-only RL	Ans	–	–	Off	64.12	68.00 ± 0.29	–
Process-terminal RL	Proc	–	–	Off	68.92	70.62 ± 0.33	–
B. 4B: Main-model attribution							
iPRM, actor-only	Proc	✓	–	Off	77.93	79.86 ± 0.27	–
iPRM + inference control	Proc	✓	–	Learned	77.86	79.91 ± 0.29	$+0.05$
RILO, controller-off	Proc	✓	✓	Off	79.04	81.10 ± 0.31	$+1.24$
Full RILO	Proc	✓	✓	Learned	**79.15**	$\mathbf{81.26 \pm 0.25}$	$\mathbf{+1.40}$
C. 3B: Train–inference 2×2 decomposition							
iPRM, actor-only	Proc	✓	–	Off	74.69	73.32 ± 0.26	–
iPRM + inference control	Proc	✓	–	Learned	74.13	73.56 ± 0.29	$+0.24$
RILO, controller-off	Proc	✓	✓	Off	76.47	75.36 ± 0.31	$+2.04$
Full RILO	Proc	✓	✓	Learned	**77.20**	$\mathbf{76.09 \pm 0.30}$	$\mathbf{+2.77}$

alone. RILO further improves consistently across all four MMAD constituent datasets, with particularly clear gains on MVTec-LOCO and GoodsAD 3), showing that the overall improvement is not dominated by a single dataset.

Zero-shot results. Without a normal reference, RILO-4B achieves the highest zero-shot average accuracy of 79.52, compared with JUDO-7B (79.41), OmniAD-7B (79.10), and the matched Qwen3-VL backbone (71.07). It also obtains the best Defect Classification score of 73.71. Relative to the matched backbone, RILO improves the overall average by 8.45 points and the defect-oriented mean from 64.08 to 77.96 (+13.89). These results show that RILO remains effective in fine-grained anomaly reasoning even without reference-image guidance.

4.2 Ablation Studies

Table 3 examines three aspects of RILO: the contribution of structured process supervision and implicit reward learning, the effect of closed-loop visual modulation on policy learning, and the additional role of inference-time control. We use the 4B model to attribute the main RILO result, while

the matched 3B experiments provide a more detailed train–inference decomposition and controller analysis. Per-seed results are reported in Appendix G.

Effect of process supervision and implicit reward learning. Starting from F-SFT (67.49), answer-only RL reaches 68.00 Avg., while the structured process target improves it to 70.62, showing the value of evidence and consistency supervision beyond final-answer correctness. Learning the same target through iPRM further raises Avg. to 73.32 (+2.70 over Process-terminal RL) without visual modulation or inference-time control, with Def. Avg. increasing from 68.92 to 74.69. Appendix H further shows that the learned reward responds to non-answer evidence errors and visual-condition changes under fixed responses, rather than relying solely on final-answer correctness.

Visual modulation improves policy learning. On the 4B main model, VM contributes primarily through training. With inference control disabled, enabling VM improves Avg. from 79.86 to 81.10 (+1.24) and Def. Avg. from 77.93 to 79.04 (+1.11). Inference control alone yields only +0.05 Avg. without VM training, but adds +0.16 after VM training, bringing Full RILO to 81.26. This positive interaction (+0.11) supports RILO as a closed-loop training mechanism that shapes the learned policy, with inference-time control providing a smaller complementary gain. The 3B decomposition exhibits the same pattern: training-time VM contributes +2.04 Avg., compared with +0.24 from inference control alone, while their combination reaches 76.09 (+2.77) with a positive interaction of +0.49. The interaction is positive across all three paired seeds (Appendix G), showing that the training-dominant effect of VM is consistent across model scales. Matched inference-control ablations further rule out repeated re-prefill and generic perturbations as explanations for the additional test-time gain; in particular, Learned exceeds the Identity re-prefill control by 0.66 ± 0.02 across three seeds The detail is shown in Appendix G. We further discussed the relationship between training stability and VM contribution in Appendix L.

4.3 Qualitative Analysis

Figure 4 shows a case where final-answer accuracy does not distinguish the two policies: both iPRM actor-only and Full RILO select the correct option B. However, the actor-only response contains localization and description errors, whereas RILO produces intermediate evidence better aligned with the observed defect while preserving the same answer.

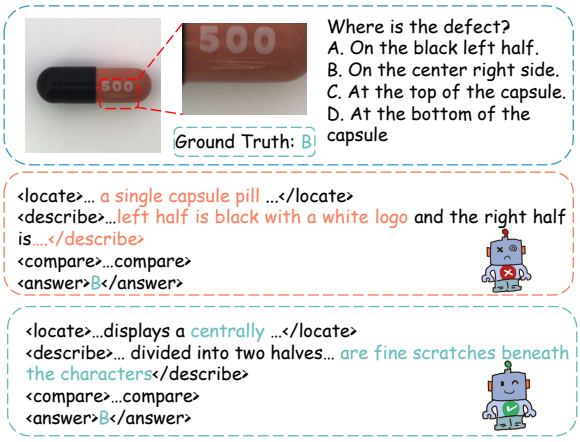


Figure 4: Intermediate evidence under equal answer correctness.

This case illustrates the learned effect of closed-loop RILO training. By feeding implicit process rewards back into rollout generation through visual modulation, RILO shapes the policy toward more evidence-consistent intermediate reasoning. This is consistent with the quantitative ablation showing that VM training improves the actor even when inference control is disabled. More case studies are shown in Appendix N

5 Conclusion

Industrial anomaly reasoning requires visual evidence to adapt with the evolving reasoning state, while standard MLLMs largely rely on fixed visual representations. RILO closes this loop by feeding implicit process rewards back into rollout generation for both policy learning and visual modulation. It achieves 81.26 and 79.52 Avg. on one-shot and zero-shot MMAD, respectively, with pronounced gains on defect-oriented tasks. Ablations attribute the gains to process credit and training-time visual modulation, while matched controls rule out re-prefilling alone. Future work will explore more efficient visual updates and broader multimodal reasoning tasks.

AI USE STATEMENT

Generative AI tools were used to assist with several aspects of this work. Specifically, they were used for code editing and debugging; preliminary mathematical reasoning and methodological discussion; reviewing experimental design and ablation settings for rigor and potential confounding factors; language editing, translation, and manuscript organization; literature discovery and reference checking; and assembling the manuscript and appendix into a single LaTeX document with cross-references. AI tools were used as assistive tools rather than as autonomous decision makers. All methodological choices, mathematical formulations, experimental protocols, reported results, and scientific conclusions were determined and verified by the authors. AI-assisted code modifications were manually inspected and validated through execution and experiments. Mathematical derivations and methodological suggestions were independently checked by the authors. Literature identified with AI assistance was verified against the original papers or official sources, and all reported results and interpretations were checked against the underlying experimental data. All AI-assisted text and translations were reviewed and revised by the authors. The authors are responsible for the final content of this submission, including all AI-assisted text, code, and artifacts.

REFERENCES

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4971–4980, 2018.

Anonymous. DCR^2-AD: Dynamic context routing and reasoning multi-modal large language model for anomaly detection. OpenReview submission, 2026. URL https://openreview.net/forum?id=qqlWHIjJ24. Under review at ICLR 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-VL technical report, 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. doi: 10.1109/CVPR.2019.00982.

Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. *International Journal of Computer Vision*, 130(4):947–969, 2022. doi: 10.1007/s11263-022-01578-9.

Remi Cadene, Corentin Dancette, Hedi Ben-Younes, Matthieu Cord, and Devi Parikh. RUBi: Reducing unimodal biases in visual question answering. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Yuhao Chao, Jie Liu, Jie Tang, and Gangshan Wu. Anomalyr1: A grpo-based end-to-end mllm for industrial anomaly detection, 2025.

Peng Chen, Chao Huang, Yunkang Cao, Chengliang Liu, Wenqiang Wang, Mingbo Yang, Li Shen, Wenqi Ren, and Xiaochun Cao. Reason-iad: Knowledge-guided dynamic latent reasoning for explainable industrial anomaly detection, 2026.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=9SkkifLopZ.

GLM-V Team, Wenyi Hong, Wenmeng Yu, et al. GLM-4.5V and GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2025.

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. doi: 10.1609/aaai.v38i3.27963.

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.

Chao Huang, Yanhui Li, Yunkang Cao, Wei Wang, Hongxi Huang, Jie Wen, Wenqi Ren, and Xiaochun Cao. M3-AD: Reflection-aware multi-modal, multi-category, and multi-dimensional benchmark and framework for industrial anomaly detection, 2026.

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. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=JDiER86r8v.

Hyunju Kang, Woohyun Lee, Jaewon Kim, and Hogun Park. JUDO: A juxtaposed domain-oriented multimodal reasoner for industrial anomaly qa, 2026.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. In *International Conference on Learning Representations*, 2025.

Kimi Team, Angang Du, Bohong Yin, et al. Kimi-VL technical report, 2025.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer, 2024.

Yanhui Li, Yunkang Cao, Chengliang Liu, Yuan Xiong, Xinghui Dong, and Chao Huang. IAD-R1: Reinforcing consistent reasoning in industrial anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 6583–6591, 2026. doi: 10.1609/aaai.v40i8.37588.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.20.

Jingyi Liao, Yongyi Su, Rong-Cheng Tu, Zhao Jin, Wenhao Sun, Yiting Li, Xun Xu, Dacheng Tao, and Xulei Yang. AD-FM: Multimodal LLMs for anomaly detection via multi-stage reasoning and fine-grained reward optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15234–15242, 2026. doi: 10.1609/aaai.v40i18.38548.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916, 2023.

Junwen Miao, Penghui Du, Yi Liu, Yu Wang, and Yan Wang. Agentiad: Tool-augmented single-agent for industrial anomaly detection, 2025.

Zeru Shi, Kai Mei, Yihao Quan, Dimitris N. Metaxas, and Ruixiang Tang. Improving visual reasoning with iterative evidence refinement, 2026.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, 2022.

Dingrong Wang, Xian Tao, Zhen Qu, Hengliang Luo, Xinyi Gong, Fei Shen, Zhengtao Zhang, and Guiguang Ding. DifferAD-R1: A difference-guided industrial anomaly localization with multimodal large language models, 2026.

Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, Lewei Lu, Haodong Duan, Yu Qiao, Jifeng Dai, and Wenhai Wang. Visualprm: An effective process reward model for multimodal reasoning, 2025.

Qiong Wu, Xiangcong Yang, Yiyi Zhou, Chenxin Fang, Baiyang Song, Xiaoshuai Sun, and Rongrong Ji. Grounded chain-of-thought for multimodal large language models, 2025.

Xinglong Yang, Zhilin Peng, Zhanzhan Liu, Haochen Shi, and Sheng-Jun Huang. History-guided iterative visual reasoning with self-correction, 2026a.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33510–33520, 2026b.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73511–73525. PMLR, 2025. URL https://proceedings.mlr.press/v267/yuan25c.html.

Jian Zhang, Runwei Ding, Miaoju Ban, and Linhui Dai. PKU-GoodsAD: A supermarket goods dataset for unsupervised anomaly detection and segmentation. *IEEE Robotics and Automation Letters*, 9(3):2008–2015, 2024. doi: 10.1109/LRA.2024.3352358.

Shifang Zhao, Yiheng Lin, Lu Han, Yao Zhao, and Yunchao Wei. Omniad: Detect and understand industrial anomaly via multimodal reasoning, 2025.

Jinguo Zhu, Weiyun Wang, Zhe Chen, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025.

Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. SPot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*, pp. 392–408, 2022. doi: 10.1007/978-3-031-20056-4_23.

APPENDIX

This appendix provides the implementation and evaluation details omitted from the main paper for space. It first makes the supervision, verifier, visual controller, and dynamic-state protocol explicit. It then specifies the additional robustness, sensitivity, efficiency, and stability analyses used to examine the method's assumptions. Unless noted otherwise, all reported accuracies follow the official MMAD protocol.

A DATA, SUPERVISION, AND LEAKAGE CONTROL

A.1 FROZEN DATA SOURCES

Structured SFT and online RL each use 1,050 QA instances. The SFT set contains 978 unique query images. The RL set contains 984 unique query images and 824 normal-reference images. The two stages share no QA instance; 143 query images occur in both stages. This limited image reuse does not cross the evaluation boundary: neither training stage overlaps with the full MMAD evaluation in QA identity, query identity, reference identity, or cross-role image identity.

The full evaluation comprises 19,080 questions, 3,968 unique queries, and 2,368 normal references. The frozen analysis set contains 700 QA instances, 700 unique queries, and 623 normal references sampled from this evaluation pool. It is therefore a subset of full MMAD rather than an independent test set, but it remains fully image-disjoint from SFT and RL. We use it only for controlled inference analyses and do not tune or update model parameters on its outcomes.

Table 4: Frozen data sources used by the reported experiments.

Split	QA	Query	Ref.	Use
SFT	1,050	978	0	tuning
Online RL	1,050	984	824	RL updates
Analysis	700	700	623	controls
Full MMAD	19,080	3,968	2,368	final eval.

A.2 SUPERVISION BOUNDARY

Each training response contains four fields, ⟨`locate`, `describe`, `compare`, `answer`⟩. Textual reference evidence is used only to construct the verifier target. It is not included in the policy prompt and is unavailable to the controller at inference. No mask, bounding box, defect crop, or human-authored reasoning trace is supplied. The normal image in the one-shot setting is a same-category reference and always precedes the query image.

B REPRODUCIBILITY DETAILS

B.1 PROMPT DESIGN

SFT presents the query image followed by the question and answer options. During one-shot RL and evaluation, the first image is a same-category normal reference and the second is the query; the prompt explicitly instructs the model to answer about the query. Every response must follow

```
<locate>region</locate>
<describe>visible evidence</describe>
<compare>evidence--option match</compare>
<answer>X</answer>, X in {A,B,C,D}.
```

RILO-3B uses one concise sentence in each evidence field. RILO-4B uses the same contract during RL but requests one precise localization sentence and two evidence-rich sentences in each of the description and comparison fields; it suggests 120–140 tokens only when supported by the image.

Algorithm 1: RILO Online Training

Input: dataset $\mathcal{D}$, actor π_θ, reward branch π_ϕ, frozen reference π_{SFT}, verifier $\mathcal{V}$, rollouts K, chunk length C, groups G
for $n = 1, \ldots, N_{\mathrm{iter}}$ **do**
  $\pi_{\theta_{\mathrm{old}}} \leftarrow \mathrm{stopgrad}(\pi_\theta)$; $\mathcal{B} \leftarrow \mathrm{Sample}(\mathcal{D})$
  $(H^n, H^{q,(0)}) \leftarrow \mathrm{Encode}(\mathcal{B})$
  for $j = 1, \ldots, K$ **do**
    $y_j \leftarrow \varnothing, \mathcal{R}_j \leftarrow \varnothing, H_j^q \leftarrow H^{q,(0)}$
    while y_j is not terminated **do**
      $\Delta y_j \leftarrow \mathrm{Decode}_C(\pi_{\theta_{\mathrm{old}}}; \mathcal{B}, y_j, H^n, H_j^q)$
      $\mathcal{R}_j \leftarrow \mathcal{R}_j \cup \{(\Delta y_j, H_j^q)\}$; $y_j \leftarrow y_j \| \Delta y_j$
      if y_j is not terminated **then**
        $\mathbf{q}_j^{\mathrm{vis}} \leftarrow \beta_{\mathrm{vis}}(\log \pi_\phi - \log \pi_{\mathrm{SFT}})_{1:|y_j|}$
        $(P_j, \mathbf{w}_j) \leftarrow \mathrm{GroupProfile}(\mathbf{q}_j^{\mathrm{vis}}; G)$
        $(H_j^q, \mathrm{KV}_j) \leftarrow \mathrm{PatchAndRePrefill}(H^{q,(0)}, P_j, \mathbf{w}_j)$
      end if
    end while
    $s_{\mathrm{proc},j} \leftarrow \mathcal{V}(z(y_j), a^\star, e^\star)$
  end for
  $(\ell_\phi, \ell_{\theta_{\mathrm{old}}}, \ell_{\mathrm{SFT}}) \leftarrow \mathrm{Replay}(\{y_j, \mathcal{R}_j\}_{j=1}^K)$
  $q^{\mathrm{cache}} \leftarrow \mathrm{stopgrad}(\ell_\phi - \ell_{\mathrm{SFT}})$
  $\mathcal{L}_{\mathrm{RM}} \leftarrow \mathrm{BCE}\left(\sigma\left(\beta_{\mathrm{RM}} \sum_t (\ell_{\phi,t} - \ell_{\mathrm{SFT},t})\right), s_{\mathrm{proc}}\right)$
  $\phi \leftarrow \phi - \eta_\phi \nabla_\phi \mathcal{L}_{\mathrm{RM}}$
  $(r^{\mathrm{imp}}, r^{\mathrm{ver}}, r^{\mathrm{kl}}) \leftarrow \left(q^{\mathrm{cache}}, s_{\mathrm{proc}} \mathbf{1}[t = T], -\lambda_{\mathrm{KL}}(\ell_{\theta_{\mathrm{old}}} - \ell_{\mathrm{SFT}})\right)$
  $A \leftarrow \mathrm{SourceRLOO}(r^{\mathrm{imp}}, r^{\mathrm{ver}}, r^{\mathrm{kl}})$
  $\ell_\theta \leftarrow \mathrm{Replay}(\pi_\theta; \{y_j, \mathcal{R}_j\}_{j=1}^K)$; $\rho \leftarrow \exp(\ell_\theta - \ell_{\theta_{\mathrm{old}}})$
  $\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}_{\mathrm{pol}}(\rho, A)$
end for
Output: optimized actor π_θ and controller $\mathcal{C}_{\mathrm{vis}}(\pi_\phi, \pi_{\mathrm{SFT}})$

Evaluation fixes the concise template for every compared checkpoint and forbids text outside the four fields.

B.2 Exact training configuration

Both backbones first undergo three epochs of SFT on the four-field targets. For each backbone, the resulting SFT adapter initializes the actor and implicit-reward branch, which are optimized separately during online RL, while a frozen copy serves as the SFT reference. Each prompt batch contains 70 prompts, with four rollouts per prompt. Actor and reward updates use mini-batches of 70 and 280 responses, respectively, with gradients accumulated over micro-batches. Table 5 summarizes the training configurations and evaluation checkpoints. Each evaluation pairs the actor and implicit-reward adapters from the same training step.

SFT, actor, and reward optimization use AdamW. The SFT and actor weight decay is 0.01, while the reward-model weight decay is zero. RL uses LoRA rank 8, scale 16, and configured dropout 0.05 on `{q,k,v,o,gate,up,down}_proj`; all visual modules are excluded. Actor and reward learning rates remain constant with no warmup. Training and rollout inference use BF16. Rollouts use temperature 1, top-$p = 1$, and top-$k = -1$; the verifier, implicit-reward, and KL sources each enter RLOO with coefficient 1. The image processor preserves aspect ratio within a 50,176–1,003,520-pixel budget. The controller never modifies the normal-reference embeddings. The principal 3B run uses framework seed '3407'; the matched multi-seed 3B ablations are reported in Section G. The strict 4B execution restores RNG state at every full-checkpoint continuation and uses 3407 as the request-local rollout seed base.

B.3 Checkpoint selection

With 1,050 QA instances and a prompt batch of 70, each epoch contains 15 rollout batches. The online RL schedules are configured for eight epochs for RILO-3B and 13 epochs for RILO-4B. The reported full evaluations use the step-119 actor and matching implicit-reward adapter for RILO-3B,

Table 5: Principal training configurations and evaluation checkpoints. The response limits refer to online RL; evaluation uses up to 192 new tokens for both backbones. The 4B run additionally enforces strict chunk-state replay as described in Section E.

Configuration	RILO-3B	RILO-4B
Backbone	Qwen2.5-VL-3B-Instruct	Qwen3-VL-4B-Instruct
SFT epochs / batch / learning rate	3 / 4 / 5×10^{-6}	3 / 4 / 5×10^{-6}
SFT LoRA $(r, \alpha, \text{dropout})$	$(8, 16, 0)$	$(8, 16, 0)$
RL QA / configured epochs	1,050 / 8	1,050 / 13
Evaluation checkpoint step	119	180
Prompt batch / rollouts per prompt	70 / 4	70 / 4
Actor / reward mini-batch	70 / 280	70 / 280
Actor / reward micro-batch	2 / 2	1 / 2
Actor / reward learning rate	3×10^{-5} / 5×10^{-5}	3×10^{-5} / 5×10^{-5}
PPO epochs / clip / entropy coefficient	1 / 0.2 / 0	1 / 0.2 / 0
KL coefficient / source coefficients	0.001 / $(1, 1, 1)$	0.001 / $(1, 1, 1)$
Actor / reward gradient clip	1 / 10	1 / 10
Maximum prompt / response (RL)	1,024 / 192 tokens	1,024 / 160 tokens
Images / target image	2 / query (index 1)	2 / query (index 1)
Controller (G, α, C)	$(20, 0.01, 64)$	$(20, 0.01, 64)$
Visual-state execution	request-local per-rollout	strict protocol-v2 replay

and the corresponding step-180 pair for RILO-4B. Adapter versions and hashes are recorded in each evaluation configuration. For the segmented 4B execution, continuation occurs only from committed full checkpoints and restores actor, reward model, optimizers, learning-rate schedulers, global step, data cursor, and RNG state. Uncommitted partial segments are excluded from the reported training trajectory.

C Deterministic Process Verifier

C.1 Strict parsing gate

A response is valid only if it consists of exactly one instance of each field in the order

$$\texttt{<locate>}\ \texttt{<describe>}\ \texttt{<compare>}\ \texttt{<answer>}, \tag{16}$$

with no text before the first field or after the last. The answer field must contain one letter in $\{A, B, C, D\}$. Missing, repeated, reordered, or unknown tags fail the parser and receive zero direct verifier reward.

C.2 Lexical evidence match

For a string z, let $T(z)$ be its lowercased alphanumeric token set after removing a fixed 20-word English stoplist and one-character tokens. For a candidate field z and a set of reference strings E, the production matcher is the maximum overlap coefficient

$$M(z, E) = \max_{e\in E} \frac{|T(z) \cap T(e)|}{\max(1, \min(|T(z)|, |T(e)|))}. \tag{17}$$

The location references combine the annotated location slot, location text, and affected part. Description references combine the description slot and visual attribute. Comparison references combine the comparison slot, correct answer text, and defect type.

Let m_{loc} and m_{desc} denote the first two field scores. The comparison score is set to zero for an incorrect selected option. Otherwise, it is the maximum of the match against the comparison references and the correct answer text:

$$\begin{aligned} m_{\text{cmp}} = \mathbf{1}[\hat{a} = a^\star] \\ \cdot \max\{M(z^{\text{cmp}}, E_{\text{cmp}}), M(z^{\text{cmp}}, E_{\text{ans}})\}. \end{aligned} \tag{18}$$

The evidence component is

$$s_{\text{evi}} = \frac{m_{\text{loc}} + m_{\text{desc}} + m_{\text{cmp}}}{3}. \tag{19}$$

C.3 ANSWER AND CONSISTENCY COMPONENTS

The answer component is

$$s_{\text{ans}} = \mathbf{1}[\hat{a} = a^\star].$$

The consistency component equals one only when the selected answer is correct, the comparison field has positive overlap with the reference evidence, and the normalized comparison string does not contain the explicit contradiction pattern "rather than [correct answer text]":

$$s_{\text{con}} = \mathbf{1}[\hat{a} = a^\star]\,\mathbf{1}[m_{\text{cmp}} > 0]\,\mathbf{1}[\neg\,\text{Contradict}(z^{\text{cmp}}, a^\star)]. \tag{20}$$

For a format-valid response, the process target is

$$s_{\text{proc}} = 0.6 s_{\text{ans}} + 0.2 s_{\text{evi}} + 0.2 s_{\text{con}}. \tag{21}$$

Because the comparison score is gated by answer correctness, an incorrect answer satisfies

$$\hat{a} \neq a^\star \quad \Longrightarrow \quad s_{\text{ans}} = s_{\text{con}} = m_{\text{cmp}} = 0.$$

Consequently,

$$s_{\text{proc}} = \frac{0.2}{3}\,(m_{\text{loc}} + m_{\text{desc}}) \leq \frac{2}{15} \approx 0.133 \qquad \text{when } \hat{a} \neq a^\star. \tag{22}$$

Thus, incorrect responses can retain limited localization or description credit, but cannot receive comparison, consistency, or answer credit.

In the principal runs, `direct_reward_mode=process_target`, so $s_{\text{proc},i}$ is used as the terminal verifier reward g_i for policy optimization. The same scalar serves as the soft target for implicit-reward learning.

The verifier is intentionally deterministic and lightweight rather than a general semantic evaluator. Its evidence matcher in Eq. 17 relies on token-set overlap and is therefore insensitive to semantic corruptions that preserve the matched token sets, while Contradict detects only the predefined contradiction template. We use the verifier as a structured training signal and do not claim general semantic verification or robustness to arbitrary or adaptive reward-hacking strategies.

C.4 PROCESS-TARGET WEIGHTING

We parameterize the process target as

$$s_{\text{proc}} = \lambda_{\text{ans}} s_{\text{ans}} + \lambda_{\text{evi}} s_{\text{evi}} + \lambda_{\text{con}} s_{\text{con}}, \tag{23}$$

with the default weighting $(\lambda_{\text{ans}}, \lambda_{\text{evi}}, \lambda_{\text{con}}) = (0.6, 0.2, 0.2)$.

The weighting is designed to keep final-answer correctness necessary for a high process score while retaining graded supervision from the evidence and consistency components. In particular, an incorrect answer forces $s_{\text{ans}} = s_{\text{con}} = m_{\text{cmp}} = 0$. Under the default weighting,

$$s_{\text{proc}} = \frac{0.2}{3}\,(m_{\text{loc}} + m_{\text{desc}}) \leq \frac{2}{15} \approx 0.133, \tag{24}$$

so partial localization or description evidence cannot compensate for an incorrect final decision. More generally, for the weightings tested below, increasing λ_{ans} decreases the maximum process credit available to an incorrect response while reducing the relative contribution of evidence and consistency supervision.

To test whether performance depends on the exact default allocation, we conduct a local sensitivity analysis on RILO-3B while keeping the evidence and consistency terms equally weighted. We compare the default $(0.6, 0.2, 0.2)$ against neighboring allocations $(0.5, 0.25, 0.25)$ and $(0.7, 0.15, 0.15)$, with all other training and evaluation settings fixed.

As shown in Table 6, the three neighboring allocations yield nearly identical performance: 76.05, 76.09, and 76.08 Avg., with a maximum spread of only 0.10 points. The default weighting therefore does not appear to be a sensitive operating point within this local range. We retain $(0.6, 0.2, 0.2)$ because it preserves a simple design in which answer correctness receives the largest weight, while evidence and consistency share the remaining supervision equally.

This experiment is intended as a local sensitivity check rather than an exhaustive search over reward compositions; it does not imply invariance to substantially different weightings.

Table 6: Local sensitivity to the process-target weights on RILO-3B. The evidence and consistency terms remain equally weighted.

λ_{ans}	λ_{evi}	λ_{con}	Avg.
0.50	0.25	0.25	76.05
0.60	0.20	0.20	76.09
0.70	0.15	0.15	76.08

D Visual-Control Profile and Patch

At a chunk boundary, the reward adapter and frozen SFT reference score the current response prefix under the current request-local visual state. For prefix token u, the implicit visual score is

$$q_u^{\text{vis}} = \beta_{\text{vis}}(\log \pi_\phi(y_u \mid c_u) - \log \pi_{\text{SFT}}(y_u \mid c_u)). \tag{25}$$

The prefix potential is $P = |y_{\le t}|^{-1} \sum_{u \le t} q_u^{\text{vis}}$. We first form token risks $r_u = \max(0, -q_u^{\text{vis}})$. If every risk is zero, the implementation uses $|q_u^{\text{vis}}|$; if these values are also zero, it returns uniform weights. Otherwise, the risk trajectory is linearly interpolated in token order to G values and normalized to unit mean:

$$\tilde{\mathbf{r}} = \text{LinearInterp}_G(\mathbf{r}), \qquad w_g = \frac{G\tilde{r}_g}{\sum_{h=0}^{G-1} \tilde{r}_h}. \tag{26}$$

This is a deterministic statistic of token-level implicit rewards; it is not a softmax and introduces no additional learned network.

For N post-merger query-image tokens, token j belongs to the contiguous token-order group

$$g(j) = \min\left(G - 1, \left\lfloor \frac{Gj}{N} \right\rfloor\right). \tag{27}$$

The implementation reconstructs each chunk state from the unmodified query embeddings and applies an additive scalar residual across the feature dimension:

$$h_j^{q,(k+1)} = h_j^{q,(0)} + \alpha P^{(k)} w_{g(j)}^{(k)} \mathbf{1}_D. \tag{28}$$

It neither replaces the embeddings nor multiplies them. The normal-reference tokens are unchanged. Since Eq. 28 changes the multimodal prefix, the decoder KV cache is invalidated and the prompt plus accumulated response is re-prefilled before generating the next chunk. The default uses $G = 20$, $\alpha = 0.01$, and $C = 64$ generated tokens per update. All profiles, weights, and patches are request-local even when different prompts share an active vLLM batch.

E Dynamic Visual-State Replay

The strict 4B implementation stores a protocol-v2 trace for each rollout. Its immutable header binds sample and rollout identity, query/reference image hashes, prompt-token hash, actor/reward/reference adapter hashes, and decoding and processor hashes. Each chunk record stores its token span, scored-state hash, current/previous potential, raw token rewards, region weights, patch scale, target image index, generated token IDs, behavior log probabilities, and patch diagnostics. A canonical JSON hash identifies the complete state.

Before an actor update, the frozen rollout snapshot replays every token under the same chunk-specific query state used for its generation. These Transformers replay log probabilities define PPO's old policy. The behavior log probabilities returned by external vLLM are retained as a cross-engine audit rather than substituted for the same-engine PPO denominator. The reward branch and frozen SFT reference replay the same chunk states for the implicit score and KL reward. When the replay actor equals the rollout snapshot, the same-engine ratio must remain within 10^{-4} of one. The cross-engine audit uses absolute log-probability error gates of 0.02 at the median, 0.16 at the 95th percentile, and 0.35 at the 99th percentile; its ratio mean and 95th-percentile limits are 0.035 and 0.16.

These strict parity gates describe the Qwen3-VL-4B principal implementation. The Qwen2.5-VL-3B ablations use the matched request-local per-rollout external-vLLM controller and are not used as evidence for the 4B replay parity thresholds.

Table 7: Three-seed stability of the reward-supervision ablation. The first four rows report absolute full-MMAD Avg.; the remaining rows report paired seed-wise gains. The last row reports the paired Def. Avg gain used in the main-text process-credit analysis.

Configuration or paired comparison	Seed 1	Seed 2	Seed 3	Mean ± Std.
F-SFT	67.18	67.53	67.76	67.49 ± 0.29
Answer-only RL	67.72	67.98	68.30	68.00 ± 0.29
Process-terminal RL	70.26	70.69	70.91	70.62 ± 0.33
iPRM actor-only	73.03	73.41	73.52	73.32 ± 0.26
Answer-only RL − F-SFT	+0.54	+0.45	+0.54	$+0.51 \pm 0.05$
Process-terminal RL − Answer-only RL	+2.54	+2.71	+2.61	$+2.62 \pm 0.09$
iPRM actor-only − Process-terminal RL	+2.77	+2.72	+2.61	$+2.70 \pm 0.08$
Def. Avg: iPRM actor-only − Process-terminal RL	+5.62	+5.80	+5.89	$+5.77 \pm 0.14$

F Evaluation Protocol

One-shot evaluation places a randomly selected same-category normal image before the query; zero-shot evaluation omits the normal reference. The four-field prompt and official choice parser are shared across matched comparisons. Unless a sensitivity variable is named explicitly, evaluation uses greedy decoding (temperature 0, top-$p = 1$), 192 maximum new tokens, $G = 20$, $\alpha = 0.01$, and $C = 64$. The 3B and 4B full one-shot evaluations use batches of 160 and 128, respectively. Batch size is an execution setting: every controller profile and patch remains associated with its own request.

Anomaly Discrimination (AD) follows MMAD's balanced protocol: normal and anomalous accuracies are averaged before combining AD with the six remaining subtasks. The overall average is the unweighted mean of these seven task scores. Defect Avg. is the unweighted mean of Defect Classification, Localization, Description, and Analysis. For paired significance analyses, we cluster bootstrap by query-image identity because multiple QA rows can share the same visual query.

G Multi-Seed Stability of the Causal Ablations

We use the same three random seeds across every matched 3B configuration and hold the training data, rollout budget, optimization schedule, checkpoint-selection rule, and evaluation protocol fixed. This paired design allows us to report not only the marginal mean and sample standard deviation of each configuration, but also the seed-wise difference for each causal comparison. Because three runs provide limited power for formal significance testing, we emphasize effect direction, paired-difference dispersion, and whether the ordering is preserved across all seeds.

The paired reporting separates absolute run-to-run variation from the stability of the ablation effect itself. The reported ordering is preserved across all three seeds: process-terminal RL exceeds answer-only RL, and iPRM actor-only exceeds process-terminal RL in every seed. The Def. Avg comparison should be interpreted in the same paired manner rather than inferred from the two marginal standard deviations.

For seed s, the interaction is computed as

$$\begin{aligned}\Delta_{\text{int}}^{(s)} = &\big(M_{\text{VM on,Ctrl on}}^{(s)} - M_{\text{VM on,Ctrl off}}^{(s)}\big) \\ &- \big(M_{\text{VM off,Ctrl on}}^{(s)} - M_{\text{VM off,Ctrl off}}^{(s)}\big).\end{aligned} \tag{29}$$

The interaction is positive for all three seeds, supporting the complementarity between training-time visual modulation and inference-time learned control in this matched decomposition.

Controller specificity is supported most directly when Learned ranks first for every seed and all paired differences in the lower block remain positive. This evidence argues against repeated re-prefilling alone, uniform scaling, or arbitrary coefficient perturbations as the primary explanation; it does not logically exclude every possible implementation-level confound.

Table 8: Three-seed 2×2 decomposition of training-time visual modulation (VM) and inference-time learned control. The upper block reports absolute Avg.; the lower block reports seed-wise main effects and the interaction.

VM_{tr}	Inference control	Seed 1	Seed 2	Seed 3	Mean $\pm$ Std.
Off	Off	73.03	73.41	73.52	73.32 ± 0.26
Off	Learned	73.24	73.63	73.81	73.56 ± 0.29
On	Off	75.02	75.44	75.62	75.36 ± 0.31
On	Learned	75.76	76.18	76.33	76.09 ± 0.30
VM effect, control off		$+1.99$	$+2.03$	$+2.10$	$+2.04 \pm 0.06$
VM effect, control on		$+2.52$	$+2.55$	$+2.52$	$+2.53 \pm 0.02$
Control effect, VM off		$+0.21$	$+0.22$	$+0.29$	$+0.24 \pm 0.04$
Control effect, VM on		$+0.74$	$+0.74$	$+0.71$	$+0.73 \pm 0.02$
Interaction		$+0.53$	$+0.52$	$+0.42$	$+0.49 \pm 0.06$

Table 9: Three-seed controller-specificity analysis. The upper block reports absolute Avg. for the learned controller and matched controls; the lower block reports the paired advantage of Learned over each alternative.

Controller or paired comparison	Seed 1	Seed 2	Seed 3	Mean $\pm$ Std.
Off	75.02	75.44	75.62	75.36 ± 0.31
Identity	75.10	75.50	75.69	75.43 ± 0.30
Uniform	75.02	75.43	75.66	75.37 ± 0.32
Shuffled	74.91	75.31	75.56	75.26 ± 0.33
Random	75.00	75.45	75.69	75.38 ± 0.35
Learned	**75.76**	**76.18**	**76.33**	$\mathbf{76.09 \pm 0.30}$
Learned $-$ Off	$+0.74$	$+0.74$	$+0.71$	$+0.73 \pm 0.02$
Learned $-$ Identity	$+0.66$	$+0.68$	$+0.64$	$+0.66 \pm 0.02$
Learned $-$ Uniform	$+0.74$	$+0.75$	$+0.67$	$+0.72 \pm 0.04$
Learned $-$ Shuffled	$+0.85$	$+0.87$	$+0.77$	$+0.83 \pm 0.05$
Learned $-$ Random	$+0.76$	$+0.73$	$+0.64$	$+0.71 \pm 0.06$

Table 10: Task-level paired gains of learned control over controller-off. A positive value indicates that the controller does not obtain its average gain by sacrificing the corresponding task family.

Metric	Seed 1	Seed 2	Seed 3	Mean $\pm$ Std.
AD	$+0.31$	$+0.45$	$+0.50$	$+0.42 \pm 0.10$
Def. Avg	$+0.65$	$+0.74$	$+0.80$	$+0.73 \pm 0.08$
Obj. Cls.	$+1.39$	$+1.58$	$+1.68$	$+1.55 \pm 0.15$
Obj. Anal.	$+0.05$	$+0.18$	$+0.25$	$+0.16 \pm 0.10$

The smaller Obj. Anal. effect should not be presented as equally strong as the Obj. Cls. improvement; however, it remains positive for all three seeds. More generally, the relevant paired effects preserve their direction across the three seeds, supporting the reported causal interpretation.

H Implicit-Reward Calibration

H.1 Field-localized textual counterfactuals

We evaluate whether the learned token-level rewards identify *where* controlled errors are introduced in a response, rather than merely assigning a nearly uniform decomposition of the terminal outcome. From the 700-example frozen analysis set, we construct four paired counterfactuals for each response by corrupting exactly one of the `locate`, `describe`, `compare`, or `answer` fields while leaving the other three fields unchanged. This yields 2,800 original–counterfactual pairs.

For each pair, we average the token-level implicit rewards over the tokens aligned to each field's character span. The drop for a field is defined as its mean reward in the original response minus its

Table 11: Field-localized implicit-reward calibration on 700 examples, comprising 2,800 paired counterfactuals. For each corruption, the localized margin is the reward drop in the corrupted field minus the mean drop across the three unchanged fields. Aggregate rows macro-average over the four corruption types.

Metric	Value
Corrupted-field drop	0.598
Unchanged-field drop	0.095
Localized margin	0.503
Cluster-bootstrap 95% CI	[0.490, 0.515]
Field-detection AUROC	0.838
Changed-field Top-1 (%)	67.18
Margin: `locate`	0.110
Margin: `describe`	0.017
Margin: `compare`	0.129
Margin: `answer`	1.757

mean reward in the corresponding counterfactual response. For a counterfactual that corrupts field c, the localized margin is

$$\Delta_c - \frac{1}{3}\sum_{f \neq c} \Delta_f,$$

where Δ_c is the drop in the corrupted field and the remaining Δ_f values are the drops in the three unchanged fields. The aggregate statistics macro-average over the four corruption types. Field-detection AUROC treats the corrupted field as the positive class and the three unchanged fields as negatives, while Changed-field Top-1 reports whether the corrupted field has the largest drop among the four fields. Confidence intervals are computed from 2,000 bootstrap replicates clustered by query-image identity.

As shown in Table 11, corruption reduces the implicit reward of the corrupted field by 0.598 on average, compared with an average drop of 0.095 in the three unchanged fields. This yields an aggregate localized margin of 0.503, with a query-image-cluster bootstrap 95% confidence interval of $[0.490, 0.515]$. The strictly positive interval indicates that the reward change is concentrated in the field where the controlled error is introduced, rather than being distributed uniformly across the response.

The ranking-based metrics provide complementary evidence. The field-detection AUROC is 0.838, and the corrupted field has the largest drop in 67.18% of the counterfactuals. Thus, the learned reward signal identifies the affected field substantially more often than would be expected from a field-insensitive decomposition, although it does not localize every corruption correctly.

Localization strength is highly non-uniform across fields. The `answer` field has the largest localized margin, at 1.757, whereas the margins for `compare` and `locate` are positive but much smaller, at 0.129 and 0.110, respectively. The `describe` margin is close to zero, at 0.017. The aggregate result should therefore be interpreted as evidence of *partial and field-dependent* localization, rather than as uniformly calibrated token-level error detection. In particular, description-level corruptions remain difficult for the learned implicit reward to isolate.

This limitation concerns the field-level calibration of the learned implicit-reward branch and should not be conflated with the explicitly lexical, template-based deterministic verifier described in Section C.

H.2 Visual-condition sensitivity under fixed responses

The field-localized analysis above evaluates whether the learned implicit reward responds to controlled textual errors. We next test a complementary question: whether the learned reward depends on the visual condition itself, rather than being determined only by response text or final-answer correctness.

We construct paired image-swap counterfactuals while keeping the response fixed. For each retained response z_i, the matched condition uses its original query image I_i, whereas the mismatched condi-

Table 12: Visual-condition sensitivity of the frozen implicit-reward branch under fixed responses. Positive Δ_{img} and PairAcc above 50% indicate preference for the visually matched condition.

Subset	Pairs	Δ_{img} [95% CI]	PairAcc (%)
All	379	$+0.20\,[+0.12, +0.28]$	70.5
Defect-oriented	280	$+0.23\,[+0.14, +0.33]$	73.0

tion replaces only the query image with an evidence-incompatible image $I_{j(i)}$. The question, answer options, response tokens, and final answer are unchanged within each pair. We obtain 379 valid pairs in total, of which 280 belong to the defect-oriented subtasks. Pair construction is independent of the implicit-reward score.

We evaluate the frozen reward branch using the response-level pre-sigmoid implicit-reward score

$$R_\phi(I, q, z) = \beta_{\text{RM}} \sum_{t=1}^{|z|} \left[\log \pi_\phi(z_t \mid I, q, z_{<t}) - \log \pi_{\text{SFT}}(z_t \mid I, q, z_{<t})\right]. \tag{30}$$

For each pair, we define

$$\Delta_{\text{img}}^{(i)} = R_\phi(I_i, q_i, z_i) - R_\phi(I_{j(i)}, q_i, z_i), \tag{31}$$

so that a positive value indicates that the reward branch assigns a higher score when the fixed response is paired with its matched visual condition. We additionally report

$$\text{PairAcc}_{\text{img}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[R_\phi(I_i, q_i, z_i) > R_\phi(I_{j(i)}, q_i, z_i)\right]. \tag{32}$$

Confidence intervals for Δ_{img} are computed with paired bootstrap resampling using the same query-image clustering convention as the other controlled analyses.

Because the response text and final answer are identical within each pair, the deterministic process target is unchanged by construction. This experiment therefore isolates whether the *learned* reward branch responds to visual-state differences that are not available to the lexical process verifier.

As shown in Table 12, the learned implicit reward consistently prefers the visually matched condition. Across all 379 pairs, the matched image increases the reward by 0.20 on average, with a paired 95% confidence interval of $[0.12, 0.28]$, and is ranked above the mismatched image in 70.5% of pairs. The same pattern holds on the 280 defect-oriented pairs, where the mean difference increases to 0.23 with a 95% confidence interval of $[0.14, 0.33]$, and PairAcc reaches 73.0%.

Since the response text, final answer, and deterministic process target are fixed within each pair, these differences cannot be explained by answer correctness or textual response content alone. They provide direct evidence that the learned implicit reward uses visual-state information and captures image–response compatibility beyond the lexical supervision used to construct its training target.

The defect-oriented subset shows a numerically larger separation than the full set, but we do not interpret this difference as statistically significant without a direct between-subset test. More generally, this audit establishes sensitivity to the visual condition, not pixel-level grounding or causal faithfulness of the generated reasoning. A lower reward under the mismatched image may reflect global image–response incompatibility rather than localization of the specific defect evidence.

I Controller Sensitivity

We vary one controller factor at a time on the same 700 prompts with the frozen 3B step-119 actor and matching PRM. The local sweep uses $G \in \{10, 20, 40\}$, $\alpha \in \{0.005, 0.01, 0.015\}$, and $C \in \{32, 64, 128\}$ around the default $(20, 0.01, 64)$. All other model, data, prompt, decoding, and batching identities are fixed. Differences and 95% intervals are paired against the default using 10,000 stratified query-image-cluster bootstrap replicates; Anomaly Discrimination retains the official class-balanced definition.

Across this finite neighborhood, point estimates remain within 0.58 accuracy points of the default and no paired interval excludes zero. The result supports local operating stability rather than a claim

Table 13: Local controller sensitivity on the frozen 700-example analysis set. Every paired 95% interval includes zero; all settings retain 100% four-field format validity.

Setting	G	α	C	Avg.	Δ Avg. [95% CI]
$\alpha = 0.005$	20	0.005	64	80.32	$+0.00\,[-0.75, +0.75]$
Default	20	0.010	64	80.32	$+0.00\,[+0.00, +0.00]$
$\alpha = 0.015$	20	0.015	64	80.49	$+0.17\,[-0.43, +0.84]$
$G = 10$	10	0.010	64	80.58	$+0.26\,[-0.43, +1.02]$
$G = 40$	40	0.010	64	79.74	$-0.58\,[-1.42, +0.16]$
$C = 32$	20	0.010	32	80.46	$+0.14\,[-0.38, +0.71]$
$C = 128$	20	0.010	128	80.76	$+0.44\,[-0.14, +1.13]$

Table 14: Matched-mass visual-profile intervention on the selected diagnostic cohort at $\alpha = 0.03$. "Learned gain" is the learned-profile accuracy minus that of the corresponding alternative; brackets denote paired 95% confidence intervals.

Profile	Acc.	Format	Learned gain [95% CI]
Identity	8.67	99.00	$+2.00\,[+0.00, +4.33]$
Learned	10.67	99.00	–
Oracle mask	10.33	99.00	$+0.34\,[-2.33, +3.00]$
Random matched	8.00	98.67	$+2.67\,[+0.00, +5.33]$
Non-defect	7.00	99.00	$+3.67\,[+1.00, +6.67]$

of global hyperparameter invariance. In particular, the downward trend at $G = 40$ and the upward point estimate at $C = 128$ remain statistically unresolved on 700 examples, so we retain the pre-specified default instead of tuning on the analysis subset.

J Visual-Profile Allocation Diagnostic

We conduct a targeted intervention analysis to examine whether the behavior of the frozen controller depends on how modulation mass is distributed across visual-token groups, rather than only on the presence of an additional visual-state perturbation.

The diagnostic cohort contains 300 difficult Defect Localization examples, balanced with 75 query images from each MMAD source dataset. Examples are selected deterministically by intervention-arm disagreement, followed by learned-profile error and sample identifier. This cohort is therefore intentionally enriched for cases in which the choice of intervention can affect the prediction. Its deliberately low accuracy should not be interpreted as an estimate of full-MMAD performance.

With the actor and implicit-reward branch frozen, we compare five intervention profiles: (i) *Identity*, which performs the matched re-prefill without modifying the query representation; (ii) *Learned*, which uses the reward-derived profile produced by RILO; (iii) *Oracle mask*, a privileged diagnostic profile constructed from the ground-truth defect mask; (iv) *Random matched*, which redistributes the same total profile mass randomly across visual-token groups; and (v) *Non-defect*, which allocates the profile toward groups outside the annotated defect region. The mask-based profiles are used only for this post-hoc diagnostic and are never available during training or standard inference.

All non-identity profiles are matched in total modulation mass and differ in how that mass is allocated across visual-token groups. We evaluate each profile at $\alpha \in \{0.01, 0.03, 0.05, 0.10\}$ while keeping the actor, reward branch, prompt, decoding configuration, and evaluation protocol fixed.

Across the intervention-strength sweep, 36.67–44.00% of complete responses change relative to Identity, while four-field format validity remains between 98.67% and 99.33%. These changes confirm that the profile interventions produce non-negligible behavioral effects. Table 14 reports the representative $\alpha = 0.03$ operating point. Paired confidence intervals are computed from 20,000 query-image-cluster bootstrap replicates stratified by source dataset.

At $\alpha = 0.03$, the learned profile obtains the highest point estimate (10.67), followed closely by the oracle-mask profile (10.33). The learned–oracle difference is only $+0.34$ points, with a paired 95%

Table 15: Inference-control cost decomposition on 700 examples using one A800. Times measure batched evaluator wall-clock and exclude model-service startup.

Mode	Time (s)	Rel.	Samples/s	Tokens/s	Peak MiB
Off	260.2	1.00×	2.690	229.84	47,793
Identity	457.5	1.76×	1.530	130.21	47,793
Full	490.4	1.88×	1.427	122.05	65,637

confidence interval of $[-2.33, +3.00]$; we therefore do not claim that either profile outperforms the other.

Relative to Identity and Random matched, the learned profile improves the point estimate by $+2.00$ and $+2.67$ points, respectively, but both paired confidence intervals touch zero. The clearest separation is against the Non-defect profile: allocating the same total profile mass away from the annotated defect region reduces accuracy by 3.67 points relative to Learned, with a paired 95% confidence interval of $[+1.00, +6.67]$.

These results provide a bounded diagnostic of profile-allocation sensitivity. On this deliberately intervention-sensitive cohort, the model's behavior depends on how modulation mass is distributed across visual-token groups, and allocating the profile toward non-defect groups is less effective than using the learned profile. The favorable point estimates of Learned relative to Identity and Random matched are consistent with this interpretation, although their intervals do not exclude zero.

We emphasize that this experiment is not a spatial-grounding evaluation. The learned profile is defined over ordered visual-token groups rather than supervised image regions, and similar task-level effects do not imply pixel-level correspondence with the ground-truth defect mask. Moreover, matched total profile mass does not by itself guarantee identical feature-space perturbation norms across profiles. We therefore use this analysis only to characterize the behavioral sensitivity of the frozen system to profile allocation, rather than to claim that the learned profile recovers the true defect region. Establishing explicit spatial correspondence between learned modulation profiles and localized defect evidence is left for future work.

K Inference-Control Cost Decomposition

We measure the cost of the optional inference-time controller using the same frozen [Qwen2.5-VL-3B step-119] actor on the same 700-example analysis set, with greedy decoding, a maximum response length of 192 tokens, $C = 64$, and the same batched evaluation schedule on one A800 GPU. Off performs standard decoding without intermediate visual control. Identity uses the same 64-token chunk boundaries and repeated multimodal re-prefill while leaving the query representation unchanged. Full follows the same schedule as Identity, additionally evaluates the request-local implicit-reward profile and applies the visual patch. Thus, Off-to-Identity isolates the cost of repeated re-prefill, whereas Identity-to-Full isolates the additional cost of online reward scoring and modulation. Reported times exclude model-service startup and measure batched evaluator wall-clock rather than single-request deployment latency.

Repeated multimodal re-prefill is the dominant source of inference overhead: Identity increases evaluator time by 75.8% relative to Off. Once this matched re-prefill schedule is incurred, online reward scoring and visual modulation add a further 7.2% over Identity (32.9 seconds), of which 25.1 seconds are recorded in the prefix-reward service. Overall, however, Full remains 88.5% slower than Off and requires 17,844 MiB more peak memory. We therefore do not characterize the complete online controller as low-overhead; the decomposition instead identifies repeated multimodal re-prefill as the principal efficiency bottleneck.

Importantly, online control is not required to obtain the main training-time benefit of RILO. On the 4B main model, the VM-trained actor improves Avg. from 79.86 to 81.10 with inference control disabled, while enabling the controller further increases performance to 81.26. Thus, when deployment latency is prioritized, the controller-off actor provides a lower-cost operating point that retains most of the visual-modulation gain. Reducing or avoiding repeated re-prefill remains the primary target for improving the efficiency of optional online control.

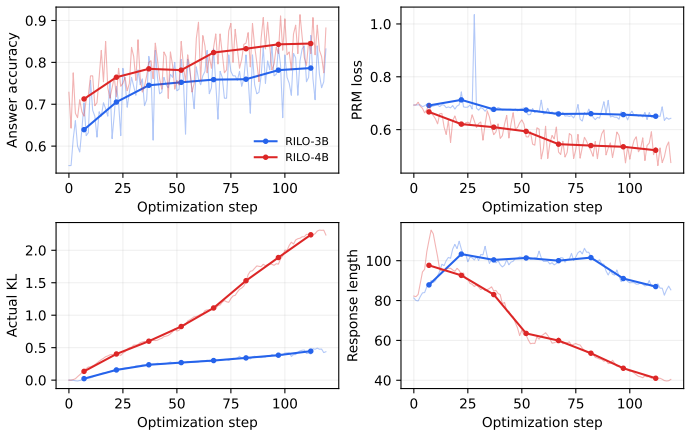


Figure 5: Per-step training diagnostics (light lines) and 15-step pass means (markers). The audit reports accuracy and reward-model improvement together with the observed KL growth and response-length contraction.

L Training Stability Analysis

Stability is computed from immutable step logs rather than rerunning training. For each 15-step data pass, we aggregate answer accuracy, actor and reward losses, KL, clip fraction, response length, and reward moments. The audit also checks finite metrics, zero/skipped updates, checkpoint continuity, and recorded OOM or restart segments. Repeated log records are deduplicated by run identity and global step; missing steps are not interpolated. Strict cross-engine parity metrics are shown only for the 4B implementation and are not merged with the 3B learning curves.

Both audited runs contain all 120 expected steps, with no interpolation, non-finite metric, or conflicting duplicate. The only zero update is the expected initialization step. Table 16 summarizes the first and last data passes. Training accuracy increases by 14.71 and 13.23 points for RILO-3B and RILO-4B, respectively, while the reward-model loss decreases. As shown in figure 5, the 4B logs also expose two trends rather than hiding them: logged KL rises through training and responses become shorter. The response contraction is accompanied by a decline in the measured VM activation rate: early rollouts frequently cross the $C = 64$ chunk boundary and trigger visual modulation, whereas later 4B rollouts do so much less often. Since VM affects subsequent generation only after a chunk boundary, this reduces the number of visual interventions available during the later stage of 4B policy optimization. This trend is consistent with the smaller training-time VM gain observed on 4B ($+1.24$) than on 3B ($+2.04$), and may partly explain the cross-scale difference. However, the two settings also differ in backbone scale, replay implementation, and response-length regime, so we do not attribute the difference solely to VM activation frequency. Its strict same-engine replay ratio error remains exactly zero at every step; the cross-engine hard parity gate passes all 120 steps (mean absolute ratio error 0.019). One failed, uncommitted 4B transaction covering steps 28–34 is retained in the incident ledger but excluded from the official curve; the committed checkpoint chain remains step-continuous.

Table 16: Stability summary from immutable training logs. Accuracy is the mean within a 15-step data pass.

Run	Steps	Pass 1	Pass 8	KL_8	$Len._8$
RILO-3B	120	63.93	78.64	0.446	87.0
RILO-4B	120	71.26	84.49	2.239	41.0

M Relation to Visual-Revisiting Methods

Table 17 organizes related methods by mechanism rather than by reported accuracy, which is often obtained under different backbones and evaluation settings. Mirage interleaves latent visual tokens with text; H-GIVR performs history-guided repeated observation; SIEVE retrieves and reinjects

salient region representations; and Reason-IAD searches latent visual patches with knowledge guidance. RILO differs in using the same learned implicit process signal for dense policy credit and online reconstruction of the request-local query state.

Table 17: Mechanism-level comparison with representative visual-revisiting methods. "Shared" indicates whether the signal for policy learning also controls within-response visual revision.

Method	Visual update	Control signal	Shared	Re-prefill
Mirage	latent-token access	generation trajectory	no	method-dependent
H-GIVR	repeated observation	reasoning history	no	yes
SIEVE	region-feature reinjection	learned salience / retrieval	no	method-dependent
Reason-IAD	latent-patch selection	knowledge-guided reward	no	iterative
RILO	grouped query reconstruction	implicit token rewards	yes	yes

N CASE STUDY

Figure 6 presents five representative cases. Successful examples illustrate fine-grained localization, logical-defect description, and object recognition; failures expose scratch–deformation confusion and compositional spatial-relation errors. They show capabilities and boundaries rather than estimate population-level performance.

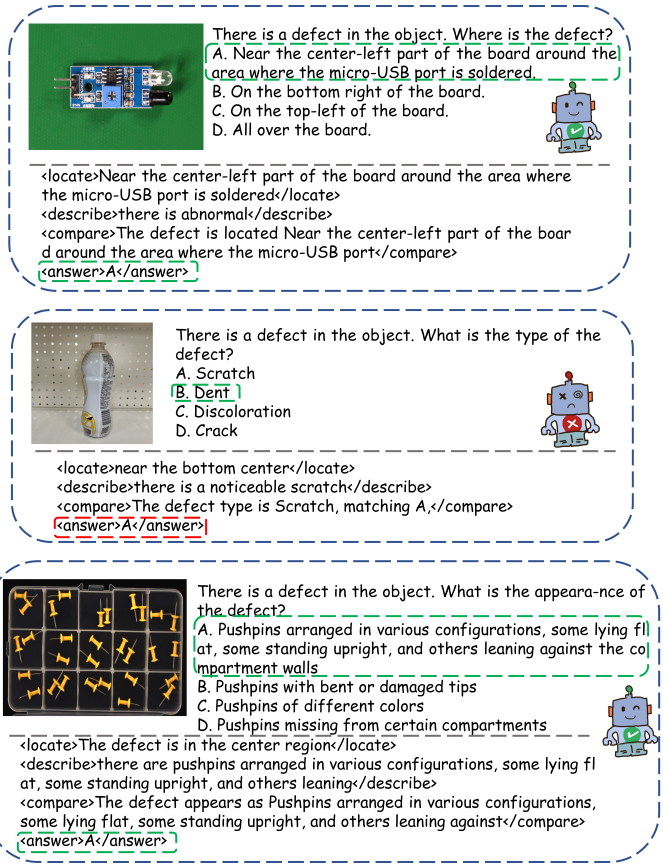


Figure 6: Qualitative case studies. Green and red indicators denote correct and incorrect final predictions, respectively. The panels report the input, question and options, structured model response, and ground-truth answer.

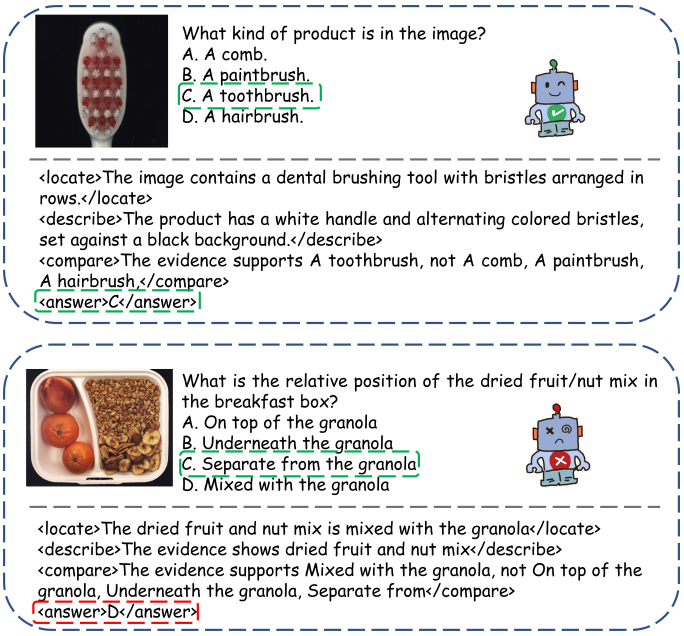


Figure 6: Qualitative case studies (continued). Green and red indicators denote correct and incorrect final predictions, respectively.