EVOROBUST: SELF-IMPROVING MLLMS ROBUSTNESS VIA CLAIM-LEVEL SELF-VERIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models lose accuracy and make more unsupported statements when images are blurred, noisy, or heavily compressed. The robustness gap is hard to close by self-improvement, because the model being improved is also the only verifier and it reads exactly the same corrupted pixels. We introduce EVOROBUST, which raises MLLM robustness via claim-level self-verification: it decomposes a free-form description into atomic claims, turns each claim into a targeted probe, re-answers every probe on the same corrupted image several times, and aggregates the resulting agreement into a claim-structured reward for Group Relative Policy Optimization. We are explicit about what this signal is: it estimates the agreement rate of a frozen verifier rather than a truth probability, repeated probing reduces decoding variance rather than recovering lost visual evidence, and policy optimization only needs the reward to rank candidates correctly. Trained on unlabeled degraded images, EVOROBUST raises the R-Bench Overall score of Qwen3-VL-4B from 0.6671 to 0.6782, improves on averaged holistic self-scoring (0.6715) under an identical optimizer, and is first or tied-first in seven of nine corrupted settings across MMMB, MMStar, and RealWorldQA without reducing clean accuracy. We will release the code and configurations shortly.

1 INTRODUCTION

Multimodal large language models (MLLMs) read pristine images well (Bai et al., 2025b; OpenAI et al., 2024; Chen et al., 2024c; Gemma Team et al., 2025; Lu et al., 2024; Deng et al., 2025), but their accuracy falls and unsupported statements become more frequent once the input suffers motion blur, low-light noise, or compression artifacts (Li et al., 2025; Qiu et al., 2025; Liu et al., 2026). Because corruption is the normal case in deployment rather than the exception, this robustness gap matters wherever a downstream decision depends on what the model claims to see (Yuan et al., 2024a; Hu et al., 2026; Tang et al., 2026).

Methods that close this gap differ less in architecture than in what they assume at adaptation time. Encoder-level alignment needs clean–corrupted image pairs or a known corruption pipeline to define its training signal (Figure 1A) (Mao et al., 2023; Schlarmann et al., 2024; Malik et al., 2025; Hossain & Imteaj, 2024; Wang et al., 2025b). Degradation-aware reasoning such as Robust-R1 (Tang et al., 2026) drops the clean counterpart but needs curated per-image annotations, including degradation parameters and reasoning chains (Figure 1B). Test-time adaptation such as Mint (Bao et al., 2025) avoids training labels but relies on a statistic of the test batch rather than on evidence about any individual image. Each assumption is reasonable in isolation, and each is hard to satisfy for the open-ended set of corruptions a deployed model actually meets.

Asking the model to self-improve its robustness, however, does not merely remove supervision; it changes the problem. If no external answer exists, the model under training is also the only verifier, and that verifier observes the same corrupted image. The most direct response is to let the model score its own output as a whole, but a single number attached to a long paragraph cannot indicate which of its statements the image supports, and one stochastic verification pass is unreliable exactly where the pixels are ambiguous. The failure mode is therefore not a missing label but an uninformative signal.

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.

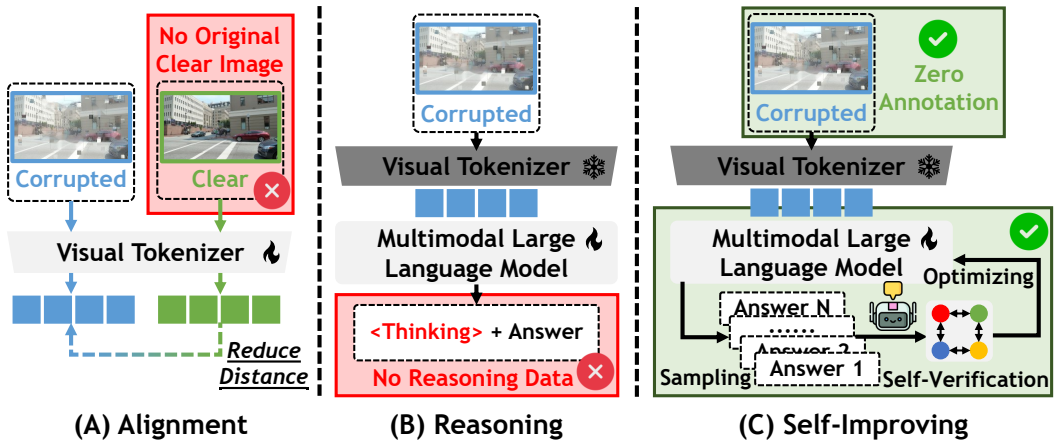


Figure 1: Three paradigms for MLLM robustness under visual corruption, distinguished by the evidence their learning signal consumes. **(A) Alignment:** clean and corrupted views of one scene are encoded together so their representations can be pulled into agreement, which presumes a clean counterpart or a known corruption process. **(B) Reasoning:** the corrupted image alone drives an explicit thinking chain and an answer, removing the clean counterpart but requiring annotated reasoning data. **(C) EvoRobust:** a self-improvement loop: the policy samples several descriptions of the corrupted image, claim-level self-verification scores them, and the resulting rewards drive optimization, with no clean counterpart, reasoning trace, or task label. The signal is nonetheless produced by a verifier reading the same degraded pixels as the policy, the constraint analyzed in Section 3.3.

Our starting point is claim-level self-verification: a long description is not one hypothesis but many, and they usually fail independently: a paragraph can identify the scene correctly while inventing the text printed on a distant sign. Splitting it into atomic claims and interrogating each one separately makes the individual judgments far simpler than judging the paragraph, and repeating each interrogation exposes where the model's own reading is unstable. We stress what this does *not* do. Because every probe is answered from the same corrupted input, no repetition recovers evidence the corruption destroyed; repetition removes decoding noise, and decomposition buys a signal that separates a correct claim from an incorrect one inside the same response (Figure 1C).

EvoRobust implements this self-improvement loop in four frozen, prompted roles built from the policy's own checkpoint: a *Fact Parser* that splits a description into atomic claims, a *Query Generator* that converts each claim into a type-matched probe with a claim-implied candidate answer, a *Voting Critic* that answers each probe M times from the corrupted image, and a *Credit Assigner* that marks each answer compatible or incompatible with the claim. The per-claim compatibility rates are averaged into a response-level reward that drives Group Relative Policy Optimization (GRPO) (Shao et al., 2024). We also make precise what this reward is: an empirical agreement rate of a frozen verifier. It coincides with truth only under assumptions that are not testable without labels, and GRPO does not need it to; standardized group advantages require only that the reward rank candidates in the right order.

Trained on unlabeled degraded images with no answers, clean references, or reasoning annotations, EvoRobust raises the R-Bench (Li et al., 2025) Overall score of Qwen3-VL-4B (Bai et al., 2025a) from 0.6671 to 0.6782, improving on its backbone in eight of nine task–severity cells. Under an identical optimizer and training set, an averaged holistic self-score in place of the claim-level reward reaches only 0.6715, which isolates the contribution of structure rather than of reinforcement learning. The same policy is first or tied-first in seven of nine corrupted settings on synthetic corruptions of MMMB (Sun et al., 2025), MMStar (Chen et al., 2024a), and RealWorldQA (xAI, 2024) while leaving clean accuracy intact. Our contributions are:

- We study self-improving MLLM robustness when the only verifier shares the learner's corrupted image: no labels, no clean counterpart, and no extra evidence.

- We implement the EVOROBUST self-improvement loop. The policy describes the corrupted image; the description is split into atomic claims; each claim is checked against the same image by repeated probes; and agreement among those checks becomes the reward that updates the policy.
- On Qwen3-VL-4B, EVOROBUST raises the R-Bench Overall score from 0.6671 to 0.6782 and beats averaged holistic self-scoring (0.6715) under the same optimizer; it is first or tied-first in seven of nine corrupted settings on MMMB, MMStar, and RealWorldQA, with no drop in clean accuracy.

2 RELATED WORK

Robustness of MLLMs under corruption. Accuracy drops and unsupported statements under perturbed or corrupted inputs are reported consistently across perturbation, hallucination, and corruption benchmarks (Ding et al., 2024; Qiu et al., 2025; Li et al., 2025; 2023; Guan et al., 2024; Liu et al., 2026). Mitigations are best distinguished by the evidence they consume. Adversarial and contrastive fine-tuning of the visual encoder aligns representations of clean and corrupted views, which presumes paired views or a corruption model even when the objective is label-free (Mao et al., 2023; Schlarmann et al., 2024; Malik et al., 2025; Hossain & Imteaj, 2024; Wang et al., 2025b). Robust-R1 (Tang et al., 2026) moves from resisting corruption to reasoning about it, at the cost of curated degradation and reasoning annotations. Test-time adaptation needs no training labels but optimizes a statistic of the test distribution; Mint (Bao et al., 2025) attributes corruption-induced errors to an embedding variance collapse and counteracts it at inference. EVOROBUST occupies a different point: a self-improvement loop that trains on unlabeled degraded images, uses no clean counterpart or corruption model, and derives its signal from claim-level self-verification of each image rather than from batch statistics.

Learning from model-generated feedback. Self-rewarding language models use one model as both policy and judge (Yuan et al., 2024b; Zhang et al., 2025), debate and self-consistency aggregate parallel samples to suppress single-pass errors (Du et al., 2024; Wang et al., 2023b), and AI-feedback pipelines such as RLAIF-V (Yu et al., 2025) turn model preferences into an iterative DPO (Rafailov et al., 2023) objective for vision-language models on clean images. Recent systems push toward little or no curated data: MM-Zero (Li et al., 2026) evolves vision-language models through a Proposer–Coder–Solver loop, Multi-Agent Evolve (Chen et al., 2025) a Proposer–Solver–Judge loop for general reasoning, R-Zero (Huang et al., 2026) a Challenger–Solver pair, and Agent0 (Xia et al., 2025) tool use. These systems mostly verify with deterministic checks such as exact match, which suit tasks with a canonical answer string. Free-form description of a corrupted image has no such form: paraphrase is pervasive, correctness is partial, and repeated reads of an ambiguous region disagree.

Atomic claim decomposition for factuality. Splitting a long generation into atomic claims and checking each separately is the standard recipe in text factuality: FActScore (Min et al., 2023) scores biographies claim by claim, SAFE (Wei et al., 2024) reasons over each claim of a long-form answer, and FacTool (Chern et al., 2023) routes claims to task-specific tools, while AMBER (Wang et al., 2023a) applies a related decomposition to MLLM hallucination. Closest to us, RLAIF-V (Yu et al., 2025) scores VLM responses by divide-and-conquer over atomic claims, but on clean images and to build preference data rather than as an optimized reward. Our Fact Parser makes the same move, and three differences fix what it can be used for. Those pipelines decompose to *evaluate* a fixed model; we decompose to build a reward that is then optimized against, so any bias of the decomposition is inherited by the policy. They resolve each claim against an external source (retrieval, search, or a tool), which is exactly the independent evidence we lack, since our verifier re-reads the same corrupted image; Proposition 1 therefore binds us and not them. And they work on text, where a claim is supported or it is not, whereas under corruption many claims are unresolvable from the pixels, which is why we score a graded agreement rate rather than a binary verdict.

Consensus under shared model bias. Self-consistency improves accuracy by marginalizing over sampled reasoning paths (Wang et al., 2023b; Wei et al., 2022), which is the intuition behind our repeated probing, but it is easy to overstate: samples drawn from one model on one input are not

independent measurements of the world, so a confident consensus can be confidently wrong, and the risk grows when the verifier inherits the policy's priors. We therefore treat aggregated verdicts as an estimate of the verifier's agreement propensity rather than as evidence about the scene, state the resulting variance floor in Section 3.3, and test the reward against holistic self-scoring in Section 4.3.

3 METHOD

3.1 PROBLEM SETTING

Let S denote the latent scene, never observed, and let x be a corrupted image of it. During the self-improvement loop we are given only a set of such images: no answers, no clean counterpart, no corruption label, and no reasoning trace. A policy π_θ produces a free-form description

$$y \sim \pi_\theta(\cdot \mid x, q_{\text{desc}}), \tag{1}$$

where q_{desc} is one fixed instruction asking for a detailed description. The learning problem is to self-improve the robustness of π_θ from a claim-level self-verification reward computable from x and y alone.

All verification is carried out by a frozen model $\mathcal{V}_\phi$, obtained by copying the initial policy checkpoint and specializing it through prompts into four roles; only π_θ is updated, so ϕ is constant throughout training. Writing U for the verifier's decoding randomness, our objective is

$$\max_\theta \; \mathbb{E}_{x\sim\mathcal{D}_{\text{deg}}} \; \mathbb{E}_{y\sim\pi_\theta(\cdot|x,q_{\text{desc}})} \; \mathbb{E}_U \big[\, R_\phi(y, x; U) \,\big], \tag{2}$$

where $\mathcal{D}_{\text{deg}}$ is a set of unlabeled degraded images and R_ϕ is the claim-level reward defined next. Making U explicit matters, because much of Section 3.3 concerns how R_ϕ behaves under resampling of it.

3.2 CLAIM-LEVEL SELF-VERIFICATION

The reward is produced by the four-stage pipeline in Figure 2. Each stage is a prompted call to $\mathcal{V}_\phi$; the exact templates are in Appendix E.

Claim decomposition. The *Fact Parser* $\mathcal{V}^{\text{fp}}$ reads the description y and emits a set of atomic claims

$$\{h_k\}_{k=1}^{K(y)} = \mathcal{V}^{\text{fp}}(y), \qquad h_k = (f_k, t_k), \tag{3}$$

where f_k is a single assertion and t_k is one of six semantic categories: object existence, object attributes, spatial relations, scene semantics, quantity, and state & action. Two properties matter. It operates on text only, so it can enforce that each claim is atomic and non-duplicated but cannot check whether the claim is true; and it decomposes only what the description asserts, so omitted content is never penalized. Appendix B gives the taxonomy and its mapping to probe types.

Probe generation. The *Query Generator* $\mathcal{V}^{\text{qg}}$ converts each claim into a probe question together with the answer that the claim implies,

$$(q_k, a_k) = \mathcal{V}^{\text{qg}}(h_k), \tag{4}$$

with the question form matched to t_k: yes/no for object presence, a single word for attributes, a relation word for spatial claims, a short phrase for scenes, an integer for quantities, and an activity phrase for states. We call a_k a *claim-implied candidate answer* rather than a reference answer: it is derived from the policy's own description and carries no independent authority. Its short canonical form is what makes the later comparison tractable, and the prompt requires at least one probe per claim index so that probes do not concentrate on a few salient objects.

Repeated probing. The *Voting Critic* $\mathcal{V}^{\text{vc}}$ answers each probe M times from the corrupted image,

$$\hat{a}_k^{(m)} \sim p_\phi(\cdot \mid x, q_k), \qquad m = 1, \ldots, M, \tag{5}$$

under a prompt that forces short answers, matches the expected answer format, and prefers an explicit *"unclear"* over a confident guess when the pixels do not support the requested detail. The M draws differ only in decoding randomness; they are repeated readings of one image by one model, not independent observations of the scene.

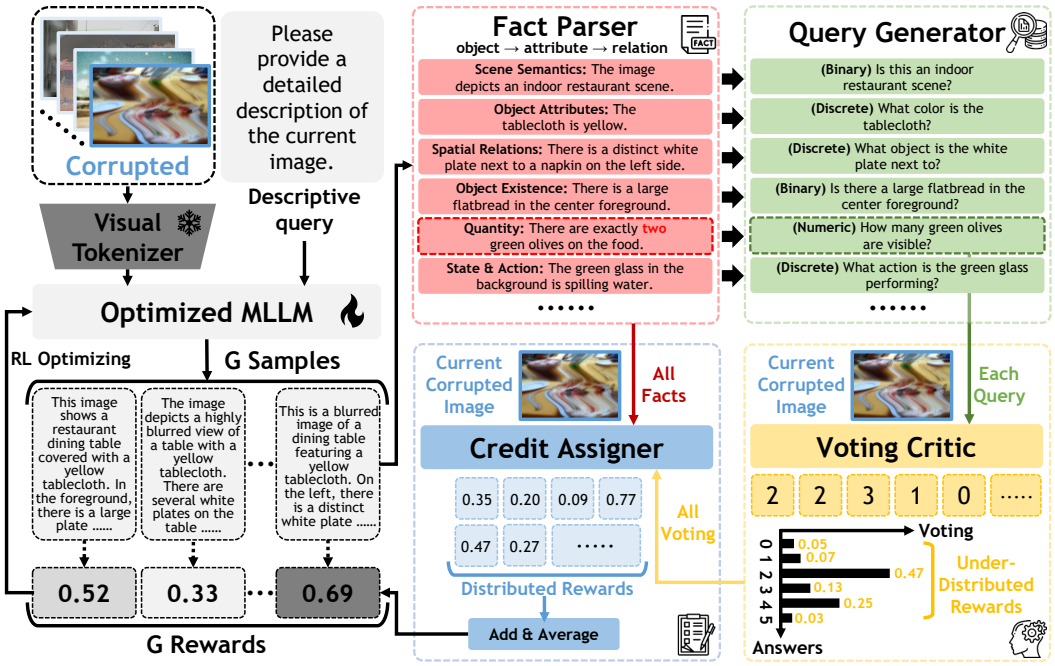


Figure 2: The EVOROBUST self-improvement loop. Given the corrupted image and one generic descriptive query, the policy samples G candidate descriptions. For each candidate, the **Fact Parser** extracts atomic claims across six semantic categories, the **Query Generator** turns every claim into a type-matched probe, the **Voting Critic** answers each probe M times on the same corrupted image, and the **Credit Assigner** marks each answer compatible or incompatible with the claim. Per-claim agreement rates are averaged into one response-level reward, and group-relative advantages update the policy. The four verification roles are frozen copies of the initial policy checkpoint; only the policy is trained.

Compatibility judging and aggregation. The *Credit Assigner* $\mathcal{V}^{\text{ca}}$ receives the claim, the probe, the candidate answer, all M critic answers, and the same corrupted image, and returns one binary verdict per critic answer,

$$V_k^{(1:M)} = \mathcal{V}^{\text{ca}}\big(x,\, h_k,\, q_k,\, a_k,\, \hat{a}_k^{(1:M)}\big) \in \{0,1\}^M. \tag{6}$$

Judging semantically rather than by string match is what makes the signal tolerant to paraphrase, and judging the M answers in one call lets the judge use the spread of the critic answers as context. The per-claim agreement rate and the response-level reward are then

$$\hat{\mu}_k = \frac{1}{M}\sum_{m=1}^{M} V_k^{(m)}, \qquad R_\phi(y,x) = \frac{1}{K(y)}\sum_{k=1}^{K(y)} \hat{\mu}_k \;\in\; [0,1], \tag{7}$$

with $R_\phi(y,x) := 0$ when decomposition returns no claim.

3.3 WHAT THE AGREEMENT REWARD DOES AND DOES NOT PROVIDE

Equation 7 is easy to misread as a probability that the description is correct. It is not, and being precise about this both prevents overclaiming and explains the design. Fix a claim, write $T \in \{0,1\}$ for its unobservable truth under S, and V for a single verdict.

Proposition 1 (Repeated probing adds no visual evidence)**.** *Every verifier output is a function of the corrupted image, the claim, and decoding randomness, $V = g_\phi(x,h,U)$ with $U \perp S \mid (x,h)$. Hence $T \to (x,h) \to V$ is a Markov chain and $I(T; V^{(1:M)} \mid x,h) = 0$ for every M.*

Algorithm 1 One EVOROBUST self-improvement step on an unlabeled corrupted image.

Require: corrupted image x; policy π_θ; frozen roles $\mathcal{V}^{\mathrm{fp}}, \mathcal{V}^{\mathrm{qg}}, \mathcal{V}^{\mathrm{vc}}, \mathcal{V}^{\mathrm{ca}}$; group size G; probes per claim M

$y^{(1:G)} \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid x, q_{\mathrm{desc}})$ ▷ group rollout on one fixed instruction; no label

for $g = 1, \ldots, G$ **do**

  $\{h_k\}_{k=1}^{K} \leftarrow \mathcal{V}^{\mathrm{fp}}(y^{(g)}), \;\; h_k = (f_k, t_k), \;\; K = K(y^{(g)})$ ▷ Fact Parser

  for $k = 1, \ldots, K$ **do**

    $(q_k, a_k) \leftarrow \mathcal{V}^{\mathrm{qg}}(h_k)$ ▷ Query Generator: probe, claim-implied answer

    $\hat{a}_k^{(m)} \sim p_\phi(\cdot \mid x, q_k), m = 1, \ldots, M$ ▷ Voting Critic: M reads of the same x

    $V_k^{(1:M)} \leftarrow \mathcal{V}^{\mathrm{ca}}(x, h_k, q_k, a_k, \hat{a}_k^{(1:M)}), \;\; \hat{\mu}_k \leftarrow \frac{1}{M}\sum_m V_k^{(m)}$ ▷ Credit Assigner

  $R^{(g)} \leftarrow \frac{1}{K}\sum_k \hat{\mu}_k$, or 0 if $K = 0$ ▷ Equation 7

$A^{(g)} \leftarrow (R^{(g)} - \mu_{\mathrm{grp}})/(\sigma_{\mathrm{grp}} + \delta)$ ▷ standardize the G rewards within the group

update θ on $A^{(1:G)}$ with the clipped surrogate and a KL penalty to the frozen reference

The statement is elementary but it fixes the ceiling of the approach: what the corruption removed from x cannot be restored by asking the same frozen model more often. Repetition acts only on U, so its value must come from elsewhere.

Proposition 2 (Aggregation estimates the verifier, at a rate set by correlation)**.** *Let $\mu = \mathbb{E}[V \mid x, h]$ be the verifier's agreement propensity, and let the M verdicts be exchangeable with variance $\sigma^2 = \mu(1-\mu)$ and pairwise correlation $\rho \in [0, 1]$. Then $\mathbb{E}[\hat{\mu} \mid x, h] = \mu$ and*

$$\mathrm{Var}(\hat{\mu} \mid x, h) = \frac{\sigma^2}{M}\big[1 + (M-1)\rho\big] \;\xrightarrow[M\to\infty]{}\; \rho\,\sigma^2, \tag{8}$$

so the effective sample size is $M_{\mathrm{eff}} = M/\big(1 + (M-1)\rho\big)$.

This is the precise sense in which repeated probing helps: as a Monte Carlo estimator of μ it suppresses the sampling noise that makes a single verification pass brittle on ambiguous regions. It also explains why the benefit saturates, since once $M \gg 1/\rho$ additional samples buy almost nothing and the variance never falls below $\rho\sigma^2$. Sharing one backbone between policy and verifier is what makes ρ large, and no choice of M removes that term (Appendix C).

Proposition 3 (Agreement separates truth only if the verifier beats chance)**.** *For claims of type t, let $\alpha_t = \Pr(V = 1 \mid T = 1, t)$ and $\beta_t = \Pr(V = 1 \mid T = 0, t)$. Then*

$$\mathbb{E}[\hat{\mu} \mid T, t] = \beta_t + (\alpha_t - \beta_t)\,T. \tag{9}$$

The reward is informative about T if and only if $\alpha_t > \beta_t$, and equals T only in the degenerate case $\alpha_t = 1$, $\beta_t = 0$. Neither α_t nor β_t is identifiable from agreement alone without labels.

Proposition 3 is why we report agreement rates rather than confidences, and it clarifies the design choices of Section 3.2: atomic decomposition, type-matched probes, and the instruction to answer *"unclear"* instead of guessing all try to move $\alpha_t - \beta_t$ away from zero by making each judgment easier, not to make the verifier correct.

What the optimizer actually needs. GRPO standardizes rewards within a group drawn for the same image, so the advantage is invariant to any positive affine rescaling of the reward inside that group (Appendix C). Calibration is therefore irrelevant; what is required is that the reward rank candidates roughly correctly,

$$\mathrm{Corr}\big(R_\phi(y, x),\, U^\star(y, S) \,\big|\, x\big) > 0 \quad \text{for } y \sim \pi_\theta(\cdot \mid x, q_{\mathrm{desc}}), \tag{10}$$

where $U^\star$ is the unobservable true utility of a description. Condition 10 cannot be checked without labels, which is why our benchmark results are the relevant evidence: an improvement measured against external ground truth indirectly tests that the surrogate ranks candidates usefully.

A known bias of averaging over claims. Because Equation 7 averages over claims, one wrong claim shifts the reward by only $1/K$, which is what localizes errors, but appending a claim changes it by $(\hat{\mu}_{\mathrm{new}} - R_\phi)/(K+1)$ and is therefore rewarded whenever the new claim is easier than average (Appendix C). With a decomposition prompt that favors high recall, this biases the signal toward descriptions that add safe content, and it never penalizes what the policy omitted.

3.4 POLICY OPTIMIZATION

We optimize Equation 2 with GRPO (Shao et al., 2024), a critic-free variant of PPO (Schulman et al., 2017) that replaces a learned value function with the spread of rewards inside a sampled group. Algorithm 1 states the resulting loop: for each image we draw G candidates from $\pi_{\theta_{\text{old}}}$, score each with Equation 7, standardize the scores within the group into advantages $A^{(g)} = (R^{(g)} - \mu_{\text{grp}})/(\sigma_{\text{grp}} + \delta)$, and take a clipped step with a KL penalty against the frozen initial checkpoint. We use $G = 16$, $M = 10$, clipping $\epsilon = 0.2$, KL weight $\beta = 0.02$, learning rate 1×10^{-6}, and three epochs. The verification roles are frozen, so no parameters are learned beyond π_θ; Appendix F shows this makes the verifier a compute cost rather than a parameter cost.

4 EXPERIMENTS

4.1 SETUP

Training. The policy and all four verification roles are initialized from Qwen3-VL-4B (Bai et al., 2025a); only the policy is updated. Adaptation uses the degraded images of the Robust-R1 corpus (Tang et al., 2026) with every human-curated annotation discarded, so no clean counterpart, degradation label, perceptual analysis, or reasoning chain remains. The setting is label-free but not curation-free: the images themselves come from an existing curated corpus. Training consumes 600 image instances on 8×NVIDIA L20 GPUs; Appendices A and F give the full protocol and cost.

Evaluation. Following the two-track protocol of Tang et al. (2026), we measure real-world corruption on R-Bench (Li et al., 2025), which scores multiple-choice (MCQ), visual question answering (VQA), and captioning (CAP) at low, mid, and high severity, and controlled synthetic corruption by degrading MMMB (Sun et al., 2025), MMStar (Chen et al., 2024a), and RealWorldQA (xAI, 2024) at three intensities. Overall is the unweighted mean of the nine R-Bench task–severity cells; Appendix A states how each task is scored.

Comparisons. We compare against base MLLMs (Qwen3-VL-4B (Bai et al., 2025a), Gemma3-4B (Gemma Team et al., 2025), InternVL-4B (Chen et al., 2024b)), degradation-aware and adversarial training (Robust CLIP (Schlarmann et al., 2024), Robust LLaVA (Malik et al., 2025), TeCoA (Mao et al., 2023), Robust-R1 (Tang et al., 2026)), test-time adaptation (Mint (Bao et al., 2025)), and label-free self-improvement (MM-Zero (Li et al., 2026), MAE (Chen et al., 2025)). Adaptation methods share the Qwen3-VL-4B backbone under one protocol; Gemma3-4B and InternVL-4B are native checkpoints. Baseline configurations reported by Tang et al. (2026) are listed in Appendix D.

4.2 RQ1: DOES THE AGREEMENT REWARD IMPROVE ITS OWN BACKBONE?

Table 1 reports the real-world corruption results. EVOROBUST reaches an Overall score of 0.6782, 0.0111 above its own backbone (0.6671) and ahead of the strongest label-free self-improvement baseline MM-Zero (0.6709) and the test-time adaptation baseline Mint (0.6682). It improves on the backbone in eight of the nine task–severity cells; the exception is VQA-mid (0.6461 against 0.6509). Against every method in the table it is first or tied-first in seven cells, with Robust-R1 leading VQA-low (0.6380 versus 0.6368) and the backbone and MM-Zero jointly leading VQA-mid.

The gains concentrate at high severity, where MCQ improves from 0.6768 to 0.6890, VQA from 0.6006 to 0.6269, and CAP from 0.6235 to 0.6336, the regime in which claims are most often ambiguous and the per-claim signal has most to distinguish. Captioning improves at every severity, and as the only free-form task CAP is also closest to the descriptive format used during adaptation, so we read this as the reward transferring within a format rather than as a general capability gain. The magnitudes are modest, roughly one point of Overall from a single run, and we do not claim more.

4.3 RQ2: DOES THE STRUCTURE OF THE REWARD MATTER?

Table 2 isolates the reward from everything else. The first two rows show the gain is not a decoding effect: drawing M test-time samples from the unadapted model and averaging their scores reaches 0.6682, essentially the backbone. The next two rows keep GRPO, the training images, and the

Table 1: **R-Bench under real-world corruption.** Adaptation uses Qwen3-VL-4B (Bai et al., 2025a); Gemma3-4B (Gemma Team et al., 2025) and InternVL-4B (Chen et al., 2024b) are native checkpoints. Overall averages all nine cells before rounding. **Bold**/underline: best/second-best, including ties. Table 4 adds results from Tang et al. (2026).

Method	MCQ			VQA			CAP			Overall
	low	mid	high	low	mid	high	low	mid	high	
Gemma3-4B	0.5823	0.5776	0.5060	0.4865	0.4630	0.4419	0.4048	0.3746	0.3480	0.4649
InternVL-4B	0.6235	0.6024	0.5914	0.4982	0.4539	0.5108	0.3667	0.3041	0.2851	0.4706
Qwen3-VL-4B	0.7353	0.7081	0.6768	0.6202	**0.6509**	0.6006	0.6938	0.6948	0.6235	0.6671
Robust CLIP	0.6059	0.4783	0.5061	0.5037	0.4715	0.5198	0.3877	0.4071	0.3624	0.4714
Robust LLaVA	0.5706	0.5217	0.5244	0.4644	0.4697	0.4952	0.4179	0.3775	0.3855	0.4697
TeCoA	0.5882	0.5342	0.4329	0.4037	0.4206	0.4371	0.2815	0.3260	0.2838	0.4120
Robust-R1	0.7118	0.6646	0.6220	**0.6380**	0.5988	0.5994	0.4370	0.4790	0.4814	0.5813
Mint	0.7353	**0.7267**	0.6707	0.6307	0.6436	0.6234	0.6827	0.6824	0.6182	0.6682
MM-Zero	0.7451	0.7143	0.6707	0.6307	**0.6509**	0.6138	0.6895	0.6969	0.6266	0.6709
MAE	0.7118	0.7205	0.6707	0.6313	0.6345	0.6084	0.6685	0.6501	0.6004	0.6551
EvoRobust	**0.7471**	**0.7267**	**0.6890**	0.6368	0.6461	**0.6269**	**0.6988**	**0.6988**	**0.6336**	**0.6782**

Table 2: **Reward-design study on R-Bench.** Adapted rows share Qwen3-VL-4B (Bai et al., 2025a), optimizer, data, and evaluation; only the reward varies. Task scores average over severities. Δ: Overall difference from full EvoRobust using displayed values. One run per row.

Reward used for adaptation	MCQ	VQA	CAP	Overall	Δ
No adaptation (Qwen3-VL-4B)	0.7067	0.6239	0.6707	0.6671	-0.0111
Test-time sampling, no adaptation	0.7067	0.6245	0.6734	0.6682	-0.0100
Holistic self-score	0.7127	0.6302	0.6695	0.6708	-0.0074
Averaged holistic self-score	0.7148	0.6330	0.6666	0.6715	-0.0067
Claim level, no decomposition	0.7130	**0.6366**	0.6669	0.6722	-0.0060
Claim level, generic probe template	0.7129	0.6364	0.6770	0.6754	-0.0028
Claim level, single probe (M=1)	0.7150	0.6343	0.6667	0.6720	-0.0062
Claim level, rule-based matching	0.7169	0.6363	0.6713	0.6748	-0.0034
EvoRobust (full)	**0.7209**	**0.6366**	**0.6771**	**0.6782**	0.0000

schedule fixed and replace only the reward with the model's holistic score for its own description, reaching 0.6708, or 0.6715 when several holistic scores are averaged to reduce variance, both below the full pipeline at 0.6782. Since the optimizer is identical, the remaining 0.0067 is attributable to how the reward is constructed rather than to reinforcement learning itself. This is the most informative comparison in the paper and also a modest one: 0.0067 from one run per configuration shows a consistent ordering, not a large effect.

The component substitutions below indicate where that structure comes from. Removing atomic decomposition costs 0.0060 and replacing repeated probing with a single pass costs 0.0062, the two largest drops, matching the two mechanisms of Section 3.3: localizing the judgment and reducing estimator variance (Proposition 2). Rule-based matching in place of semantic judging costs 0.0034 and a generic probe template costs 0.0028, consistent with their role in Proposition 3 of making individual judgments easier rather than changing what is measured. Every substitution lowers Overall, but the spread across the four is only 0.0034, so we read them as small and consistent contributions and do not claim any component is necessary. The probe sweep has the shape Proposition 2 predicts: 0.6720 at M=1, 0.6750 at M=5, 0.6782 at M=10, then 0.6774 at M=20 (Table 6); saturation is expected once verdicts are correlated, and we use M=10 throughout.

4.4 RQ3: Does the Adapted Policy Transfer to Controlled Corruptions?

Adaptation used real-world degradations, so the synthetic benchmarks test transfer, and Figure 3 shows it largely holds. EvoRobust has the highest clean accuracy on all three datasets (84.75,

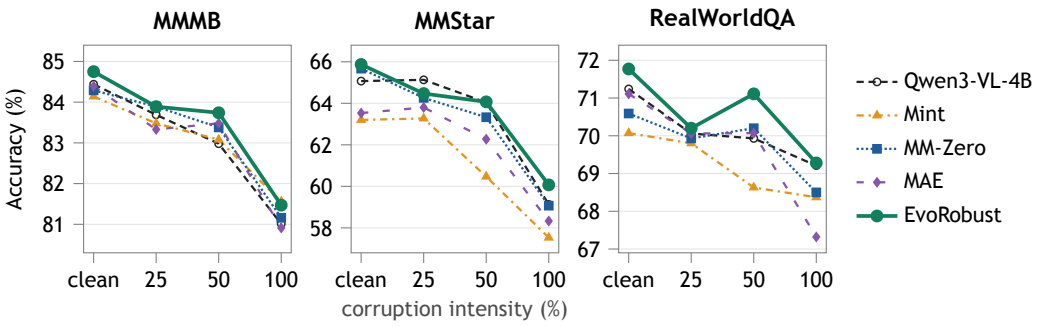


Figure 3: **Synthetic-corruption accuracy** on Qwen3-VL-4B (Bai et al., 2025a); vertical scales differ. EVOROBUST leads on all three clean settings and leads or ties in seven of nine corrupted settings. Table 5 lists all methods.

65.87, 71.77), so the self-improvement loop does not trade clean for corruption performance, and it is first or tied-first in seven of nine corrupted cells. The exceptions are Mint at full intensity on MMMB (81.56 versus 81.47) and the unadapted backbone at 25% on MMStar (65.13 versus 64.47); full-intensity margins over the backbone are 0.46, 0.94, and 0.07 points. From clean to full intensity it loses 3.28, 5.80, and 2.49 points against 3.43, 5.94, and 2.03 for the backbone, so it raises accuracy under corruption while the evidence that it degrades more gracefully is mixed.

4.5 RQ4: SCOPE AND COST

Section 3.3 does not depend on a particular backbone: on InternVL3.5-4B (Wang et al., 2025a), under the identical label-free protocol, the pipeline raises R-Bench Overall from 0.4964 to 0.5121 (Table 7), the gain sitting in MCQ and VQA while CAP moves only from 0.3390 to 0.3428. One extra family shows the recipe is not Qwen3-VL-specific, not that it is general.

Verification, not the policy update, dominates cost: 3.27M logical calls, 2.94M (89.9%) of them Voting Critic calls, and 237.85 GPU-hours (Appendix F). This is Proposition 2 in practice: M sets both the estimator's variance and the dominant cost, which is why saturation at $M{=}20$ matters. Deployment is unaffected: the policy is an ordinary checkpoint and no verification role runs at test time.

Limitations. The reward is a frozen verifier's agreement rate, not a truth probability. By Proposition 3 it tracks truth only where that verifier beats chance, and because the policy and the four roles share one backbone and the same corrupted image, a shared mistake can produce a stable $\hat{\mu}{=}1.00$ indistinguishable from a correct consensus (Appendix G.4).

Averaging over extracted claims rewards appending a claim easier than the current average and never penalizes omissions. We did not measure the resulting composition drift; Appendix H expands both caveats.

5 CONCLUSION

We presented EVOROBUST, a framework for improving MLLM robustness through claim-level self-verification on unlabeled degraded images. By decomposing descriptions into atomic claims, generating targeted probes, and aggregating repeated judgments, the framework connects policy optimization to feedback about individual visual assertions. This decomposition localizes disagreement within a description, while repeated probing reveals which assertions the model reads consistently and which produce variable answers. Its frozen verification roles share the policy's initial checkpoint, allowing adaptation without task labels, clean references, or an external teacher. Verification is confined to adaptation, so the resulting policy can be used directly at inference without running the verification pipeline. The reward measures agreement with the verifier's reading of the image, with reliability limited by shared biases and incomplete visual evidence. Future work can explore more diverse verification signals, adaptive probing, and broader visual conditions.

AI USE STATEMENT

This section discloses how language models were used in preparing the submission, as required. General-purpose large language models were used solely as writing assistance: rewriting sentences for readability, tightening phrasing, polishing transitions, and correcting typos. No language model was used to generate scientific claims, design experiments, produce technical content, or decide on methodology. All results, figures, analyses, and conclusions were produced and verified by the authors, who take full responsibility for the content.

REPRODUCIBILITY STATEMENT

Appendix E gives the verification-role prompts and Table 3 the claim taxonomy; the adaptation loop is Algorithm 1 and the optimizer is specified in Section 3.4. Appendix A states the experimental protocol, Appendix C the derivations, Appendix F the training cost, and Appendix G one execution trace; every value in Figure 3 is in Table 5 and is plotted from `figs/fig_synthetic.tex`. Decoding temperatures, batch sizes, and sequence-length caps are in the configuration files to be released with the code upon acceptance.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a. URL https://arxiv.org/abs/2511.21631.

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

Wenxuan Bao, Ruxi Deng, and Jingrui He. Mint: A simple test-time adaptation of vision-language models against common corruptions. In *Advances in Neural Information Processing Systems*, volume 38, pp. 58666–58700. Curran Associates, Inc., 2025. doi: 10.52202/085713-1964. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/54df19dc823e2cacff6d26640bac6c10-Paper-Conference.pdf.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*, volume 37, pp. 27056–27087. Curran Associates, Inc., 2024a. doi: 10.52202/079017-0850. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2f8ee6a3d766b426d2618e555b5aeb39-Paper-Conference.pdf.

Yixing Chen, Yiding Wang, Siqi Zhu, Haofei Yu, Tao Feng, Muhan Zhang, Mostofa Patwary, and Jiaxuan You. Multi-Agent Evolve: LLM self-improve through co-evolution. *arXiv preprint arXiv:2510.23595*, 2025. URL https://arxiv.org/abs/2510.23595.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024b. URL https://arxiv.org/abs/2412.05271.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24185–24198, 2024c. URL https://openaccess.thecvf.com/content/CVPR2024/html/Chen_InternVL_Scaling_up_Vision_Foundation_Models_and_Aligning_for_Generic_CVPR_2024_paper.html.

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. FacTool: Factuality detection in generative AI – a tool augmented framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*, 2023. URL https://arxiv.org/abs/2307.13528.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. URL https://arxiv.org/abs/2505.14683.

Peng Ding, Jingyu Wu, Jun Kuang, Dan Ma, Xuezhi Cao, Xunliang Cai, Shi Chen, Jiajun Chen, and Shujian Huang. Hallu-PI: Evaluating hallucination in multi-modal large language models within perturbed inputs. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 10707–10715, 2024. doi: 10.1145/3664647.3681251. URL https://doi.org/10.1145/3664647.3681251.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. HallusionBench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14375–14385, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Guan_HallusionBench_An_Advanced_Diagnostic_Suite_for_Entangled_Language_Hallucination_and_CVPR_2024_paper.html.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1.5-VL technical report. *arXiv preprint arXiv:2505.07062*, 2025. URL https://arxiv.org/abs/2505.07062.

Md Zarif Hossain and Ahmed Imteaj. Sim-CLIP: Unsupervised siamese adversarial fine-tuning for robust and semantically-rich vision-language models. *arXiv preprint arXiv:2407.14971*, 2024. URL https://arxiv.org/abs/2407.14971.

Chengyin Hu, Xiang Chen, Zhe Jia, Weiwen Shi, Fengyu Zhang, Jiujiang Guo, and Yiwei Wei. A semantic decoupling-based two-stage rainy-day attack for revealing weather robustness deficiencies in vision-language models. *arXiv preprint arXiv:2601.13238*, 2026. URL https://arxiv.org/abs/2601.13238.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-Zero: Self-evolving reasoning LLM from zero data. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=96apU6YzSO.

Jianyu Lai, Sixiang Chen, Yunlong Lin, Tian Ye, Yun Liu, Song Fei, Zhaohu Xing, Hongtao Wu, Weiming Wang, and Lei Zhu. SnowMaster: Comprehensive real-world image desnowing via MLLM with multi-model feedback optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4302–4312, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Lai_SnowMaster_Comprehensive_Real-world_Image_Desnowing_via_MLLM_with_Multi-Model_Feedback_CVPR_2025_paper.html.

Chunyi Li, Jianbo Zhang, Zicheng Zhang, Haoning Wu, Yuan Tian, Wei Sun, Guo Lu, Xiongkuo Min, Xiaohong Liu, Weisi Lin, Xiao-Ping Zhang, and Guangtao Zhai. R-Bench: Are your large multimodal model robust to real-world corruptions? *IEEE Journal of Selected Topics in Signal Processing*, 19(7):1349–1361, 2025. doi: 10.1109/jstsp.2025.3558652. URL https://doi.org/10.1109/jstsp.2025.3558652.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne 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. URL https://aclanthology.org/2023.emnlp-main.20/.

Zongxia Li, Hongyang Du, Chengsong Huang, Xiyang Wu, Lantao Yu, Yicheng He, Jing Xie, Xiaomin Wu, Zhichao Liu, Jiarui Zhang, and Fuxiao Liu. MM-Zero: Self-evolving multi-model vision language models from zero data. *arXiv preprint arXiv:2603.09206*, 2026. URL https://arxiv.org/abs/2603.09206.

Jinming Liu, Zhaoyang Jia, Jiahao Li, Bin Li, Xin Jin, Wenjun Zeng, and Yan Lu. When MLLMs meet compression distortion: A coding paradigm tailored to MLLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=YDRoTtmXu1.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. DeepSeek-VL: Towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024. URL https://arxiv.org/abs/2403.05525.

Hashmat Shadab Malik, Fahad Shamshad, Muzammal Naseer, Karthik Nandakumar, Fahad Khan, and Salman Khan. Robust-LLaVA: On the effectiveness of large-scale robust image encoders for multi-modal large language models. *arXiv preprint arXiv:2502.01576*, 2025. URL https://arxiv.org/abs/2502.01576.

Chengzhi Mao, Scott Geng, Junfeng Yang, Xin Wang, and Carl Vondrick. Understanding zero-shot adversarial robustness for large-scale models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2212.07016.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. URL https://arxiv.org/abs/2410.21276.

Xinkuan Qiu, Meina Kan, Yongbin Zhou, and Shiguang Shan. Benchmarking multimodal large language models against image corruptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9014–9023, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Qiu_Benchmarking_Multimodal_Large_Language_Models_Against_Image_Corruptions_ICCV_2025_paper.html.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

Christian Schlarmann, Naman Deep Singh, Francesco Croce, and Matthias Hein. Robust CLIP: Unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43685–43704. PMLR, 2024. URL https://proceedings.mlr.press/v235/schlarmann24a.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

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

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=HvoG8SxggZ.

Hai-Long Sun, Da-Wei Zhou, Yang Li, Shiyin Lu, Chao Yi, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, De-Chuan Zhan, and Han-Jia Ye. Parrot: Multilingual visual instruction tuning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57984–58007. PMLR, 2025. URL https://proceedings.mlr.press/v267/sun25ad.html.

Jiaqi Tang, Jianmin Chen, Wei Wei, Xiaogang Xu, Runtao Liu, Xiangyu Wu, Qipeng Xie, Jiafei Wu, Lei Zhang, and Qifeng Chen. Robust-R1: Degradation-aware reasoning for robust visual understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(11):9421–9429, 2026. doi: 10.1609/aaai.v40i11.37902. URL https://ojs.aaai.org/index.php/AAAI/article/view/37902.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, and Jitao Sang. AMBER: An LLM-free multi-dimensional benchmark for MLLMs hallucination evaluation. *arXiv preprint arXiv:2311.07397*, 2023a. URL https://arxiv.org/abs/2311.07397.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025a. URL https://arxiv.org/abs/2508.18265.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2203.11171.

Zeyu Wang, Cihang Xie, Brian Bartoldson, and Bhavya Kailkhura. Double visual defense: Adversarial pre-training and instruction tuning for improving vision-language model robustness. *arXiv preprint arXiv:2501.09446*, 2025b. URL https://arxiv.org/abs/2501.09446.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. Long-form factuality in large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 80756–80827. Curran Associates, Inc., 2024. doi: 10.52202/079017-2567. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/937ae0e83eb08d2cb8627fe1def8c751-Paper-Conference.pdf.

xAI. Grok-1.5 vision preview, 2024. URL https://x.ai/news/grok-1.5v.

Peng Xia, Kaide Zeng, Jiaqi Liu, Can Qin, Fang Wu, Yiyang Zhou, Caiming Xiong, and Huaxiu Yao. Agent0: Unleashing self-evolving agents from zero data via tool-integrated reasoning. *arXiv preprint arXiv:2511.16043*, 2025. URL https://arxiv.org/abs/2511.16043.

Tianyu Yu, Haoye Zhang, Qiming Li, Qixin Xu, Yuan Yao, Da Chen, Xiaoman Lu, Ganqu Cui, Yunkai Dang, Taiwen He, Xiaocheng Feng, Jun Song, Bo Zheng, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. RLAIF-V: Open-source AI feedback leads to super GPT-4V trustworthiness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19985–19995, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yu_RLAIF-V_Open-Source_AI_Feedback_Leads_to_Super_GPT-4V_Trustworthiness_CVPR_2025_paper.html.

Fan Yuan, Chi Qin, Xiaogang Xu, and Piji Li. HELPD: Mitigating hallucination of LVLMs by hierarchical feedback learning with vision-enhanced penalty decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1768–1785. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.emnlp-main.105. URL https://aclanthology.org/2024.emnlp-main.105/.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57905–57923. PMLR, 2024b. URL https://proceedings.mlr.press/v235/yuan24d.html.

Shimao Zhang, Xiao Liu, Xin Zhang, Junxiao Liu, Zheheng Luo, Shujian Huang, and Yeyun Gong. Process-based self-rewarding language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 18097–18110. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.930. URL https://aclanthology.org/2025.findings-acl.930/.

APPENDIX OVERVIEW

The appendix is organized so that a reader can check the setup before the results and the results before the interpretation. Appendix A states the experimental protocol: what initializes each role, what is and is not removed from the adaptation data, the optimizer and hardware, how R-Bench scores each task, and what our comparisons leave uncontrolled. Appendix B defines the claim taxonomy and its mapping onto probe types. Appendix C derives the variance formula of Proposition 2, the composition-bias identity of Section 3.3, and the sense in which GRPO is invariant to rescaling the reward. Appendix D gives the complete result tables, including the scores originally reported by each baseline on its own backbone, the full synthetic-corruption numbers behind Figure 3, the probe-count sweep, and the backbone-transfer run. Appendix E reproduces the prompt templates of the four verification roles. Appendix F reports the training-time cost accounting. Appendix G walks through one real execution on a heavily distorted street scene and, in Appendix G.4, draws the failure modes that trace exposes. Appendix H discusses limitations and Appendix I outlines future work.

A EXPERIMENTAL PROTOCOL

This appendix gathers in one place the facts a reader needs in order to judge what our numbers compare. Nothing here is new: it is assembled from Sections 3.1, 3.4, and 4.1, where the same facts are distributed across three subsections and are easy to lose track of. The last part, Appendix A.5, is the one we would most want a reader to see, because it lists what the experiments do *not* control.

A.1 INITIALIZATION AND WHAT IS TRAINED

The policy π_θ and all four verification roles $\mathcal{V}^{\mathrm{fp}}$, $\mathcal{V}^{\mathrm{qg}}$, $\mathcal{V}^{\mathrm{vc}}$, and $\mathcal{V}^{\mathrm{ca}}$ are initialized from the same Qwen3-VL-4B checkpoint (Bai et al., 2025a). The roles differ from one another only in their prompts (Appendix E), and their parameters ϕ are held fixed for the entire run: only π_θ receives gradients.

Two consequences are worth naming. The verifier is a compute cost rather than a parameter cost, which is why Appendix F accounts for it in logical calls and GPU-hours rather than in parameters. And because ϕ never moves, the verifier the policy is optimized against at the end of training is the same verifier it faced at the start, so the target does not drift with the policy. The backbone-transfer run of Table 7 repeats the same arrangement with InternVL3.5-4B (Wang et al., 2025a) supplying both the policy and the four roles.

A.2 ADAPTATION DATA: LABEL-FREE BUT NOT CURATION-FREE

Adaptation uses the degraded images of the Robust-R1 corpus (Tang et al., 2026). That corpus ships each degraded image together with human-curated material: the clean counterpart, the degradation type and its parameters, a perceptual analysis, and a step-by-step reasoning chain. We discard all of it. What reaches the training loop is the corrupted pixels and one fixed descriptive instruction q_{desc}, with no answer, no reference caption, and no label of any kind.

We state the scope of that claim carefully because it is easy to overread. Discarding the annotations makes the procedure label-free, in the sense that nothing in the reward depends on a human judgment about any particular image. It does not make the procedure curation-free: someone decided which scenes and which degradations belonged in that corpus, and we inherit the selection. These experiments therefore do not show that the loop behaves the same way on an arbitrary uncurated image stream.

A.3 OPTIMIZER AND HARDWARE

Adaptation runs GRPO (Shao et al., 2024) as written in Algorithm 1, with group size $G = 16$, $M = 10$ probes per claim, clipping $\epsilon = 0.2$, KL weight $\beta = 0.02$ against the frozen initial checkpoint, learning rate 1×10^{-6}, and three epochs. The run processes $N = 600$ image instances on $8\times$ NVIDIA L20 GPUs and takes 237.85 GPU-hours, or 29.73 hours of wall clock; Table 8 gives the full accounting. Settings the paper does not fix, including the Voting Critic's sampling temperature, the batch size, and the maximum generation length, are left to the configuration files we

will release upon acceptance rather than restated here, so that every number printed in the paper is one we measured or set deliberately.

A.4 How R-Bench is scored

R-Bench (Li et al., 2025) evaluates three task formats at three corruption severities. MCQ is scored as binary correctness against the annotated option. VQA and CAP have no single correct string, so they are scored by a judge model along three axes (completeness, precision, and relevance), with the evaluation repeated and the scores averaged into the interval $[0, 1]$. The Overall column reported throughout the paper is the unweighted mean of the nine task–severity cells, computed before rounding, so a method can lead on Overall while losing individual cells.

Two properties of this protocol matter for how our results should be read. CAP is the only free-form task, and it is also the format closest to the descriptive instruction used during adaptation, which is the basis for the reading in Section 4.2 that the reward transfers within a format. And because VQA and CAP are judged by a model, their scores inherit that judge's behaviour; this is external to our method and identical across every row of Table 1, but it means the comparison is between methods under one judge, not against ground truth.

A.5 What is not controlled

Three things are not controlled, and we prefer to list them explicitly rather than leave them to be inferred.

First, every configuration in every table is a single training run. We did not repeat any configuration with a different seed, so we have no estimate of run-to-run variance and no confidence intervals. Differences of a few thousandths of an Overall point should be read as ties, which includes several of the rankings in Table 2 and some of the per-cell marks in Table 1.

Second, verifier compute is not matched across the rows of Table 2. A holistic self-score costs one judging call per candidate description and its averaged variant costs a few; the full pipeline costs one decomposition call, $(1 + r_q)$ probe-generation calls, $\bar{K}M$ critic calls, and $\bar{K}$ judging calls per candidate (Appendix F). The 0.0067 gap between 0.6715 and 0.6782 is therefore evidence that a claim-structured reward outperforms a holistic one under an identical optimizer; it is not evidence that structure rather than verifier budget produced the difference. The $M{=}1$ row is the closest thing we have to a budget-reduced control, and it still issues the same decomposition and probe-generation calls plus $\bar{K}$ critic and $\bar{K}$ judging calls per candidate, so it is not compute-matched either. A properly compute-matched comparison would need a run we did not perform.

Third, the baselines in Table 1 are rebuilt on a common backbone rather than run in their authors' original configurations. Rebuilding a method this way removes the confound of backbone strength but can change the method in ways we cannot audit, which is why Table 4 reproduces the baseline configurations reported by Tang et al. (2026) alongside ours instead of replacing it.

B Claim Taxonomy

This appendix answers a question the main text leaves open: what exactly counts as a claim, and how does a claim of a given kind become a probe? The answer matters because the reward of Equation 7 weights all claim types equally, so the taxonomy fixes what the reward is an average over.

The Fact Parser labels every claim with one of six semantic categories, and the Query Generator maps each category onto a probe form whose answers are short enough to compare. The two prompts use different surface strings for the same categories, which Table 3 makes explicit. The taxonomy is a practical choice aimed at literal, visually checkable content, not a claim about the space of visual statements; Appendix H discusses the content it does not cover.

C Derivations for Section 3.3

Section 3.3 states three small results without working them out: the variance of the agreement estimator under correlated verdicts, the effect on the reward of appending one claim, and the sense

Table 3: **The claim taxonomy.** Every atomic claim extracted from a description carries one of six semantic categories, and each category fixes the form of the probe question used to test it. The first column gives the display names used in Figure 2; the next two give the literal `type` strings emitted by the decomposition and probe prompts of Appendix E, which differ from each other and from the display names. The fourth column states why the category is sensitive to corruption, and the fifth gives an example. The reward of Equation 7 weights all six categories equally.

Category	Claim	Probe	Why corruption affects it	Example claim
Object existence	`object`	`existence`	Blur hides small objects and noise can be read as objects.	"There is a traffic light in the image."
Object attributes	`attribute`	`attribute`	Colour shifts and flattened texture change apparent attributes.	"The car is red."
Spatial relations	`relation`	`spatial`	Reduced contrast merges boundaries and confuses relative position.	"The cup is left of the keyboard."
Scene semantics	`scene`	`scene`	Lost global illumination makes the setting ambiguous.	"The image shows an indoor office."
Quantity	`quantity`	`count`	Merged or occluded instances cause miscounting.	"There are three visible windows."
State & action	`state`	`state`	Motion blur erases action cues and fine pose detail.	"The person is walking."

in which GRPO does not care about the scale of the reward. None is deep, but each carries an assumption that is worth seeing, so we give the derivations here and name the assumptions as we use them.

C.1 Variance of the agreement estimator (Proposition 2)

Fix a claim h and an image x, and write $V^{(1)}, \dots, V^{(M)}$ for the M verdicts the Credit Assigner returns. We assume the verdicts are *exchangeable* given (x, h): each has the same marginal mean $\mu = \mathbb{E}[V \mid x, h]$ and the same marginal variance $\sigma^2 = \mu(1-\mu)$, and every pair has the same correlation $\rho = \mathrm{Corr}(V^{(i)}, V^{(j)})$ for $i \neq j$, with $\rho \in [0, 1]$. Unbiasedness is immediate from the common mean, $\mathbb{E}[\hat{\mu} \mid x, h] = \frac{1}{M}\sum_m \mathbb{E}[V^{(m)} \mid x, h] = \mu$. For the variance, expand the double sum and use $\mathrm{Cov}(V^{(i)}, V^{(j)}) = \rho\sigma^2$ for $i \neq j$:

$$\begin{aligned} \mathrm{Var}(\hat{\mu} \mid x, h) &= \frac{1}{M^2}\Big[\sum_{m=1}^{M} \mathrm{Var}\big(V^{(m)}\big) + \sum_{i \neq j} \mathrm{Cov}\big(V^{(i)}, V^{(j)}\big)\Big] \\ &= \frac{1}{M^2}\Big[M\sigma^2 + M(M-1)\rho\sigma^2\Big] = \frac{\sigma^2}{M}\big[1 + (M-1)\rho\big]. \end{aligned} \tag{11}$$

Two readings follow. Letting $M \to \infty$ gives $\mathrm{Var}(\hat{\mu}) \to \rho\sigma^2$, a floor that no amount of repeated probing removes and that vanishes only when $\rho = 0$. Equating the expression to σ^2/M_{eff}, the variance an independent sample of size M_{eff} would have, gives

$$M_{\text{eff}} = \frac{M}{1 + (M-1)\rho} \xrightarrow[M \to \infty]{} \frac{1}{\rho}, \tag{12}$$

so a budget of M correlated probes is worth at most $1/\rho$ independent ones however large M becomes.

We flag the assumptions rather than hide them. Exchangeability is a modelling choice, not something we verify: the M critic answers are drawn from one prompt and one decoding distribution, which makes it plausible, but the Credit Assigner then judges all M answers inside a single call (Appendix E.4), so the verdicts are produced jointly and the case study shows identical critic answers receiving different verdicts (Appendix G.4). A single common pairwise correlation is a further simplification. We also do not estimate ρ; doing so would require replicating the whole M-draw procedure many times per claim, which we did not run, and it is why Appendix I lists measuring ρ as future work rather than reporting a value.

C.2 Composition bias of averaging over claims

Let a description contain K claims with agreement rates $\hat{\mu}_1, \ldots, \hat{\mu}_K$, so that $R_\phi = \frac{1}{K}\sum_k \hat{\mu}_k$ by Equation 7. Append one further claim with agreement rate $\hat{\mu}_{\text{new}}$ and call the new reward R'_ϕ. Then

$$R'_\phi = \frac{K R_\phi + \hat{\mu}_{\text{new}}}{K+1} = R_\phi + \frac{\hat{\mu}_{\text{new}} - R_\phi}{K+1}, \qquad \text{so} \qquad R'_\phi > R_\phi \iff \hat{\mu}_{\text{new}} > R_\phi. \tag{13}$$

The identity is exact and it has three consequences. A claim is rewarded purely for being easier to agree with than the current average, independently of whether it adds anything the reader wants; the decomposition prompt of Appendix E.1 explicitly asks for high recall, which supplies a ready stock of such claims. The size of the effect decays as $1/(K+1)$, so the incentive is strongest for short descriptions and weak once a description is long; with the observed $\bar{K} = 30.59$ claims per description, a single appended claim moves the reward by at most $1/(\bar{K}+1)$. By the same factor, a single unsupported claim is penalized by at most $R_\phi/(K+1)$, which is exactly the localization that makes the signal usable and also the reason the penalty is small.

The asymmetry that matters is that the sum runs over claims the Fact Parser found *in* y. There is no term for content the description never mentioned, so omission is invisible to the reward: a description can raise its score by saying more, never by saying the right things. We did not measure the resulting length or composition drift over training, and Appendix H records that as a gap in the evaluation.

C.3 Why GRPO is indifferent to the scale of the reward

Take a group of G candidates for one image with rewards $R^{(1)}, \ldots, R^{(G)}$, group mean μ_{grp}, group standard deviation σ_{grp}, and advantages $A^{(g)} = (R^{(g)} - \mu_{\text{grp}})/(\sigma_{\text{grp}} + \delta)$ with δ the numerical stabilizer of Section 3.4. Rescale every reward in the group by the same positive affine map, $\tilde{R}^{(g)} = cR^{(g)} + d$ with $c > 0$. The group statistics transform as $\tilde{\mu}_{\text{grp}} = c\mu_{\text{grp}} + d$ and $\tilde{\sigma}_{\text{grp}} = c\,\sigma_{\text{grp}}$, so the offset d cancels and

$$\tilde{A}^{(g)} = \frac{c\big(R^{(g)} - \mu_{\text{grp}}\big)}{c\,\sigma_{\text{grp}} + \delta} = \lambda\, A^{(g)}, \qquad \lambda = \frac{c\,(\sigma_{\text{grp}} + \delta)}{c\,\sigma_{\text{grp}} + \delta} > 0. \tag{14}$$

With $\delta = 0$ the advantages are exactly unchanged. With $\delta > 0$ they are multiplied by one positive constant shared by the whole group, so their signs and their ordering are unchanged and the rescaling acts only as a group-wise change of step size. Either way, no candidate's advantage changes sign and no pair changes places.

This is the precise reason calibration is irrelevant to the argument of Section 3.3. A reward that systematically overstates or understates how often claims are true, but preserves the within-group ordering of candidates, drives the same update. It is also the reason Proposition 3 does not by itself sink the method: not knowing α_t and β_t would be fatal if the reward had to be read as a probability, and it is not fatal if the reward only has to rank.

The invariance should not be overextended. It is a statement about a group of candidate responses, not about individual claims. Proposition 3 gives an affine relation between $\mathbb{E}[\hat{\mu} \mid T, t]$ and T whose coefficients depend on the claim type t, so when a response mixes types its reward is an average of differently scaled per-claim signals and is not an affine function of any single underlying quantity. Invariance removes the need to calibrate the reward scale; it does not remove the need for Condition 10, which remains an assumption we can test only indirectly, through benchmark scores measured against external ground truth.

D Complete Result Tables

This appendix reports every number behind the main-text tables and figure, including the rows we omitted there for legibility and the originally published configurations we chose not to rank against our unified evaluation.

R-Bench, including originally reported configurations. Table 4 extends Table 1 with baseline scores reported in Table 1 of Tang et al. (2026). These rows are context only and are not ranked

against the rest, because they differ in backbone, training data, and evaluation harness simultaneously. They are nonetheless informative about how much of a published method's behaviour depends on its original backbone: Robust LLaVA in its released configuration produces captions that the R-Bench judge scores near zero (0.0068 at low severity), whereas the same method rebuilt on Qwen3-VL-4B reaches 0.4179. Gemma3-4B and InternVL-4B appear only as native checkpoints.

Table 4: **Full R-Bench results.** The upper block reproduces baseline scores from Table 1 of Tang et al. (2026); the lower block is the evaluation of Table 1 (native 4B MLLMs plus adaptation methods on Qwen3-VL-4B (Bai et al., 2025a)). Ranking marks apply within the lower block only.

Method	MCQ			VQA			CAP			Overall
	low	mid	high	low	mid	high	low	mid	high	
Reported on original backbones										
Robust CLIP	0.4705	0.4658	0.4024	0.4503	0.4339	0.4743	0.2290	0.2219	0.1983	0.3718
Robust LLaVA	0.3352	0.2608	0.3048	0.2607	0.2212	0.2443	0.0068	0.0065	0.0067	0.1830
TeCoA	0.4647	0.4223	0.4024	0.4687	0.3994	0.4461	0.2111	0.2195	0.1937	0.3586
Robust-R1	0.6529	0.6391	0.6097	0.4914	0.4909	0.4980	0.4068	0.3781	0.3484	0.5017
4B MLLMs / Qwen3-VL-4B adaptation										
Gemma3-4B	0.5823	0.5776	0.5060	0.4865	0.4630	0.4419	0.4048	0.3746	0.3480	0.4649
InternVL-4B	0.6235	0.6024	0.5914	0.4982	0.4539	0.5108	0.3667	0.3041	0.2851	0.4706
Qwen3-VL-4B	0.7353	0.7081	0.6768	0.6202	**0.6509**	0.6006	0.6938	0.6948	0.6235	0.6671
Robust CLIP	0.6059	0.4783	0.5061	0.5037	0.4715	0.5198	0.3877	0.4071	0.3624	0.4714
Robust LLaVA	0.5706	0.5217	0.5244	0.4644	0.4697	0.4952	0.4179	0.3775	0.3855	0.4697
TeCoA	0.5882	0.5342	0.4329	0.4037	0.4206	0.4371	0.2815	0.3260	0.2838	0.4120
Robust-R1	0.7118	0.6646	0.6220	**0.6380**	0.5988	0.5994	0.4370	0.4790	0.4814	0.5813
Mint	0.7353	**0.7267**	0.6707	0.6307	0.6436	0.6234	0.6827	0.6824	0.6182	0.6682
MM-Zero	0.7451	0.7143	0.6707	0.6307	**0.6509**	0.6138	0.6895	0.6969	0.6266	0.6709
MAE	0.7118	0.7205	0.6707	0.6313	0.6345	0.6084	0.6685	0.6501	0.6004	0.6551
EvoRobust	**0.7471**	**0.7267**	**0.6890**	0.6368	0.6461	**0.6269**	**0.6988**	**0.6988**	**0.6336**	**0.6782**

Synthetic corruption, all methods. Table 5 gives the numbers behind Figure 3 together with the baselines omitted from the plot. One entry is anomalous and we flag it rather than smoothing it over: TeCoA, as reported in Table 2 of Tang et al. (2026), scores higher at 25% corruption (65.71) than on clean MMMB (57.17), which we could not reconcile with the surrounding rows.

Probe count and backbone transfer. Table 6 gives the probe-count sweep discussed in Section 4.3 and Table 7 the InternVL3.5-4B run discussed in Section 4.5.

E Prompt Templates of the Verification Roles

This appendix is the specification of the method: because the four roles differ from one another only in their prompts, the prompts below are the whole of what distinguishes a Fact Parser from a Credit Assigner. A reader checking whether a design choice discussed in Section 3.2 is really implemented should look here.

All four roles share the Qwen3-VL-4B checkpoint (Bai et al., 2025a) and are specialized only by these prompts; no parameters are trained beyond the policy. Placeholders in curly braces are filled at run time by the output of the preceding stage. The role names used in the paper describe the function of each stage and do not always match the wording inside the prompt, which is reproduced as implemented rather than cleaned up for presentation.

E.1 Fact Parser

The Fact Parser turns the description y into the claim set $\{h_k\}$. The system message fixes the output to a JSON array, and the user message enforces a strict type-to-element schema matching Table 3, a recall checklist, and constraints against duplicates. Note that the high-recall instruction interacts with the averaging bias discussed in Section 3.3: it improves coverage of the description but also encourages many easily supported claims.

Table 5: **Full synthetic-corruption results**, accuracy (%) on clean images and at three corruption intensities. The upper block is reproduced from Table 2 of Tang et al. (2026); remaining native 4B MLLMs are unadapted checkpoints; adaptation methods use Qwen3-VL-4B (Bai et al., 2025a). Ranking marks apply within the lower block only, matching Figure 3.

Method	MMMB				MMStar				RealWorldQA			
	clean	25%	50%	100%	clean	25%	50%	100%	clean	25%	50%	100%
Reported on original backbones												
Robust CLIP	58.83	58.28	57.97	53.33	33.00	32.26	31.80	29.46	43.26	42.48	42.61	41.43
TeCoA	57.17	65.71	56.11	51.76	30.46	30.60	30.73	28.06	40.00	39.73	39.47	38.69
Robust-R1	81.41	79.49	79.04	75.35	56.86	54.40	53.60	49.53	67.71	66.40	67.05	63.26
4B MLLMs / Qwen3-VL-4B adaptation												
Gemma3-4B	71.01	70.30	70.20	69.14	43.93	43.20	42.60	41.33	55.42	54.77	53.72	52.81
InternVL-4B	77.97	77.47	76.66	74.59	51.53	50.26	49.60	46.93	57.38	58.16	57.64	54.90
Qwen3-VL-4B	84.44	83.69	82.98	81.01	65.07	**65.13**	64.04	59.13	71.24	70.06	69.93	69.21
Robust CLIP	75.05	73.94	72.63	67.27	42.53	42.67	42.27	40.53	50.07	50.07	49.67	47.06
Robust LLaVA	73.28	72.63	71.77	66.41	41.73	40.47	40.07	38.53	50.07	50.33	50.46	48.50
TeCoA	72.12	71.46	70.56	65.51	37.67	37.53	38.07	36.00	45.88	46.01	46.14	44.31
Robust-R1	79.29	77.63	77.53	76.31	58.13	56.53	56.93	51.73	69.02	68.37	67.45	65.36
Mint	84.14	83.48	83.08	**81.56**	63.20	63.27	60.47	57.53	70.07	69.80	68.63	68.37
MM-Zero	84.29	**83.89**	83.38	81.16	65.67	64.26	63.33	59.07	70.59	69.93	70.20	68.50
MAE	84.39	83.33	83.48	80.91	63.53	63.80	62.27	58.33	71.11	70.06	70.07	67.32
EvoRobust	**84.75**	**83.89**	**83.74**	81.47	**65.87**	64.47	**64.07**	**60.07**	**71.77**	**70.20**	**71.11**	**69.28**

Table 6: **Probe-count sweep.** R-Bench Overall after adaptation with M Voting Critic draws per probe, all else fixed. Gains flatten at M=10 and do not rise at M=20, matching Proposition 2 for correlated verdicts. One run per row.

M	Overall
1	0.6720
3	0.6723
5	0.6750
10 (default)	**0.6782**
20	0.6774

Table 7: **Backbone transfer.** The same self-improvement pipeline applied to InternVL3.5-4B (Wang et al., 2025a), which supplies both the policy and the four frozen verification roles, evaluated on R-Bench. Δ is the Overall gain over the unadapted backbone. One extra model family, one run.

Backbone	MCQ	VQA	CAP	Overall	Δ
InternVL3.5-4B	0.6630	0.4871	0.3390	0.4964	–
+ EvoRobust	**0.6848**	**0.5088**	**0.3428**	**0.5121**	+0.0157

Prompt: Fact Parser

[System] You are an expert at analyzing image descriptions and extracting atomic visual elements. You MUST output ONLY a valid JSON array, with no other text.

[User] Analyze the following image description and decompose it into atomic elements. Aim for 15–30+ elements when the description is rich. MORE elements is better, provided they are non-duplicate and grounded.

Granularity rule. One object instance per item; one attribute value per item; one relation per item; one count per item.

Strict schema rules (type ↔ element MUST match):

- `object`: `element` is the object name (short noun phrase); `detail` lists concrete visual facts of *that* object only.
- `attribute`: `element` is the attribute word (color, material, pattern, size, shape, texture, lighting, . . .); `detail` names the anchor object and the value.
- `scene`: `element` is the scene/setting name; `detail` may add time-of-day or atmosphere but stays scene-level.

- `relation`: `element` is the relation word (e.g. `left_of`, `above`, `holding`, `wearing`); `detail` names both ends in natural language.
- `quantity`: `element` is `count`; `detail` states what is counted and the number (or *unclear*).
- `state`: `element` is the action/posture/expression word (e.g. `walking`, `sitting`, `eyes_open`); `detail` names the anchor subject (person/animal) and the observed action, posture, or expression.

High-recall checklist. People (each person separately; clothing); objects (furniture, devices, food/drink, decor, windows, screens); attributes (colors, materials, patterns, lighting, screen content); relations (above/-below/left/right/holding/wearing/looking_at/near); quantities (explicit or implied); states/actions (visible actions, postures, or expressions of people/animals).

Hard constraints. Every item must have non-empty `element` and `detail`; at least one element per type must appear when applicable; objects must NOT be labeled as `scene`; hedging-only items are forbidden; the same fact must not be duplicated under multiple types.

Description: `{description}`

Output a single JSON array; each item has keys `"type"`, `"element"`, `"detail"`. Output ONLY the JSON array.

E.2 Query Generator

The Query Generator converts the claim set into probes (q_k, a_k). Because a rich description can yield 20–30 claims, the prompt requires at least one probe for every claim index, which prevents the probes from concentrating on the first few claims. The field named `expected_answer` in the prompt is the claim-implied candidate answer a_k of Section 3.2; it is derived from the policy's own description and is not a ground-truth answer.

Prompt: Query Generator

[System] You are an expert examiner. Your task is to create a visual question-answering test based on a structured description of a DEGRADED image (the real image suffers from noise, blur, or low light). Output ONLY a valid JSON array.

[User] Let N be the number of items in the `Elements` list.

Hard rule on balanced coverage. You MUST output at least one question for every index $0, 1, \ldots, N-1$, in ascending order. Multiple questions per index are allowed.

Each question targets a single element (named by `element_index`) and aligns with its type:

- `existence`: whether a specific object is present.
- `attribute`: color, texture, material, or shape of an object (ask only what survives degradation).
- `spatial`: relative position of two objects.
- `scene`: global setting or environment type.
- `count`: number of specific items.
- `state`: action, posture, or expression of people/animals.

Elements (from the candidate description; the image is degraded): `{elements_json}`

Output ONE JSON array; each item has keys `"element_index"`, `"question"`, `"expected_answer"`.

E.3 Voting Critic

The Voting Critic is queried M times per probe at non-zero temperature. The system message below is reproduced verbatim; the accompanying user message supplies the degraded image together with the single probe question q_k. Three properties matter for the estimator in Proposition 2: brevity, so that answers are comparable; literalness, so that an unreadable detail produces *"unclear"* instead of a confident guess (Ding et al., 2024; Yuan et al., 2024a); and type-aware formatting.

Prompt: Voting Critic

[System] You are the Voting Critic: answer verification questions using ONLY pixels visible in the degraded image. Reply with at most 1–24 words; no reasoning steps, no "I think", no full sentences unless required by the answer.

Core rules.

- Answer the SAME question; do not switch to a different object, attribute, or topic.
- Prefer literal, checkable facts over storytelling. If the image is too blurry for the asked detail, reply *"unclear"* or *"cannot tell"* rather than guessing.
- Never invent objects, counts, colors, or actions that are not supported by the image.
- If several salient activities co-occur, list all of them in order of prominence, separated by "`;` ".

Format by question type.

- Yes/No: start with "Yes" or "No"; optionally add up to 6 words of evidence.
- Count: a non-negative integer, or *"unclear"*.
- Color: one color word, or *"unclear"*.
- Object name: a short noun phrase matching the asked region.
- Location: a short phrase (e.g. "on the table", "left side").
- Activity: list visible human actions first; combine multiple briefly with "`;` ".

Degraded image. If a detail is occluded or unreadable, prefer *"unclear"* over a confidently wrong specific answer.

E.4 Credit Assigner

The Credit Assigner sees the claim, the probe, the candidate answer, all M critic answers, and the same degraded image, and returns one verdict per critic answer. Judging the answers in a single call rather than one at a time lets the judge use the spread of the critic answers as context, which is convenient but also means the M verdicts are not produced independently; this is one concrete source of the correlation ρ in Proposition 2.

Prompt: Credit Assigner

[System] You are a visual consistency judge. You see the SAME degraded image as the policy and the Voting Critic. Given an atomic element and MULTIPLE critic samples (from separate vision passes on the same image), decide for EACH critic sample independently whether the element, that critic sample, and the image are mutually compatible. Output ONLY a JSON array of strings, one per critic sample in order, each being exactly "`consistent`" or "`inconsistent`".

[User] For the atomic element, verification question, reference answer, and the `{n_samples}` critic samples shown below, decide the consistency of EACH critic sample.

Atomic element (from the candidate description):

- Type: `{elem_type}`
- Element: `{elem_name}`
- Detail: `{elem_detail}`

Verification question: `{question}`
Reference answer (hint, may be incomplete): `{expected_answer}`
Critic samples (`{n_samples}` independent vision passes on this image): `{samples_list}`

Decision rules.

- *consistent* if the element/detail is supported by the image AND the critic sample does not contradict the image. Wording differences between critic samples and reference are tolerated.
- *inconsistent* only if (i) the critic sample and the element/detail clearly contradict each other, or (ii) the element/detail is clearly wrong or unsupported by the image.

Output ONLY a JSON array of `{n_samples}` strings, one per critic sample in the same order.

F Training Cost

This appendix says what the method costs to run and where the cost goes. The short answer is that almost nine calls in ten are Voting Critic calls, so the price of the approach is the price of repeated

probing; anyone deciding whether to reproduce or extend the method should size their budget from that term and not from the policy update.

We count one logical call as one generation or judging request issued to the policy or to a verification role, which makes the accounting independent of batching. Let N be the number of image instances processed before the final checkpoint, G the candidates per image, $\bar{K}$ the mean claims per candidate, M the probes per claim, and r_q the observed retry rate of the Query Generator when its output fails schema validation. The total is

$$C_{\text{train}} = NG\Big[\underbrace{1}_{\text{policy}} + \underbrace{1}_{\text{decompose}} + \underbrace{(1+r_q)}_{\text{probe}} + \underbrace{\bar{K}M}_{\text{critic}} + \underbrace{\bar{K}}_{\text{judge}}\Big]. \tag{15}$$

Equation 15 assumes one probe per claim; the prompt permits more, so the critic and judge terms are lower bounds. With N=600, G=16, $\bar{K}$=30.59, M=10, and r_q=0.7656, the run issues 3.27M logical calls, of which 3.26M are frozen-role reward calls. The Voting Critic term alone, $NG\bar{K}M = 2.94$M, is 89.9% of the total, so the bottleneck is repeated image-grounded probing rather than the policy update.

Table 8: **Training cost** of the final EVOROBUST checkpoint, adapted on 600 unlabeled degraded images. A logical call is one generation or judging request issued to the policy or to a frozen verification role, counted before batching, so these counts are implementation-independent; GPU-hours, wall-clock time, and throughput are measured from the batched implementation and aggregated over the eight devices. Voting Critic calls alone are 89.9% of the total. This cost is incurred once during adaptation and not at test time.

Cost item	**Value**
Processed image instances (N)	600
Candidate descriptions (NG)	9,600
Mean claims per description ($\bar{K}$)	30.59
Voting Critic logical calls ($NG\bar{K}M$)	2.94M
Total frozen-role reward calls	3.26M
Total logical calls including policy rollout	3.27M
Hardware	8× NVIDIA L20 GPUs
GPU-hours / wall-clock time	237.85 / 29.73 hours
Tokens seen / mean throughput	7.77M / 73.06 tokens/s

This is an offline training cost. The learned policy is evaluated as an ordinary checkpoint, so deployment uses the same single-model interface as the base MLLM and none of the verification roles run at test time. The cost-performance trade-off is therefore governed by the verifier budget during training, chiefly M, which is what makes the saturation in Table 6 economically relevant.

G CASE STUDY

This appendix follows one real claim-level self-verification execution end to end, from the corrupted image through the policy's description, the extracted claims, the generated probes, the ten critic answers per probe, and the judge's verdicts, to the agreement rates that become the reward. It is the most concrete thing in the paper: everything Section 3.2 describes in symbols appears here as actual text produced by the model, and Appendix G.4 then reads off the four ways the signal misbehaves.

The example is a heavily manipulated street scene (Figure 4): a glitch-art street corner carrying a red *"COPY"* watermark, coloured overlay lines, and geometric distortion. We chose it because the policy's description mixes correct global content with unreadable fine detail, which is exactly where per-claim scoring behaves differently from a single response-level score.

G.1 STAGE 0: POLICY DESCRIPTION

This is the object everything else in the trace is derived from: one sample y from the policy, produced from the corrupted image and the single fixed instruction used throughout adaptation. Reading it



(a) Corrupted image x: the only input to the policy and to all four verification roles.

(b) Clean counterpart, shown for the reader only and never accessed by the method.

Figure 4: **The street scene used in the walkthrough of Appendix G.** A glitch-art street corner carrying a red *"COPY"* watermark, coloured overlay lines, and geometric distortion. Only (a) is available to EVOROBUST: the policy describes it, and all four verification roles read the same pixels when scoring that description. We reproduce (b) so that the reader can see which claims are in fact grounded, an assessment the method itself cannot make because it never has access to a clean counterpart.

against the clean counterpart in Figure 4b shows the pattern the method is built for, in which the global content is right and the fine detail is not.

Given x and the descriptive query *"Please provide a comprehensive and detailed description of this image. Include all visible objects, their attributes, spatial relationships, the overall scene, and any quantities you can observe."*, the policy returns a long description; the salient excerpts follow.

Trace: Stage 0: policy description y

Overall scene. A surreal, digitally manipulated suburban street corner under a clear blue sky; buildings, trees, and road signs bend unnaturally; bright daylight with clear shadows.

Traffic signage. A large, red, octagonal *STOP* sign tilted in the upper right; below it a yellow rectangular warning sign with black text reading *"DRIVER CROSSING TRAFFIC DOES NOT STOP"*.

Vehicles. A dark blue sedan is parked on the left, angled toward the viewer. In the foreground a trash bin shows graffiti that looks like *"B-H-O"*; a portion of a black car is visible at the bottom right.

Infrastructure. Palm trees curving into spirals; power lines running diagonally across the top; a slanted wooden fence in front of a dark-roofed house.

Graphic overlays. A large bold red word *"COPY"* is superimposed centrally; multiple colored lines (purple, cyan, yellow, green) intersect across the scene; several small rectangular fragments float in the upper right.

G.2 STAGES 1–2: CLAIMS AND PROBES

Here the description becomes a set of separately checkable questions. The point to watch is how uneven the resulting set is: one paragraph produces thirty claims of very different difficulty, and the reward will treat them all alike.

Decomposition yields $K{=}30$ claims: 1 scene, 9 object, 16 attribute, 2 relation, 1 quantity, and 1 state. The composition is itself informative, since attribute claims dominate and the reward in Equation 7 weights all types equally. Six probes are shown below; they were chosen to span the range of outcomes and are therefore not a random sample.

Trace: Stages 1–2: probes and claim-implied candidate answers

`Q[0]`	*What is the general setting of the image?* → *suburban street corner under clear blue sky… chaotic due to digital distortion.*
`Q[13]`	*What colors are the colorful lines?* → *purple, cyan, yellow, green.*
`Q[14]`	*What text is on the yellow road sign?* → *DRIVER CROSSING TRAFFIC DOES NOT STOP.*
`Q[15]`	*What is the text content on the trash bin?* → *B-H-O.*
`Q[18]`	*What shape are the power lines?* → *stretched and bent.*
`Q[29]`	*How many color lines are there?* → *multiple.*

G.3 Stages 3–4: Probing and Judging

This is where the numbers that make up the reward are produced. Table 9 is the summary; the paragraph below it describes the raw critic answers behind each row, because the spread of those answers is what the agreement rate is measuring.

Each probe is answered $M{=}10$ times on x. The raw answers fall into four patterns. Q[0] is answered consistently by all ten samples with varied wording ("urban street with stop sign, parked car", "street corner with residential houses"), so the global scene survives the corruption. Q[13] and Q[29] oscillate between the described palette and a plausible superset that adds pink or red, and between five and six lines. Q[18] is answered "straight" by all ten samples, contradicting the description's "stretched and bent". Q[14] and Q[15] are hard reads: for the trash bin the samples are {"B-H-O", "unclear", "B-40", "unclear", "5640", "B-H-O", "1440", "1440", "640", "B-H-O"}, and for the yellow sign only two or three of ten recover the full string while the rest drift to visually similar variants such as *OPPOSING* or *TURNING*.

Table 9: **Judge verdicts and agreement rates for six probes of the case study.** Each row is one atomic claim taken from the policy's description of the corrupted street scene. The verdict pattern shows what the Credit Assigner returned for each of the $M{=}10$ Voting Critic answers to that probe, consistent (✓) or inconsistent (×), and $\hat{\mu}_k$ is their mean, the per-claim agreement rate of Equation 7. These six probes were selected to span the range of outcomes from 1.00 to 0.00, so their mean is not an estimate of the response-level reward over all 30 claims.

Probe	Claim (summary)	Verdict pattern	$\hat{\mu}_k$
Q[0]	scene: suburban street corner	✓✓✓✓✓✓✓✓✓✓	1.00
Q[13]	attribute: purple, cyan, yellow, green lines	✓✓ × ×✓ × ×✓✓✓	0.60
Q[14]	attribute: "DRIVER CROSSING…DOES NOT STOP"	✓ × × × × × ✓ × ✓×	0.30
Q[15]	attribute: trash-bin graffiti "B-H-O"	✓ × × × ×✓ × × × ✓	0.30
Q[18]	attribute: power lines are *stretched and bent*	× × × × × × × × ××	0.00
Q[29]	quantity: *multiple* coloured lines	×✓✓✓ × ✓ × ✓✓×	0.60

Two observations are favourable. Decomposition localizes the penalty: the unsupported "bent power lines" claim receives $\hat{\mu} = 0.00$ without affecting the credit given to the correct scene claim, which is behaviour a single response-level score cannot produce. And repeated probing turns unstable reads into intermediate values rather than arbitrary binary flags, which is Proposition 2 in practice. The same trace also exposes four ways the signal misbehaves, which we take up next.

G.4 Observed failure modes

The trace above is worth reading for what goes wrong in it as much as for what goes right. Each of the following is visible in the six probes of Table 9, and each corresponds to a term in the analysis of Section 3.3 rather than to an implementation bug we could fix by patching a prompt.

The M verdicts are not conditionally independent. Probe Q[29] asks how many coloured lines there are. Several of the ten critic answers are the identical string "five", yet they do not all receive the same verdict. Since the Credit Assigner judges all ten answers inside one call (Appendix E.4), the verdict on one answer is conditioned on the others, so identical inputs need not produce identical outputs. This is the correlation ρ of Proposition 2 appearing in its most concrete form, and it is the reason we describe $\hat{\mu}_k$ as a noisy property of the verifier rather than a property of the image. Judging each answer in a separate call would remove this particular coupling, at the price of the cross-answer context the prompt was designed to use and of M times as many judging calls.

Claims about the corruption are rewarded like claims about the scene. The description asserts a red *"COPY"* watermark and coloured overlay lines. Those are artifacts of the manipulation, not contents of the street corner, but they are genuinely present in the corrupted pixels, so the critic confirms them and the judge marks the confirmations consistent: Q[13] earns $\hat{\mu} = 0.60$ for the line colours. The reward is behaving exactly as defined, since it asks whether the image supports the claim and the image does. It is the interpretation that needs care: a description can raise its score by describing the damage, and nothing in the reward distinguishes a claim about the scene from a claim about what was done to the image of it.

Mid-range agreement is usually honest uncertainty, not partial correctness. Q[14] and Q[15] both score $\hat{\mu} = 0.30$, and both are hard text reads under occlusion and distortion. The critic answers for the trash bin are {"B-H-O", "unclear", "B-40", "unclear", "5640", "B-H-O", "1440", "1440", "640", "B-H-O"}, and for the yellow sign only two or three of ten recover the full string while the rest drift to visually similar variants. A rate of 0.30 here does not mean the claim is thirty percent true. It means the verifier could resolve the detail about three times in ten, which is the signature Proposition 1 predicts for a detail the corruption has partly destroyed: repetition measures the instability, it does not reduce it.

Stable consensus is only as good as the shared prior. Q[18] is the clearest case of the reward working: all ten critic samples answer "straight", contradicting the description's claim that the power lines are "stretched and bent", and the claim collapses to $\hat{\mu} = 0.00$. Agreement caught an unsupported statement. The danger is the symmetric case. Policy and verifier share one checkpoint, so a claim the backbone is disposed to make is also a claim the backbone is disposed to confirm, and such a claim would produce a stable $\hat{\mu} = 1.00$ that is indistinguishable, from inside the method, from the stable 1.00 that Q[0] earns for a scene reading that is in fact correct. The trace contains no instance we can point to, because identifying one would require the ground truth the method does not use; that this failure cannot be detected from the agreement rates alone is the point.

H LIMITATIONS

This appendix expands the two Limitations paragraphs at the end of Section 4.

Agreement is not truth. The reward is a frozen-verifier agreement rate (Proposition 3): it tracks truth only where that verifier beats chance. Policy and the four roles share one backbone and the same corrupted image, so a shared mistake can yield a stable $\hat{\mu}$=1.00 indistinguishable from a correct consensus (Proposition 2; Appendix G.4). Condition 10 is untestable without labels (Appendix C.3).

Composition bias. Averaging over extracted claims rewards appending a claim easier than the current average and never penalizes omissions (Appendix C.2). We did not measure length or composition drift, and we did not add an omission term.

Scope of evidence and cost. Results are single-run; verifier compute is not matched against holistic self-scoring, so the 0.0067 gap in Table 2 conflates structure with budget (Appendix A.5). Transfer covers one extra backbone; training costs 3.27M logical calls and 237.85 GPU-hours (Appendix F). The taxonomy targets literal, checkable content, so aesthetic, affective, and counterfactual claims largely fall outside it.

I FUTURE WORK

The limitations above point at three specific next steps rather than at a general call for more work, and we list them in the order in which we think they would pay off. Each targets a quantity the analysis of Section 3.3 names: the verdict correlation ρ, the rank-informativeness condition of Equation 10, and the probe budget M that dominates cost.

Decorrelated verification. Proposition 2 identifies verdict correlation as the binding constraint, which makes decorrelation more valuable than more sampling. Verifiers drawn from different model families (Chen et al., 2024c; Gemma Team et al., 2025; Bai et al., 2025b; Deng et al., 2025) or a stronger external verifier (OpenAI et al., 2024; Guo et al., 2025) attack ρ directly, and estimating ρ empirically would turn the choice of M into a measurement rather than a hyperparameter.

Measuring rank-informativeness directly. Condition 10 is what the optimizer needs, and it can be tested without retraining by measuring, on a labelled subset, how often the agreement reward orders two candidate descriptions the same way an external metric does. That would evaluate the surrogate rather than the end-to-end system.

Budget allocation and broader inputs. A fixed M spends the same budget on claims whose answers never disagree and on genuinely ambiguous ones; allocating probes by observed disagreement (Sui et al., 2025) would cut the dominant cost term. Extending the loop to video and embodied inputs requires temporally consistent decomposition and cross-frame probing, and applying it to other shifts such as weather (Hu et al., 2026), desnowing (Lai et al., 2025), or coding artifacts (Liu et al., 2026) would test generality beyond the corruptions studied here.