ByteDance

PRISM: Priority-aware Rubric Internalization via Structured Multimodal Data Synthesis

Xiaomin He[1,2,*,†] Dongling Xiao[1,*] Jiahao Xie[1] Ruiqi Lu[1]
Qianle Wang[1] Zhongbin Guo[1,†] Wanxuan Sun[1,✉]

[1] ByteDance [2] Peking University

Abstract

Real-world multimodal instructions often bundle multiple requirements with unequal importance, yet most multimodal training data still reduce instruction following to answering one self-contained question. We study this gap through **rubric comprehension**, which casts the model not as a generator measured against rubrics but as an **executor** that follows them: given an image and a typed, prioritized rubric, the model must verify each rule before producing an overall judgment. To support this setting, we propose **PRISM**, a four-stage data synthesis framework that produces persona–task pairs, prefix-guided rule sets, quality-filtered rubrics, and structured verification traces. We further introduce **PRISM-Eval**, whose Loose and Strict metrics use deterministic matching against fixed labels and therefore require no inference-time judge model. With only 10K synthesized samples, PRISM lifts Qwen3-VL-4B from 9.5% to 30.1% Strict accuracy on PRISM-Eval while preserving average performance on general benchmarks, and the gains transfer to four additional open-source MLLMs across dense and MoE architectures, suggesting that structured rubric supervision is a scalable path toward multi-rule, priority-aware multimodal instruction following.

✉ **Email:** 2401210613@stu.pku.edu.cn

1. Introduction

Multimodal Large Language Models (MLLMs) have made remarkable progress, approaching or surpassing human-level performance on standard benchmarks of VQA, captioning, and document understanding [1, 2]. Yet as these models move from controlled benchmarks to real-world deployment, a class of user instructions consistently exposes their limits: instructions whose intent is not a single question, but a bundle of requirements that must be jointly satisfied.

Each such requirement is a rule (Figure 1). Since rules differ in importance, faithfully interpreting the instruction is not a single-question answer: the model must parse every rule and reason about their relative priorities. Such instructions appear in rule-based discriminative tasks (e.g., product quality inspection, advertisement compliance review) and constrained generation. This work focuses on the former, using discriminative verification as a measurable proxy for **multi-rule, priority-aware comprehension**.

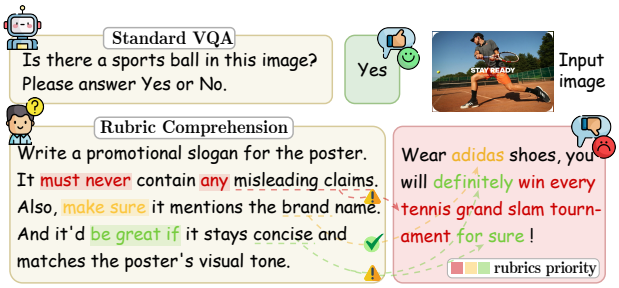


Figure 1: ***From standard VQA to rubric comprehension.*** *For instance, when asked to write a poster slogan, the user may have multiple requirements.*

* Equal contribution. ✉ Corresponding author.
† Work done during an internship at ByteDance.

To systematically train and evaluate this capability, we operationalize it as a discriminative proxy task that we call **rubric comprehension**. Here the user's set of rules is organized into a structured, prioritized collection, a **rubric**, and the model must produce a judgment for each rule it contains.

The core obstacle is a supervision gap: existing multimodal instruction-tuning datasets [3–6] largely follow the paradigm of "understand the image and answer a question" and rarely require interpreting a structured set of prioritized rules. Crucially, scale alone cannot close this gap. Even the strongest closed-source MLLMs remain far from solving rubric comprehension (§4), indicating that the bottleneck lies in the *kind* of supervision rather than in model capacity.

Our key insight is a change of perspective. Prior work predominantly places rubrics on the *judgment* side: they serve either as inference-time criteria for benchmark scoring [7, 8] or as training-time reward signals [9, 10]. We instead place rubrics on the model-input side and train the model to apply them as an **executor**: it parses every rule, accounts for priority, and produces both rule-level judgments and an overall verdict. Teaching this executor-side behavior requires supervision that explicitly factorizes an instruction into typed, prioritized rules and demands a verdict on each, precisely what current data lack.

To operationalize this insight, we propose **PRISM**, a data synthesis framework for teaching MLLMs to **internalize** multi-rule instructions rather than treat them as one-shot prompts. We first design a structured rubric representation that organizes rules into five types (Perceptual, Reasoning, Content, Format, Linguistic) and three priority levels (Critical, Important, Optional). On top of this, our four-stage pipeline comprises (1) persona–task discovery, (2) prefix-guided rule completion, (3) rule quality assessment, and (4) structured response generation. We pair the framework with **PRISM-Eval**, a companion evaluation protocol supporting both Loose and Strict rule-based automatic assessment without relying on external judge models.

With only 10K synthesized samples, PRISM substantially outperforms prior data-synthesis baselines on PRISM-Eval and closes much of the gap to the strongest closed-source frontier, while preserving general-benchmark performance. The same gains transfer to four additional MLLMs covering diverse architectures and scales.

Our contributions are summarized as follows:

- We formulate rubric comprehension as an executor-side task and design a fixed-label evaluation protocol (Loose / Strict accuracy) whose scoring requires no external judge model.
- We propose **PRISM**, a four-stage data synthesis framework that produces typed, priority-aware training signals for rubric comprehension.
- PRISM substantially improves in-domain Strict accuracy across five diverse MLLM backbones while preserving general-benchmark performance.

2. Related Work

Multimodal Data Synthesis. Automated production of multimodal instruction data has largely followed two routes. **Early efforts** [3–5] expand sample complexity and reasoning diversity through evolution, collective Monte-Carlo tree search, or staged "summary–caption–reasoning–conclusion" chain-of-thought. These methods all rely on carefully designed prompt templates or handcrafted rules to steer the generation direction. **OASIS** [6], inspired by Magpie [11] in the text-only domain, feeds an MLLM only an image and the dialogue-template prefix and lets it freely complete instructions and responses. This removes prompt engineering, but leaves the data distribution entirely determined by

the model's own biases. PRISM inherits the simplicity of prefix completion but adds intent-guided prefixes and rule-level quality assessment, steering synthesis toward multi-rule, priority-aware rubric comprehension.

Rubric-Based Methods. Rubrics—structured, multi-dimensional evaluation criteria—have been widely adopted in the LLM community, yet almost exclusively on the **judgment** side. On the evaluation side, IFEval [7], InFoBench [12], and AdvancedIF [8] decompose complex instructions into programmatically verifiable or fine-grained sub-criteria that serve as scoring yardsticks. On the training side, RIFL [8], Rubrics-as-Rewards [9], and Rubric-ARM [10] convert rubrics into multi-dimensional RL rewards, while RuscaRL [13] further uses rubrics as decaying rollout-time scaffolding. Across these works, rubrics remain on the judgment side; PRISM instead studies the comparatively underexplored setting in which rubrics are supplied as model inputs and jointly supervise rule-level verification and priority-aware aggregation.

3. Method

We first formalize the rubric comprehension task and its evaluation protocol (§3.1), then describe our data synthesis pipeline **PRISM** (§3.2), and finally instantiate it into the training and evaluation pools together with their composition (§3.3).

3.1. Problem Formulation

We cast rubric comprehension as a conditional generation task. The input is a pair $(I, \mathcal{R})$, where I is an image and $\mathcal{R} = \{(t_i, p_i, c_i)\}_{i=1}^{N}$ is a rubric of N rules, each rule specified by a type $t_i \in \mathcal{T}$, a priority $p_i \in \{\text{Critical}, \text{Important}, \text{Optional}\}$, and a description c_i. The model produces a structured trace $\tau = (g, \{(j_i, e_i)\}_{i=1}^{N}, y)$: visual grounding g, per-rule verifications (j_i, e_i) with $j_i \in \{\text{Pass}, \text{Fail}\}$, and aggregated conclusion y that respects the priority hierarchy. PRISM learns $p_\theta(\tau \mid I, \mathcal{R})$ via SFT on synthesized $(I, \mathcal{R}, \tau)$ tuples.

Evaluation Protocol. Given a model output $\hat{\tau}$, we measure rubric comprehension at two granularities. Let y and $\hat{y}$ denote the gold and predicted overall verdicts, and let j_i and $\hat{j}_i$ denote the corresponding rule-level labels. For a rubric with N rules,

$$\text{Loose} = \mathbb{1}[\hat{y} = y], \qquad \text{Strict} = \mathbb{1}[\hat{y} = y] \prod_{i=1}^{N} \mathbb{1}[\hat{j}_i = j_i].$$

Strict is our primary metric because it requires the overall verdict and every rule judgment to be correct. "Judge-free" refers specifically to scoring: predictions are matched deterministically against fixed labels, without invoking an inference-time LLM judge; it does not assume that synthesized labels are intrinsically error-free.

3.2. PRISM: Data Synthesis Framework

Given an image I sampled from the Cambrian dataset [14], PRISM synthesizes $(I, \mathcal{R}, \tau)$ tuples through four stages: (1) persona–task discovery, (2) prefix-guided rule completion, (3) rule quality assessment, and (4) structured response generation. We use frozen generators with stage-specific prompts: Seed2.0 Lite [15] for Stages 1, 3, and 4, and Qwen3-VL-30B [2] for Stage 2 rule completion. This separation makes rule generation distinct from downstream filtering and label synthesis. The overall pipeline is shown in Figure 2.

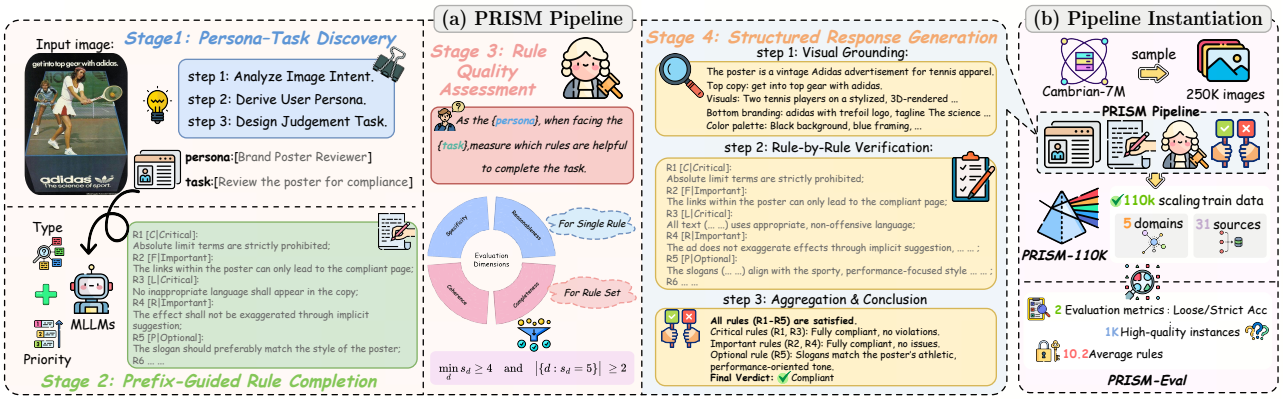


Figure 2: ***Overview of the PRISM data synthesis pipeline.*** *Part* ***(a)*** *shows the four stages of PRISM: (1) Persona–Task Discovery, (2) Prefix-Guided Rule Completion, (3) Rule Quality Assessment, and (4) Structured Response Generation, which produces the supervision trace* $\tau=(g,\{(j_i,e_i)\},y)$*. PRISM is employed to generate* ***PRISM-110K*** *for SFT and* ***PRISM-Eval*** *for evaluation, as shown in part* ***(b)****.*

Stage 1: Persona–Task Discovery. Given image I, we prompt Seed2.0 Lite to first extract its *image intent*—the most plausible real-world use case for I—and then derive an associated *persona* (a role identity) and *task* (a task description). For instance, a geometry diagram may yield the persona "middle-school math test-paper proofreader" and the task "check whether this geometry diagram meets standard exam requirements". This anchoring keeps downstream rubrics grounded in plausible deployment scenarios rather than artificial constructs; the full prompt is reproduced in Figure 9.

Stage 2: Prefix-Guided Rule Completion. Conditioned on the persona and task, we construct a structured prefix that encodes our rule-type taxonomy and prompt Qwen3-VL-30B to complete a rubric of multi-dimensional rules. The type space $\mathcal{T}$ used in §3.1 is instantiated as five categories:

- **P (Perceptual):** *object presence, attributes, counts, spatial layout.*
- **R (Reasoning):** *comparison, arithmetic, deduction over visual or symbolic content.*
- **C (Content):** *topical scope and factual coverage of the answer.*
- **F (Format):** *schema, layout, section ordering.*
- **L (Linguistic):** *style, tone, register, terminology.*

Each rule additionally carries a priority p_i (Critical / Important / Optional) that reflects its relative importance. Rules are produced sequentially, each emitted with a leading `[Type|Priority]` tag that drives the generator to span the taxonomy and avoid homogenization. Compared to unconstrained free-form generation, this prefix-guided mechanism markedly improves rule diversity and structural completeness; the exact prefix template is shown in Box 1.

Prefix Template

```
<|im_start|>user
<|vision_start|><|image_pad|><|vision_end|>
Suppose you are a {persona}, {task}. Please make your judgment based on the
following rubrics.
Note: each rule is tagged [Type|Priority].
Types: P(Perceptual) / R(Reasoning) / C(Content) / F(Format) / L(Linguistic).
Priority: Critical / Important / Optional.
[Rubrics]:<|im_end|>
```

Stage 3: Rule Quality Assessment. To ensure data fidelity, we prompt Seed2.0 Lite to score each rubric on a 1–5 scale along four dimensions:

- **Specificity:** *each rule is unambiguous and admits a clear binary judgment.*
- **Reasonableness:** *each rule has genuine inspection value under the persona and task.*
- **Completeness:** *the rule set covers the dimensions the scenario requires.*
- **Coherence:** *the rules are mutually consistent and contradiction-free.*

A two-tier threshold then routes each rubric: a relaxed threshold admits it into the *training pool*, while a strict threshold admits it into the *evaluation pool* that seeds PRISM-Eval (§3.3). Quality scoring uses the frozen Seed2.0 Lite model. For each $(I, \mathcal{R})$ it emits one JSON object with a 1–5 score and a short rationale per dimension. We parse the JSON and discard any rubric whose scoring response is malformed or incomplete, so a candidate enters the funnel below only if all four dimension scores are recovered.

Concretely, each rubric receives a score $s_d \in \{1, 2, 3, 4, 5\}$ for $d \in \{\text{Spec.}, \text{Reas.}, \text{Comp.}, \text{Coh.}\}$. The training pool admits a rubric iff

$$\min_d s_d \geq 4 \quad \text{and} \quad \big|\{d : s_d = 5\}\big| \geq 2,$$

while a strict threshold ($s_d = 5$ for all d) selects a candidate evaluation pool. We deduplicate this candidate pool against the training pool at both the image-source and rule-pattern level; the surviving rubrics form PRISM-Eval and are excluded from the training pool. This in-process gating shifts quality control from post-hoc curation to synthesis time, yielding a cleanly stratified rubric pool ready for downstream use.

Stage 4: Structured Response Generation. For each $(I, \mathcal{R})$ admitted to the training pool, we synthesize the supervision trace τ defined in §3.1 by prompting Seed2.0 Lite to emit a structured response with three explicit fields:

1. **Visual Grounding (g):** A holistic description of I that grounds the elements relevant to $\mathcal{R}$.
2. **Rule-by-Rule Verification ($\{(j_i, e_i)\}$):** For each rule in `[Type|Priority]` order, the model analyzes the corresponding evidence in I and emits a Pass/Fail judgment j_i with a justification e_i.
3. **Aggregation & Conclusion (y):** A final verdict synthesized from per-rule judgments under priority precedence.

The target overall verdict follows a fixed priority-aware policy. Any failed Critical rule is an immediate veto. If all Critical rules pass, Important and Optional rules receive weights 3 and 1, respectively:

$$s = \frac{3N_{\text{Important,Pass}} + N_{\text{Optional,Pass}}}{3N_{\text{Important}} + N_{\text{Optional}}}, \qquad y = \begin{cases} \text{Fail}, & \exists\, i :\ p_i = \text{Critical} \ \wedge\ j_i = \text{Fail}, \\ \text{Fail}, & s < 0.5, \\ \text{Pass}, & \text{otherwise}. \end{cases}$$

When no non-Critical rules are present, passing all Critical rules yields Pass. The formula is used to construct the supervision target but is intentionally not supplied at joint inference time: learning the mapping from prioritized rule judgments to the final verdict is part of the behavior PRISM is designed to internalize.

Rubric-based RL methods treat rubrics as a sparse reward signal over free-form outputs [9, 10, 13].

We instead use rubrics as a *structural template* for the learner's own reasoning trace, supplying fine-grained per-rule supervision. Type tags teach the learner to differentiate perceptual, inferential, content, format, and linguistic constraints. Priority annotations induce a verification ordering that prioritizes high-stakes rules at inference. The full output-format prompt is reproduced in Figure 11.

3.3. Pipeline Instantiation

Applying the pipeline in §3.2 to ∼250K Cambrian images—each treated as an independent sample yielding at most one rubric—produces ∼150K candidate rubrics, of which ∼110K pass Stage 3 quality filtering (a 72.6% admission rate). These rubrics form two complementary pools that share the same generation procedure and structural schema but serve complementary roles.

PRISM-110K. PRISM-110K is the full supervision pool for fine-tuning, providing per-rule, type- and priority-aware training signals at scale. All downstream training subsets in §4—both the 10K used in main experiments and the 10K–80K sweep used in the scaling study—are sampled uniformly at random from PRISM-110K without any rebalancing across types or priorities, so any observed performance bias reflects what the synthesis pipeline naturally produces rather than post-hoc curation.

PRISM-Eval. PRISM-Eval is a 1,000-sample evaluation set used exclusively for assessment. It retains only the highest-quality rubrics (5/5 on all four dimensions) and is deduplicated against the training pool, so no evaluation rubric overlaps with training data. Scoring requires no inference-time judge: predictions are compared deterministically with fixed rule-level and overall labels under the protocol in §3.1.

Yield statistics. Stage 2 produces approximately 150K candidate rubrics; the remaining images are dropped because Stage 2 returns too few rules to constitute a usable rubric. Stage 3 then applies the two-tier quality filtering above, admitting roughly 72.6% of the candidates. The 1,000 rubrics that meet the strict threshold (s_d=5 for all d), after deduplication, form **PRISM-Eval**; the rubrics that meet only the relaxed threshold form **PRISM-110K**, our full supervision pool, from which all training subsets in §4 (e.g. the 10K used in main experiments and the 10K–80K used in the scaling study) are randomly sampled. PRISM-110K and PRISM-Eval are disjoint and share only the generation procedure, so there is no train–test overlap. The relaxed-vs-strict gap acts as a quality buffer: relaxed admission keeps the supervision pool abundant, while the strict cut reserves the cleanest rubrics for evaluation. Table 1 summarizes this generation funnel.

Table 1: *PRISM data-generation funnel. Each Cambrian image is one independent sample yielding at most one rubric.*

Stage	Output	Count
Source images	Cambrian images	∼250 K
Stage 1–2	Candidate rubrics	∼150 K
Stage 3 (relaxed)	PRISM-110K (train)	∼110 K
Stage 3 (strict)	PRISM-Eval (eval)	1 K

Independent human validation. We validate a stratified 100-example subset of PRISM-Eval (967 rule labels). Two annotators relabel every rule and overall verdict blind to original gold labels; a third adjudicates disagreements. As Figure 3 shows, independent agreement reaches 91.31% at rule level (Cohen's κ=0.795) and 91.00% for overall verdicts (κ=0.819). Original labels agree with adjudicated human labels on 98.55% of rules and 92.00% of verdicts. Critical rules are especially stable: annotators agree on 99.27%, and 409 of 410 original Critical labels are retained. The estimated correction rate is 1.45% (95% CI: 0.73–2.27%).

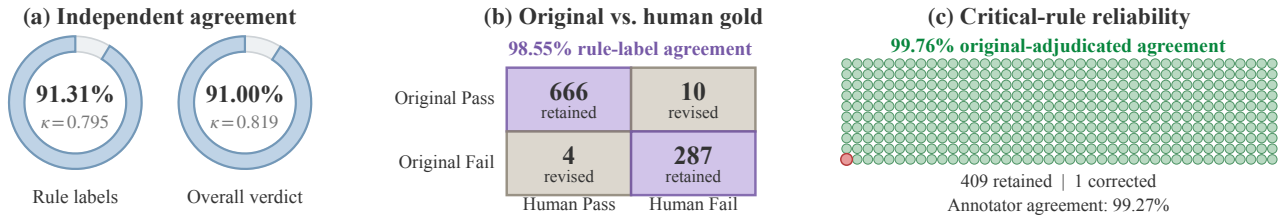
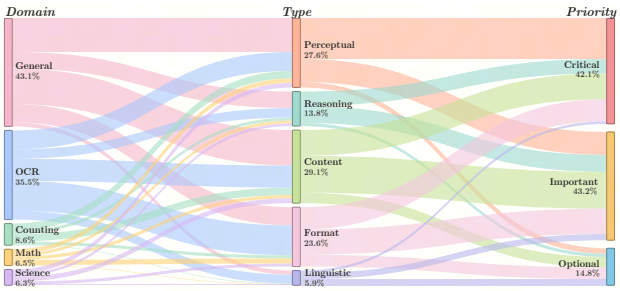


Figure 3: ***Independent human validation of PRISM-Eval.*** ***(a)*** *Agreement between two independent annotators.* ***(b)*** *Original and adjudicated human rule labels agree on 98.55% of 967 rules.* ***(c)*** *Critical rules show particularly high reliability: 409 of 410 original labels are retained.*

Using the adjudicated labels as an alternative gold view yields the same conclusion as the full benchmark. On Qwen3-VL-4B, PRISM raises per-rule accuracy from 67.32% to 82.11%, Loose from 36.30% to 70.90%, and Strict from 8.70% to 30.00%. On Qwen3.5-9B, the corresponding scores rise from 70.37% to 84.84%, 49.70% to 69.70%, and 16.80% to 33.40%. Thus, the improvement persists when evaluation is anchored to independently adjudicated human labels.

(a) Structural coverage



(b) Semantic diversity

Unit	N	Exact	Max similarity	
			$\geq .95$	$\geq .98$
Task	10K	0.00	1.59	≈0.00
Rule	100,561	0.66	2.59	0.40
Context. rule	100,561	0.00	1.79	0.23

Figure 4: ***Diversity of PRISM-110K.*** ***(a)*** *The Sankey diagram connects image domain, rule type, and priority.* ***(b)*** *The adjacent table reports exact-duplicate and semantic near-neighbor rates (%); contextualized rules include their persona–task context.*

Diversity & composition. Figure 4 examines diversity from complementary structural and semantic views. The pipeline covers all five rule types and three priority levels across General, OCR, Counting, Math, and Science images. Semantically, all 10,000 sampled persona–task pairs are unique; exact duplication is 0.66% for standalone rules and 0% after incorporating persona–task context. Only 2.59% of rules have a cross-sample neighbor at cosine similarity at least 0.95, falling to 0.40% at 0.98. Together, these results indicate that the synthesis process does not collapse to a small collection of repeated templates. Figure 5 further summarizes the domain and source-dataset composition of PRISM-110K.

Engineering practices and efficiency. All four stages are stateless, per-image calls to frozen endpoints: Qwen3-VL-30B for Stage 2 and Seed2.0 Lite for Stages 1, 3, and 4. Because each image is processed independently, the pipeline is embarrassingly parallel: throughput scales near-linearly with the number of concurrent requests, and no cross-sample state or synchronization is required. With a request concurrency of 40, generating the full corpus over ∼250K Cambrian images takes roughly 18 hours end-to-end, including all four stages. Malformed or under-length responses are dropped rather than retried (the dominant Stage 2 failure mode is returning too few rules; see Table 1), which keeps the pipeline simple and avoids long-tail stalls. The same procedure can be applied to new image pools without retraining the synthesis models, although independent validation remains necessary when the source distribution changes.

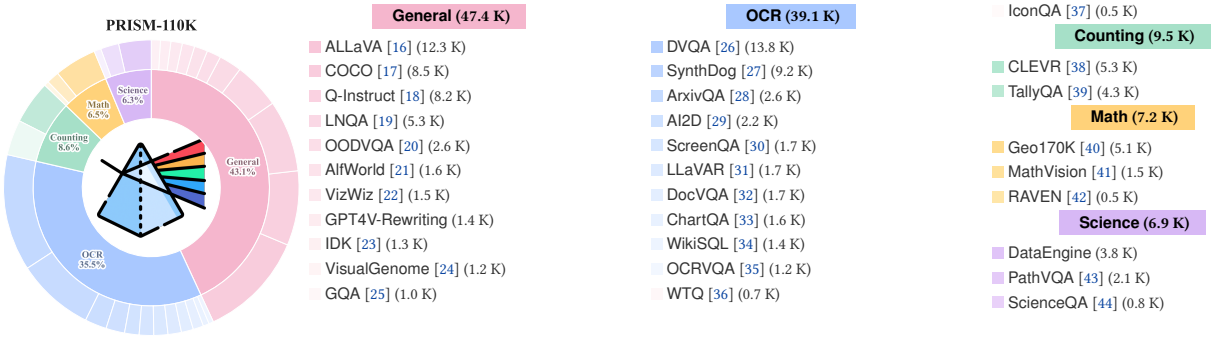


Figure 5: ***PRISM-110K data composition. Left:*** *Domain-level distribution of PRISM-110K. The inner ring follows the verified domain proportions (General 43.1%, OCR 35.5%, Counting 8.6%, Math 6.5%, Science 6.3%), while the outer ring preserves the corresponding child-dataset proportions within each domain.* ***Right:*** *All the data sources in the PRISM-110K as well as the ones filtered in data curation.*

4. Experiments

4.1. Experimental Setup

Base Models. To assess whether PRISM's gains transfer across learner architectures, we run experiments on five MLLMs spanning the Qwen and InternVL families and a wide parameter range: Qwen3-VL-4B [2], InternVL3.5-14B [45], and three Qwen3.5 variants—Qwen3.5-9B, Qwen3.5-27B, and Qwen3.5-35B-A3B [46]—covering both dense and mixture-of-experts architectures from 4B to 35B.

Training Recipe. All experiments perform full-parameter SFT on top of LLaVA-NeXT-100K [47], into which the per-method incremental data is mixed. To ensure a fair comparison, we adopt a single training recipe across every backbone and every comparison group: AdamW, peak learning rate 1×10^{-5}, cosine decay with 0.05 warmup ratio, global batch size 128, 1 training epoch, and a maximum sequence length of 8192. To improve compute efficiency on heterogeneous-length samples, we enable *sequence packing*, which concatenates short samples up to the length budget so that no compute is wasted on padding tokens. We use full-parameter SFT throughout (no LoRA / no parameter freezing); the vision encoder is updated jointly with the language backbone.

Evaluation. We evaluate **in-domain** on **PRISM-Eval** and **out-of-domain** on 14 public benchmarks grouped into *Perception*, *Document*, and *Reasoning* suites, whose unweighted average forms **Gen. Avg**:

- **Perception:** MMBench_EN [48], MME [49], MMStar [50], MMMU [51], AI2D [29], covering broad visual perception, world knowledge, and diagram understanding.
- **Document:** OCRBench [52], InfoVQA [53], CharXiv descriptive and CharXiv reasoning [54], targeting OCR, text-rich images, and chart/document comprehension.
- **Reasoning:** DynaMath [55], MathVision_MINI [41], LogicVista [56], VisuLogic [57], CountBenchQA [58], probing multimodal mathematical, logical, and counting reasoning.

The three suites are intentionally chosen to be *disjoint* from rubric-style supervision, so that a non-trivial portion of each benchmark probes capabilities that PRISM never explicitly trains for, providing a single scalar that captures whether rubric-internalization training comes at the cost of general MLLM ability.

PRISM-Eval statistics and parsing. PRISM-Eval is nearly balanced at the sample level (505 Pass / 495 Fail overall verdicts) and contains 10,253 rule judgments (7,507 Pass / 2,746 Fail). An all-Pass predictor therefore obtains 50.5% Loose but only 13.7% Strict, illustrating why complete rule-level correctness is substantially more demanding than majority prediction. Our deterministic parser accepts semantically equivalent JSON object and list serializations, extracts rule judgments and the overall verdict separately, and counts any missing or invalid required label as incorrect. On Qwen3-VL-4B, parse failures occur for 27/1,000 Baseline outputs and 9/1,000 PRISM outputs; across the full leaderboard, the fraction ranges from 0.2% for the best-formatted closed-source outputs to 7.0% for LLaVA-family models.

4.2. Leaderboard of PRISM-Eval

Before introducing PRISM training, we use PRISM-Eval to characterize the current state of rubric comprehension across representative closed- and open-source MLLMs. As reported in Table 2, we ask whether parameter scale alone can close the gap to closed-source frontiers.

Two findings stand out from Table 2. First, even GPT-5.4 reaches only 42.1% Strict, and no open-source model crosses the 30% Strict line. Second, larger backbones within the evaluated model families provide only modest Strict gains relative to the remaining gap. Thus, under the present protocol, model scale alone does not recover reliable per-rule, priority-aware judgment. These observations motivate a training signal that explicitly factorizes complex instructions into typed, prioritized rules and demands a verdict on each.

Table 2: ***Evaluation of various MLLMs on PRISM-Eval.*** *Loose/Strict accuracy (%); section best in* ***bold****.*

Model	Param	Loose	Strict
Closed-Source MLLMs			
GPT-5.4 [1]	-	**82.7**	**42.1**
Gemini-3-Flash [59]	-	80.5	32.0
Seed2.0 Lite [15]	-	77.8	38.3
Open-Source MLLMs			
Qwen3-VL [2]	2B	57.4	16.5
	4B	63.5	20.6
	8B	67.1	23.5
Qwen3.5 [46]	2B	57.0	15.9
	4B	59.0	21.5
	9B	63.6	25.1
	27B	63.9	**29.3**
	35B-A3B	61.8	25.1
InternVL3.5 [45]	2B	61.8	14.4
	4B	64.0	15.9
	8B	62.8	16.8
	14B	63.6	18.3
	30B-A3B	**67.8**	18.9
LLaVA-1.5 [60]	7B	51.6	14.0
LLaVA-NeXT [47]	8B	52.3	14.3

4.3. Main Results

Comparison with Data Synthesis Baselines. We first ask whether existing data-synthesis recipes can already impart rubric comprehension. The upper block of Table 3 compares PRISM with five representative baselines on Qwen3-VL-4B, all sharing the same recipe and incremental data volume atop LLaVA-NeXT-100K: **LLaVA-CoT** [5] (visual instruction data emphasizing reasoning chains), **MMEvol** [3] (iteratively evolved instruction data), **Mulberry** [4] (collective reasoning data emphasizing path diversity), **Cambrian** [14] (curated visual QA), and **OASIS** [6] (prefix completion without prompt templates). All five cluster tightly around the no-incremental-data baseline on PRISM-Eval. Increasing the volume of generic visual instruction data, however diversified, does not by itself induce per-rule, priority-aware judgment. PRISM departs from this cluster because it injects *structural* supervision (typed rules, priority, and per-rule verification) rather than additional samples. Crucially, this in-domain gain leaves Gen. Avg essentially unchanged, indicating no measurable degradation on general benchmarks. Finer-grained breakdowns by priority, rule type, and rubric size are provided in §4.5, while controlled format variants in §4.7 test whether the gain persists under changes to the rubric representation.

Table 3: *Main results. The upper block compares data synthesis methods on Qwen3-VL-4B (each adds 10K on top of LLaVA-NeXT-100K); green deltas on the PRISM row are gains over Baseline. The lower block evaluates transfer of PRISM supervision across learner backbones (each row uses 10K samples from PRISM-110K). Deltas are reported on Gen. Avg and the in-domain PRISM-Eval metrics.*

Model / Method	General Avg	PRISM-Eval		Perception					Document			Reasoning				
		Loose	Strict	MMB	MME	MMS	MMMU	AI2D	OCR	Info	ChXiv$_{d/r}$	Dyn	MV	Logic	Visu	Count
Comparison with data synthesis methods (Qwen3-VL-4B, LLaVA-NeXT-100K +10K synthesized samples)																
Baseline	60.30	36.7	9.5	84.7	83.0	62.7	53.7	84.4	79.5	69.8	70.6/38.6	41.4	23.0	39.8	24.3	88.7
LLaVA-CoT	60.68	53.0	14.7	84.0	83.0	62.1	55.0	84.8	80.9	70.4	71.9/39.0	42.2	22.4	38.0	25.5	90.3
MMEvol	60.66	51.4	14.2	83.6	83.6	62.2	54.9	84.0	81.3	69.9	73.2/38.0	41.2	22.0	39.6	26.7	89.1
Mulberry	60.69	54.7	14.8	83.4	83.6	63.0	55.1	84.4	81.4	69.6	71.4/37.6	41.4	22.0	39.1	27.4	90.3
Cambrian	60.34	52.9	14.9	83.5	83.1	61.7	52.3	83.5	82.3	70.5	73.6/37.9	41.7	24.7	37.8	23.9	88.9
OASIS	60.57	54.5	15.0	84.5	83.5	63.4	54.0	84.4	80.5	69.3	71.2/38.6	41.1	22.4	38.3	25.6	90.1
PRISM (Ours)	60.87 +0.6	71.1 +34.4	30.1 +20.6	85.0	83.0	62.5	53.8	83.9	80.5	68.8	72.6/39.7	41.6	24.7	39.4	25.9	90.8
Transfer across learner backbones (LLaVA-NeXT-100K +10K synthesized samples)																
InternVL3.5-14B	65.10	65.8	19.7	84.9	86.3	67.2	61.9	85.6	82.4	77.9	80.5/43.6	49.7	32.2	46.8	27.1	85.4
w. PRISM	65.56 +0.5	66.3 +0.5	28.8 +9.1	85.2	85.5	67.9	61.8	85.8	82.4	78.8	81.2/44.1	49.4	34.5	49.9	27.5	83.8
Qwen3.5-9B	66.63	49.5	16.6	87.5	87.4	68.1	57.8	89.1	87.5	73.8	79.0/57.4	61.2	27.3	43.8	24.8	88.1
w. PRISM	68.17 +1.5	69.6 +20.1	33.2 +16.6	87.4	89.0	69.9	60.3	89.1	87.2	74.0	80.0/57.2	71.8	27.6	44.7	27.5	88.7
Qwen3.5-35B-A3B	68.83	62.8	23.9	88.6	88.6	69.3	62.6	90.8	87.8	78.3	81.4/59.6	68.3	27.6	43.8	27.4	89.5
w. PRISM	70.45 +1.6	68.9 +6.1	36.2 +12.3	88.7	89.1	71.1	64.3	90.4	88.4	79.4	83.6/60.8	77.6	29.6	44.7	28.5	90.1
Qwen3.5-27B	71.61	63.6	27.8	88.9	89.1	73.7	68.2	91.2	87.5	80.1	83.0/61.1	72.2	36.5	52.6	26.8	91.6
w. PRISM	72.52 +0.9	72.3 +8.7	38.4 +10.6	89.8	91.1	72.5	66.6	91.0	88.4	80.1	85.9/63.2	79.9	32.6	52.8	27.6	93.8

Transfer across Learner Backbones. The lower block tests whether this effect is backbone-specific. Adding the same 10K synthesized samples to four MLLMs spanning 9B–35B parameters in both dense and MoE configurations raises Strict accuracy on *every* backbone, while leaving Gen. Avg unchanged or slightly higher. Since PRISM contributes only a tiny fraction of the total training tokens, this directional consistency suggests that rubric comprehension occupies a capability axis largely orthogonal to what these backbones absorb during pre-training—structured supervision can be slotted in without disturbing existing skills. Notably, the strongest configuration rises well above the best open-source result in Table 2 and narrows the gap to the closed-source models with only 10K synthesized samples, suggesting that the binding constraint is the *kind* of supervision, not the parameter count.

4.4. What Makes PRISM Work?

We isolate each design choice in PRISM along three axes—*data* (what enters the model), *procedure* (how the model reasons), and *rubric structure* (what the labels carry). All variants are trained with the same recipe on Qwen3-VL-4B with 10K synthesized samples; results are summarized in Table 4.

Data vs. Procedure. Removing **quality filtering** barely affects Loose accuracy but lowers Strict and Gen. Avg, consistent with *data-purity* interference: low-quality rubrics rarely flip the overall verdict (so Loose is robust), but they inject ambiguous or self-contradictory rules that degrade per-rule judgment. Removing **structured CoT** shows the opposite asymmetry: Loose and Gen. Avg fall the most, while Strict drops by the same margin as the filtering ablation. This is consistent with structured CoT acting as a *procedural prior* (Visual Grounding → Rule-by-Rule → Aggregation) that keeps reasoning aligned with the task structure across both rubric and non-rubric inputs; without it, the model still recognizes individual rules at strict granularity but loses the holistic verification habit that general-benchmark transfer relies on. The two effects are complementary rather than redundant—filtering safeguards what enters the model, structured CoT shapes how the model reasons over what it has learned.

Table 4: *Component ablations on Qwen3-VL-4B (10K synthesized samples). Red deltas on General Avg and PRISM-Eval mark drops relative to PRISM (Full); the Shuffled-rules row controls for surface ordering rather than removing a component.*

Configuration	General Avg	PRISM-Eval		Perception					Document			Reasoning				
		Loose	Strict	MMB	MME	MMS	MMMU	AI2D	OCR	Info	ChXiv$_{d/r}$	Dyn	MV	Logic	Visu	Count
PRISM (Full)	**60.87**	**71.1**	**30.1**	85.0	83.0	62.5	53.8	83.9	80.5	68.8	72.6/39.7	41.6	24.7	39.4	25.9	90.8
Data side: what enters the model.																
w/o Quality Filtering	60.49 -0.38	71.0 -0.1	29.4 -0.7	84.5	82.5	62.7	53.3	83.8	81.8	69.2	71.9/38.1	40.9	24.0	39.3	25.4	89.4
Procedure side: how the model reasons.																
w/o Structured CoT	60.10 -0.77	69.2 -1.9	29.4 -0.7	84.5	82.4	62.5	52.7	84.2	79.9	69.7	71.1/38.5	40.9	21.7	39.2	24.8	89.3
Rubric structure: semantic priority vs. surface order.																
w/o Priority	60.45 -0.42	66.8 -4.3	23.6 -6.5	85.3	85.8	62.4	53.9	84.2	81.5	68.0	71.2/39.1	41.0	21.4	40.3	25.1	89.9
Shuffled rules	60.84 -0.03	70.5 -0.6	30.3 +0.2	84.7	84.6	62.1	54.1	84.4	81.3	68.7	72.2/38.5	41.9	22.7	41.6	26.3	91.0

Priority vs. Order. The remaining two rows probe what the rubric structure itself contributes. Stripping the **Priority** field while keeping types and the per-rule trace causes the largest single-component drop in our entire ablation: Strict falls by -6.5 pp and Loose by -4.3 pp, far exceeding either of the data- or procedure-side ablations. Without priority annotations, the model treats every rule as equally decisive and loses the Critical-first verification ordering induced at Stage 4 (§3.2); the resulting drop is precisely the gain we will dissect by Priority bucket in §4.5. In contrast, when we retrain on rubrics whose rules have been **randomly shuffled** within each sample (types and priorities preserved), performance is statistically indistinguishable from PRISM Full (30.3 vs. 30.1 Strict, 70.5 vs. 71.1 Loose). The two rows draw a clean line: it is the *semantic* structure of priority labels that PRISM depends on, not the *surface* order in which rules happen to be listed.

Aggregation Internalization. The aggregation formula is deliberately withheld at inference time because mapping prioritized rule judgments to the final verdict is part of the capability that PRISM is intended to learn. To isolate this component, we perform an oracle-policy diagnostic: we retain each model's predicted rule labels, discard its generated overall verdict, and recompute that verdict using the fixed policy in §3.2. This diagnostic gives the policy to an external program and is therefore not an equal-condition baseline. As Table 5 shows, programmatic re-aggregation raises the Baseline's Loose accuracy from 36.7% to 47.5% ($+10.8$ pp), but changes PRISM only from 71.1% to 73.6% ($+2.5$ pp). When every predicted rule label is correct, PRISM's own overall verdict is correct in 302/344 cases (87.8%), compared with 95/143 (66.4%) for the Baseline. The smaller oracle-policy gain and higher conditional consistency indicate that PRISM has largely internalized the priority-aware aggregation behavior.

Table 5: ***Aggregation diagnostic on Qwen3-VL-4B*** *(%). Generated/re-aggregated verdict accuracy and generated-verdict accuracy when all rule predictions are correct.*

Model	Generated / Re-aggregated	All rules correct
Baseline	36.7 →47.5 +10.8	66.4
PRISM	**71.1** →**73.6** +2.5	**87.8**

4.5. Where Do the Gains Come From?

The headline results in Table 3 report a single Strict number per model, leaving open *where* the gains come from. We dissect PRISM-Eval errors along three orthogonal axes—rubric *Priority*, rule *Type*, and rubric *Size* (number of rules). Because the same evaluation set carries information at three different granularities, we report per-rule accuracy for Priority and Type, which operate on rule instances, and Strict accuracy for Size, which operates on complete samples.

We compare the Qwen3-VL-4B **Baseline** (LLaVA-NeXT-100K only), our **PRISM**-trained variant (Baseline + 10K PRISM samples; the same checkpoint reported in Table 3), and the strongest closed-source model on the leaderboard, **GPT-5.4**.

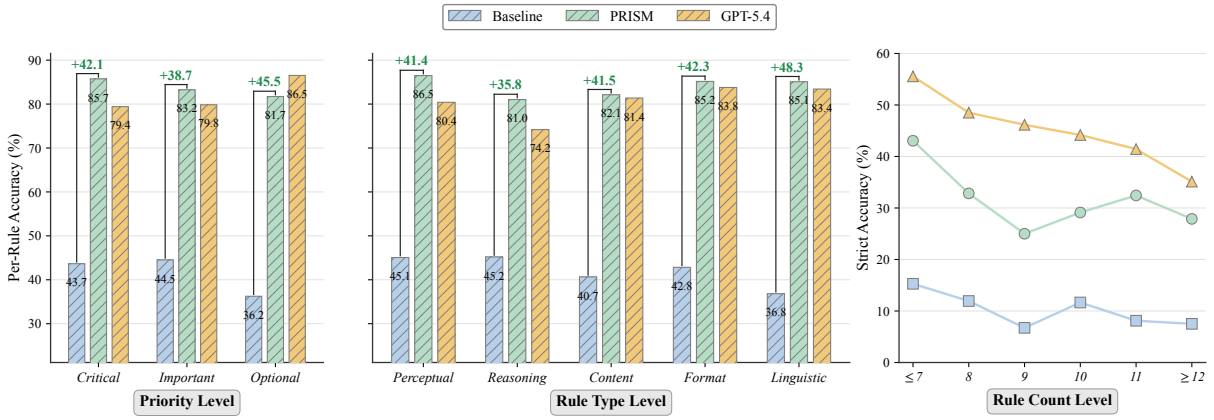


Figure 6: ***Error breakdown on PRISM-Eval. Baseline*** *denotes Qwen3-VL-4B fine-tuned on LLaVA-NeXT-100K, and* ***PRISM*** *denotes the same backbone fine-tuned on LLaVA-NeXT-100K + 10K PRISM samples;* ***GPT-5.4*** *is the strongest closed-source reference. We report per-rule accuracy stratified by* Priority Level *(left), per-rule accuracy stratified by* Rule Type Level *(middle), and per-sample Strict accuracy stratified by* Rule Count Level *(right).*

By Priority. PRISM lifts per-rule accuracy across all three priority buckets: from 43.7% to 85.7% on Critical, 44.5% to 83.2% on Important, and 36.2% to 81.7% on Optional. The resulting gains of +42.1, +38.7, and +45.5 pp show that the improvement is broad rather than confined to one priority level. Critical rules nevertheless attain the highest absolute accuracy, which matters because any Critical failure can determine the overall verdict. Together with the −6.5 pp Strict drop in the **w/o Priority** ablation, these results support the value of explicitly encoding priority semantics.

By Rule Type. The Baseline's weakest axes are *Reasoning* and *Content*—both require composing visual evidence into a fact-grounded judgment beyond mere perception. PRISM substantially reduces per-rule error on both, and also improves Format and Linguistic rules. This supports the role of the typed taxonomy in §3.2: explicit `[Type]` tags push the model to differentiate Perceptual ("check what is in the image"), Reasoning ("infer over it"), and Content ("check topic coverage"), rather than collapsing them into a single answering mode. GPT-5.4 also lags PRISM on Reasoning in this evaluation, suggesting the axis remains challenging and that typed supervision directly targets it.

By Rule Count. Strict accuracy declines mechanically with rubric size, since every rule must be judged correctly. What matters is the shape of the gap: PRISM beats the Baseline by a wide margin on small rubrics (≤ 7), and the lead persists at ≥ 12, where the Baseline approaches zero. The Strict gain in Table 3 is therefore not an artifact of short rubrics; the advantage remains visible as the number of jointly evaluated rules increases.

4.6. How Far Does Structured Supervision Scale?

We investigate the effect of training-set size (from 10K to 80K synthesized samples) on Qwen3-VL-4B performance. The two curves diverge in shape. **Gen. Avg follows a shallow inverted-U**, peaking at a moderate scale before retreating, whereas **in-domain Strict increases monotonically**. This pattern is consistent with the general-benchmark benefit saturating after the structured "ground–verify–aggregate" procedure is learned, while additional

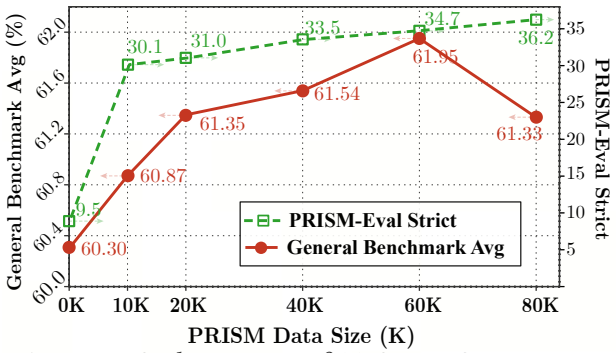


Figure 7: *Scaling curves of PRISM on Qwen3-VL-4B.*

rubric data continues to improve fidelity on the target task. In practice, a moderate pool gives the best Gen. Avg trade-off, whereas larger pools favor in-domain Strict accuracy.

4.7. How Robust Is PRISM to Equivalent Representations?

We construct three controlled PRISM-Eval variants, changing exactly one factor at a time while preserving images, rule identities, priorities, and gold judgments: **Paraphrase** rewrites each rule with equivalent wording; **Natural-language tags** replaces compact `[Type|Priority]` markers with explicit natural-language descriptions; and **Output format** replaces the required JSON serialization with a line-based rule list. Table 6 compares each PRISM-trained model with its corresponding Baseline. PRISM retains a positive Strict gain in every condition on both 4B and 9B. The gains are particularly strong under rule paraphrasing and natural-language tags, showing that the improvement is not tied to fixed rule wording or compact tag tokens. Transfer to a new output schema is positive but more modest and model-dependent, separating input-side rubric comprehension from output-serialization difficulty.

Table 6: ***Strict accuracy under controlled representation changes*** *(%). Base → PRISM.*

Variant	4B	9B
Original	9.5 →**30.1** +20.6	16.6 →**33.2** +16.6
Paraphrase	8.4 →**30.2** +21.8	4.2 →**35.5** +31.3
Nat.-lang. tags	3.0 →**30.4** +27.4	5.2 →**29.0** +23.8
Output format	10.0 →**17.2** +7.2	20.7 →**23.7** +3.0

4.8. Can Inference-Time Strategies Replace Training?

We evaluate two training-free alternatives on the same 4B and 9B Baseline checkpoints. **2-shot trace prompting** prepends one Pass and one Fail training exemplar, each demonstrating visual grounding, rule-wise verification, and aggregation. The stronger **Sequential** strategy queries every rule independently and applies the explicit priority-aware policy programmatically. As Figure 8 shows, both approaches improve over joint zero-shot inference, confirming that demonstrations and decomposition are useful. Nevertheless, PRISM remains substantially stronger in Strict accuracy: 30.1% vs. 21.0% for Sequential on 4B, and 33.2% vs. 24.1% on 9B. Sequential also requires one call per rule—10.253 calls per sample on average—plus external aggregation, whereas PRISM jointly predicts all rule judgments and the overall verdict in one call without receiving the aggregation formula at inference time. Thus, inference-time structure recovers part of the capability, but does not match the accuracy or efficiency of internalizing the procedure through training.

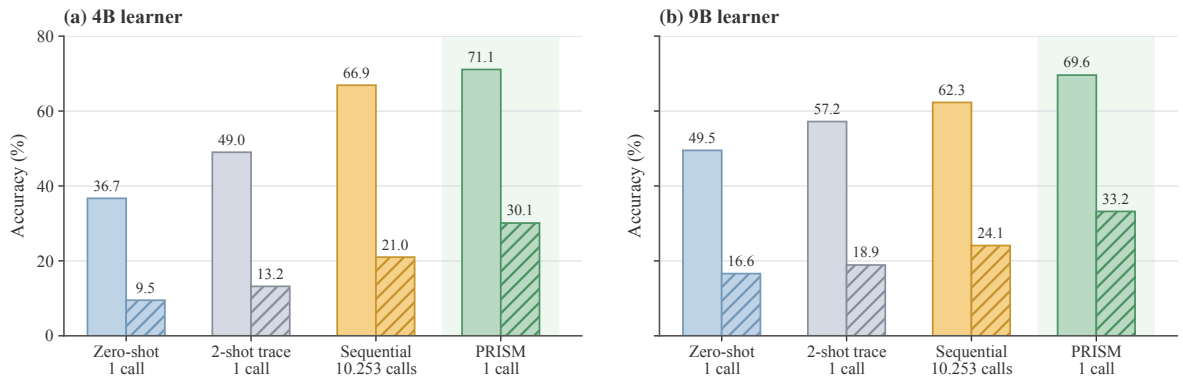


Figure 8: ***Training-free inference strategies on PRISM-Eval.*** *Two-shot trace prompting and Sequential rule-wise querying recover part of the capability, but PRISM achieves the highest Loose and Strict accuracy with a single joint call. Hatching denotes Strict accuracy.*

5. Conclusion

We formulate **rubric comprehension** as an executor-side task that requires a model to verify multiple prioritized rules and produce an overall judgment within a single multimodal instruction. We introduce **PRISM**, a four-stage data synthesis framework that generates typed, priority-aware rubrics and structured verification traces, yielding the training pool **PRISM-110K** and the instance-disjoint evaluation set **PRISM-Eval**. Across five MLLM backbones spanning dense and MoE architectures from 4B to 35B, PRISM improves in-domain Strict accuracy by 9.1–20.6 percentage points without measurable degradation on the evaluated general benchmarks. Independent human adjudication supports the reliability of the fixed labels, controlled representation variants demonstrate robustness beyond the original rule wording and tags, and training-free baselines recover only part of the gain at greater inference cost. Together, these results establish structured rubric supervision as an effective and scalable approach to priority-aware multimodal verification.

Limitations

PRISM currently relies on supervised fine-tuning, leaving reinforcement learning or preference optimization with rule-level feedback unexplored. Our experiments focus on single-image, static verification; extending structured rubric supervision to video and multi-turn settings will require temporal and interaction-aware rubric formulations. Finally, exact all-rule correctness remains challenging as rubric size increases, motivating stronger visual reasoning and more reliable tracking of multiple constraints.

References

[1] OpenAI. Introducing GPT-5.4. https://openai.com/index/introducing-gpt-5-4/, 2025. Blog post.

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

[3] Run Luo, Haonan Zhang, Longze Chen, Ting-En Lin, Xiong Liu, Yuchuan Wu, Min Yang, Yongbin Li, Minzheng Wang, Pengpeng Zeng, Lianli Gao, Heng Tao Shen, Yunshui Li, Xiaobo Xia, Fei Huang, and Jingkuan Song. MMEvol: Empowering multimodal large language models with evol-instruct. *arXiv preprint arXiv:2409.05840*, 2024. URL https://arxiv.org/abs/2409.05840.

[4] Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Feng, Yibo Wan, Shunyu Xiao, Shuai Liu, Yijie Guo, Weihao Zhang, et al. Mulberry: Empowering MLLM with o1-like reasoning and reflection via collective monte carlo tree search. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Spotlight.

[5] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. LLaVA-CoT: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2024. doi: 10.48550/arXiv.2411.10440. URL https://arxiv.org/abs/2411.10440.

[6] Letian Zhang, Quan Cui, Bingchen Zhao, and Cheng Yang. Oasis: One image is all you need for

multimodal instruction data synthesis. *arXiv preprint arXiv:2503.08741*, 2025. URL https://arxiv.org/abs/2503.08741.

[7] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

[8] Yun He, Wenzhe Li, Hejia Zhang, Songlin Li, Karishma Mandyam, Sopan Khosla, Yuanhao Xiong, Nanshu Wang, Selina Peng, Beibin Li, Shengjie Bi, Shishir G. Patil, Qi Qi, Shengyu Feng, Julian Katz-Samuels, Richard Yuanzhe Pang, Sujan Gonugondla, Hunter Lang, Yue Yu, Yundi Qian, Maryam Fazel-Zarandi, Licheng Yu, Amine Benhalloum, Hany Awadalla, and Manaal Faruqui. Rubric-based benchmarking and reinforcement learning for advancing LLM instruction following. *arXiv preprint arXiv:2511.10507*, 2025. URL https://arxiv.org/abs/2511.10507.

[9] Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025. doi: 10.48550/arXiv.2507.17746. URL https://arxiv.org/abs/2507.17746.

[10] Ran Xu, Tianci Liu, Zihan Dong, Tony Yu, Ilgee Hong, Carl Yang, Linjun Zhang, Tuo Zhao, and Haoyu Wang. Alternating reinforcement learning for rubric-based reward modeling in non-verifiable LLM post-training. *arXiv preprint arXiv:2602.01511*, 2026. URL https://arxiv.org/abs/2602.01511.

[11] Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned LLMs with nothing. *arXiv preprint arXiv:2406.08464*, 2024. URL https://arxiv.org/abs/2406.08464.

[12] Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. InFoBench: Evaluating instruction following ability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13025–13048, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.772. URL https://aclanthology.org/2024.findings-acl.772/.

[13] Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Kongcheng Zhang, Jiale Zhao, Jingwen Yang, Yihe Zhou, Jianwei Lv, Tongya Zheng, Hengtong Lu, Wei Chen, Yan Xie, and Mingli Song. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for general LLM reasoning. *arXiv preprint arXiv:2508.16949*, 2025. URL https://arxiv.org/abs/2508.16949.

[14] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024.

[15] ByteDance. Seed 2.0 Model Card. https://lf3-static.bytednsdoc.com/obj/eden-cn/lapzild-tss/ljhwZthlaukjlkulzlp/seed2/0214/Seed2.0%20Model%20Card.pdf, 2026. Model card.

[16] Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4v-synthesized data for a lite vision-language model. *arXiv preprint arXiv:2402.11684*, 2024.

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[18] Haoning Wu, Zicheng Zhang, Erli Zhang, Chaofeng Chen, Liang Liao, Annan Wang, Kaixin Xu, Chunyi Li, Jingwen Hou, Guangtao Zhai, et al. Q-instruct: Improving low-level visual abilities for multi-modality foundation models. *arXiv preprint arXiv:2311.06783*, 2023.

[19] Jordi Pont-Tuset, Jasper R. R. Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *ECCV*, 2020.

[20] Haoqin Tu, Chenhang Cui, Zijun Wang, Yiyang Zhou, Bingchen Zhao, Junlin Han, Wangchunshu Zhou,

Huaxiu Yao, and Cihang Xie. How many unicorns are in this image? a safety evaluation benchmark for vision llms. *arXiv preprint arXiv:2311.16101*, 2023.

[21] Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, et al. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *arXiv preprint arXiv:2405.10292*, 2024.

[22] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *CVPR*, 2018.

[23] Sungguk Cha, Jusung Lee, Younghyun Lee, and Cheoljong Yang. Visually dehallucinative instruction generation: Know what you don't know. *arXiv preprint arXiv:2402.09717*, 2024.

[24] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2016.

[25] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019.

[26] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *CVPR*, 2018.

[27] Geewook Kim, Teakgyu Hong, Moonbin Yim, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Donut: Document understanding transformer without ocr. In *ECCV*, 2022.

[28] Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models. *arXiv preprint arXiv:2403.00231*, 2024.

[29] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pages 235–251. Springer, 2016.

[30] Yu-Chung Hsiao, Fedir Zubach, Maria Wang, et al. Screenqa: Large-scale question-answer pairs over mobile app screenshots. *arXiv preprint arXiv:2209.08199*, 2022.

[31] Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. Llavar: Enhanced visual instruction tuning for text-rich image understanding. *arXiv preprint arXiv:2306.17107*, 2023.

[32] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *WACV*, 2021.

[33] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *ACL*, 2022.

[34] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*, 2017.

[35] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *International Conference on Document Analysis and Recognition*, 2019.

[36] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *ACL*, 2015.

[37] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. In *NeurIPS*, 2021.

[38] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross

Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[39] Manoj Acharya, Kushal Kafle, and Christopher Kanan. Tallyqa: Answering complex counting questions. In *AAAI*, 2019.

[40] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*, 2023.

[41] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.

[42] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. Raven: A dataset for relational and analogical visual reasoning. In *CVPR*, 2019.

[43] Xuehai He, Yichen Zhang, Luntian Mou, Eric P. Xing, and Pengtao Xie. Pathvqa: 30000+ questions for medical visual question answering. *CoRR*, abs/2003.10286, 2020.

[44] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022.

[45] 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*, 2025.

[46] Qwen Team. Qwen3.5. https://qwen.ai/blog?id=qwen3.5, 2026. Blog post.

[47] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024.

[48] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024.

[49] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Advances in Neural Information Processing Systems*, 38, 2026.

[50] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024.

[51] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9556–9567, 2024.

[52] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024.

[53] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.

[54] Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697, 2024.

[55] Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. In *International Conference on Learning Representations*, volume 2025, pages 48337–48383, 2025.

[56] Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*, 2024.

[57] Weiye Xu, Jiahao Wang, Weiyun Wang, Zhe Chen, Wengang Zhou, Aijun Yang, Lewei Lu, Houqiang Li, Xiaohua Wang, Xizhou Zhu, et al. Visulogic: A benchmark for evaluating visual reasoning in multi-modal large language models. *arXiv preprint arXiv:2504.15279*, 2025.

[58] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching CLIP to Count to Ten. *arXiv preprint arXiv:2302.12066*, 2023.

[59] Google. Gemini 3 Flash: Frontier intelligence built for speed. https://blog.google/products/gemini/gemini-3-flash/, 2025. Blog post.

[60] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Prompt Used for Persona–Task Generation

[SYSTEM PROMPT]

You are a multimodal task design expert. Your task is to design a reasonable user persona and a complex rule-based judgment task for a given image.

Core Concept

A complex rule-based judgment task refers to: given an image, requiring a systematic inspection of the image content based on multiple rules, and providing a comprehensive judgment result. Such tasks are widely present in real-world scenarios like content moderation, compliance checking, and quality assessment.

Generation Strategy

Please reason step-by-step following this chain of thought:

Step 1: Analyze Image Intent. First, understand the "usage intent" of this image—that is, the functional purpose for which it was created or used. Image intent is not a description of visual content (e.g., "a blue poster") or physical category (e.g., "indoor photo"); rather, it addresses: why was this image produced, what purpose might it serve in a real-world scenario, and who are its likely target audience or users?

Step 2: Derive User Persona. Based on the image intent, conceive a reasonable user role—who would need to examine, inspect, or judge this image in their life or work? *Requirements:* (i) Be specific to a profession or role, rather than a generic "user"; (ii) there should be a natural causal relationship between the persona and the image intent.

Step 3: Design Judgment Task. Based on the persona, design a complex rule-based judgment task that this role might perform in real-world settings. *Requirements:*

- The task is a high-level description explaining "from what aspects" the image needs to be inspected, but does not need to list the specific rules.
- The task should be concise and clear, have a non-trivial difficulty level, require reference to multiple rules in order to be answered, and cannot be resolved by visual observation alone.
- The aspects inspected should be understandable and judgeable based on general domain common sense and world knowledge, and should not rely on specific industry professional regulations, technical standards, or domain terminology (e.g., avoid requiring knowledge of legal article numbers, medical diagnostic standards, or engineering specification parameters).
- The task should not degenerate into a simple yes/no classification task, but require inference combined with the image content.
- The task should have a real-world demand background in practical scenarios.
- The final judgment result should be a binary conclusion of "compliant / non-compliant" or "pass / fail".

Output Format

Strictly output a JSON object:

```
{
  "image_intent": "<Brief analysis of image intent, 1-2 sentences>",
  "persona": "<User persona name>",
  "task": "<High-level description of the multi-criteria judgment task, explaining the aspects to be inspected and the final judgment goal, without listing specific rules>"
}
```

Figure 9: *Prompt used in Stage 1 to jointly extract image intent, derive a domain-specific persona, and propose a multi-criteria judgment task.*

Prompt Used for Rubric Quality Assessment

[SYSTEM PROMPT]

You are a multimodal data quality assessment expert. Your task is to evaluate the quality of a set of automatically generated evaluation criteria (rubrics).

Input Information

You will receive the following information:

- Image: An image to be judged.
- Persona: Describes the user role performing the task.
- Task: A description of a complex rule-based judgment task, requiring systematic inspection of the image based on multiple rules.
- Rubrics: A set of numbered binary judgment rules (R1, R2, ...) automatically generated by the model. Each rule specifies a pass/fail criterion and is tagged with a Type (Perceptual, Reasoning, Content, Format, or Linguistic) and a Priority level (Critical, Important, or Optional).

Evaluation Dimensions

Please evaluate the rubric set along the following four dimensions, providing a score of 1–5 and a brief rationale for each.

Dimension 1: Specificity. *Object: each rule.* Is each rule precisely described, with actionable judgment criteria and clear "compliant / non-compliant" boundaries? Are the judgment objects identifiable in the image (i.e., presence or absence can be determined)?

Dimension 2: Reasonableness. *Object: each rule.* Under the given persona and task, is each rule logically grounded and of practical inspection value? Reasonableness is judged by the persona/task, *not* by whether the inspected object actually appears in the image.

Dimension 3: Completeness. *Object: the rule set.* Does the rubric cover all judgment dimensions required by the task, including those implied by the task description and realized only through image content?

Dimension 4: Coherence. *Object: the rule set.* Are the rules semantically independent, free of redundant repetition and logical contradictions, and as a whole well-aligned with the persona and task?

For each dimension, use the following anchored scale: 5 = excellent / negligible defects; 4 = strong with minor issues on isolated rules; 3 = mixed, with several clearly problematic rules; 2 = majority of rules problematic on this dimension; 1 = near-complete failure on this dimension.

Output Format

Strictly output a JSON object with one entry per dimension:

```
{
  "specificity": {"score": <1-5>, "rationale": "..."},
  "reasonableness": {"score": <1-5>, "rationale": "..."},
  "completeness": {"score": <1-5>, "rationale": "..."},
  "coherence": {"score": <1-5>, "rationale": "..."}
}
```

Figure 10: *Prompt used in Stage 3 rubric quality assessment.*

Prompt Used for Structured Response Generation

[OUTPUT PROMPT]

[Output Requirements]:

The rubrics above contain numbered rules (R1, R2, ...), each tagged [Type|Priority].

- Type guides your reasoning approach: P (Perceptual) = visual evidence, R (Reasoning) = logical inference, C (Content) = coverage check, F (Format) = structural check, L (Linguistic) = style/tone check.
- Priority determines weight: Critical (any fail → overall Fail), Important (major quality factor), Optional (bonus).

Think step by step:

Step 1 — Visual Grounding. Identify and describe the visual elements in the image that are relevant to the task and rubrics.

Step 2 — Rule-by-Rule Verification. Evaluate each rule in priority order (Critical first, then Important, then Optional). For each rule R_i [Type|Priority]:

- Apply type-appropriate reasoning: cite visual evidence for [P], perform logical inference for [R], check information scope for [C], verify structure for [F], assess language style for [L].
- Conclude with Pass or Fail and a brief justification.

Step 3 — Aggregation & Conclusion. Aggregate all rule results. If any Critical rule is Fail, overall result is Fail. Otherwise, weigh Important and Optional results. State the decisive rule(s) and the overall judgment.

Strictly output a JSON object in the following format:

```
{
  "thinking_process": "Step 1: ...\nStep 2:\n- R1 [Type|Priority]: Pass/Fail. Justification...\n- R2 [Type|Priority]: Pass/Fail. Justification...\n...\nStep 3: ...",
  "final_answer": {
    "visual_grounding": "Key visual elements relevant to the task",
    "rule_results": {"R1": "Pass", "R2": "Fail", ...},
    "decisive_rules": "R1, R3",
    "overall_result": "Pass/Fail"
  }
}
```

Figure 11: *Prompt used in Stage 4 to elicit the structured discriminative trace $\tau = (g, \{(j_i, e_i)\}, y)$. Type tags steer the per-rule reasoning mode, while priority tags impose a Critical-first verification order and an aggregation rule under priority precedence.*