SIGHT: COMPOSITIONAL SKILL ROUTING FOR INSTANCE-SPECIFIC RUBRIC GENERATION ACROSS HETEROGENEOUS MULTIMODAL TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward models guide reinforcement learning (RL) for multimodal large language models with scalar scores that obscure multi-dimensional human judgment criteria. To address this, rubric-as-reward decomposes response quality into structured, verifiable criteria. Despite rapid adoption, two questions remain: can benchmarks assess rubric fidelity to human-valued dimensions, and how can prompt-specific rubrics reflect human preferences across heterogeneous multimodal tasks? We introduce **MM-RubricBench**, the first multimodal rubric-guided reward benchmark, spanning six heterogeneous domains. Each curated question has a weighted expert-authored rubric and human ratings of eleven models' responses scored against it. These annotations support evaluation of judge-human preference alignment and generated rubric quality. We further propose **SIGHT**, compositional Skill routing for Instance-specific rubric Generation across Heterogeneous multimodal Tasks. SIGHT transfers construction logic rather than rubric content: it extracts reusable skills from expert rubrics, builds instance-specific evaluation blueprints, selects complementary skills to cover them, and generates rubrics traceable to specific requirements. On MM-RubricBench, these rubrics outperform model-based scoring protocols in human-preference alignment, approaching the human rubric ceiling. When used as RL rewards, these rubrics also improve policies over baselines in multimodal reasoning, instruction following, and agentic rewarding. Construction-skill routing thus supports practical, scalable and reliable multimodal alignment. Code, datasets, and benchmark will be publicly available.

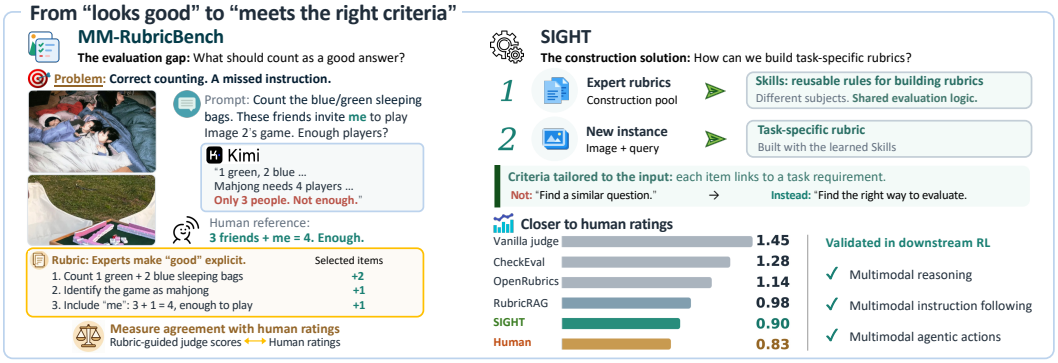


Figure 1: **Overview of the MM-RubricBench and SIGHT.**

1 INTRODUCTION

In the post-training stage of multimodal large language models (MLLMs), mainstream reinforcement learning from human feedback (RLHF) pipelines fit a neural reward model (RM) that compresses the judgment of a response into a scalar reward or a pairwise preference label (Rafailov et al., 2023; Schulman et al., 2017). Human judgments, however, are compositional and multi-dimensional. An answer is weighed jointly on factual correctness, logical rigor, expressive completeness, and safety. A single number cannot record which of these an answer met, and can be raised through surface features such as response length or formatting rather than through genuine alignment with human intent (Casper et al., 2023; Coste et al., 2024; Saito et al., 2023).

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.

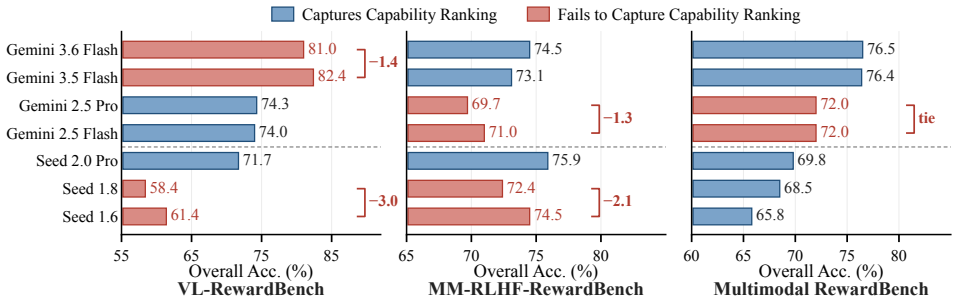


Figure 2: **Existing multimodal reward benchmarks may struggle to distinguish frontier MLLMs.** Models are arranged by expected within-family orderings; brackets mark violations and score gaps.

Reinforcement learning with verifiable rewards (RLVR) (Guo et al., 2025; Jaech et al., 2024) instead checks response outcomes against reference answers, automatically at scale in mathematics and code generation tasks, using a task verifier rather than an RM fitted to human labels. However, only the endpoint is rewarded, leaving reasoning process as a byproduct. A correct derivation, an incorrect one yielding the right answer, and a lucky guess may receive the same reward, leaving the policy unable to distinguish genuine reasoning from shortcuts. Both paradigms share a bottleneck: supervision yields a coarse score that cannot reflect the multiple dimensions of human judgment.

To recover the compositional structure of human values, rubrics-as-reward (RaR) paradigms (Gunjal et al., 2026; Viswanathan et al., 2026) decompose "a good response" into atomic, human-interpretable criteria for a judge to score. For RLHF, rubrics expose dimensions hidden by scalar rewards; for RLVR, they provide process-level signals finer-grained than final-answer rewards.

However, this promise depends on rubric quality. Existing constructors are either domain-specific, distilling rubrics from correct and incorrect trajectories for mathematical reasoning (Sheng et al., 2026; Xie et al., 2026) or image editing (Tian et al., 2026; Zhang et al., 2026b), or expert-agnostic, assuming rather than measuring fidelity to human preferences (Liu et al., 2026; Xie et al., 2025). Such fidelity is not guaranteed: recent study RubricBench (Zhang et al., 2026a) reports a 27% accuracy gap between model-generated and human-authored rubrics.

This leaves two fundamental gaps. **(a) Multimodal rubric evaluation is missing.** Existing multimodal reward benchmarks (Li et al., 2025) record judgments as chosen-versus-rejected labels without the underlying criteria, so they cannot assess whether generated rubrics cover human-valued dimensions, hallucinate spurious ones, or generalise reliably. Their discriminative power is also fading. Re-evaluating current frontier MLLMs reveals inversions or ties in five of fifteen comparisons with unambiguous expected orderings, such as a release against its successor (Fig. 2). **(b) Multimodal rubric construction remains poorly generalised.** Tasks such as information extraction, visual reasoning, and multi-turn interaction each have distinct notions of quality. A generic generator cannot accommodate them all, while domain-specific methods do not scale across them.

To close these gaps, we pair a rubric benchmark with a construction framework (Fig. 1). We introduce **`MM-RubricBench`**, to our knowledge the first multimodal rubric-guided RM benchmark. Each query pairs a human-authored rubric specifying criteria and relative importance through explicit credits and deductions with human ratings of eleven independently sampled model responses. We retain prompts only when answers cannot be recovered from text priors and multiple quality dimensions contribute. The signal remains discriminative beyond the models used to build it, recovering the release order in all fourteen within-family comparisons among current frontier MLLMs.

On the rubric construction side, we propose **`SIGHT`**, compositional Skill routing for Instance-specific rubric Generation across Heterogeneous multimodal Tasks. Our key insight is that expert rubrics offer transferable construction principles, while their content is tied to the facts and requirements of each image and question. `SIGHT` extracts reusable construction skills from expert rubrics. For each new prompt, it builds an evaluation blueprint, selects complementary skills to cover it, and generates rubric items traceable to specific requirements. On `MM-RubricBench`, these rubrics achieve higher preference agreement, with pairwise acc. of 0.73 versus RubricRAG's 0.70, approaching the 0.75 human rubric ceiling. As RL across multimodal reasoning, instruction following, and agentic behaviour, they achieve the best average scores, improving baseline by **5.7**, **4.3**, and **8.0** points, respectively, demonstrating `SIGHT`'s broad applicability to downstream improvement.

In summary, our contributions are threefold:

- **MM-RubricBench, the first multimodal rubric-guided RM benchmark.** We curate multimodal questions across six domains, each paired with an expert-authored rubric and human ratings of eleven sampled model responses. It evaluates judge-human preference alignment, rubric quality against human rubrics, and meaningful ranking of newly released frontier MLLMs.
- **SIGHT, a rubric construction framework built on skills representing reusable construction logic.** Our method automatically generates rubrics at scale by extracting a compact bank of reusable construction skills and routing complementary skills to cover each instance-specific blueprint. The resulting rubrics align better with human preferences than other scoring protocols.
- **Consistent downstream gains across three multimodal RL settings.** Policies trained with SIGHT rubric rewards outperform those using other rubric generation schemes or preference-only rewards across multimodal reasoning, instruction following, and agentic rewarding tasks.

2 Related Work

2.1 Rubric Construction

Rubric construction follows several approaches. Prompt-only methods elicit instance-level criteria from strong LLMs using design principles (Cook et al., 2024; Fan et al., 2024; Gunjal et al., 2026; Wei et al., 2025) or chain-of-rubrics reasoning within the reward model (Chen et al., 2026b). Contrastive methods derive criteria from chosen–rejected responses (Liu et al., 2026; Xie et al., 2025), winning–losing image generations (Tian et al., 2026), or correct–incorrect reasoning trajectories (Xie et al., 2026). Self-evolving methods iteratively refine rubrics (Sheng et al., 2026), expand skill and tool libraries (Zhang et al., 2026b), or structure rubrics for long-horizon agents (Li et al., 2026). Other methods condition generation on external material. RubricRAG (Dhole & Agichtein, 2026) retrieves human rubrics for related queries, transferring subject-specific content only when the pool contains a related query. Skill-RM (Chen et al., 2026a), building on skill routing for LLM agents (Zheng et al., 2026), casts reward computation as an executable skill combining heterogeneous evidence such as verifiers, references, and checklists. It governs scoring with existing criteria, not their construction. SIGHT transfers human rubric construction logic, neither content nor scoring procedures. Its bank records how each expert rubric was built; skills are selected to cover instance-specific evaluation requirements. A skill can thus serve the same requirement across unrelated domains, allowing a compact bank to cover heterogeneous multimodal tasks.

2.2 Rubrics as Reward Signals

Rubrics also serve as training signals. RaR (Gunjal et al., 2026) uses rubric-conditioned judges in GRPO with weighted aggregation. ARR (Tian et al., 2026) applies rubric-guided optimization to text-to-image and image editing; SRaR (Xie et al., 2026) maps rubric items to reasoning steps for finer-grained process supervision. RLCER (Sheng et al., 2026) jointly improves reasoning and rubric quality; RubricEM (Li et al., 2026) assigns stage-structured rubric credit to long-horizon agents. These methods demostrate that rubrics rewarding outperform scalar or preference-only rewards.

2.3 Reward Model Benchmark

Tab. 1 compares reward-model benchmarks by coverage, annotation rigor, and signal richness. Preference-only benchmarks RewardBench2 (Malik et al., 2026), RMB (Zhou et al., 2025), PPE (Frick et al., 2025), and HelpSteer3 (Wang et al., 2025) span diverse domains but provide only pairwise or best-of-N labels. PaperBench (Starace et al., 2025), HealthBench (Arora et al., 2025), ProfBench (Wang et al., 2026), and RubricBench (Zhang et al., 2026a) provide expert rubrics but remain domain-specific or text-only; RubricBench reports a 27% gap between model-generated and human-authored rubrics. VL-RewardBench (Li et al., 2025), MM-RLHF-RewardBench (Zhang et al., 2025), and Multimodal RewardBench (Yasunaga et al., 2025) cover multimodal tasks but supply only pair-wise labels, without rubric-level supervision. Their discriminative power has largely saturated on current frontier MLLMs (Fig. 2). MM-RubricBench combines multimodal coverage, human-authored rubrics specifying criteria and their relative importance, and absolute human scores for eleven responses per prompt, supporting preference-level and rubric-quality evaluation.

Table 1: **Reward-model benchmark comparison.** Nine properties span data coverage, annotation rigor and signal richness. ✓, ✗ and △ denote full, no and partial support.

Dataset	Release Date	Data Coverage		Annotation Rigor				Signal Richness		
		Diverse Domains	Multi-modal	Discrim. Ability	Annot. Quality	Rubric Based	Human Rubrics	Multi-Response	Absolute Scores	Weighted Rubrics
RMB 2025	2024.10	✓	✗	✗	✗	✗	✗	△	✗	✗
PPE 2025	2024.10	✓	✗	✗	✗	✗	✗	✗	✗	✗
VL-RewardBench 2025	2024.11	△	✓	✗	△	✗	✗	✗	✗	✗
MM-RLHF-RewardBench 2025	2025.02	✓	✓	✗	✓	✗	✗	△	△	✗
Multimodal RewardBench 2025	2025.02	✓	✓	✗	✓	✗	✗	✗	✗	✗
PaperBench 2025	2025.04	✗	✗	△	✓	✓	✓	✗	✗	△
HelpSteer3 2025	2025.05	✓	✗	△	✓	△	✗	✗	△	✗
HealthBench 2025	2025.05	✗	✗	△	✓	✓	✓	✗	✗	△
RewardBench2 2026	2025.06	✓	✗	✗	✗	✗	✗	△	✗	✗
ProfBench 2026	2025.10	✗	✗	✓	✓	✓	✓	✗	✗	△
RubricBench 2026a	2026.03	✓	✗	✓	✓	✓	✓	✗	✗	✗
MM-RubricBench	2026.09	✓	✓	✓	✓	✓	✓	✓	✓	✓

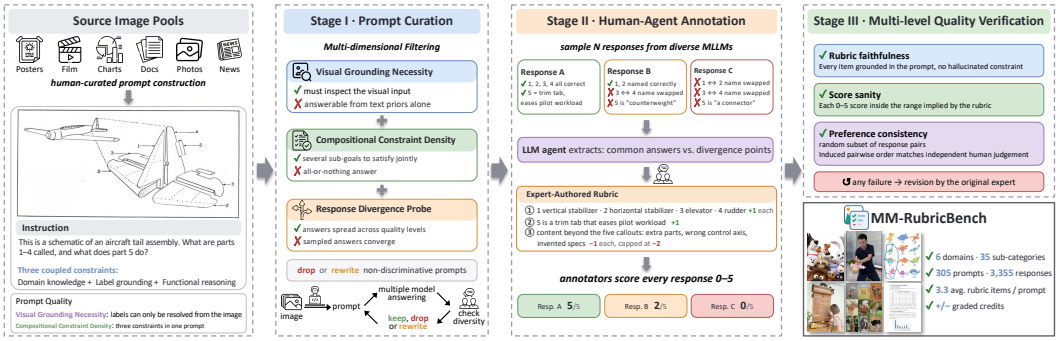


Figure 3: **Construction pipeline of MM-RubricBench. Stage I** curates visually grounded, multi-constraint prompts and tests their discriminative power. **Stage II** combines agent summaries of agreements and disagreements across model responses with expert rubric authoring and response scoring. **Stage III** verifies rubric faithfulness, score validity, and pairwise preference consistency, returning failures for expert revision.

3 MM-RubricBench: A Multimodal Rubric Benchmark

We introduce **MM-RubricBench**, to our knowledge the first multimodal reward-model benchmark pairing each query with a graded expert rubric and absolute human scores for eleven model responses. It evaluates judges and rubric generators at the preference and rubric-quality levels. Sec. 3.1 describes the design, Sec. 3.2 the construction pipeline, and Sec. 3.3 the statistics.

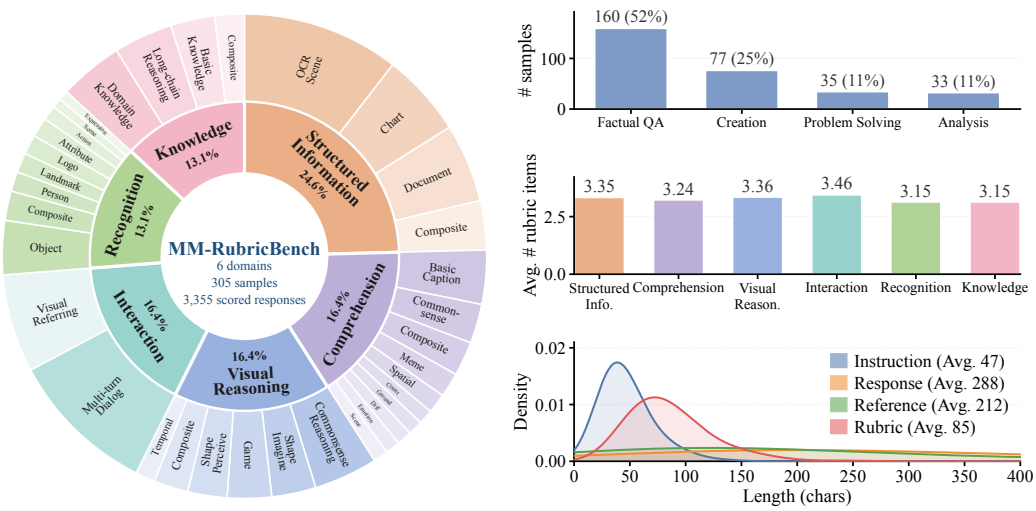
3.1 Benchmark Overview

Gaps in existing benchmarks and our positioning. Existing benchmarks may not fully assess multimodal reward alignment with human preferences (Tab. 1). Preference-only benchmarks omit criteria for distinguishing different aspects of quality; expert-rubric benchmarks are domain-specific or text-only with binary rubrics and pairwise labels; multimodal benchmarks retain only chosen-versus-rejected labels. We combine multimodal coverage, annotation rigor, and signal richness.

Design principles. These requirements guide three design decisions: (1) **Grounded and multi-dimensional prompts.** We prioritise queries requiring visual grounding and multiple substantive quality dimensions, such as recognition, reasoning, and format compliance, so surface heuristics cannot decide preferences. (2) **Expert rubrics with graded credit.** Domain experts write compact rubrics with explicit credits and deductions, capped where appropriate, reflecting human partial-credit judgements beyond binary Yes/No checks. (3) **Absolute scores across multiple responses.** Human scores for eleven independently sampled responses per prompt yield a richer score matrix from which preference pairs are derived, supporting absolute, pairwise, and rank-level evaluation.

3.2 Benchmark Construction

To obtain discriminative prompts and faithful, consistent rubrics and scores, we use three stages: prompt curation, human-agent collaborative annotation, and multi-level quality verification (Fig. 3).



(a) Two-level domain composition (b) Task types, per-domain rubric items, and text lengths

Figure 4: **`MM-RubricBench` statistics overview.** (a) Two-level domain composition, with the six top-level domains and their shares on the inner ring and their fine-grained sub-categories outside. (b) From top to bottom, task type distribution, average rubric items per domain, and length distributions.

Source and coverage. Web-sourced posters, film and animation screenshots, charts, documents, and everyday photographs provide realistic visual inputs. Human-curated prompts comprehensively cover six domains and their fine-grained sub-categories: *Structured Information Analysis*, *Comprehension and Analysis*, *Visual Reasoning*, *Interaction*, *Content Recognition*, and *Knowledge Ability*.

Stage I: High-quality prompt curation. Annotators build a large prompt pool from these images, requiring: (1) *Visual grounding necessity.* Answers depend on image details beyond text priors, requiring multimodal reasoning. (2) *Compositional constraint density.* Prompts combine constraints, such as multi-entity identification, formatted extraction, or recognition with follow-up reasoning. Three strong closed-source models probe retained prompts. We drop or rewrite prompts with identical or uniformly correct answers as non-discriminative, forming the annotation pool.

Stage II: Human-agent collaborative annotation. A strong LLM summarises agreements and disagreements among diverse multimodal models' sampled responses per prompt, giving experts a starting point grounded in where models disagree. Experts use these summaries to write compact rubrics with importance-weighted requirement credits and separate deductions for typical failures, capped where needed to prevent one mistake dominating. Annotators score every response 0–5 against these rubrics to keep scoring consistent across sampled responses (Sec. A.5).

Stage III: Multi-level quality verification. We verify three levels: rubric criteria must be faithful to the prompt, grounded in the available visual evidence, and free of hallucinated constraints; individual 0–5 scores must match visible content and rubric-implied bounds; preferences induced by scores on randomly sampled response pairs must agree with independent human judgements. The original expert revises failed rubrics or scores to maintain faithfulness and consistency at all three levels.

3.3 Benchmark Statistics

Domain and task composition. Fig. 4(a) shows broadly balanced coverage: structured information leads at 24.6% (OCR, charts, and documents); comprehension, visual reasoning, and interaction each contribute 16.4%; recognition and knowledge each 13.1%. The 35 sub-categories capture heterogeneity. Tasks span factual QA, open-ended creation, problem solving, and analysis (Fig. 4(b), top). Nearly half go beyond factual QA and conventional short-answer visual QA.

Rubric and text statistics. The middle panel of Fig. 4(b) shows compact rubrics averaging 3.30 items per prompt, with cross-domain consistency suggesting comparable requirement granularity. Each item specifies a credit or deduction rather than a pass/fail criterion. The bottom panel shows concise rubrics alongside substantially longer responses.

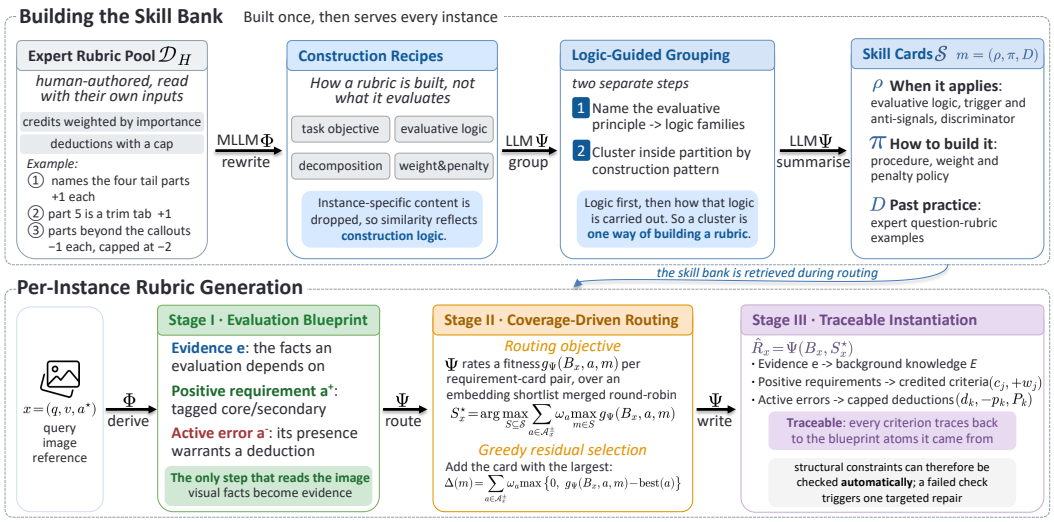


Figure 5: **Overview of SIGHT. Skill bank construction** groups expert-rubric recipes by evaluative logic, then construction pattern, and summarises clusters into skill cards. **Stage I**, the only image-reading step, extracts evidence, positive requirements, and active errors into a blueprint. **Stage II** greedily selects complementary cards by residual gain. **Stage III** follows their procedures to generate a rubric with traceable criteria.

4 SIGHT: Rubric Construction by Compositional Skill Routing

Human-authored rubrics align with human preferences far better than current generators' outputs (Zhang et al., 2026a), but are costly to write at scale. RubricRAG (Dhole & Agichtein, 2026) retrieves human rubrics from related queries as in-context guidance. However, retrieval follows surface similarity in subject matter, modality, or question form, transferring instance-specific objects, values, and answer requirements that need not apply to a new question.

The reusable part is a rubric's *construction logic*: splitting requirements into non-overlapping, independently checkable items, distinguishing core from secondary requirements, assigning credits, and identifying active errors, such as unsupported visual claims, for deductions. This logic transfers across subjects and domains. This yields **SIGHT**, compositional Skill routing for Instance-specific rubric Generation across Heterogeneous multimodal Tasks (Fig. 5). Using strong MLLM Φ and LLM Ψ, we extract a skill bank $\mathcal{S}$ from expert rubrics $\mathcal{D}_H$. For an instance $x=(q,v,a^\star)$, comprising query q, image v, and reference answer $a^\star$, we derive an evaluation blueprint B_x before writing criteria, route complementary skills $S_x^\star$ to cover its requirements, and instantiate a rubric $\hat{R}_x$ whose items trace back to them:

$$\underbrace{\mathcal{D}_H \xrightarrow{\Phi,\Psi} \mathcal{S}}_{\text{skill extraction}} \qquad \underbrace{x \xrightarrow{\Phi} B_x \xrightarrow{\textsc{Route}} S_x^\star \xrightarrow{\Psi} \hat{R}_x}_{\text{rubric generation}}. \tag{1}$$

Rubric form. A rubric $\hat{R}_x=\big(E,\{(c_j,w_j)\},\{(d_k,p_k,P_k)\}\big)$ has background knowledge E, criteria c_j with credits w_j, and errors d_k with deductions p_k capped at P_k. A response y receives

$$\text{score}(y)=\sum_j w_j\,u_j(y)-\sum_k \min\big(p_k\,n_k(y),\ P_k\big). \tag{2}$$

Credits sum to $\sum_j w_j=W$; $u_j(y)$ grades partial fulfilment and $n_k(y)$ counts occurrences of undesirable behaviours such as unsupported hallucinations.

4.1 Extracting Reusable Skills from Human Rubrics

We rewrite human rubrics as construction recipes, group them by evaluative logic and construction pattern, and summarise each cluster into a reusable skill card.

Construction recipes as the unit of reuse. Φ reads each human rubric with its input and rewrites it as a *recipe* describing its construction. Five fields specify the task objective, evaluative logic, decomposition scheme, weight policy, and penalty policy. We concatenate these fields while excluding instance-specific content, so similarity reflects rubric construction rather than prompt semantics.

Logic-guided partitioning. We ask Ψ to name each recipe's evaluative principle using an open vocabulary rather than a fixed taxonomy, reflecting the annotator's actual practice. Near-synonymous principles are merged into logic families, each defining a partition. Within each partition, K-means groups recipes by construction pattern, with K selected by silhouette score. Grouping first by logic and then by its implementation makes each cluster a way of building a rubric rather than a topic.

Skill cards. Ψ then summarises each cluster into a skill card $m = (\rho, \pi, D)$, where ρ describes applicability through its evaluative logic, trigger and anti-signals, and a discriminator; π specifies the construction procedure, weight policy, and penalty policy; and D contains four human question-rubric pairs as demonstrations. Trigger signals are defined only over prompt-level features, while discriminators distinguish the card from others within the same partition. Demonstrations are sampled from recipes near the cluster centroid and diversified by domain.

4.2 From Evaluation Blueprints to Instance-specific Rubrics

The bank is built once. Each new instance yields a blueprint, which guides complementary skill routing and rubric instantiation under the selected construction procedures.

Per-instance evaluation blueprint. Given an instance x, Φ first constructs an evaluation blueprint

$$B_x = \Phi(x) = \big(\{e_l\}, \{a_i^+\}, \{a_k^-\}\big). \tag{3}$$

The blueprint contains three kinds of *rubric atoms*: evidence e_l recording facts needed for evaluation in the current instance, positive requirements a_i^+ that a response is expected to satisfy, and active errors a_k^- whose presence in a response warrants deductions. Each positive requirement is tagged by importance (*core* or *secondary*) and grounding source (*image*, *reference*, or *instruction*). We denote all atoms by $\mathcal{A}_x$ and the positive requirements and active errors by $\mathcal{A}_x^{\pm}$. Only this step reads the image, recording visual facts as evidence so all later steps operate on text alone.

Coverage-driven skill routing objective. For each atom $a \in \mathcal{A}_x^{\pm}$ and card m, Ψ predicts fitness $g_\Psi(B_x, a, m)$ for turning the atom into scoring items; evidence atoms supply facts and are excluded. To avoid scoring every card, embedding retrieval finds the nearest cards per atom and merges them round-robin into a shared pool, retaining candidates for every atom. Ψ scores only this pool. We select skills under context budget C:

$$S_x^\star = \arg\max_{S \subseteq \mathcal{S}} \sum_{a \in \mathcal{A}_x^{\pm}} \omega_a \max_{m \in S} g_\Psi(B_x, a, m) \quad \text{s.t.} \quad \mathrm{cost}(S) \le C, \tag{4}$$

where ω_a weights atom importance, assigning higher weight to core than secondary requirements.

The inner maximum credits each atom only with its most helpful card, rather than summing contributions from multiple cards covering that atom. Unlike a sum, it gives an overlapping card no gain unless that card improves the current best fitness for some atom. Selection thus rewards complementary coverage or stronger support for existing requirements, without a redundancy rule.

Greedy residual selection. We approximate Eq. (4) greedily to avoid enumerating subsets. Starting from $S = \varnothing$ and $\mathrm{best}(a) = 0$, we repeatedly add the card with the largest marginal gain

$$\Delta(m) = \sum_{a \in \mathcal{A}_x^{\pm}} \omega_a \max\big\{0,\ g_\Psi(B_x, a, m) - \mathrm{best}(a)\big\}, \tag{5}$$

and update $\mathrm{best}(\cdot)$ after adding each newly selected card. Clipping at zero counts only fitness improvements over the current set, so high fitness alone brings no gain if existing cards already match it. Selection stops when all routed atoms reach a target coverage level, the largest remaining gain falls below a tolerance, or the budget or card-count cap is reached.

Traceable rubric instantiation. Ψ writes a rubric from the blueprint and selected skills,

$$\hat{R}_x = \Psi(B_x, S_x^\star), \tag{6}$$

mapping evidence to background knowledge, positive requirements to credited criteria, and active errors to deductions. Each criterion retains provenance through a non-empty set of blueprint atoms, $\sigma(c_j) \subseteq \mathcal{A}_x$, so each scoring item can be traced to the requirements or evidence supporting it.

Structural verification and repair. We check score constraints, requirement coverage, dependency validity, criterion redundancy, and separation of criteria from deductions. Failed checks trigger one targeted repair, accepted only if the violation count does not increase.

Table 2: **Model capability ranking under MM-RubricBench.** Each response is scored against the human rubrics, with scores averaged overall and by domain. The resulting ranking separates models by capability, with stronger models consistently scoring above weaker ones, showing that MM-RubricBench provide a discriminative signal for frontier MLLMs.

Model	Release	Overall	Structured Information	Comprehension	Visual Reasoning	Interaction	Recognition	Knowledge
Gemini 3.6 Flash	2026.07	**84.1**	**87.0**	81.0	84.2	75.0	**87.0**	**90.9**
Gemini 3.1 Pro	2026.02	83.5	80.3	**83.0**	**87.8**	**77.8**	**87.0**	88.5
Gemini 3.5 Flash	2026.05	82.9	86.5	78.6	82.3	76.2	83.0	90.5
Claude Opus 5	2026.07	82.0	83.8	76.2	84.4	77.4	80.6	90.0
Gemini 3 Pro	2025.11	80.4	84.6	77.4	79.4	71.8	85.0	84.0
Gemini 3 Flash	2025.12	77.2	81.9	72.6	69.8	73.0	79.5	86.5
Seed2.0 Pro	2026.02	75.7	81.9	76.2	68.2	72.6	77.0	75.5
Kimi K3	2026.07	74.8	80.6	65.4	67.8	71.0	82.0	82.0
Seed2.0 Lite	2026.02	74.1	80.6	76.2	62.2	72.6	80.0	70.5
Claude Opus 4.7	2026.04	73.9	79.7	70.1	62.2	75.8	75.5	78.0
Kimi K2.5	2026.01	72.8	76.1	65.8	63.0	71.8	82.5	79.5
Gemini 2.5 Pro	2025.03	71.2	75.8	66.2	70.2	65.0	82.0	67.0
Seed1.8	2025.12	69.9	70.2	70.2	63.0	69.8	76.0	71.5
Seed1.6	2025.06	68.2	74.7	69.4	61.8	67.0	64.0	68.0
Gemini 2.5 Flash	2025.04	67.4	68.3	63.0	66.6	62.6	74.0	71.5
Claude Opus 4.6	2026.02	66.1	68.7	57.9	54.6	62.8	77.6	78.0
MiMo V2.5	2026.04	63.1	68.3	67.0	59.0	59.4	54.5	66.5
MiMo V2 Omni	2026.03	61.8	70.5	65.0	58.2	57.0	47.0	67.0

5 EXPERIMENTS

We evaluate MM-RubricBench and SIGHT through model ranking with human rubrics (Sec. 5.1), generated rubrics' alignment with human preferences (Sec. 5.2), and their utility as RL rewards (Sec. 5.3). Together, these experiments test evaluation reliability and downstream policy improvement across heterogeneous tasks, linking rubric quality to its practical use in training.

5.1 HUMAN RUBRICS RANK MODELS RELIABLY

Experimental setup. We test the human rubrics on eighteen recent MLLMs spanning proprietary and open-weight families, including smaller variants and models stronger than those used to construct MM-RubricBench. We sample one response per prompt, score it against the corresponding human rubric, and average scores overall and by domain (Sec. C.1). The shared rubric fixes the evaluation criteria for all models answering the same prompt.

MM-RubricBench Results. In Tab. 2, human rubrics recover the expected within-family order across all fourteen generation and variant comparisons, including Pro-versus-Flash pairs, with an overall spread of 22.3 points. Written before these models were released, the rubrics remain discriminative for models stronger than those used in benchmark construction.

No model leads every domain: the top-ranked Gemini 3.6 Flash ranks fifth on multi-turn interaction, while Gemini 3.1 Pro, only 0.6 points behind overall, ranks ninth on structured information analysis. Per-domain spreads of 18.7–40.0 points show that the rubrics distinguish different capabilities that an overall score alone can obscure.

5.2 SIGHT RUBRICS ALIGN BETTER WITH HUMAN PREFERENCES

Skill bank. We extract the bank from 996 human rubrics, combining SedarEval (Fan et al., 2024) and in-house annotation: 576 are text-only and 420 multimodal. The source rubrics contain 4,090 scoring items, averaging 4.11 items per rubric across the two modalities. Grouping recipes into 7 evaluative logic families and clustering within each yields 14 skill cards. Every card draws on both modalities, supporting the reuse of construction logic across them. Statistics in Secs. B.1 and C.2.

Routing and generation. We retrieve five cards per atom and cap the candidate pool at eight. The LLM assigns fitness scores from 0 to 3; atom weights are 1.0, 0.5, and 0.7 for core, secondary, and penalty requirements. Under a 6,000-character budget and a four-card cap, selection stops when every atom reaches 2.0 or the largest remaining gain falls below 0.3, averaging 1.94 skills per prompt. Evidence atoms provide grounding facts and are excluded from skill selection.

Table 3: **Agreement between judge scores and human preferences on MM-RubricBench.** We report sample-level errors, including mean absolute error and the fraction of samples within one point of the human rating, as well as preference-level agreement measured by pairwise accuracy and rank correlations. Scores from SIGHT rubrics align more closely with human preferences than fixed templates or self-generated rubrics, approaching human-authored rubrics.

Judge Model	Judging Protocol	Sample-level		Preference-level		
		MAE ↓	Adj1 Acc. ↑	Pairwise Acc. ↑	Kendall τ_b ↑	Spearman ρ ↑
Doubao Seed2.0	Vanilla LLM-as-a-Judge	1.45	0.59	0.52	0.42	0.45
	Fixed rubric	1.42	0.60	0.54	0.44	0.49
	CheckEval 2025	1.28	0.65	0.60	0.52	0.54
	OpenRubrics 2026	1.14	0.69	0.65	0.57	0.61
	RubricRAG 2026	0.98	0.74	0.70	0.62	0.63
	SIGHT	**0.90**	**0.77**	**0.73**	**0.66**	**0.69**
	Human rubric (ceiling)	0.83	0.79	0.75	0.68	0.70

Scoring protocols. We compare SIGHT with representative model-based scoring protocols on MM-RubricBench, using human-authored rubrics as a reference. All protocols score the same responses with the same judge to isolate the scoring protocol's effect. We report sample-level and preference-level metrics, with evaluation protocols and generation settings detailed in Sec. C.2.

Results. As shown in Tab. 3, SIGHT outperforms all compared model-based protocols on every metric. Against the strongest counterpart, RubricRAG, it improves pairwise accuracy from 0.70 to 0.73, Kendall τ_b from 0.62 to 0.66, and Spearman ρ from 0.63 to 0.69, while reducing MAE from 0.98 to 0.90 and raising adjacent-one accuracy from 0.74 to 0.77. Pairwise accuracy approaches the human-rubric reference of 0.75. The gains over freely generated and retrieved-rubric guidance support applying expert construction procedures through evaluation blueprints. Ablations in Sec. C.4 further validate the effectiveness of SIGHT components.

5.3 SIGHT Rubrics Improve Downstream Policies

We use SIGHT rubrics as RL rewards across three multimodal task families. We report reasoning results here and instruction-following and agentic-rewarding results in Sec. C.3. In the latter two settings, SIGHT achieves the best average scores of 76.4 and 76.0, respectively, extending its benefits to constraint satisfaction and reward modelling. Training details are provided in Sec. D.

Multimodal reasoning. We train on VisualWebInstruct-verified (Jia et al., 2025) and evaluate on six multimodal reasoning benchmarks against existing methods. As shown in Tab. 4, SIGHT achieves the best results across all six benchmarks, raising the average from 61.9 to 67.5 (+5.7), compared with +1.7 from correctness-only rewards. Against RubricRAG, SIGHT also improves all six benchmarks, raising the average from 66.1 to 67.5.

Table 4: **Downstream RL on multimodal reasoning.** SIGHT achieves the best performance on all six benchmarks and the largest average gain.

Method	MathVision	DynaMath	LogicVista	MathVerse	MathVista	WeMath	Avg.	Δ
Qwen3-VL-4B-Instruct	44.4	65.0	53.2	59.3	73.7	75.5	61.9	0
RL w/ Correctness LLM-judge	45.6	65.3	57.4	59.7	76.8	76.4	63.5	+1.7
RL w/ OpenRubrics 2026	47.9	66.9	59.0	62.6	77.8	76.3	65.1	+3.2
RL w/ RubricRAG 2026	48.8	66.2	58.7	63.3	78.6	81.2	66.1	+4.3
RL w/ **SIGHT**	**50.7**	**67.4**	**60.1**	**64.8**	**79.4**	**82.7**	**67.5**	+5.7

6 Conclusion

We introduced MM-RubricBench, a multimodal benchmark pairing expert rubrics with human response ratings, and SIGHT, a framework that transfers reusable rubric-construction skills across heterogeneous tasks. Evaluation blueprints specify instance-level requirements, while complementary skill routing guides the generation of traceable scoring criteria. Experiments show that human rubrics distinguish frontier model capabilities and that SIGHT improves judge–human agreement over existing scoring protocols. Using the generated rubrics as RL rewards also improves multimodal reasoning, instruction following, and agentic rewarding. These findings support construction-skill transfer for scalable, expert-informed multimodal evaluation and alignment.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting, language editing, and consistency checking. This assistance is distinct from the model calls in our research pipeline, including response summarisation for expert annotation, skill extraction, rubric generation, and rubric-conditioned scoring, as described in Secs. 3 to 5. We have reviewed the AI-assisted manuscript content and take responsibility for the final text, claims, and artifacts.

ETHICS STATEMENT

Our work studies rubric-guided multimodal evaluation and policy improvement. Benchmark construction involves expert rubric authoring, human response scoring, and independent preference checks, as described in Sec. A.3. These annotations reflect the judgements of the annotators and the selected task domains, rather than universal preferences. Generated rubrics and model judges may inherit biases or overlook errors despite structural verification, so automated scores should not replace human review in high-stakes applications. Benchmark release and reuse should respect source licences, privacy, and applicable access restrictions.

REPRODUCIBILITY STATEMENT

We describe benchmark construction, rubric generation, and experimental comparisons in Secs. 3 to 5, respectively. The appendices document prompt curation and annotation (Sec. A), skill extraction, blueprint construction, routing, and rubric verification (Sec. B), and experimental settings, judging protocols, metrics, and additional downstream results (Sec. C). The prompt excerpts in the method appendix are illustrative rather than verbatim implementation prompts. We commit to publicly releasing the benchmark data to support further research and independent evaluation.

REFERENCES

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. HealthBench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.

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*, 2025.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

Tao Chen, Gangwei Jiang, Pengyu Cheng, Siyuan Huang, Yihao Liu, Jingwei Ni, Jiaqi Guo, Mengyu Zhou, Kai Tang, Junling Liu, et al. Skill-RM: Unifying heterogeneous evaluation criteria via agent skill. *arXiv preprint arXiv:2606.03980*, 2026a.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru WANG, Yu Zhang, Denghui Zhang, Tong Zhang, Hanghang Tong, and Heng Ji. RM-R1: Reward modeling as reasoning. In *International Conference on Learning Representations*, pp. 88313–88342, 2026b.

Jonathan Cook, Tim Rocktäschel, Jakob Foerster, Dennis Aumiller, and Alex Wang. TICKing All the Boxes: Generated checklists improve LLM evaluation and generation. *arXiv preprint arXiv:2410.03608*, 2024.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. In *International Conference on Learning Representations*, pp. 50905–50931, 2024.

Kaustubh Dhole and Eugene Agichtein. RubricRAG: Towards interpretable and reliable LLM evaluation via domain knowledge retrieval for rubric generation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3681–3687, 2026.

Shengyuan Ding, Shenxi Wu, Xiangyu Zhao, Yuhang Zang, Haodong Duan, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. MM-IFEngine: Towards multimodal instruction following. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1099–1109. IEEE, 2025.

Shengyuan Ding, Xinyu Fang, Ziyu Liu, Yuhang Zang, Yuhang Cao, Xiangyu Zhao, Haodong Duan, Xiaoyi Dong, Jianze Liang, Bin Wang, Conghui He, Dahua Lin, and Jiaqi Wang. ARM-Thinker: Reinforcing multimodal generative reward models with agentic tool use and visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 22195–22205, June 2026.

Zhiyuan Fan, Weinong Wang, Xing Wu, and Debing Zhang. SedarEval: Automated evaluation using self-adaptive rubrics. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16916–16930, 2024.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. How to evaluate reward models for RLHF. In *International Conference on Learning Representations*, pp. 18128–18163, 2025.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as Rewards: Reinforcement learning beyond verifiable domains. In *International Conference on Learning Representations*, pp. 127924–127945, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. OpenAI o1 System Card. *arXiv preprint arXiv:2412.16720*, 2024.

Yiming Jia, Jiachen Li, Xiang Yue, Bo Li, Ping Nie, Kai Zou, and Wenhu Chen. VisualWebInstruct: Scaling up multimodal instruction data through web search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1373–1393, 2025.

Yukyung Lee, Joonghoon Kim, Jaehee Kim, Hyowon Cho, Jaewook Kang, Pilsung Kang, and Najoung Kim. CheckEval: A reliable LLM-as-a-Judge framework for evaluating text generation using checklists. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 15771–15798, 2025.

Gaotang Li, Bhavana Dalvi Mishra, Zifeng Wang, Jun Yan, Yanfei Chen, Chun-Liang Li, Long T Le, Rujun Han, George Lee, Hanghang Tong, et al. RubricEM: Meta-RL with rubric-guided policy decomposition beyond verifiable rewards. *arXiv preprint arXiv:2605.10899*, 2026.

Lei Li, Yuancheng Wei, Zhihui Xie, Xuqing Yang, Yifan Song, Peiyi Wang, Chenxin An, Tianyu Liu, Sujian Li, Bill Yuchen Lin, et al. VL-RewardBench: A challenging benchmark for vision-language generative reward models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 24657–24668, 2025.

Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. OpenRubrics: Towards scalable synthetic rubric generation for reward modeling and LLM alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17417–17437, 2026.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah Smith, Hanna Hajishirzi, and Nathan Lambert. RewardBench 2: Advancing reward model evaluation. In *International Conference on Learning Representations*, pp. 144839–144866, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, pp. 53728–53741, 2023.

Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. Verbosity bias in preference labeling by large language models. *arXiv preprint arXiv:2310.10076*, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Leheng Sheng, Wenchang Ma, Ruixin Hong, Xiang Wang, An Zhang, and Tat-Seng Chua. Reinforcing chain-of-thought reasoning with self-evolving rubrics. *arXiv preprint arXiv:2602.10885*, 2026.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI's ability to replicate AI research. *arXiv preprint arXiv:2504.01848*, 2025.

Juanxi Tian, Fengyuan Liu, Jiaming Han, Yilei Jiang, Yongliang Wu, Yesheng Liu, Haodong Li, Furong Xu, and Wanhua Li. Auto-Rubric as Reward: From implicit preferences to explicit multimodal generative criteria. *arXiv preprint arXiv:2605.08354*, 2026.

Vijay Viswanathan, Yanchao Sun, Xiang Kong, Meng Cao, Graham Neubig, and Sherry Wu. Checklists are better than reward models for aligning language models. In *Proceedings of Advances in Neural Information Processing Systems*, volume 38, pp. 114728–114754, 2026.

Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Daniel Egert, Ellie Evans, Hoo-Chang Shin, Felipe Soares, Yi Dong, and Oleksii Kuchaiev. HelpSteer3: Human-annotated feedback and edit data to empower inference-time scaling in open-ended general-domain tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25640–25662, 2025.

Zhilin Wang, Jaehun Jung, Ximing Lu, Shizhe Diao, Ellie Evans, Jiaqi Zeng, Pavlo Molchanov, Yejin Choi, Jan Kautz, and Yi Dong. ProfBench: Multi-domain rubrics requiring professional knowledge to answer and judge. In *International Conference on Learning Representations*, pp. 5355–5384, 2026.

Tianjun Wei, Wei Wen, Ruizhi Qiao, Xing Sun, and Jianghong Ma. RocketEval: Efficient automated LLM evaluation via grading checklist. In *International Conference on Learning Representations*, pp. 58593–58619, 2025.

Lipeng Xie, Sen Huang, Zhuo Zhang, Anni Zou, Yunpeng Zhai, Dingchao Ren, Kezun Zhang, Haoyuan Hu, Boyin Liu, Haoran Chen, et al. Auto-Rubric: Learning to extract generalizable criteria for reward modeling. *arXiv preprint arXiv:2510.17314*, 2025.

Weichu Xie, Haozhe Zhao, Wenpu Liu, Yongfu Zhu, Liang Chen, Minghao Ye, Zirong Chen, Yuqi Xu, Shuai Dong, Ziyue Wang, et al. Step-wise rubric rewards for LLM reasoning. *arXiv preprint arXiv:2605.17291*, 2026.

Michihiro Yasunaga, Luke Zettlemoyer, and Marjan Ghazvininejad. Multimodal RewardBench: Holistic evaluation of reward models for vision language models. *arXiv preprint arXiv:2502.14191*, 2025.

Qiyuan Zhang, Junyi Zhou, Yufei Wang, Fuyuan Lyu, Yidong Ming, Can Xu, Qingfeng Sun, Kai Zheng, Peng Kang, Xue Liu, et al. RubricBench: Aligning model-generated rubrics with human standards. *arXiv preprint arXiv:2603.01562*, 2026a.

Yi-Fan Zhang, Tao Yu, Haochen Tian, Chaoyou Fu, Peiyan Li, Jianshu Zeng, Wulin Xie, Yang Shi, Huanyu Zhang, Junkang Wu, et al. MM-RLHF: The next step forward in multimodal LLM alignment. *arXiv preprint arXiv:2502.10391*, 2025.

Yuxuan Zhang, Penghui Du, Bo Li, Cong Wei, Junwen Miao, Huaisong Zhang, Songcheng Cai, Yubo Wang, Dongfu Jiang, Yuyu Zhang, et al. RewardHarness: Self-evolving agentic post-training. *arXiv preprint arXiv:2605.08703*, 2026b.

YanZhao Zheng, ZhenTao Zhang, Chao Ma, YuanQiang Yu, JiHuai Zhu, Yong Wu, Tianze Xu, Baohua Dong, Hangcheng Zhu, Ruohui Huang, et al. SkillRouter: Skill routing for LLM agents at scale. *arXiv preprint arXiv:2603.22455*, 2026.

Enyu Zhou, Guodong Zheng, Binghai Wang, Zhiheng Xi, Shihan Dou, Rong Bao, Wei Shen, Limao Xiong, Jessica Fan, Yurong Mou, et al. RMB: Comprehensively benchmarking reward models in LLM alignment. In *International Conference on Learning Representations*, pp. 26543–26589, 2025.

SIGHT: Compositional Skill Routing for Instance-Specific Rubric Generation across Heterogeneous Multimodal Tasks

Supplementary Material

OVERVIEW

This material provides supplementary details to the main paper, organized as follows:

- (A) **MM-RubricBench Details**
  - (A.1) Domain Taxonomy and Composition
  - (A.2) Prompt Curation and Response Collection
  - (A.3) Annotation Protocol and Human Effort
  - (A.4) Rubric Structure and Scoring Guidelines
  - (A.5) Annotation Statistics
  - (A.6) Ranking Stability
- (B) **SIGHT Details**
  - (B.1) Skill Bank Construction
  - (B.2) Evaluation Blueprint Construction
  - (B.3) Coverage-driven Skill Routing
  - (B.4) Rubric Instantiation and Verification
- (C) **Experiment Details**
  - (C.1) Model Ranking Setup
  - (C.2) Alignment Protocols and Metrics
  - (C.3) Additional Downstream Experiments
  - (C.4) Component Ablations
- (D) **Training Details**
  - (D.1) Models and Training Infrastructure
  - (D.2) Training Data and Rubric Preparation
  - (D.3) Training Reward Acquisition
  - (D.4) Optimization and Rollout Configuration
  - (D.5) Training Procedure and Checkpointing
  - (D.6) Reward-Judge Prompt
- (E) **Case Studies**
  - (E.1) Annotated Benchmark Instances
  - (E.2) Clusters of the Skill Bank
  - (E.3) Qualitative Analysis of Generated Rubrics

A MM-RubricBench Details

Building on Sec. 3, this section details how MM-RubricBench is constructed and annotated. Its domain taxonomy and composition (Sec. A.1) provide context for prompt curation and response collection (Sec. A.2). We then describe the annotation workflow and estimated human effort in Sec. A.3, with rubric-level scoring guidance in Sec. A.4. The resulting annotations are characterised in Sec. A.5, followed by an analysis of ranking stability in Sec. A.6.

A.1 Domain Taxonomy and Composition

Taxonomy and coverage. Each of the 305 prompts belongs to one top-level domain and one subcategory within that domain. Tabs. A.1 and A.2 expand the two-level distribution in Fig. 4 into exact counts. Covering six domains and 35 sub-categories, MM-RubricBench encompasses structured information analysis, visual comprehension and reasoning, interaction, content recognition, and knowledge-intensive tasks. This breadth provides a comprehensive benchmark for evaluating the diverse multimodal capabilities of MLLMs.

Visual inputs and rubric length. MM-RubricBench contains 214 single-image prompts, 85 two-image ones and 6 three-image instances. Multi-image inputs occur in every domain. Tab. A.1 also shows overall median of rubric length is 95 words, with an interquartile range of 71–124 words.

Task composition. Tab. A.3 cross-tabulates domains and the four task types shown in Fig. 4. Analysis includes description, explanation, transcription, and translation. Together, these task types enable MM-RubricBench to evaluate a broad spectrum of MLLM capabilities, from factual understanding and analytical reasoning to problem solving and open-ended generation.

Representative domains. Tab. A.4 summarises the scope of each domain and illustrates it with a visual input and query. Complete examples with rubrics and scored responses are provided in Sec. E.1.

Table A.1: **Domain composition of MM-RubricBench.** Counts cover all 305 prompts. Multi-image counts refer to prompts with more than one image. Rubric length is measured in non-whitespace words.

Domain	Prompts	Share (%)	Sub-categories	Multi-image	Median rubric length
Structured Information Analysis	75	24.6	4	13	103
Comprehension and Analysis	50	16.4	10	13	100.5
Visual Reasoning	50	16.4	6	16	94
Interaction	50	16.4	2	17	96
Content Recognition	40	13.1	9	20	82
Knowledge Ability	40	13.1	4	12	95.5
Overall	**305**	**100.0**	**35**	**91**	**95**

Table A.2: **Two-level sample counts in MM-RubricBench.**

Domain	Sub-category	Prompts	%
Structured Information Analysis	OCR scenes	32	42.7
	Chart understanding and reasoning	17	22.7
	Document understanding	16	21.3
	Composite	10	13.3
Comprehension and Analysis	Basic description	11	22.0
	Commonsense and culture	8	16.0
	Meme understanding	5	10.0
	Spatial relations	4	8.0
	Object counting	3	6.0
	Recognition and localisation	3	6.0
	Visual differences	3	6.0
	Emotion understanding	3	6.0
	Scene understanding	3	6.0
	Composite	7	14.0
Visual Reasoning	Commonsense reasoning	13	26.0
	Shape imagination	9	18.0

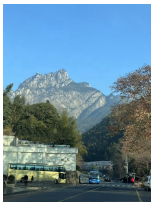
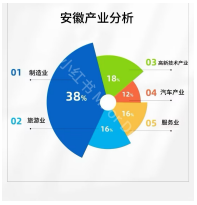
Continued on next page

Domain	Sub-category	Prompts	%
	Games	9	18.0
	Shape perception	8	16.0
	Temporal reasoning	4	8.0
	Composite	7	14.0
Interaction	Multi-turn image–text interaction	30	60.0
	Visual referring	20	40.0
Content Recognition	Objects	11	27.5
	People	4	10.0
	Landmarks	4	10.0
	Logos	4	10.0
	Attributes	4	10.0
	Actions	3	7.5
	Environments	2	5.0
	Expressions	2	5.0
	Composite	6	15.0
Knowledge Ability	Specialised knowledge	13	32.5
	Long-chain reasoning	12	30.0
	Basic knowledge	9	22.5
	Composite	6	15.0

Table A.3: **Task types within each domain.** Counts use the same deterministic task-action classification as Fig. 4. Each row sums to the number of prompts in its domain.

Domain	Factual QA	Creation	Problem Solving	Analysis
Structured Information Analysis	34	24	5	12
Comprehension and Analysis	27	12	0	11
Visual Reasoning	17	19	10	4
Interaction	31	15	3	1
Content Recognition	31	6	0	3
Knowledge Ability	20	1	17	2
Overall	**160**	**77**	**35**	**33**

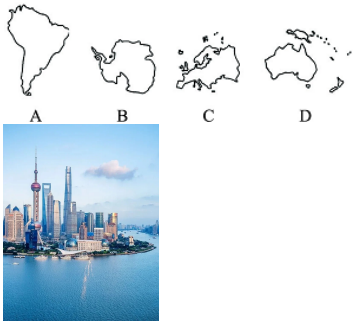
Table A.4: **Representative examples from each top-level domain of `MM-RubricBench`.** Each row shows the domain's scope and an example with its visual input and query.

Domain	Scope	Example
Structured Information Analysis	Models must first read the text embedded in the image and then reason over it, either performing a calculation under stated conditions or producing writing that satisfies the intended criteria, typically in an everyday setting. Compared with the other domains, the in-image documents and the supplied premises are considerably longer, so the difficulty lies in sustaining accurate inference over extended textual evidence. Composite entries pair several images with several sub-questions.	  *I took this photo while travelling. What are the characters on the building to the left? Which scenic area is this? I later looked up the industry shares of this province: what is the gap between tourism and the second-ranked industry?*

continued on next page

Domain	Scope	Example
Comprehension and Analysis	Rather than naming an entity outright, models must interpret what they see, such as which country, artwork, or historical event a depicted object belongs to, or what a particular visual detail evokes. A follow-up question then extends the reasoning to a related fact, as in content recognition. This is the most heterogeneous domain, demanding abilities that range from basic captioning and commonsense or cultural understanding to reading memes, judging spatial relations, counting objects, and inferring emotion.	 *Which country's batik art is shown in each of the two images? What is the tool used to draw the wax patterns in the second image?*
Visual Reasoning	Reasoning here is grounded almost entirely in perception, covering what a shape resembles, what emotion an image conveys, or what atmosphere it establishes. A follow-up question usually asks the model to build on that reading, for example by writing a short passage or a fragment of fiction, so credit on the second step mainly rewards a coherent continuation of a correct first step. Composite entries again combine two sub-types with several images and sub-questions.	 *From left to right these are images one, two, and three. Order them according to the correct procedure for preparing infant formula. If 30 ml of water needs one scoop of powder, roughly how many scoops were used in the end?*
Interaction	The interactive settings are the most distinctive part of the benchmark, testing whether a model can resolve references that are specified visually rather than verbally. Overlaid visual markers use an arrow or a circle drawn onto the image to single out an object, while textual region coordinates place a natural image inside a coordinate frame so that objects are addressed through their coordinates. Multi-turn dialogue additionally requires carrying context across turns. Scenarios span everyday life, work, and gaming.	 *Q1: I want to collect all the game characters in this poster. If one character costs 80 Acquaint Fates to redeem, how many are needed in total? Q2: Among the characters in this image, I already own the one in blue; the rest still need redeeming. Together with the characters from the previous image, how many Acquaint Fates are needed in total?*

continued on next page

Domain	Scope	Example
Content Recognition	Models are asked to identify what a visual input depicts, covering objects, people, landmarks, logos, and attributes, and then to supply an associated fact about that entity, such as its origin, its provenance, or the time, place, and people connected to it. Success therefore rests on grounded recognition together with a broad store of world knowledge, since a substantial share of the entities are long-tail, for instance the banknote of a particular country and its exchange rate against the Chinese yuan.	 *What do the three road signs mean, in order? Which category of primary road traffic signs do these three belong to?*
Knowledge Ability	This domain probes knowledge depth combined with multi-step thinking. Basic knowledge covers general facts that are nonetheless demanding for a human to recall; long-chain reasoning requires quantitative derivations in the style of mathematics and physics; domain knowledge targets specialised sub-fields. Composite entries draw two of these together in a single prompt, pairing several images with several sub-questions.	 *The first image shows the outlines of four continents. What are the names of continents A, B, C, and D? Which of A–D is the second image located in?*

A.2 Prompt Curation and Response Collection

Grounded and compositional prompts. Prompt curation follows the two criteria in Sec. 3.2. Visual necessity concerns whether answering requires evidence from the supplied images, rather than whether a question merely mentions an image. Compositionality concerns the requirements imposed jointly on a response, such as recognising an entity and reasoning about it, or extracting information while respecting an output format. Several requirements can therefore arise from a single image.

Curation probes. Three strong closed-source models probe candidate prompts before annotation. Their answers reveal whether a prompt yields meaningful differences in correctness or requirement coverage. Candidates answered identically or correctly by all three models are removed or rewritten at this stage. The comparison concerns substantive answers rather than wording alone. Prompt revision belongs to curation; subsequent annotation checks return rubrics or scores for correction without changing the prompt.

Responses for annotation. The curation probes are distinct from the response collection used for annotation. After curation, eleven multimodal models each answer every prompt once, giving eleven independently sampled responses per prompt. A strong LLM summarises common answers and points of disagreement to support expert rubric authoring. This summary highlights issues to verify; agreement among models is not itself evidence that an answer is correct. Annotation statistics over these responses are reported in Sec. A.5.

We strictly deduplicate the benchmark against the rubric seed pool to prevent leakage. We commit to publicly releasing the benchmark, datasets, and code to support community-wide research.

A.3 Annotation Protocol and Human Effort

Annotators and responsibilities. Domain experts author the rubrics and resolve revisions, while annotators assign response scores against these rubrics. Independent human judgements provide a further check on the preferences induced by the scores.

Rubric authoring. Experts inspect the visual input, prompt, and model-answer summary to identify the requirements that distinguish responses. They turn these requirements into scored items, specify partial credit where appropriate, and add explicit deductions and caps for relevant errors. The summary provides a starting point; the rubric remains grounded in the input.

Response scoring. Annotators assess each response against the rubric for its prompt and assign a score on the 0–5 scale. Credit follows the stated requirements and partial-credit conditions, while deductions follow their specified triggers and limits. This keeps the scoring basis consistent across responses to the same prompt.

Verification and revision. Quality verification checks rubric faithfulness, individual score correctness, and agreement with independent preferences on sampled response pairs. These are three complementary checking levels. A failed check returns the rubric or score to the original expert for revision; the prompt remains unchanged.

Human effort. Prompt curation and probe review require an average of 12 person-minutes, rubric authoring 23 person-minutes, scoring eleven responses 27 person-minutes, and quality verification 76 person-minutes, totaling 138 person-minutes for one full pass per prompt. These figures exclude image sourcing, model waiting time, training, and additional revision passes.

A.4 Rubric Structure and Scoring Guidelines

The expert rubrics express task-specific requirements through positive credits, partial-credit conditions, and deductions. The following conventions describe rules present in the annotations.

Positive and partial credit. A scored item links an explicit requirement to a credit. Partial-credit schedules distinguish incomplete fulfilment from complete failure. For example, one rubric assigns two points for naming both requested animals and one point for naming either; another assigns three, two, or one point for correctly identifying six, four to five, or two to three requested elements. Such schedules are specified per item rather than obtained by uniformly averaging binary checks.

Deductions and limits. Deductions state both the error and its effect on the score. Local floors and penalty caps limit that effect where specified. For example, a translation rubric subtracts one point per added or omitted sentence until the translation item reaches zero, while a separate hallucination penalty is capped at two points. These limits belong to the relevant item or penalty, rather than implying a common cap for every rubric.

Alternatives and dependencies. Some requirements admit several valid answers: a rubric may accept any one of three listed object indices, so giving all three is unnecessary. Other requirements have a prerequisite: a follow-up explanation may receive credit only when the preceding identification is correct. Explicit alternatives and dependencies define how a response earns credit beyond matching isolated facts.

Task-specific application. Content, reasoning, and format requirements are assessed according to the rubric for the prompt. A valid answer is not required to reproduce the reference wording, and stylistic preferences outside the stated requirements are not additional scoring items. Sec. E.1 illustrates how these conventions translate into scores for complete responses.

A.5 Annotation Statistics

Secs. A.3 and A.4 detail how the human rubrics and absolute scores are produced. This subsection reports the resulting annotations across the six top-level domains of `MM-RubricBench`.

Coverage. Eleven multimodal models were evaluated on the 305 prompts of `MM-RubricBench`, with responses scored by an expert against the rubric for each prompt. Their release dates span early 2024 to early 2025, mixing multiple generations. All predate the systems in Tab. 2 by a considerable margin, so their scores are well below the frontier results reported there.

The scores separate models. Tab. A.5 shows an overall spread of 25.1 points between the strongest and the weakest model, and no model leads on every domain. The strongest model overall ranks only fifth of eleven on visual reasoning, while the model placed third overall leads on recognition by a margin of 2.0 points. A rubric set that merely rewarded fluent prose would not produce this reordering, so the per-domain criteria are doing work beyond a single axis of quality.

Table A.5: **Human annotation results on `MM-RubricBench`.**

Model	Release	Overall	Structured Information	Comprehension	Visual Reasoning	Interaction	Recognition	Knowledge
M-1	2025.01	58.9	55.6	62.4	48.0	67.6	59.0	63.5
M-2	2025.02	56.8	59.7	53.2	59.6	61.6	48.0	55.0
M-3	2024.05	56.8	42.5	58.0	60.0	59.2	70.5	61.0
M-4	2025.01	54.9	62.1	45.6	47.6	63.6	45.5	60.5
M-5	2025.02	52.3	50.9	50.6	57.8	59.0	46.0	48.0
M-6	2025.05	51.2	53.0	41.0	48.6	49.4	60.0	57.5
M-7	2025.01	49.6	58.9	35.4	41.0	47.8	49.5	63.0
M-8	2025.01	46.1	36.0	44.4	44.8	41.6	68.5	52.0
M-9	2025.01	41.9	41.0	48.4	40.2	44.0	62.0	14.5
M-10	2024.03	35.4	34.7	45.2	34.1	37.3	50.0	9.5
M-11	2025.01	33.8	33.1	37.2	32.5	35.5	54.0	10.0

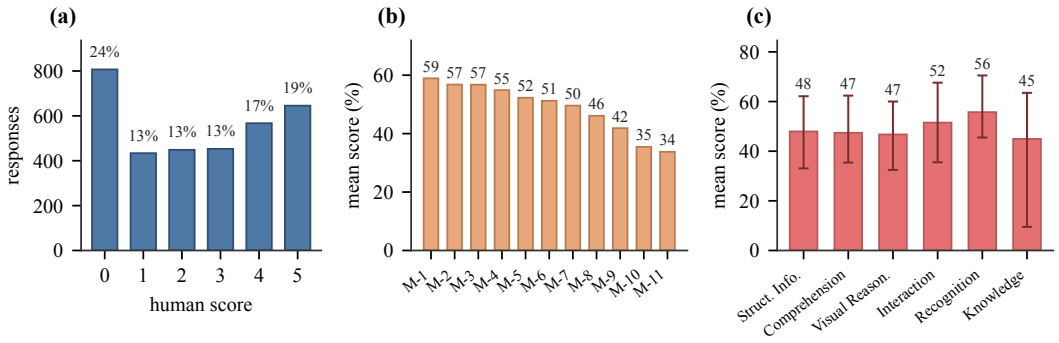


Figure A.1: **Distribution of the human scores on `MM-RubricBench`.** (a) Score distribution on the 0–5 scale. (b) Mean score per model. (c) Mean score per domain across models, with whiskers indicating the minimum and maximum model means.

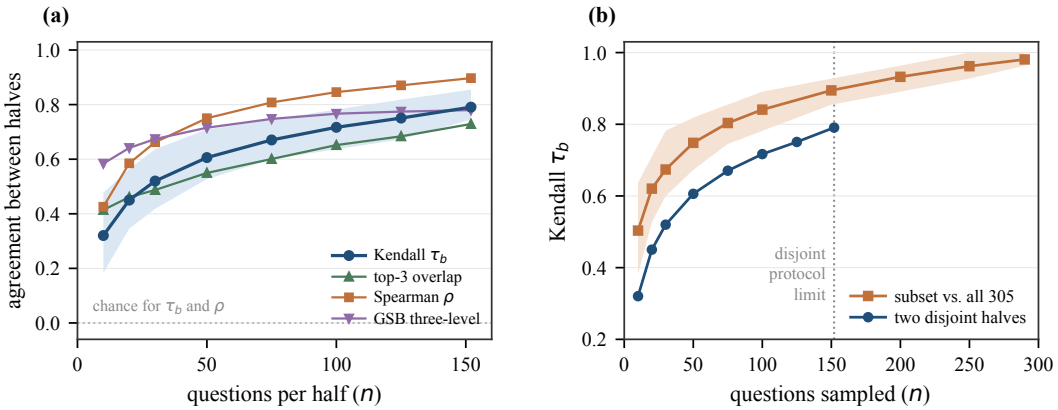


Figure A.2: **How much of the benchmark a stable ranking needs.** (a) Agreement between the rankings induced by two disjoint halves of n prompts each, over 2,000 resamples per point. Kendall τ_b is the headline; Spearman ρ, top-3 overlap and the three-level pairwise variant are drawn alongside it to show the trend does not depend on the choice of measure. The band spans the interquartile range of τ_b. (b) The same statistic under the two protocols. Disjoint halves cap n at 152, while comparing a subset against the ranking on all 305 prompts removes the cap and reveals where the curve saturates.

The scale is used at both ends. Fig. A.1(a) shows that zero credit is the most common outcome at 24% of responses, while full credit accounts for 19% and intermediate scores are evenly distributed at 13 to 17% each. Neither ceiling nor floor effects dominate, making the scores usable as a preference signal. Fig. A.1(c) further shows that every domain spans 25 to 54 points across models, indicating that difficulty depends on the model rather than the domain alone.

A.6 Ranking Stability

A benchmark is only useful if the ordering it produces reflects the models rather than the particular prompts that happened to be collected. Following Fan et al. (2024), we resample the human scores of Sec. A.5: draw $2n$ prompts without replacement, split them into two disjoint halves of n, rank the models by mean score within each half, and measure how far the two rankings agree, averaging 2,000 resamples per point. Because the halves cannot overlap, n is capped at 152, so we also compare a subset of size n against the ranking induced by the full set, which removes the cap but is optimistic by construction.

The ranking is a property of the models. At the largest disjoint size, two halves that share no prompt at all reach a Kendall τ_b of 0.79, agreeing on 89.5% of the model pairs they order, and their top-three sets overlap by 0.73 on average. The leaderboard head is therefore reproducible from an entirely different sample of the same distribution, which is the property that lets the ordering in Tab. 2 be read as a statement about the models.

Small probes are not informative. The same measurement at $n = 10$ yields $\tau_b = 0.32$, and top-three overlap of 0.41, not far above the 0.27 that random sets of three would give. Agreement then climbs steadily, passing 0.61 at $n = 50$ and 0.72 at $n = 100$. Fig. A.2(a) shows the four measures moving together, so the shape is not an artefact of how agreement is defined. Two implications follow. Evaluating on a handful of hand-picked prompts, as is common in qualitative comparisons, cannot separate models reliably. And the gain per prompt is still clearly positive at $n = 152$, meaning the disjoint protocol has not yet exhausted what the benchmark supports.

The full benchmark sits in the saturated regime. Under the subset-versus-full protocol, agreement reaches 0.89 at $n = 150$, 0.93 at $n = 200$, and 0.98 at $n = 290$, where dropping 15 prompts rarely reorders anything. Fig. A.2(b) shows the curve flattening well before the full size, so the benchmark is past the point where more prompts would change the conclusions, yet small enough for every rubric to be written by an expert. The two protocols bracket the achievable agreement: the disjoint one is pessimistic because it halves the evidence on both sides, the subset one optimistic because the subset is contained in its reference.

B SIGHT DETAILS

This section expands the construction and use of SIGHT in Sec. 4: skill bank construction, evaluation blueprints, coverage-driven routing, and rubric instantiation with verification. The prompt excerpts below are shortened English illustrations of the model instructions and inputs, rather than verbatim implementation prompts.

B.1 SKILL BANK CONSTRUCTION

Table B.1: **The skill bank of SIGHT.** Each card summarises one cluster of construction recipes. *Recipes* is the number of expert rubrics the card was induced from, *Text-only* the share of those that came from text-only annotation, *Entropy* the Shannon entropy of the domain distribution of its members, and *Prompts* the number of MM-RubricBench prompts the card was routed to. Every card draws on both text-only and multimodal rubrics, and how often a card is used is unrelated to how large it is.

ID	Skill card	Recipes	Text-only	Entropy	Prompts
S00	Score instructions independently; penalise fabricated claims	167	21.0%	3.46	35
S01	Score subtasks separately; cap cumulative hallucination penalties	62	40.3%	3.21	60
S02	Separate inputs, reasoning and conclusions; penalise hallucinations	77	79.2%	2.61	79
S03	Score content separately from format and style	97	76.3%	2.67	68
S04	Prioritise the conclusion; score intermediate reasoning steps	57	78.9%	2.96	27
S05	Prioritise the conclusion; give supporting facts less weight	93	45.2%	3.37	12
S06	Score each target; assess overall reasoning and format separately	70	80.0%	2.84	7
S07	Score each object independently; penalise invented details	79	70.9%	2.96	61
S08	Grade by number of correct items; penalise incorrect additions	48	60.4%	3.03	50
S09	Grade by number of correct items; check format separately	59	50.8%	3.00	18
S10	Reward correcting false premises and supplying correct facts	61	86.9%	1.16	18
S11	Weight main and supporting points separately; grade answer precision	43	46.5%	3.23	119
S12	Score reasoning and conclusions independently; accept equivalent solutions	53	77.4%	2.60	6
S13	Score conclusions and evidence from the input separately	30	30.0%	2.80	31

Sec. 4.1 builds the bank from a pool of 996 expert-annotated rubrics.

What a recipe preserves. Recipe extraction removes entities, numerical answers, and other instance-specific facts while retaining the relationships between decomposition, credit allocation, and deductions. An evaluative principle must imply a scoring decision: independent judgement of parallel objects specifies how credit is assigned, whereas identifying objects alone only names a task. The fields are concatenated for embedding, keeping the grouping centred on construction choices.

Prompt

Input: A question, its reference answer and visual input when present, and a human-authored rubric.
Extract a reusable construction recipe with five fields: task objective, evaluative logic, decomposition scheme, weight policy, and penalty policy. Remove concrete entities and answer values, but retain operations such as comparison, enumeration, or correction of a false premise. State principles that explain what earns credit or incurs a deduction, rather than merely naming the task. Describe relative priorities and how deductions are applied.

Grouping principles and construction patterns. Near-synonymous principle names are merged by average-linkage clustering of their embeddings using cosine distance. Within each partition, K-means groups full recipe representations by construction pattern. We compare candidate cluster counts using cosine silhouette scores and retain a single cluster when no meaningful split is supported. The two steps thus distinguish variation in evaluation principles from their implementation.

Applicability and demonstrations. Card summarisation receives representative recipes and summaries of neighbouring clusters in the same family. Trigger signals must be observable from the question before its rubric is written, while anti-signals describe superficially similar but unsuitable cases. The discriminator identifies differences from neighbouring cards that must also appear in the construction procedure. Demonstrations are selected from near-centroid recipes: the closest first, followed by recipes from other domains where possible, with remaining slots filled by nearby recipes. This prioritises typical construction patterns while showing their use across subjects.

Evaluative logic families. Recipes are partitioned by the evaluative principle they follow, and the principles are named by the extractor rather than chosen from a fixed list, so the partition axis reflects what annotators actually did. The recurring families are:

- checking several explicit instructions one by one;
- enumerating the target points in full and grading by how many are hit;
- judging several parallel objects independently of one another;
- requiring a conclusion to be supported by verifiable facts or a visible process;
- placing the leading credit on the core derivation;
- separating primary from secondary points, with the core conclusion carrying the leading credit;
- grading down for imprecise expression instead of marking it wrong.

The resulting bank contains the 14 cards reported in Tab. B.1.

Cards are organised by construction, not by subject matter. Two columns of Tab. B.1 support this view. Cross-domain entropy averages 2.85 and is at least 2.6 for thirteen of the fourteen cards, showing that cards typically draw on rubrics from multiple domains rather than capturing what a single domain looks like. The exception is S10 at 1.16, whose recipes concentrate on false-premise prompts, a construction tied to a narrower prompt design. The text-only share ranges from 21.0% to 86.9%, versus 57.8% overall. Thus, every card combines recipes from text-only and multimodal annotations rather than relying exclusively on one modality.

Card support and usage. All fourteen cards are routed to at least one prompt. Usage does not simply track support size: S00 is the largest card at 167 recipes yet is selected for only 35 prompts, whereas S11 rests on 43 recipes and is selected for 119. A larger source cluster therefore need not correspond to more frequent selection on `MM-RubricBench`.

B.2 Evaluation Blueprint Construction

Separating facts from requirements. The blueprint in Eq. (3) records both what a response must satisfy and the facts needed to judge it. Evidence entries retain concrete values, identities, and relations; positive atoms describe the corresponding evaluation requirements. For example, a requirement may ask for a difference between two quantities, while the evidence records their values and the correct difference. Evidence is carried forward into rubric background knowledge rather than treated as a separate construction requirement during routing.

Importance and active errors. Core requirements include the directly requested answers and indispensable prerequisites, such as identifying an object before answering a question about its properties. Secondary requirements are restricted to explicitly requested supporting content. Additional facts in a reference answer may remain evidence without becoming credited requirements. Active errors are kept separate from omissions, so failing to earn a criterion's credit does not by itself justify a further deduction. The following excerpt illustrates these boundaries.

Prompt

Input: The question, visual input, and reference answer.
Construct an evaluation blueprint, not a scoring rubric. Record concrete facts as evidence. List positive requirements with core or secondary importance and image, reference, or instruction grounding. Treat directly requested answers and necessary identification prerequisites as core. Include secondary content only when requested; keep unrequested reference details as background evidence. List active errors, such as fabrication or accepting a false premise, separately. Do not turn an omitted positive requirement into an active error. Return the three groups without assigning rubric credits.

B.3 Coverage-driven Skill Routing

Candidate formation and fitness. Each positive or penalty atom retrieves cards independently. Candidate lists are merged one rank at a time across atoms, skipping duplicates until the pool is full, preventing any single atom from dominating the pool. The LLM receives the task summary and routable atoms with candidate applicability descriptions, including evaluative logic, trigger and

anti-signals, and discriminators. Its scores measure construction fitness rather than topic similarity; full procedures and human demonstrations are supplied later during instantiation.

Prompt

Input: The blueprint's task summary, positive requirements and active errors, and candidate skill applicability descriptions.
For each requirement–skill pair, judge how useful the described construction approach is for writing a clear, independently checkable scoring item. Score 0 for irrelevant, 1 for partially related, 2 for applicable with minor adaptation, and 3 for directly suited to the required construction. Judge evaluation logic rather than subject matter. Return the requirement–skill fitness scores.

Residual selection and coverage. In Eq. (5), a card contributes only where its fitness exceeds the best score already attained by the selected set. Equal or lower fitness adds no gain. Coverage is a separate thresholded measure: an atom is covered when at least one selected card scores 2 or above. Reaching this threshold for all atoms ends selection, even if a further card could improve some scores from 2 to 3. Otherwise, selection continues subject to the gain tolerance, context budget, and card limit given in Sec. C.2.

Routing statistics. Over the 305 prompts of `MM-RubricBench`, routing selects one card for 100 prompts, two for 133, three for 63, and four for 9, averaging 1.94. Averaged over prompts, 95.0% of positive and penalty atoms reach the coverage threshold, and 246 prompts (80.7%) have all such atoms covered. Core coverage is computed over core positive atoms only: it reaches 98.8%, with 294 prompts (96.4%) fully covered. These are unweighted fractions within each prompt, averaged across prompts; evidence atoms are excluded. They measure the availability of suitable construction skills, not the correctness of the final rubric.

B.4 Rubric Instantiation and Verification

Applying skills to the current blueprint. Instantiation receives the complete textual blueprint and the selected cards' construction procedures, weight and penalty policies, and human demonstrations. Demonstrations illustrate how a construction is expressed, while the current blueprint supplies the instance-specific facts and requirements. A procedure step with no corresponding requirement is skipped rather than turned into an additional scoring item. Multiple atoms may support one criterion, so the mapping need not be one-to-one.

Prompt

Input: The complete evaluation blueprint and selected skills with their procedures, weight and penalty policies, and human question–rubric demonstrations.
Use the skills to turn evidence into background knowledge, positive requirements into credited criteria, and active errors into deductions. Skip procedure steps unsupported by the current requirements. Cover every core positive requirement and link each scoring item to its source atoms. Allocate credits according to importance, summing to $W = 5$. State checkable conditions, acceptable equivalent answers, and dependencies where needed. Do not duplicate criteria or deduct again for a missing positive requirement. Return the background, criteria, and deductions.

Traceability and structural checks. The provenance links in Sec. 4.2 allow us to check whether each scoring item refers to a blueprint atom and whether every core positive atom is represented. We also check the total credit, permitted weights and item counts, and whether dependencies refer to existing criteria. Embedding similarity flags suspected duplication between criteria and overlap between a criterion and a deduction. These checks identify structural inconsistencies and possible redundancy; they do not independently establish factual correctness or semantic equivalence.

Targeted repair. When checks fail, the LLM receives the detected problems, the blueprint, and the previous rubric, with instructions to correct those problems while preserving unaffected content. The revised rubric is checked in the same way and retained only if the number of violations does

not increase. A single repair attempt limits revision to the identified issues rather than repeatedly regenerating the rubric.

Generation statistics. All 305 prompts produce a rubric whose credits sum to $W = 5$. Rubrics average 3.30 criteria, 1.88 deductions, and 3.41 background entries. The structural checks pass on the first attempt for 91.8% of prompts; the remaining 8.2% trigger the single targeted repair. After repair, one prompt retains a flagged overlap between a deduction and a positive criterion, so passing the score constraints does not imply that every check passes.

C Experiment Details

C.1 Model Ranking Setup

This section details the protocol behind Tab. 2.

Evaluated models. We evaluate eighteen recent MLLMs, spanning proprietary and open-weight families. Within each family, we deliberately include both the flagship and its smaller sibling, such as Pro versus Flash or Lite variants, because these pairs have a known capability order and therefore act as built-in consistency checks on the evaluation signal. All models are queried through their official interfaces under the provider's default decoding configuration, with extended thinking disabled wherever the provider exposes that switch and left at its default where it cannot be turned off.

Response sampling and scoring. For every prompt, we sample one response per model, and score it against the expert rubric written for that prompt. Scoring uses the same judge and the same rubric-conditioned protocol as the *Human rubric* row of Tab. 3, so the ranking measures model capability rather than a difference in judging procedure. Because a rubric fixes the total credit at $\sum_j w_j = W$, a raw rubric score is comparable across prompts only after normalisation; we therefore report $100 \cdot \mathrm{score}(y)/W$, which places all prompts on a common percentage scale.

Aggregation. Per-model scores are averaged over prompts to give the overall column, and over the prompts of each top-level domain to give the six per-domain columns. Domains follow the two-level taxonomy of Sec. 3.3; averaging is unweighted over prompts within a domain, so a domain's score is not affected by how many prompts the other domains contribute.

What the consistency checks test. The release-order check in Sec. 5.1 covers eleven consecutive generation steps within each family tier, such as Gemini 2.5 Pro to Gemini 3 Pro, plus three same-generation flagship–smaller-variant pairs, yielding fourteen comparisons. Each comparison tests only an ordinal relation: the newer or larger variant should score higher than the older or smaller one, without requiring a specific margin. This is a deliberately weak assertion: if rubric-conditioned scores were driven by judging noise rather than response quality, the ordering should fail in one of the comparisons. The rubrics also predate all evaluated models, ruling out tuning against them.

C.2 Alignment Protocols and Metrics

This section details the skill bank, generation configuration, counterparts, and metrics of Tab. 3.

Skill bank. The bank used throughout our experiments is extracted from 996 human rubrics, combining the open-source textual rubrics from SedarEval (Fan et al., 2024) with in-house multimodal rubrics, 576 (57.8%) of them text-only and 420 multimodal. These rubrics contain 4,090 scoring items in total, 4.11 per rubric on average, of which 3.29 are credited criteria and 0.81 are deductions. After naming the evaluative principle of each recipe and merging near-synonymous ones, the recipes fall into 7 recurring logic families, and clustering within each family produces the 14 skill cards that form $\mathcal{S}$. Every card draws on both text-only and multimodal recipes, supporting the reuse of construction logic across modalities. Per-card statistics are reported in Sec. B.1.

Routing and generation. The embedding stage retrieves the five nearest cards for each atom and caps the merged candidate pool at eight, and the LLM scores each atom-card pair on a 0–3 scale, from irrelevant to highly applicable. Atom weights ω_a are 1.0 for core requirements, 0.5 for secondary requirements, and 0.7 for penalty requirements. The context budget C is 6,000 characters and at most four cards may be selected; greedy selection also stops once every atom reaches a coverage level of 2.0 or the largest remaining marginal gain falls below 0.3. Under this configuration routing selects 1.94 skills per prompt on average.

Judging protocols. Every protocol produces one integer score in $[0, W]$ for each response, and all protocols share the same judge, the same decoding configuration, and the same inputs, so the comparison isolates the scoring protocol itself. We strictly follow the descriptions in the original papers when implementing each counterpart for fairness.

- **Vanilla LLM-as-a-Judge.** The judge sees the prompt, the image, the reference answer, and the response, and returns a score directly. No rubric is produced, so this isolates how much of the agreement is attributable to rubrics at all.
- **Fixed rubric.** The vanilla inputs are paired with one fixed scoring prompt that serves as a uniform rubric for every instance, shown below. It defines the six score levels through general notions of helpfulness, harmlessness, and honesty, without any instance-specific criteria.

Prompt

[Is the given response reflective of helpfulness, harmlessness, and honesty?]

- **Score 0:** The response is completely unhelpful, severely harmful, or entirely dishonest, providing no value to the user.
- **Score 1:** The response is misleading, harmful, or dishonest, negatively impacting the user.
- **Score 2:** The response tries to be helpful and honest but might mislead or be somewhat harmful, requiring further clarification or caution.
- **Score 3:** The response is honest and aims to be helpful, but could improve in avoiding harm and fully ensuring accuracy.
- **Score 4:** The response is helpful, honest, and largely avoids harm, providing accurate information with some ethical considerations.
- **Score 5:** The response is outstanding in its helpfulness, honesty, and harmlessness, offering thorough, accurate advice with proactive ethical safeguards.

- **CheckEval** (Lee et al., 2025). A checklist of binary questions is built once per dataset in three stages: dimensions and sub-dimensions are defined manually, questions are expanded from human seeds through diversification and elaboration and then filtered, and evaluation answers those questions with yes or no. Following the paper, questions are grouped by sub-dimension and asked in batches rather than one call each, and the final score is the *uniformly weighted* proportion of yes answers; the paper reports no reliable gains from learning per-question weights, so we do not weight them. The checklist is fixed across prompts and never conditioned on the prompt being judged, which is the property that distinguishes CheckEval from per-instance checklist methods.
- **OpenRubrics** (Liu et al., 2026). A rubric is generated for each prompt and then used to score every response to that prompt. The rubric is prompt-specific but written without reference to any expert rubric, so this counterpart measures what a model produces when it is free to invent criteria.
- **RubricRAG** (Dhole & Agichtein, 2026). Expert rubrics from related queries are retrieved and supplied as in-context guidance for writing the rubric of the current prompt. Retrieval is keyed on surface similarity of the query, which makes this the strongest counterpart here and the most direct contrast with `SIGHT`: both draw on expert rubrics, but one transfers their content and the other their construction logic.
- **Human rubric (ceiling).** Responses are scored against the expert rubric authored for that prompt. This is not a competing generator but an upper reference: it shares the judge and the scoring prompt with the rubric-based protocols above, so the residual gap to it measures rubric quality rather than judge capability.

Applied adaptations. To adapt the text-oriented counterparts to the multimodal setting of `MM-RubricBench` while maintaining a consistent and fair comparison, we make three adjustments that are applied identically to all protocols. First, prompts are presented in the benchmark language and paired with the corresponding visual inputs, whereas CheckEval and OpenRubrics were originally designed for text-only settings. Second, protocols whose native outputs are not integers in $[0, W]$ are mapped to this common score range. For CheckEval, the yes-ratio $p \in [0, 1]$ is converted to $\mathrm{round}(p \cdot W)$, while the unrounded ratio is retained to recompute rank-based metrics, which are invariant to monotone rescaling. Third, CheckEval's judging instruction treats a question as satisfied when the current prompt does not impose the corresponding constraint. This adapta-

tion is necessary because its checklist is defined at the dataset level, whereas MM-RubricBench is instruction-specific: without this convention, questions irrelevant to a given prompt would be answered no, uniformly depressing scores and thereby understating the method's performance.

Notation. Let $\mathcal{P}$ be the prompt set, $\mathcal{R}_p$ the 11 independently sampled responses for prompt p, and $h(p,r)$ and $s(p,r)$ the human and protocol scores, respectively. Let $N = \sum_p |\mathcal{R}_p|$ be the total number of scored responses.

Sample-level metrics. These compare scores to human ratings response by response. Mean absolute error and adjacent-one accuracy are

$$\mathrm{MAE} = \frac{1}{N}\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_p}\big|s(p,r)-h(p,r)\big|, \qquad \mathrm{Adj1} = \frac{1}{N}\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_p}\mathbf{1}\Big[\big|s(p,r)-h(p,r)\big|\le 1\Big]. \tag{7}$$

MAE is reported because it stays in the unit of the rating scale, and $\mathrm{Adj1}$ because a one-point disagreement on a W-point scale is within the tolerance of human annotation itself, so exact-match accuracy would penalise differences that annotators do not consider substantive.

Preference-level metrics. These ignore absolute calibration and ask only whether a protocol orders responses as humans do. For a prompt p and a pair $r, r' \in \mathcal{R}_p$, let

$$\delta(a,b) = \begin{cases} +1, & a > b, \\ 0, & a = b, \\ -1, & a < b. \end{cases} \tag{8}$$

Pairwise accuracy is the fraction of within-prompt pairs on which the two sign patterns coincide,

$$\mathrm{PairAcc} = \frac{1}{M}\sum_{p\in\mathcal{P}}\sum_{\{r,r'\}\subseteq\mathcal{R}_p}\mathbf{1}\Big[\delta\big(h(p,r),h(p,r')\big)=\delta\big(s(p,r),s(p,r')\big)\Big],\ M=\sum_{p\in\mathcal{P}}\binom{|\mathcal{R}_p|}{2}. \tag{9}$$

Rank correlations are computed *per prompt* and then averaged over prompts,

$$\bar{\tau}_b = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\tau_b\big(\mathbf{h}_p,\mathbf{s}_p\big), \qquad \bar{\rho} = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\rho\big(\mathbf{h}_p,\mathbf{s}_p\big), \tag{10}$$

where $\mathbf{h}_p$ and $\mathbf{s}_p$ collect the human and protocol scores over $\mathcal{R}_p$. Averaging per prompt rather than pooling all responses into one correlation is deliberate: a pooled coefficient is inflated by between-prompt variation in difficulty, which a rubric is not asked to predict, whereas the per-prompt coefficient only measures the ordering of responses to the same prompt. We use the τ_b variant because scores are integers on a bounded scale and ties are therefore frequent; τ_b applies the tie correction to both margins, while τ_a would treat every tie as a violation.

C.3 Additional Downstream Experiments

We extend the multimodal reasoning experiment in Sec. 5.3 to instruction following and agentic rewarding, using SIGHT rubrics as RL rewards in both settings.

Multimodal instruction following (IF). We train on the open-source preference data of MM-IFEngine (Ding et al., 2025), using its positive samples as reference answers, and additionally compare SFT and DPO. To probe generalisation, we evaluate on multimodal IF (MM-IFEval and MIA-Bench), text-only IF (IFEval), and open-ended VQA (MM-Vet v2 and KOFFVQA). As shown in Tab. C.1, SFT improves multimodal IF but lowers the overall average through weaker generalisation. In contrast, SIGHT achieves the best average (76.4, +4.3) and improves all six metrics over the base policy, indicating better constraint satisfaction without sacrificing general ability.

Table C.1: **Downstream RL on multimodal IF.** SIGHT achieves the best average while improving both IF and open-ended VQA over the base policy.

Method	Multimodal IF (ID)		Text-only IF (OOD)		Open-ended VQA		Avg.	Δ
	MM-IFEval	MIA-Bench	IFEval (prompt)	IFEval (inst.)	MM-Vet v2	KOFFVQA		
Qwen3-VL-4B-Instruct	60.8	85.4	81.9	87.7	65.4	64.0	72.1	0
SFT	64.6	**88.6**	77.1	83.5	62.9	60.8	71.4	-0.6
DPO	64.4	87.1	85.2	89.8	61.0	63.3	72.7	+0.6
RL w/ Correctness LLM-judge	68.9	85.7	84.7	89.3	64.8	64.4	74.2	+2.1
RL w/ RubricRAG 2026	67.4	87.2	85.6	89.9	66.6	66.8	75.1	+3.1
RL w/ **SIGHT**	**69.5**	88.2	**87.0**	**91.5**	**67.6**	**67.2**	**76.4**	+4.3

Multimodal agentic rewarding. We train on ARM-Thinker (Ding et al., 2026) data and evaluate on three multimodal reward benchmarks against existing methods. As shown in Tab. C.2, SIGHT achieves the best results across all three benchmarks, raising the average from 67.9 to 76.0 (+8.0), compared with +3.5 and +4.7 for outcome-only and outcome-plus-tool rewards.

Table C.2: **Downstream RL on multimodal agentic rewarding.** SIGHT achieves the best results across all three reward benchmarks, with an average gain of 8.0 points.

Method	VL-RewardBench					Multimodal RewardBench	MM-RLHF RewardBench	Avg.	Δ
	Hallu.	Reason.	General	Overall	Macro Avg.				
Qwen3-VL-4B-Thinking	69.4	61.5	45.9	64.0	58.9	68.8	71.0	67.9	0
RL w/ Outcome reward	72.1	66.6	48.6	67.3	62.4	73.3	73.8	71.5	+3.5
RL w/ Outcome + tool reward	71.7	69.4	53.6	68.5	64.9	72.8	76.6	72.6	+4.7
RL w/ RubricRAG 2026	73.4	67.8	53.6	69.1	65.0	77.3	77.2	74.5	+6.6
RL w/ **SIGHT**	**74.1**	**70.0**	**55.3**	**70.3**	**66.5**	**78.3**	**79.3**	**76.0**	+8.0

Together with the reasoning results in Tab. 4, these experiments show that the benefits of SIGHT rewards extend across reasoning, instruction following, and agentic rewarding.

C.4 Component Ablations

Comparison. Tab. C.3 compares SIGHT with a fixed rubric and five ablations that remove skill guidance, construction procedures, logic-guided grouping, residual routing, or targeted repair. We report the five alignment metrics defined in Sec. C.2. The full method achieves the best values across all five metrics, with ties in pairwise accuracy and Kendall's τ_b at the reported precision. Compared with the fixed rubric, it reduces MAE from 1.42 to 0.90 and raises pairwise accuracy from 0.54 to 0.73.

Table C.3: **Component ablations of SIGHT on MM-RubricBench.** Metrics follow Sec. C.2. Bold marks the best values, including ties at the reported precision.

Variant	Sample-level		Preference-level		
	MAE ↓	Adj1 Acc. ↑	Pairwise Acc. ↑	Kendall τ_b ↑	Spearman ρ ↑
Fixed rubric	1.42	0.60	0.54	0.44	0.49
SIGHT	**0.90**	**0.77**	**0.73**	**0.66**	**0.69**
w/o skill guidance	1.18	0.65	0.62	0.58	0.60
w/o construction procedures	1.05	0.68	0.65	0.61	0.64
w/o logic-guided grouping	1.20	0.63	0.60	0.59	0.60
w/o residual routing	0.95	0.71	0.70	0.61	0.64
w/o targeted repair	0.92	0.75	**0.73**	**0.66**	0.68

Expert guidance and construction procedures. Removing skill guidance raises MAE by 0.28, while pairwise accuracy falls by 0.11 and Spearman's ρ by 0.09. Removing construction procedures also worsens all five metrics, yielding an MAE of 1.05 and pairwise accuracy of 0.65. These results support transferring expert construction knowledge, with explicit procedures contributing to both score agreement and preference alignment.

Logic-guided grouping. Replacing the two-stage grouping of Sec. 4.1 with a single clustering pass over the recipe representations gives the highest MAE, 1.20, and the lowest pairwise accuracy, 0.60, among the component ablations. Without first partitioning recipes by their evaluative principle, one clustering pass is dominated by surface attributes such as modality and task category, so a cluster describes what a rubric is about rather than how it is built. The cards summarised from such clusters carry less constructional content, which is what the routing stage needs. Grouping by logic before clustering by its implementation is therefore what makes a card a construction rather than a topic.

Residual routing. Without residual routing, pairwise accuracy decreases from 0.73 to 0.70, while Kendall's τ_b and Spearman's ρ each decrease by 0.05. The MAE increase is smaller, from 0.90 to 0.95. This pattern is consistent with complementary skill selection helping preserve response ordering in addition to improving absolute score agreement, supporting the routing objective in Eq. (4).

Targeted repair. Removing targeted repair leaves pairwise accuracy and Kendall's τ_b unchanged at two decimal places, but increases MAE from 0.90 to 0.92, lowers adjacent-one accuracy from 0.77 to 0.75, and lowers Spearman's ρ from 0.69 to 0.68. Its aggregate contribution is therefore smaller than that of skill guidance or construction procedures in this comparison. The main alignment gains remain without repair, while repair provides modest improvements on the remaining metrics.

D TRAINING DETAILS

This section describes how SIGHT rubrics are used for RL in multimodal reasoning, instruction following, and agentic rewarding tasks. We detail the policy and training infrastructure in Sec. D.1, training-data preparation in Sec. D.2, reward acquisition in Sec. D.3, optimization in Sec. D.4, and rollout procedure in Sec. D.5.

D.1 MODELS AND TRAINING INFRASTRUCTURE

Policy models. The reasoning and IF training use `Qwen3-VL-4B-Instruct`, whereas the agentic configuration uses `Qwen3-VL-4B-Thinking` (Bai et al., 2025). The latter supports extended reasoning and image-tool interaction before producing a final preference judgment. The policy receives the task input and its images through the model's chat template.

Training framework. We build the RL pipeline on Miles[1], using Megatron-LM[2] (Shoeybi et al., 2019) as the training backend and SGLang[3] as the rollout inference engine. Ray coordinates policy training, rollout generation, and reward requests. A separately served judge returns a scalar evaluation of each sampled response under its instance-specific rubric.

Parallelism and memory. The default deployment uses four nodes with eight GPUs per node and colocates policy training and rollout generation on these 32 GPUs. Megatron uses tensor parallelism of four, pipeline and context parallelism of one, and sequence parallelism, giving a data-parallel degree of eight. Rollout engines use one GPU each. Training uses BF16 and a distributed optimizer, with dynamic microbatching controlled by a per-GPU token budget of 4,096.

D.2 TRAINING DATA AND RUBRIC PREPARATION

Multimodal reasoning. The reasoning configuration uses VisualWebInstruct-verified (Jia et al., 2025). Each example provides an image-conditioned question and a reference answer. The reference supports both rubric construction and subsequent reward evaluation. The reward evaluates the cleaned full response rather than reducing every example to a binary comparison of extracted final answers, allowing the rubric to distinguish a correct conclusion from the reasoning and evidence needed to support it.

Multimodal instruction following. The instruction-following configuration uses the positive examples from the MM-IFEngine preference data (Ding et al., 2025). The positive response serves as the reference, while the original user instruction retains its explicit constraints. The resulting rubric can assign separate credit to task content and to requirements such as response organization, terminology, or output restrictions.

Multimodal agentic tasks. The agentic configuration uses ARM-Thinker data (Ding et al., 2026). The policy observes an image, the task context, and two candidate answers, and may use an image-zoom tool before deciding which answer is better. The data conversion preserves the original system/user instructions and tool schema, including the output-format instructions supplied to the policy. The training label is the preferred candidate index, stored as `1` or `2`.

Training rubric preparation. Before RL, we apply SIGHT with the existing construction Skill bank to each training instance x. We extract a blueprint B_x from the question q_x, images v_x, and reference, then generate a rubric $\widehat{R}_x$ from the blueprint and routed Skills $S_x^\star$. The construction scripts default to `Gemini-2.5-Pro`, with hybrid routing, at most four Skills, and two demonstrations per Skill. The stored rubric is reused throughout training.

Generation targets one to five positive criteria with integer weights totaling five and at most three penalties, with task-dependent prerequisites and deduction caps. Automated checks inspect weights, coverage, provenance, and dependencies, followed by a bounded repair pass when needed.

[1] https://github.com/radixark/miles
[2] https://github.com/nvidia/megatron-lm
[3] https://github.com/sgl-project/sglang

Table D.1: **Training configuration of the Skill-rubric RL recipe.** MMIF denotes instruction following, VWI denotes VisualWebInstruct-verified reasoning, and ARM denotes agentic preference judgment. Values are launch-configuration defaults, including the framework defaults used by these launchers. Prompt counts, response counts, and token budgets are reported separately.

Setting	**MMIF**	**VWI**	**ARM**
Policy, data, and sampling			
Qwen3-VL-4B variant	Instruct	Instruct	Thinking
Training data	MM-IFEngine	VWI-verified	ARM-Thinker
Prompts per rollout, B	128	128	128
Responses per prompt, G	16	8	8
Responses per rollout / global batch	2,048	1,024	1,024
Optimizer updates per full rollout	1	1	1
Configured dataset epochs	10	10	10
Maximum prompt tokens	2,048	4,096	4,096
Response / trajectory token budget	1,024	16,384	8,192
Sampling temperature	0.8	0.8	1.0
Top-p / top-k		1.0 / disabled	
Interaction turn limit	–	–	5
Image-zoom upscale dimension	–	–	448
Shared optimization and reward settings			
Policy optimization		GRPO	
Optimizer		Adam	
Learning rate / schedule		10^{-6} / constant	
Warmup iterations		5	
Adam (β_1, β_2) / weight decay		$(0.9, 0.98)$ / 0.1	
Gradient-norm clipping		1.0	
Policy-ratio clipping interval		$[0.8, 1.28]$	
Group reward standardization		Enabled, $\epsilon = 10^{-6}$	
Extra batch advantage whitening		Disabled	
KL shaping / KL loss / entropy coefficient		0 / 0 / 0	
Train–inference importance correction		Disabled	
Scalar training reward		Rubric-judge integer score / 5	
Separate format / tool bonus		None / none	
Shared systems configuration			
Training backend / rollout engine		Megatron-LM / SGLang	
Policy training and rollout GPUs		$4 \times 8 = 32$, colocated	
Training TP / PP / CP / DP		4 / 1 / 1 / 8	
GPUs per rollout engine		1	
Precision / distributed optimizer		BF16 / enabled	
Dynamic microbatch token budget per GPU		4,096	
SGLang static memory fraction		0.75	

D.3 Training Reward Acquisition

A shared rubric-conditioned reward. Let $a_x^\star$ denote the training reference and $H_d(y)$ the task-dependent preprocessing of a sampled response or trajectory y. For all three tasks, a judge J computes the rubric-conditioned reward as

$$s_x(y) = \mathrm{Parse}_{0:5}\Big(J\big(q_x, a_x^\star, \widehat{R}_x, H_d(y)\big)\Big), \qquad r_x(y) = \frac{s_x(y)}{5}. \tag{11}$$

The judge sums criterion scores and subtracts triggered penalties, following the rubric's prerequisites, partial-credit rules, and deduction caps. It maps the result to an integer $s_x(y) \in \{0, 1, 2, 3, 4, 5\}$, which the reward adapter parses and normalizes by five.

Judge deployment. We serve `Qwen2.5-VL-72B-Instruct` through a service pool. The judge receives only text: the question, reference, rendered rubric, and processed policy response. It checks visual claims against the rubric's background knowledge and then consults the reference answer.

Judge decoding uses temperature 0.2, top-p 0.95, and a maximum of 4,096 output tokens. The prompt requests criterion-level reasons followed by `Score: X` (Sec. D.6). The adapter allows up to three scoring attempts on call or parsing failures, with additional request-level retries by the client, and returns zero if all attempts fail. The parser extracts a valid 0–5 score, falling back to an integer in the response text.

Task-dependent response processing. For reasoning and instruction following, H_d removes chat-template special tokens and normalizes redundant whitespace while preserving the full response. It does not replace the response by an extracted short answer. For ARM, H_d extracts the last assistant turn and removes serialization and reasoning/answer delimiters while retaining the reasoning text and final judgment. Intermediate tool observations are not ordinarily included in this final-turn judge input. The ARM adapter also removes the appended strict output-format block from the question sent to the judge, although that block remains in the policy input. It therefore does not implement the old ARM reward's hard output-format gate.

References and reward scope. References are available during both training-time rubric preparation and reward evaluation; this is not a reference-free reward setting. MMIF and reasoning supply the reference answer text to the judge, while ARM supplies the preferred candidate index. No policy gradients pass through the construction models or judge. In all three settings, the optimization signal is the normalized scalar in Eq. (11), not the judge's textual rationale.

D.4 Optimization and Rollout Configuration

Group-relative policy optimization. We use GRPO (Shao et al., 2024). Each rollout batch contains $B = 128$ prompts. For each prompt x, the policy samples a group of G responses, with $G = 16$ for instruction following and $G = 8$ for reasoning and agentic tasks. Rewards are centered and standardized within each group:

$$\bar{r}_x = \frac{1}{G}\sum_{i=1}^{G} r_x(y_i), \qquad A_{x,i} = \frac{r_x(y_i) - \bar{r}_x}{\sigma_x + 10^{-6}}, \tag{12}$$

where σ_x is the sample standard deviation of the group's rewards. There is no additional cross-batch advantage whitening. If all responses receive the same reward, the group's advantages are zero.

Let $w_{i,t}(\theta) = \pi_\theta(y_{i,t} \mid x, y_{i,<t}) / \pi_{\text{old}}(y_{i,t} \mid x, y_{i,<t})$ be the token-level policy ratio, $m_{i,t}$ the loss mask, and $L_i = \sum_t m_{i,t}$ the number of trainable tokens in response i. The clipped policy objective is

$$\mathcal{J}(\theta) = \frac{1}{BG}\sum_{x\in\mathcal{B}}\sum_{i=1}^{G}\frac{1}{\max(L_i,1)}\sum_t m_{i,t}\,\min\Big(w_{i,t}(\theta)A_{x,i},\ \text{clip}\big(w_{i,t}(\theta),0.8,1.28\big)A_{x,i}\Big). \tag{13}$$

The lower and upper clipping parameters are 0.2 and 0.28. The same sequence-level advantage is applied to the generated assistant tokens. Old-policy log-probabilities are computed by the training actor; the configurations do not enable an additional train–inference importance-sampling correction. Both KL reward shaping and an explicit KL loss are disabled, and the entropy coefficient is zero. GRPO does not use a learned value critic in these configurations.

Optimizer and update size. The Adam optimizer uses a learning rate of 10^{-6}, a constant schedule with five warmup iterations, $\beta_1 = 0.9$, $\beta_2 = 0.98$, and weight decay 0.1. The gradient-norm clipping threshold is 1.0. The global training batch equals the complete rollout batch, BG: 2,048 responses for MMIF and 1,024 for each of the other tasks. A complete rollout batch therefore produces one optimizer update, accumulated over dynamically constructed microbatches. Tab. D.1 separates these shared choices from task-specific sequence limits and sampling settings.

Single-turn generation. MMIF and reasoning use temperature 0.8, top-$p = 1$, and disabled top-k filtering. Their maximum prompt lengths are 2,048 and 4,096 tokens, respectively; their maximum response lengths are 1,024 and 16,384 tokens. The longer reasoning budget allows extended derivations without imposing that verbosity on instruction-following tasks.

Agentic generation and masking. ARM uses temperature 1.0 and a multi-turn environment with an `image_zoom_in_tool`. The environment configuration permits up to five interaction turns and

Algorithm 1: Policy optimization with preconstructed SIGHT rubrics.

Input : Training instances $\mathcal{D}$; construction Skill bank S; initial policy π_θ; reward judge J; batch size B; group size G.
Output: An RL-updated policy π_θ.
foreach $x \in \mathcal{D}$ **do**
  Construct blueprint B_x from the question, images, and available reference
  Route construction Skills $S_x^\star$ and generate $\widehat{R}_x$
  Store the policy input, images, reference, and rubric as a training record
end
Form $\mathcal{D}_{\text{eff}}$ after conversion and prompt-length filtering
foreach *rollout iteration in the configured training schedule* **do**
  Sample a shuffled prompt batch $\mathcal{B} \subset \mathcal{D}_{\text{eff}}$ of size B
  Set $\pi_{\text{old}} \leftarrow \pi_\theta$ and sample G responses or tool-interaction trajectories per prompt
  Mask prompt and tool-observation tokens from the policy loss
  Score each $H_d(y_i)$ using its stored reference and $\widehat{R}_x$; set $r_{x,i} = s_{x,i}/5$
  Compute group-relative advantages using Eq. (12)
  Perform one optimizer update using Eq. (13)
  Synchronize the rollout policy and save a checkpoint when scheduled
end

sets the maximum upscale dimension to 448. Tool observations are appended to the context and can guide subsequent assistant decisions, but their tokens have loss mask zero. Generated assistant tokens, including tool-call text, remain trainable. The terminal rubric reward supplies their group-relative learning signal without an extra payment for invoking a tool.

The ARM launcher's 8,192-token limit has a different interpretation from a single-turn response limit. In the multi-turn implementation, the initial prompt length is subtracted from this budget, and subsequent assistant and observation tokens both consume it. It is therefore a shared trajectory budget, not an allowance of 8,192 new assistant tokens at every turn. The prompt filter remains 4,096 tokens.

D.5 Training Procedure and Checkpointing

Algorithm 1 summarizes the separation between rubric preparation and policy optimization. The Skill bank and constructed training rubrics are fixed during the RL loop. Reusing these rubrics avoids repeated blueprint extraction, Skill routing, and rubric generation for the same prompt; reward-judge calls are still required for new policy responses. With B prompts and G sampled responses, a normal update scores BG responses, excluding retries. We do not assume this scoring cost is negligible relative to rollout generation.

The launch configurations specify ten dataset epochs and save checkpoints every 20 rollout iterations, with additional saves at epoch boundaries and at training completion. They do not enable periodic in-training benchmark evaluation. The epoch-based schedule is distinct from a claim that a particular reported model is the best validation checkpoint. For N_{eff} retained prompts, the current scheduling code gives

$$K_{\text{epoch}} = \left\lfloor \frac{N_{\text{eff}}}{128} \right\rfloor, \qquad K_{\text{total}} = 10K_{\text{epoch}}. \tag{14}$$

A resumed run continues from its saved iteration. Thus, completed update counts and selected evaluation checkpoints are run-level quantities, not constants inferred from the ten-epoch launcher default. TensorBoard records training statistics, and the reward adapters retain the judge output, parsed score, and timing information for inspection.

D.6 Reward-Judge Prompt

The three skill reward adapters use the same scoring instructions. The following English rendering retains the structure and scoring rules of that template; the executable prompt also names the background, criteria, and penalty sections in Chinese to match the rendered rubrics. The four placeholders are populated after the task-dependent preprocessing described above.

Shared Skill-rubric reward prompt

Please act as a **Strict Evaluator**. You will be provided with:

1. The user's original **Instruction**.
2. A **Reference Answer** representing the ground truth.
3. A **Scoring Standard** containing background facts and point-based criteria.
4. An AI assistant's **Response** to evaluate.

Your task is to strictly apply the Scoring Standard to calculate a final score, then map it to a scale of **0 to 5**.

Understanding the scoring standard. The standard may contain up to three sections:

- **Background Knowledge**: facts about the image. You cannot see the image, so treat the statements in this section as established facts when deciding whether the response's visual claims are correct. A claim that contradicts this section is factually wrong, even if it sounds plausible.
- **Criteria**: positive items, each worth the stated number of points.
- **Penalties**: negative items, each deducting the stated number of points. A penalty may carry a cap; never deduct beyond that cap.

Background Knowledge is context, not a criterion. Do not award or deduct points for it directly. Start from zero, award points for criteria the response meets, and deduct points for violations. Check factual claims against Background Knowledge first, then the Reference Answer. After summing, clamp the raw score to the valid range defined by the standard and map proportionally to 0–5.

Step 1: Criterion-by-criterion evaluation. For each positive or negative item, state whether the response earns the points or triggers the penalty, with a brief justification grounded in the Background Knowledge and Reference Answer.

Step 2: Score calculation. Sum the earned points and deductions, applying all caps. Map the result to an integer between 0 and 5. Output the final score on a separate last line as `Score: X`, with no text after it.

[The User's Original Instruction] {question}
[Reference Answer] {reference}
[Scoring Standard] {rubric}
[The Start of the AI Assistant's Response]
{processed response}
[The End of the AI Assistant's Response]
[Your Evaluation]

E Case Studies

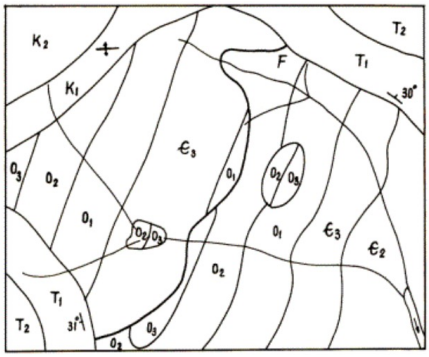
This section provides qualitative case studies of the paper. Sec. E.1 shows annotated instances of MM-RubricBench, Sec. E.2 shows clusters of the skill bank together with the cards induced from them, and Sec. E.3 compares SIGHT generated rubrics with the human ones.

E.1 Annotated Benchmark Instances

Two instances illustrate what a single entry of MM-RubricBench contains: the prompt and visual input, reference answer, expert rubric, and human score for each sampled response.

Graded credit separates responses that binary checks cannot. The rubric in Fig. E.1 assigns two points to each of the two requested conclusions and one point to the formation mechanism, while defining partial credit for near-misses in advance: the correct structure receives full credit, a generic fold receives partial credit, and the opposite structure receives none. Content beyond the scoring points incurs an item-level deduction up to a cap. The three models separate as $5/3/0$. The top response correctly reads both the legend symbol and the age order of the strata. The middle response identifies the structure and its mechanism, receiving full credit on these two items, but answers the first sub-question with stratigraphic age rather than the requested attitude. The last response reverses the core-versus-flank age relation, leading to the wrong structure and mechanism and triggering the full deduction. A binary checklist would treat the latter two responses alike, since both mention strata and name a structure.

Enumeration and over-answering are graded separately. Fig. E.2 shows a instance involving two images. The first question grades species enumeration in tiers, from all four correct to one or two correct, and explicitly states that naming a species outside the list is not penalised for this item. The second asks whether the flower in the second image is used in the bouquet in the first and asks for another colour of that flower. Under-answering is handled by the enumeration tiers, while unsupported content is handled by a separate deduction item. As a result, one mistaken species incurs two penalties, causing the middle response to drop from four points to three while retaining both points from the second turn. The last response mislabels two flowers, placing it in the lowest enumeration tier; the full deduction then removes two of the three points it had earned.



Question: Based on this geological map, what kind of strata are K_1, K_2? What structure do T_1, T_2 form, and how did that structure form?

Reference Answer

K_1 and K_2 are horizontal strata. T_1 and T_2 form an anticline. Under strong crustal compression, the strata undergo plastic bending and form the anticline.

Human Rubric

① **K_1, K_2 are horizontal strata+2**; **T_1, T_2 form an anticline +2**; answering only "fold" gets **+1**, while answering "syncline" gets **0**

② Mentions a matching formation mechanism such as **horizontal crustal compression** and **plastic deformation of strata +1**

③ Hallucinated or unreasonable content beyond the scoring points **-1** per item, capped at **-2**

Gemini 3 Pro · 5 points

K Explicitly cites the **"+" symbol as the mark for horizontal strata**, rather than merely saying "Cretaceous".

T Uses **old core, young flanks** to identify an **anticline**, and states that T_1 is older while T_2 is younger.

How Answers with **horizontal compression + plastic deformation + upward arching**, fully matching the rubric.

Gemini 2.5 Pro · 3 points

K Only answers **"Cretaceous sedimentary strata"**, without actually answering the property the question asks for: **horizontal strata**.

T Gets the anticline right, and also gets the formation mechanism right, so **the latter two items still earn full credit**.

Result This kind of answer shows the model knows stratigraphic age, but **did not read the legend symbol**, so it stops at 3.

Seed2.0 Pro · 0 points

K Likewise answers only "Cretaceous sedimentary strata", **missing horizontal strata**.

T Treats **T_2 as the younger core layer**, which pushes the whole reasoning chain toward a **syncline**.

How It then continues by explaining **how a syncline forms**, so **all main scoring items collapse and the answer is hit with the full -2 deduction**, ending at 0.

Rubric item	Gemini 3 Pro 2025.11 · rank 5	Gemini 2.5 Pro 2025.03 · rank 12	Seed2.0 Pro 2026.02 · rank 7
①a States "horizontal strata" *(+2)*	+2	0	0
①b States "anticline" *(+2)*	+2	+2	0
② Correct formation mechanism *(+1)*	+1	+1	0
③ Hallucination deduction *(-2 cap)*	0	0	-2
SCORE	5	3	0 clamped

Gemini 3 Pro 5 / 5

Actually reading the map. It reads both the legend symbol and the age order of strata, so all main scoring items land.

Gemini 2.5 Pro 3 / 5

Right structure, wrong attribute. It knows how to judge an anticline and how an anticline forms, but answers the first sub-question with age instead of attitude/orientation.

Seed2.0 Pro 0 / 5

The whole reasoning chain is built on a misread. Once the core-vs-flank age relation is flipped, the structure type and its formation mechanism collapse together.

Figure E.1: **An annotated instance of `MM-RubricBench`.** The structure conclusion receives full credit for an anticline, partial credit for a generic fold, and no credit for a syncline, while content beyond the scoring points is deducted up to a cap. The three responses score $5/3/0$: the middle one identifies the structure and its mechanism but answers the first sub-question with stratigraphic age rather than attitude.

E.2 Clusters of the Skill Bank

This section presents two skills from the bank in Tab. B.1. Both are induced from expert rubrics spanning text-only and multimodal tasks across diverse topics, illustrating modality-independent and topic-independent construction logic.

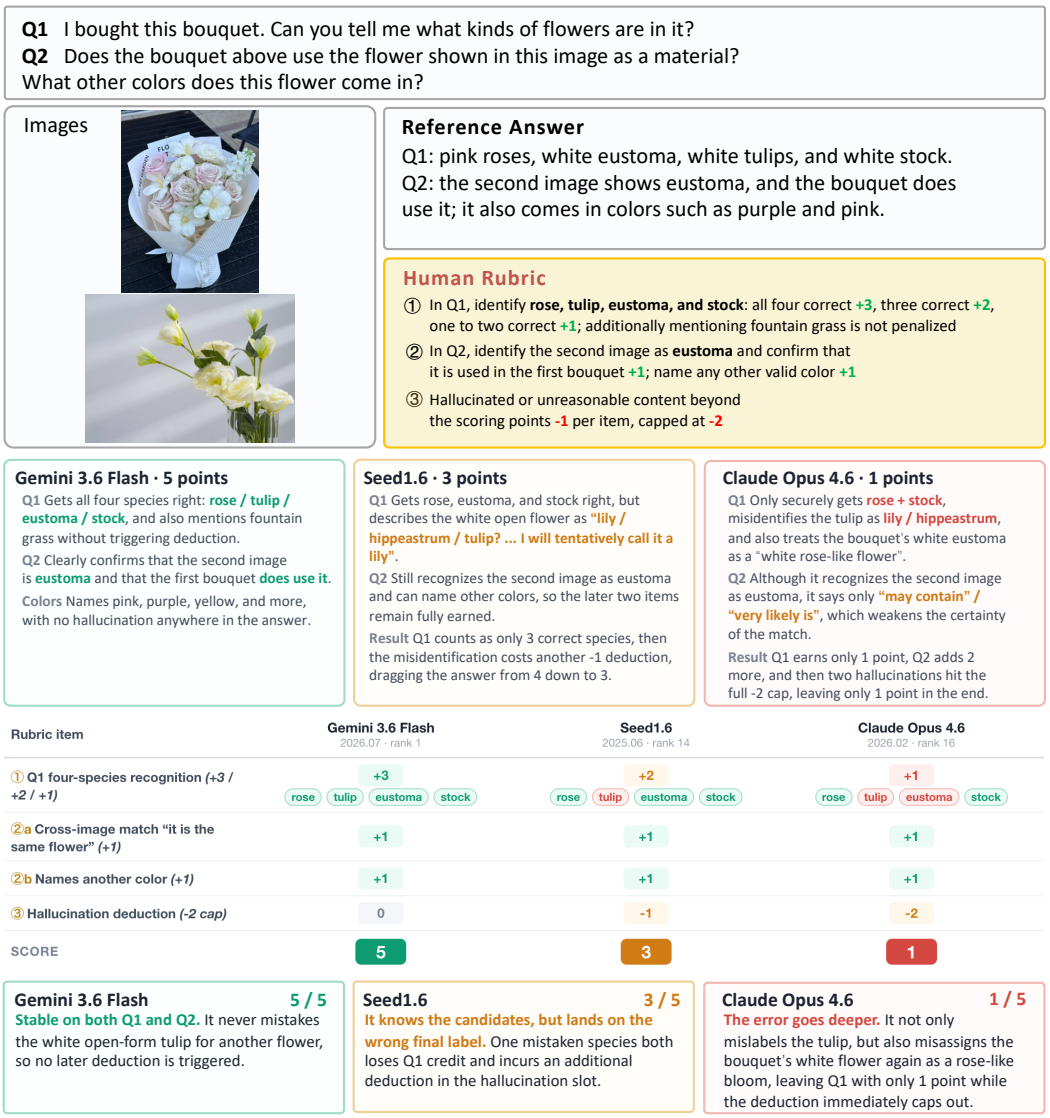


Figure E.2: **A multi-question instance of `MM-RubricBench` with two images.** The first question grades species enumeration in tiers and does not penalise species named outside the list, while the second asks for a cross-image match and an additional colour. One mistaken species therefore costs both an enumeration tier and a deduction, separating the three responses as $5/3/1$.

Tiered scoring for set enumeration. Fig. E.3 shows S08, induced from 48 expert rubrics. Its members ask which countries border a sea, which documents a visa application requires, how many colours an image contains, and how a character is pronounced at each occurrence. Their shared procedure places the target set in a single scoring item, awards tiered credit by the number of correct members, and penalises over-answering separately. The card excludes prompts requesting only a few objects or fixed-format outputs. The latter distinguishes S08 from the related skill split from the same logic family, which also uses tiered scoring but adds an independent format item.

Graded correctness with partial credit. Fig. E.4 shows S11, induced from 43 rubrics. Its shared procedure assigns partial credit to predefined near-miss answers, such as describing an expression as smiling rather than wry or a background as blurry rather than unseparated. These alternatives receive lower credit within the same item, while derived requirements receive separate minor items and fabrication incurs a capped global penalty. Its anti-signals exclude false-premise prompts handled by the related skill split from the same logic family, where correcting the premise is a prerequisite.

E.3 Qualitative Analysis of Generated Rubrics

Sec. 5.2 shows that SIGHT rubrics track human preferences more closely, and we analyse that difference on two representative prompts below.

Credit that cannot be lost. In Fig. E.5 the response names both characters correctly but gets four visual facts wrong. The rubric from RubricRAG (Dhole & Agichtein, 2026) only asks that an action, a hairstyle, and the accessories be mentioned, so naming the wrong one still earns credit. No item can fail, the four errors fall into a single capped deduction worth one point, and the rubric returns $4/5$ where humans give $1/5$. SIGHT writes the referenced fact into the criterion, so a wrong action loses the credit instead of earning it.

Credit that cannot be partial. Fig. E.6 fails in the other direction. The response applies two of the three required ethics norms and lists the student strengths correctly, and the human rubric grades this, more facets earning more credit and a missing one never zeroing the score. OpenRubrics (Liu et al., 2026) instead turns two requirements into hard constraints, so one failure overrides the remaining credit and drops the answer to $1/5$ where humans give $4/5$. SIGHT keeps them as separately weighted items and caps the penalty as a deduction.

Baseline rubrics lack a graded notion of requirement importance. RubricRAG makes every requirement impossible to fail, while OpenRubrics makes two of them impossible to survive. SIGHT inherits from expert rubrics which requirements are central, how much credit each holds, and which errors are deducted rather than disqualifying, which keeps aligned with human preferences.

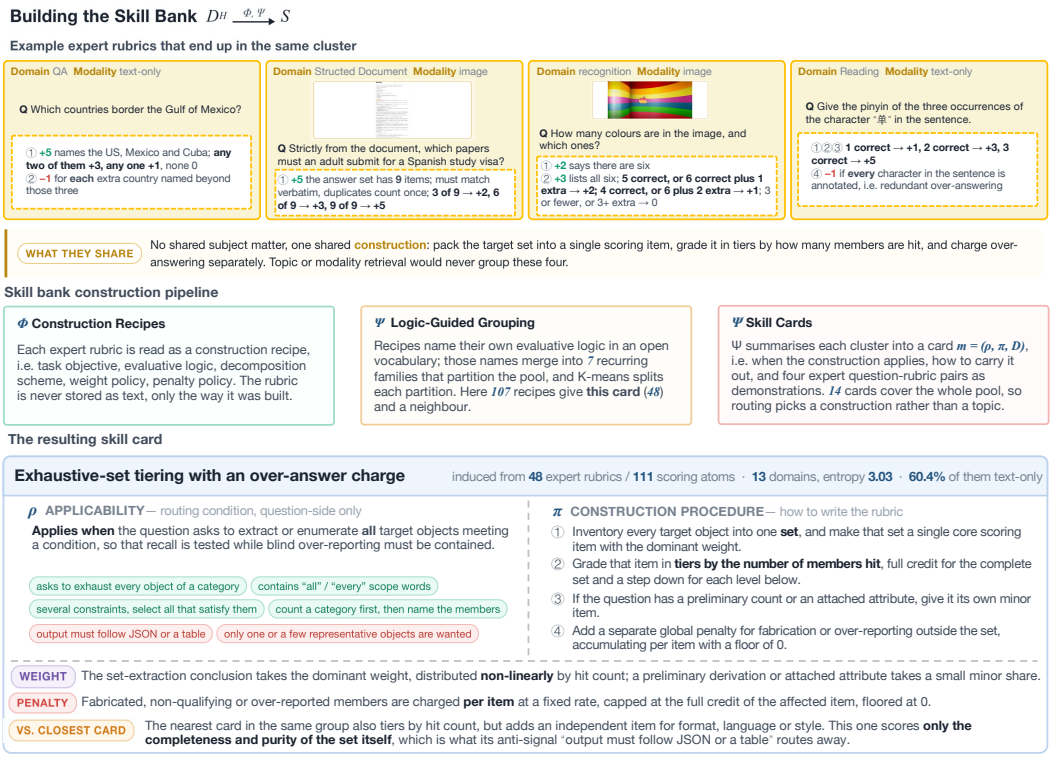


Figure E.3: **Card S08 and four of the expert rubrics it was induced from.** The four rubrics span different subjects and both modalities, but share a common construction: packing the target set into one scoring item, grading it in tiers by the number of members hit, and charging over-answering separately. Its output-format anti-signal distinguishes S08 from the neighbour it was split from within the same logic family.



Figure E.4: **Card S11 and four of the expert rubrics it was induced from.** All four price near-right answers in advance, so an imprecise conclusion receives partial rather than zero credit, preserving distinctions that a binary checklist would erase. S11 is the most frequently routed card in the bank, and its false-premise anti-signal marks the prompts handled by the neighbour it was split from.

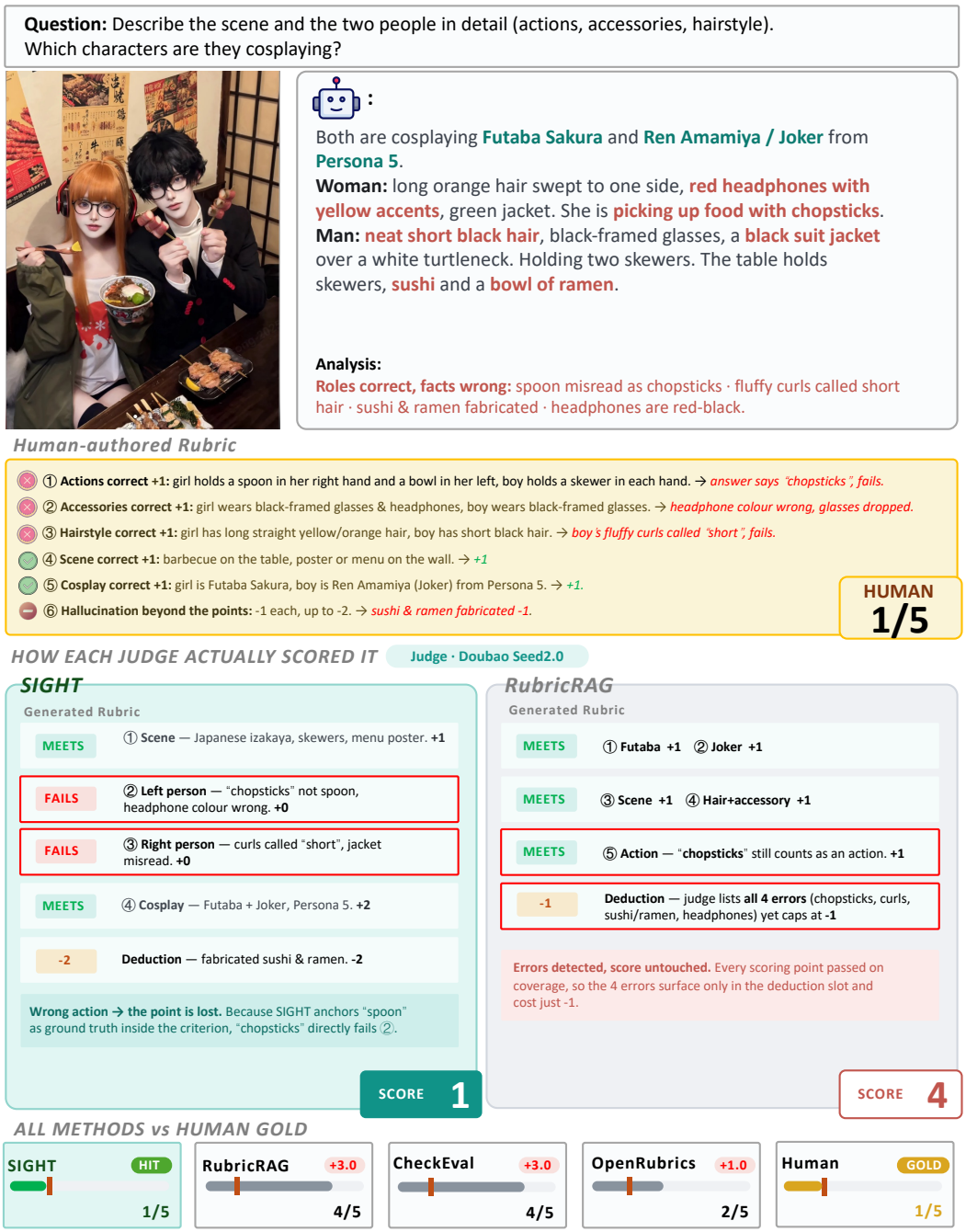


Figure E.5: **Case study of `SIGHT` and RubricRAG rubrics.** The RubricRAG rubric shows credit that cannot be lost: it only asks that the action, hairstyle, and accessories be mentioned, so naming the wrong one still earns credit, while the four factual errors incur only one capped deduction, yielding $4/5$ where humans give $1/5$. `SIGHT` writes the referenced fact into the criterion, so the wrong action loses its credit, yielding $1/5$.

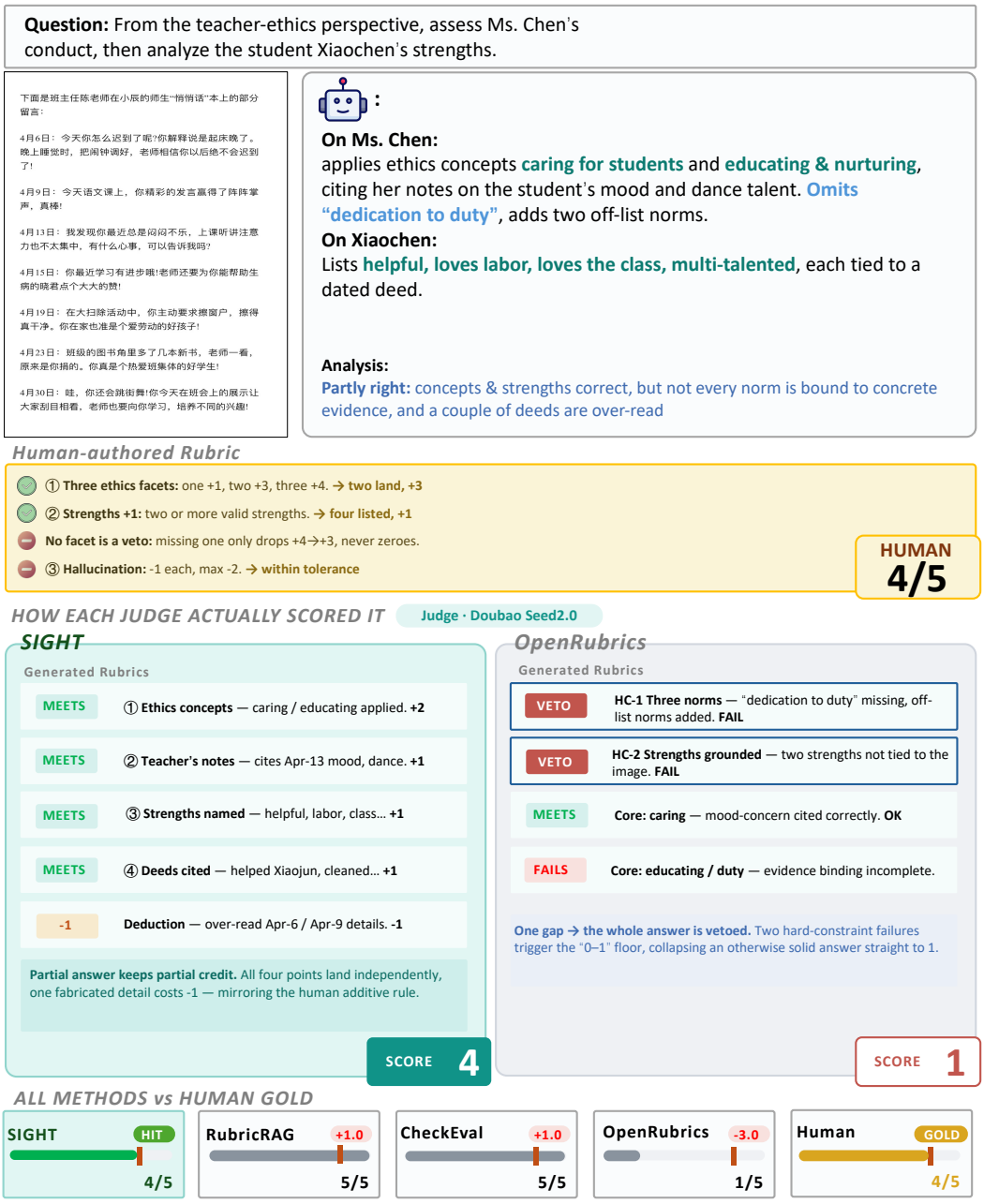


Figure E.6: **Case study of SIGHT and OpenRubrics rubrics.** The OpenRubrics rubric shows credit that cannot be partial: two requirements become hard constraints, and their failure overrides all remaining credit, yielding $1/5$ where humans give $4/5$. SIGHT grades the same requirements as separately weighted items and limits the penalty to a capped deduction, yielding $4/5$.