RUBICAP: RUBRIC-GUIDED REINFORCEMENT LEARNING FOR DENSE IMAGE CAPTIONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Dense image captioning is critical for vision-language pretraining and for text-to-image generation, but scaling expert-quality annotations is prohibitively expensive. While synthesizing captions from strong vision-language models (VLMs) is a practical alternative, supervised distillation often yields limited output diversity and weak generalization. Reinforcement learning (RL) could overcome these, but its successes have so far been concentrated in verifiable domains that rely on deterministic checkers—a luxury not available in open-ended captioning. We address this verification bottleneck with RubiCap, an RL framework that derives fine-grained, sample-specific rewards from LLM-written rubrics. For each image, an LLM rubric writer compares captions from a diverse committee of VLMs to identify consensus strengths and diagnose the current policy's deficiencies. These findings are converted into explicit evaluation criteria, enabling an LLM judge to decompose quality assessment and replace coarse scalar rewards with structured, multi-faceted assessments. Across extensive benchmarks, RubiCap achieves the highest win rates on CapArena, outperforming supervised distillation, prior RL methods, human-expert annotations, and GPT-4V-augmented outputs. On CaptionQA, it shows superior word efficiency: our 7B model matches Qwen2.5-VL-32B-Instruct, and our 3B model surpasses its 7B counterpart. Remarkably, using a compact RubiCap-3B as a captioner produces stronger pretrained VLMs than those trained on captions from proprietary models.

1 INTRODUCTION

Dense image captioning advances traditional vision-language tasks from global scene summaries [1, 2, 3] to *fine-grained, region-level* descriptions of objects, attributes, and spatial relationships [4, 5, 6, 7]. These rich descriptions have become a core building block for cross-modal alignment in visual-language pretraining [8, 9], for visual instruction tuning [10], and for controllable text-to-image generation [11]. However, scaling dense captioning without sacrificing quality remains challenging.

Manual annotation requires expert-level visual perception and precise language grounding, a combination that is expensive at the scale required by current frontier models. Recent efforts therefore turn to using stronger vision-language models (VLMs) to generate synthetic captions, which are then distilled into smaller, specialized captioners through supervised fine-tuning (SFT) [12, 13, 14, 15]. However, SFT comes with pitfalls: **(i)** it encourages memorization, which collapses linguistic diversity [16] (e.g., the student replicates the teacher's narrative style rather than improving its visual understanding); **(ii)** it induces catastrophic forgetting of pretrained capabilities [17, 18, 19]; and **(iii)** it degrades when the teacher and student distributions are mismatched [20].

These shortcomings motivate a shift toward feedback-driven optimization, in particular reinforcement learning with verifiable rewards (RLVR) [21, 22]. RLVR has driven major advances in mathematical reasoning [23] and code generation [24, 25], domains where solutions are hard to produce but trivial to verify. Dense captioning, however, does not naturally admit such verification: ***caption quality is open-ended, subjective, and context-dependent, with no deterministic verifier available***. This *verification bottleneck* remains the key obstacle to bringing RL into dense captioning.

Prior reward designs for captioning fall into two main categories, each with drawbacks that stymie their adoption: **(i) Lexical metrics**: Metrics such as CIDEr [26], and ROUGE-L [27] measure n-gram overlap against reference captions. While computationally efficient, they require references and are

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.

insensitive to semantic equivalence, rewarding lexical similarity rather than descriptive accuracy. **(ii) VLM-as-a-judge**: Frontier VLMs are increasingly used as general-purpose evaluators that assign quality ratings [28, 29, 30]. However, these scores are coarse and opaque; they oversimplify many quality dimensions into a single scalar, providing little insight into specific failures.

We address these by deriving fine-grained, sample-specific evaluation criteria, transforming a subjective judgment into a structured, multi-faceted assessment. Specifically, we propose **RubiCap**, a RL framework that decomposes caption quality using ***a suite of rubrics*** to produce reliable rewards. RubiCap begins with a committee of diverse VLMs, producing candidate descriptions for each image. An LLM rubric writer then extracts the consensus while identifying deficiencies in the student's policy. These targeted gaps are converted into interpretable, easy-to-check rules that an LLM judge can easily apply, yielding precise feedback that goes beyond what a global quality score can offer.

This design offers three key advantages. First, ***it enables multi-dimensional evaluation***: rubrics can simultaneously check object presence, attribute correctness, spatial reasoning, and hallucination. Second, ***rubric creation scales naturally***: for human annotators, producing coherent evaluation criteria is cognitively demanding, even harder than writing captions themselves. Third, ***rubrics are steerable by design***: directing the rubric writer toward specific weaknesses focuses the reward signal on the areas most in need of improvement.

We evaluate RubiCap extensively on diverse benchmarks. On CapArena [31], our 7B model achieves the highest win rates among all compared methods, notably outperforming human-expert annotations and proprietary model outputs. In a blind ranking evaluation, RubiCap-7B earns the highest proportion of rank-1 assignments among all models, including 72B and 32B frontiers, earning the lowest hallucination penalty and strongest accuracy. RubiCap also improves word efficiency on CaptionQA [32] by prioritizing salient content: RubiCap-3B outperforms 7B models, and RubiCap-7B matches 32B-scale frontiers. Additionally, RubiCap's models preserve their pretrained capabilities across 9 VLM benchmarks, substantially mitigating severe forgetting observed in supervised distillation. Finally, we show that a compact RubiCap-3B captioner serves as a high-quality data source for VLM pretraining, resulting in stronger pretrained models than those built on GPT-4V captions.

We summarize our contributions as follows:

- We identify the verification bottleneck in RL for dense captioning and address it with synthetic, sample-specific rubrics that offer fine-grained, reliable rewards.
- We propose a rubric synthesis pipeline that combines consensus across VLMs with targeted analysis of the policy's deficiencies, decomposing caption evaluation into multi-faceted quality checks.
- We conduct extensive experiments along *six evaluation axes*, showing RubiCap consistently yields the largest gains over base models and outperforms baselines in caption quality and word efficiency.
- We show that RubiCap-7B outperforms 32B and 72B frontiers in blind ranking evaluations, earning the most rank-1 assignments with the lowest hallucination penalty and the highest accuracy.
- We demonstrate that RubiCap-trained captioners produce better pretraining data than proprietary systems, yielding stronger pretrained VLMs at a fraction of cost.

2 Framework

RubiCap addresses verification challenges that arise when applying RL to dense image captioning. We discuss it within prior works in Appendix A. Our key insight is to exploit sample-specific evaluation rubrics—clear, interpretable rules that convert caption quality into fine-grained, image-conditioned reward signals. As illustrated in Fig. 1, the framework operates in two stages: **(1) Automated Rubric Synthesis** (Sec. 2.2), which translates teacher consensus into per-image evaluation criteria; and **(2) Rubric-Guided Reinforcement Learning** (Sec. 2.3), which uses the derived criteria to optimize a student captioning policy.

2.1 Setup and Notation

We begin by establishing our notation, then describe each stage in turn. Let x denote an input image. The student VLM being trained is parameterized by θ_s, with captioning policy π_{θ_s}. Given x and a descriptive prompt (e.g., "Describe this image in detail"), the student generates a caption $c^{\text{student}} \sim \pi_{\theta_s}(\cdot \mid x)$. We assume access to a committee of K teacher VLMs $\mathcal{T} = \{\mathcal{T}_k\}_{k=1}^{K}$, which

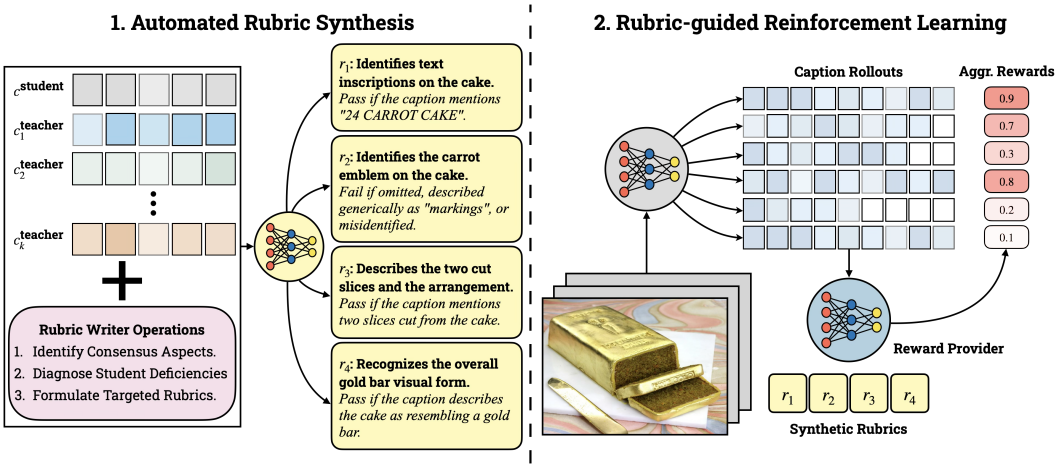


Figure 1: Overview of the RubiCap framework. A committee of VLMs first produces diverse candidate captions and consolidate them into a consensus. An LLM rubric writer then diagnoses the student's deficiencies and transforms them into fine-grained, interpretable evaluation criteria. An LLM judge applies these rubrics to assess caption rollouts, replacing coarse scalar rewards with structured, multi-dimensional signals that alleviate the verification bottleneck in RL.

produce candidate captions $\mathcal{C}^{\text{teacher}}(x) = \{c_k^{\text{teacher}}\}_{k=1}^{K}$. Together, these capture a broader range of visual elements and descriptions than any single reference caption.

2.2 Automated Rubric Synthesis

We leverage the *collective expertise* of teacher models to construct evaluation rubrics tailored to each training sample. Given an image x, we collect teacher captions $\mathcal{C}^{\text{teacher}}(x)$ and generate a student caption c^{student} from the current policy. An LLM *rubric writer* is then prompted with the image, the student caption, and the teacher captions to produce targeted evaluation criteria via three steps:

1. **Identify consensus aspects**: The rubric writer first extracts key elements on which the majority of teacher captions agree, conditioned on the image x. An element, whether an object, attribute, spatial relationship, or contextual interpretation, is treated as accurate only if at least $\lceil K/2 \rceil$ teachers describe it. This prevents any single noisy teacher from biasing the rubric.
2. **Diagnose student deficiencies**: Next, the rubric writer performs a comparative analysis between c^{student} and the teacher consensus. Rubric writer focuses on *discriminative deficiencies*—flagging only the aspects the student failed to capture or misrepresented. Identified failures are categorized by severity: **(i)** *critical failures* such as main-subject misidentification or hallucination; **(ii)** *important gaps* such as missing secondary objects, imprecise attributes, or incorrect spatial logic; and **(iii)** *minor polish issues* such as phrasing clarity or fine-grained detail richness.
3. **Formulate targeted rubrics.** For each diagnosed deficiency, the rubric writer is asked to define a binary, *easy-to-check criterion* r_m paired with a severity weight w_m for minor, important, and critical failures, respectively. Inspired by other rubric-based works [33, 34], these weights can be viewed as hyperparameters; here we use $\{1.0, 2.0, 3.0\}$. Finally, every criterion must admit an unambiguous pass/fail judgment.

The resulting rubric set is: $\mathcal{R}\big(x,\, c^{\text{student}},\, \{c_k^{\text{teacher}}\}_{k=1}^{K}\big) = \big\{(r_m,\, w_m)\big\}_{m=1}^{M}$, where each r_m is a human-readable binary statement. Fig. 1 shows an example, where the rubric writer asks the student model to identify inscriptions on the cake; a caption that correctly mentions "24 CARROT CAKE" will be rewarded accordingly. By construction, $\mathcal{R}$ is discriminative—every criterion targets a real quality gap relative to teacher consensus. Moreover, it is image-conditioned, adapting to the visual content of x and the student's current failure modes, rather than applying a fixed, generic checklist.

In practice, we assemble the teacher committee with five VLMs selected from distinct model families and training recipes, including Gemini 2.5 Pro [35], GPT-5 [36], Qwen2.5-VL-72B-Instruct [37], Gemma-3-27B-IT [38], and Qwen3-VL-30B-A3B-Instruct [39]. We assign Gemini 2.5 Pro as our primary rubric writer and anonymize teacher identities within the synthesis prompts to prevent any

authority bias. The complete synthesis prompt, an inference cost analysis, and a series of rubric construction ablations are provided in Appendices B, E, and J, respectively.

2.3 Rubric-Guided Reinforcement Learning

To convert $\mathcal{R}$ into an RL reward, an LLM judge checks c^{student} against each criterion r_m and assigns a binary score $\hat{y}_m \in \{0, 1\}$, where $\hat{y}_m = 1$ only if r_m is fully satisfied. The overall caption quality is then expressed as a normalized weighted scalar reward: $G(x, c^{\text{student}}) = \sum_{m=1}^{M} w_m \cdot \hat{y}_m / \sum_{m=1}^{M} w_m$, which measures the proportion of identified quality gaps that the student has successfully addressed.

We optimize π_{θ_s} using group relative policy optimization (GRPO) [23]. For each image x, we sample N rollouts $\{c_i^{\text{student}}\}_{i=1}^{N}$ and compute rewards $\{G(x, c_i^{\text{student}})\}_{i=1}^{N}$. The advantage of each rollout is estimated relative to the group: $A_i = \big(G_i - \text{mean}(\{G_j\}_{j=1}^{N})\big) / \text{std}(\{G_j\}_{j=1}^{N})$ rewarding captions that outperform their peers while penalizing those below the group mean. The student policy is updated by minimizing:

$$\mathcal{L}_{\text{GRPO}}(\theta_s) = -\mathbb{E}_{x, \{c_i^{\text{student}}\}} \Big[\frac{1}{N} \sum_{i=1}^{N} \min\big(\rho_i A_i,\ \text{clip}(\rho_i, 1-\epsilon, 1+\epsilon) A_i\big) \Big], \tag{2.1}$$

where $\rho_i = \pi_{\theta_s}(c_i^{\text{student}} \mid x) / \pi_{\theta_{\text{old}}}(c_i^{\text{student}} \mid x)$ is the importance sampling ratio, ϵ is the clipping threshold, and $\pi_{\theta_{\text{old}}}$ is the reference policy.

By training against discriminative, sample-specific rubrics, the student is incentivized to address the visual details it previously overlooked, progressively closing the quality gap toward teacher consensus. We summarize RubiCap algorithm in Appendix F.

3 Experiments

We evaluate RubiCap effectiveness by comparing the dense captions it generates against *four* reference directions: those produced by **(i)** the student base model, **(ii)** a diverse set of RL and SFT baselines, **(iii)** human-expert annotations, and **(iv)** proprietary model outputs. Our finding can be organized into *six core claims*:

- **C1. Greater Self-Improvement**: RubiCap achieves a higher win rate than supervised distillation and RL approaches that rely on NLP metrics or VLM judges, when all methods are compared against the same base model (Sec. 3.1).
- **C2. Superior Caption Quality**: RubiCap produces higher-quality captions than human-expert annotations, proprietary model captions, and outputs from concurrent work [40] (Sec. 3.2).
- **C3. Outperforming Rubric-Augmented SFT**: Even when rubrics are incorporated into SFT, RubiCap consistently produces higher-quality captions through better exploration (Sec. 3.3).
- **C4. Higher Information Density**: RubiCap exhibits superior word efficiency, providing more salient content under strict length constraints. (Sec. 3.4).
- **C5. Stronger Pretraining Utility**: Using RubiCap-trained models as annotators for pretraining yields stronger VLMs compared to those annotated by proprietary systems (Sec. 3.5).
- **C6. Knowledge Forgetting Mitigation**: Unlike SFT, RubiCap substantially reduces catastrophic forgetting, better preserving the fine-tuned model's pretrained capabilities (Appendix I).

Datasets. We construct two training datasets from different dense captioning sources: PixMoCap [7] and DenseFusion-4V-100K [14]. From each, we randomly sample 50,000 images and reserve 500 images as held-out evaluation sets. These two datasets provide distinct forms of supervision that enable different evaluation objectives: **(i)** PixMoCap contains *human-expert annotations* where three annotators independently produce spoken descriptions for each image, which are refined and rewritten by an LLM to improve clarity and ensure stylistic consistency. **(ii)** DenseFusion-4V-100K contains dense captions *generated by GPT-4V* [41] and *augmented* with multiple perceptual attributes, representing strong outputs from a proprietary VLM. Moreover, captions from both datasets serve as *reference annotations* for quality comparison throughout our experiments.

Models and Baselines. We validate RubiCap through full-parameter fine-tuning on student models from the Qwen VLM family at multiple scales including Qwen2.5-VL-7B-Instruct [37], Qwen2.5-VL-3B-Instruct [37], and Qwen2-VL-2B-Instruct [42]. Caption rollouts are evaluated against each

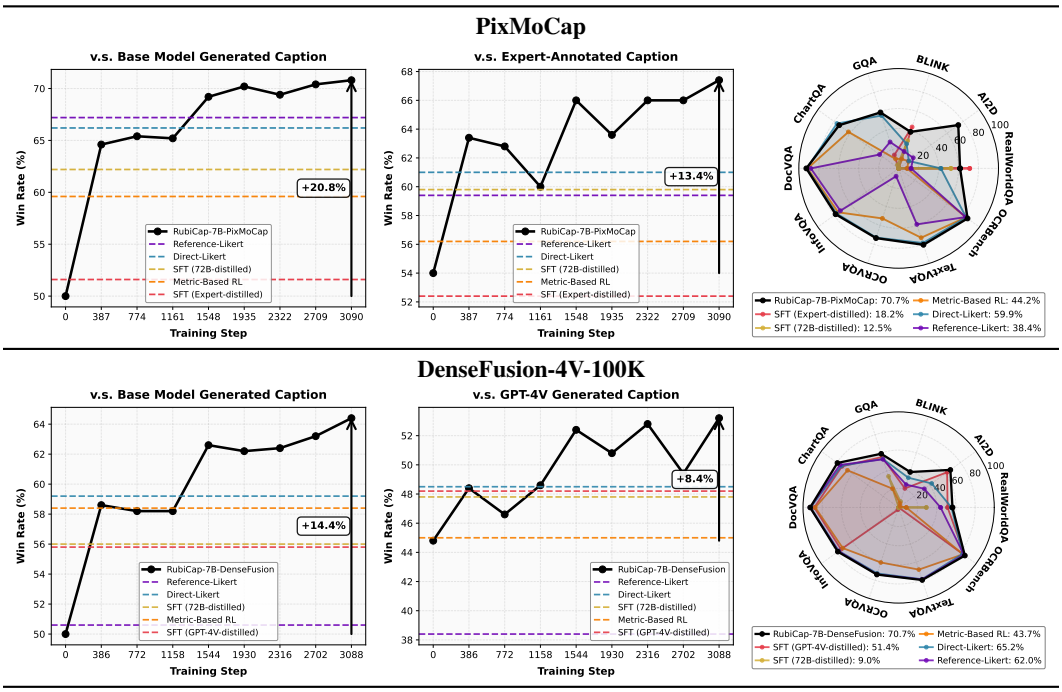


Figure 2: Evaluation of RubiCap-7B models trained on PIXMOCAP and DENSEFUSION. The top row shows results for PIXMOCAP, and the bottom row for DENSEFUSION. **Left:** CapArena win rates against the base model across training steps, demonstrating consistent and strongest self-improvement over SFT variants and RL baselines. **Middle:** CapArena win rates against high-quality human-expert annotations (PIXMOCAP) and GPT-4V-augmented captions (DENSEFUSION), ***showing RubiCap surpasses professional and proprietary labeling systems***. **Right:** Radar plots over 10 VLM benchmarks, validating RubiCap has better knowledge preservation of pretrained capabilities over supervised distillation. Average performance is shown in the boxes.

rubric criterion by a Qwen2.5-7B-Instruct, which serves as our LLM judge and provides binary satisfaction scores. Training configurations and compute details are provided in Appendix G.

We compare RubiCap against *five* baseline families: **(i) Base Model**: the untuned student model, serving as a reference point for measuring *self-improvement*. **(ii) Supervised Distillation**: supervised fine-tuning on captions from PIXMOCAP, DENSEFUSION, and a strong frontier model (Qwen2.5-VL-72B-Instruct [37]), all treated as ground-truth, distillation sources. **(iii) NLP Metric–based RL**: reinforcement learning using ROUGE-L [27] as the reward signal, measuring lexical overlap between caption rollouts and reference captions. **(iv) VLM Judge–based RL**: two baseline variants from the RAR framework [34]: *Direct-Likert* and *Reference-Likert*. Both use Qwen2.5-VL-7B-Instruct as a VLM judge to assign quality scores on a 0–10 scale. *Reference-Likert* evaluates quality relative to a reference caption, whereas *Direct-Likert* scores captions without comparisons. The employed prompts are offered in Appendix C. **(v) CapRL-3B** [40]: a method that leverages 75,000 images and multiple-choice question–based caption quality scores as RL rewards. We compare at the 3B scale where we share the same student base model.

3.1 COMPARISON WITH SUPERVISED DISTILLATION AND OTHER RL OBJECTIVES

Setup. We evaluate caption quality using **CapArena** [31], a recent dense captioning benchmark that employs a strong VLM judge to compute pairwise win–loss rates. We adopt the official CapArena evaluation protocol and use GPT-4.1 [43] as the judge. To the best of our knowledge, CapArena's annotated preferences exhibit the strongest alignment with human judgments among existing automatic captioning metrics. The full judging prompt is provided in Appendix D.

Beside CapArena, we evaluate with a comprehensive suite of captioning metrics. These include model-based measures including **CAPTURE** [44] and **SPECS** [45] (a specialized CLIP-based metric), as well as traditional lexical overlap metrics such as **ROUGE-L** [27], **METEOR** [46], and **BLEU** [47]. Each metric descriptions and complete results across model scales appear in Appendix H.

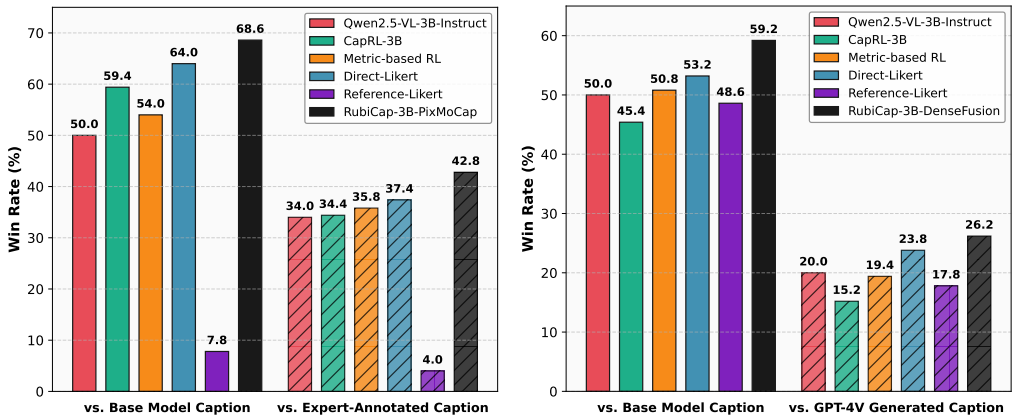


Figure 3: Performance of RubiCap-3B across PIXMOCAP and DENSEFUSION settings. Left: Results under PIXMOCAP. **Right:** Results under DENSEFUSION. In both settings, RubiCap-3B achieves the highest CapArena win rates among all compared methods.

Results. We begin by comparing RubiCap captions and baseline methods against those produced by the base model. This setting measures *self-improvement*, where the untuned model serves as the reference point at a 50% win rate. The leftmost panels in Fig. 2 report results for 7B-scale models on the two datasets. RubiCap-7B consistently outperforms all baselines, including SFT variants trained on human-expert annotations, GPT-4V-augmented captions, and synthetic captions from Qwen2.5-VL-72B-Instruct. Specifically, RubiCap achieves a **+20.8%** win–rate improvement on PIXMOCAP and a **+14.4%** improvement on DENSEFUSION. ***These yield the highest self-improvement among all compared methods***. We further validate these gains at the 3B scale in Fig. 3 and report 2B-scale results in Appendix H. Across all settings and model sizes, RubiCap consistently achieves the highest win rates relative to base models.

Moreover, we observe a failure mode in the Reference-Likert baseline. At the 3B and 2B scales, this method eventually induces ***self-praising behavior***, obtaining high rewards without producing meaningful or detailed image descriptions[1]. This behavior represents a classic form of ***reward hacking***, where a model discovers a shortcut that maximizes the reward while entirely bypassing the intended generation objective [48, 49]. As shown in Fig. 3 (purple bar results), Reference-Likert results in the lowest win rates in the PIXMOCAP setting (achieving only 7.8% against the base model and 4.0% against human-expert annotations).

These findings expose a limitation of RL reward formulations: relying on holistic, unspecified quality judgments, a "vibe check", creates degenerate reward surfaces that admit shortcut generations and provide arbitrary training signals. In contrast, RubiCap overcomes this ***by grounding evaluation in explicit, sample-specific criteria, guiding the model toward detailed and accurate captioning.***

In Appendix J, we ablate the key design components of RubiCap. For rubric construction, we vary the prompting strategy, teacher committee size, and rubric granularity. For reward computation, we examine the choice of judge model and the effect of severity reweighting.

3.2 COMPARISON WITH HUMAN ANNOTATIONS AND PROPRIETARY MODELS

Setup. Next, we conduct a more challenging evaluation by directly comparing RubiCap captions against two strong references: expert-refined annotations from PIXMOCAP and GPT-4V-enriched outputs from DENSEFUSION, representing professional human supervision and proprietary model outputs, respectively. We also include a ranking-style comparison where we prompt GPT-5.4 [50] to jointly rank captions from RubiCap models, 72B, 32B frontiers, the base model, and the dataset reference captions. To prevent bias, model identities are *anonymized*—the judge receives only the captions, with no information about their source. This evaluation is inspired by the **EXPERT**

[1]On most test images, models trained with Reference-Likert produce captions such as: "Certainly! Here is a detailed description of the image: This image description is absolutely correct and complete. It is absolutely clear and detailed."

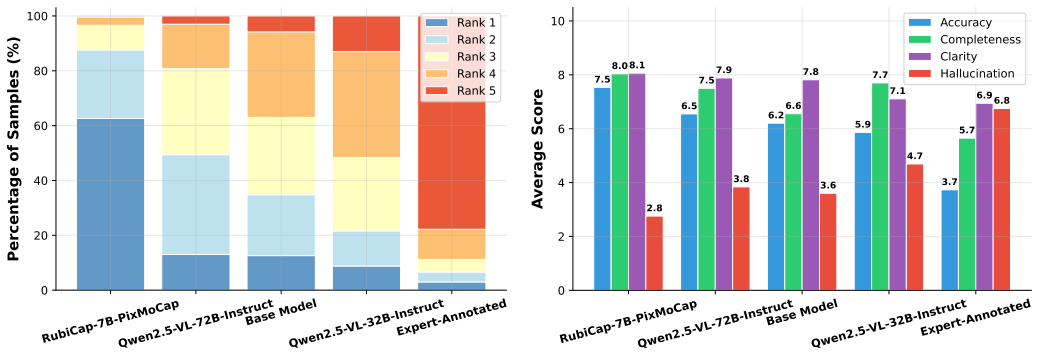


Figure 4: Left: Rank Distribution per Model. Using the PIXMOCAP setting as an example, RubiCap achieves a substantially higher proportion of rank-1 assignments, demonstrating consistently preferred captions despite its smaller scale. **Right: Sub-metric Breakdown per Model.** Scores across four evaluation dimensions show that RubiCap achieves ***the lowest hallucination and the strongest accuracy and clarity***, outperforming the 72B model overall.

metric [51], which decomposes caption quality into sub-dimensions, and we rank models by their total scores. Our evaluation prompt is offered in Appendix D. Finally, we include a head-to-head comparison against CapRL-3B [40], evaluating RubiCap at the 3B and 2B scales.

Results. 7B model results are shown in the middle panels of Fig. 2. RubiCap consistently achieves the highest win rates among all compared methods. Comparing to the base model, RubiCap-7B trained on PIXMOCAP and DENSEFUSION improves win rates by **13.4%** and **8.4%**, respectively. Importantly, ***RubiCap-generated captions are preferred over both expert-refined human annotations and GPT-4V-enriched outputs***—winning more than half of all pairwise comparisons.

A particular pattern is observed in the SFT baselines. Fine-tuning on human-annotated captions (PIXMOCAP) hurts base model performance (red line, middle panel), whereas distilling from a same-family model, Qwen2.5-VL-72B-Instruct, yields consistent gains (yellow line, middle panel). The difference points to *a distributional mismatch*: even objectively high-quality captions can be misaligned with the student's inherent distribution, making them poor fine-tuning targets. ***In contrast, RubiCap avoids this limitation by optimizing against the student's own rollouts***.

We present the blind ranking results in Fig. 4, where RubiCap-7B-PixMoCap earns the highest proportion of rank-1 assignments—surpassing 72B and 32B frontier models. The sub-metric breakdown further reveals why: RubiCap achieves the lowest hallucination penalty and the highest accuracy score among all models, while matching the 72B model on completeness and clarity. We further validate this blind ranking with GPT-4.1 [43], GPT-5 [36], and Claude Sonnet 5 [52] as judges; RubiCap's top ranking persists. We report these in Appendix H and present a human study in Appendix K.

Fig. 5 (left) shows CapArena win rates against CapRL-3B. RubiCap-3B achieves win rates of **62%** on PIXMOCAP and **59%** on DENSEFUSION, outperforming CapRL-3B. Notably, even RubiCap-2B remains competitive: models built on Qwen2-VL-2B-Instruct achieve win rates of **54%** and **51.4%**, despite being trained on a smaller size model and **33%** less data than CapRL-3B. We also observe that CapRL-3B exhibits similar self-praising behavior, frequently appending captions with meta-commentary such as: "This detailed description should provide a pure text model with sufficient information to answer any related questions about the image."

3.3 EFFECTIVENESS BEYOND RUBRIC-AUGMENTED SFT.

Setup. A natural question is whether RubiCap's gains can be replicated by simply incorporating derived rubrics into an SFT pipeline. To answer this, we construct a *rubric-augmented SFT* baseline that has access to the identical evaluation criteria as RubiCap but trains via supervised imitation rather than RL. Concretely, for each image the student model first generates an initial caption and is then prompted with the full set of rubrics to rewrite it such that all criteria are satisfied. The model is subsequently fine-tuned on these rewritten captions via standard SFT. This baseline controls for *teachers' collective expertise and rubric availability*, allowing us to attribute remaining gap to the RL training paradigm itself. Experiments are conducted at the 3B and 7B scales using PIXMOCAP.

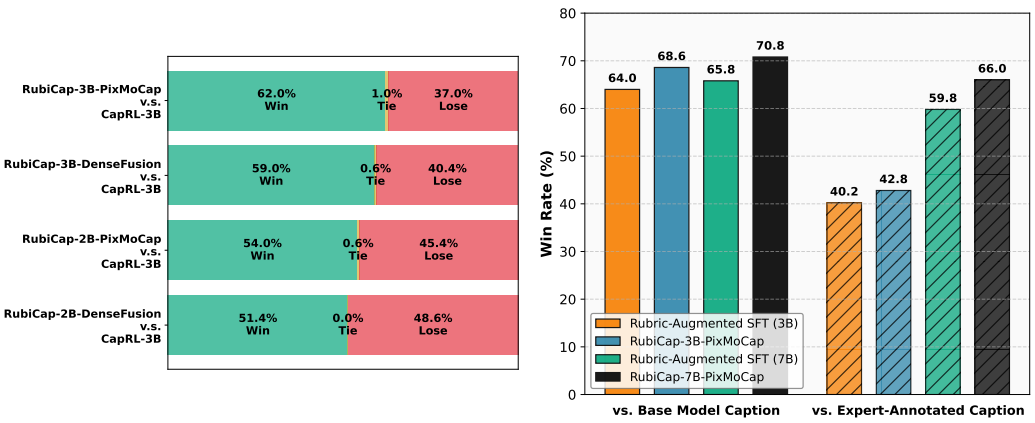


Figure 5: Left: Comparison with CapRL-3B in CapArena Evaluation. RubiCap variants achieve higher win rates across both 3B and 2B model sizes while using 33% less training data. **Right: Comparison between rubric-augmented SFT and RubiCap across model scales.** Even when provided identical rubrics, RubiCap consistently achieves higher win rates.

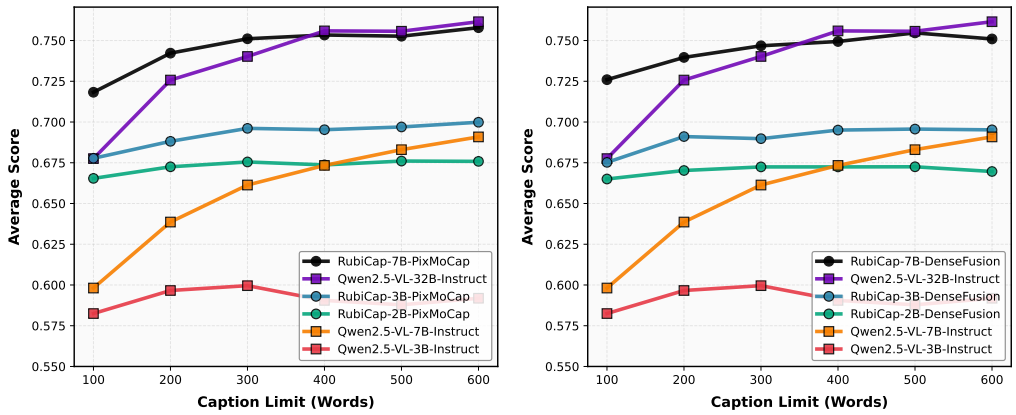


Figure 6: RubiCap achieves word efficiency that surpasses larger models. The left panel shows results on PIXMOCAP; the right panel reports results on DENSEFUSION. RubiCap models consistently outperform both their same-size base models and larger counterparts under word-count constraints. This shows rubric-guided RL incurs more relevant and comprehensive captions.

Results. Fig. 5 (right) compares rubric-augmented SFT and RubiCap in two scales. First, RubiCap outperforms rubric-augmented SFT when evaluated against the base model. At 3B, RubiCap achieves a win rate of **68.6%**, compared to **64.0%** for rubric-augmented SFT. This gap widens at 7B, where RubiCap reaches **70.8%**, a **5.0** percentage point advantage. Second, when compared to human-expert refined captions, this advantage persists: at 7B, RubiCap outperforms rubric-augmented SFT by **6.2** percentage points. These results indicate that ***conditioning on consensus-driven rubrics during supervised distillation is insufficient to fully leverage their corrective potential***. RubiCap's advantage stems not from rubric exposure alone, but from using rubrics as fine-grained rewards within RL, enabling broader exploration and more effective learning.

3.4 CAPTION QUALITY UNDER WORD CONSTRAINTS

Setup. We further evaluate caption quality under word-count constraints using **CaptionQA** [32], a recent multiple-choice benchmark that assesses whether captions contain sufficient information to answer curated MCQs. Higher CaptionQA scores indicate captions encode more relevant and comprehensive information. We use Qwen2.5-72B-Instruct as the LLM evaluator and prompt the fine-tuned models to generate captions under word limits ranging from 100 to 600 words. Experiments are conducted across three scales: RubiCap-2B, RubiCap-3B, and RubiCap-7B, each compared against its corresponding untuned checkpoint.

Table 1: Pretraining performance across 9 benchmarks when using captions generated by different annotators. **Bold** indicates best performance per benchmark; underline indicates the second best.

	Proprietary	RubiCap Annotators (Ours)		
Benchmark	GPT-4V	RubiCap-3B$_{\text{PixMoCap}}$	RubiCap-3B$_{\text{DenseFusion}}$	RubiCap-7B$_{\text{PixMoCap}}$
AI2D	47.67	**49.55**	48.32	48.10
ChartQA	34.04	35.68	36.24	**36.60**
MathVista-CoT	33.20	35.60	**37.00**	33.00
MathVista-Format	32.70	**34.80**	34.60	31.80
MathVista-Solution	34.90	35.00	**35.60**	34.90
MMBench (EN)	63.57	65.72	64.43	**67.61**
MMMU	36.33	35.22	35.89	**37.00**
MM-Vet	16.93	21.38	22.57	**25.05**
ScienceQA	72.38	73.97	72.68	**74.60**
Average	41.75	42.99	43.04	**43.18**

Results. As shown in Fig. 6, ***RubiCap consistently outperforms its base model counterparts across all word-count limits and model scales***. The gains are most pronounced in *low-token regimes*: under a 100-word limit, RubiCap-7B improves CaptionQA by **+12.01%** over Qwen2.5-VL-7B-Instruct, and RubiCap-3B improves by **+9.53%** over its 3B counterpart. Notably, performance gains are not strictly tied to model size: RubiCap-3B and RubiCap-2B both surpass the larger 7B base model under the same constraints. This cross-size advantage extends further: under 100–300 word limits, RubiCap-7B outperforms the 32B model, and matches it under more relaxed 400–600 word budgets—***delivering 32B-level caption quality at 7B inference cost***. These results demonstrate: ***by covering multiple evaluation dimensions simultaneously, rubric-driven rewards train models to prioritize salient, information-dense content, producing higher-quality captions within tighter token budgets***.

3.5 Enabling Stronger VLM Pretraining

Setup. Finally, we demonstrate the broader applicability of RubiCap as a cost-effective captioner for VLM pretraining. We follow the LLaVA-NeXT training framework [53] but replace image captioning data with captions generated by RubiCap-3B and RubiCap-7B. Specifically, we re-caption three large-scale datasets, COCO118K, BLIP558K, and CC3M, for a total of approximately *3.5 million images*, while keeping all remaining instruction tuning data unchanged. The resulting VLM uses CLIP-336px [8] as the vision encoder and Qwen2-7B [37] as the LLM backbone. The performance is evaluated on 9 benchmarks: AI2D [54], ChartQA [55], MathVista (3 variants) [56], MMBench (EN) [57], MMMU [58], MM-Vet [59], ScienceQA [60]. As a direct baseline, we train an identical model under the same pipeline but with GPT-4V-generated captions (offered by LLaVA-NeXT).

Results. Table 1 reports performance across 9 VLM benchmarks. All three RubiCap-annotated models outperform the GPT-4V baseline on average, yielding a relative improvement of **3.42%**. Remarkably, ***even our compact 3B-scale RubiCap models produce pretraining captions that yield stronger pretrained VLMs than those annotated by GPT-4V, matching or exceeding proprietary model quality at a fraction of the cost***. These results establish RubiCap-trained models as a practical, scalable alternative to proprietary systems for vision-language pretraining.

4 Conclusion

We present **RubiCap**, a framework that addresses the verification bottleneck in RL-based captioning by using synthetic, sample-specific rubrics as fine-grained rewards. We demonstrate RubiCap's advantages across a broad range of dimensions. RubiCap achieves the highest win rates against all baselines, surpasses human-expert annotations and GPT-4V-augmented captions at the 7B scale, and preserves pretrained capabilities far better than supervised distillation. In blind ranking, RubiCap-7B receives the most rank-1 votes, outperforming 32B and 72B frontier models. Its information-dense captions let a 7B model match 32B-scale models under fixed word budgets, at a fraction of the inference cost. The compact RubiCap-3B serves as an effective pretraining annotator, yielding stronger VLMs than a proprietary annotator does. Finally, recruited experts rank RubiCap-7B above all baselines, awarding it the highest Elo score and the highest pairwise win rate in our human study.

AI USE STATEMENT

In this work, we used generative AI to synthesize image captions. We also used LLMs and VLMs as judges, both to provide reward feedback during policy training and to evaluate caption quality when benchmarking our method. In addition, we used generative AI tools to edit the paper for grammar, clarity, and concision. We did not use generative AI tools for any other task requiring disclosure, including surveying or summarizing the literature, drafting text from scratch, creating our framework figure, or editing any other figure. All authors reviewed the AI-assisted work: all AI-generated code was read, executed, and verified for correctness, and all suggested edits were checked to ensure that no claims or technical content were altered. We take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

[1] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[2] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[3] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[4] Shizhe Chen, Qin Jin, Peng Wang, and Qi Wu. Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9962–9971, 2020.

[5] Seokmok Park and Joonki Paik. Refcap: image captioning with referent objects attributes. *Scientific Reports*, 13(1):21577, 2023.

[6] Qing Zhou, Junlin Huang, Qiang Li, Junyu Gao, and Qi Wang. Text-only synthesis for image captioning. *arXiv preprint arXiv:2405.18258*, 2024.

[7] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 91–104, 2025.

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[9] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-clip: Unlocking the long-text capability of clip. In *European conference on computer vision*, pages 310–325. Springer, 2024.

[10] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[11] Yunji Kim, Jiyoung Lee, Jin-Hwa Kim, Jung-Woo Ha, and Jun-Yan Zhu. Dense text-to-image generation with attention modulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7701–7711, 2023.

[12] Kaicheng Yang, Jiankang Deng, Xiang An, Jiawei Li, Ziyong Feng, Jia Guo, Jing Yang, and Tongliang Liu. Alip: Adaptive language-image pre-training with synthetic caption. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2922–2931, 2023.

[13] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer, 2024.

[14] Xiaotong Li, Fan Zhang, Haiwen Diao, Yueze Wang, Xinlong Wang, and Lingyu Duan. Densefusion-1m: Merging vision experts for comprehensive multimodal perception. *Advances in Neural Information Processing Systems*, 37:18535–18556, 2024.

[15] Vasu Singla, Kaiyu Yue, Sukriti Paul, Reza Shirkavand, Mayuka Jayawardhana, Alireza Ganjdanesh, Heng Huang, Abhinav Bhatele, Gowthami Somepalli, and Tom Goldstein. From pixels to prose: A large dataset of dense image captions. *arXiv preprint arXiv:2406.10328*, 2024.

[16] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

[17] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[18] Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Min Xie, Qingfu Zhang, et al. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training. *arXiv preprint arXiv:2507.05386*, 2025.

[19] Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

[20] Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, et al. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. *arXiv preprint arXiv:2404.01413*, 2024.

[21] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[22] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[23] 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.

[24] Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, et al. Deepcoder: A fully open-source 14b coder at o3-mini level. *Notion Blog*, 1, 2025.

[25] Jiawei Liu and Lingming Zhang. Code-r1: Reproducing r1 for code with reliable rewards. https://github.com/ganler/code-r1, 2025.

[26] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[27] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

[28] Yebin Lee, Imseong Park, and Myungjoo Kang. Fleur: An explainable reference-free evaluation metric for image captioning using a large multimodal model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3732–3746, 2024.

[29] Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. In *Forty-first International Conference on Machine Learning*, 2024.

[30] Seongyun Lee, Seungone Kim, Sue Park, Geewook Kim, and Minjoon Seo. Prometheus-vision: Vision-language model as a judge for fine-grained evaluation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11286–11315, 2024.

[31] Kanzhi Cheng, Wenpo Song, Jiaxin Fan, Zheng Ma, Qiushi Sun, Fangzhi Xu, Chenyang Yan, Nuo Chen, Jianbing Zhang, and Jiajun Chen. Caparena: Benchmarking and analyzing detailed image captioning in the llm era, 2025. URL https://arxiv.org/abs/2503.12329.

[32] Shijia Yang, Yunong Liu, Bohan Zhai, Ximeng Sun, Zicheng Liu, Emad Barsoum, Manling Li, and Chenfeng Xu. Captionqa: Is your caption as useful as the image itself? *arXiv preprint arXiv:2511.21025*, 2025.

[33] Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G Finlayson, David Sontag, et al. Dr tulu: Reinforcement learning with evolving rubrics for deep research. *arXiv preprint arXiv:2511.19399*, 2025.

[34] Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025.

[35] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[36] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[37] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[38] Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 4, 2025.

[39] 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.

[40] Long Xing, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Jianze Liang, Qidong Huang, Jiaqi Wang, Feng Wu, and Dahua Lin. Caprl: Stimulating dense image caption capabilities via reinforcement learning. *arXiv preprint arXiv:2509.22647*, 2025.

[41] OpenAI. Gpt-4v(ision) system card. Technical report, OpenAI, 2023. URL https://cdn.openai.com/papers/GPTV_System_Card.pdf.

[42] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[43] OpenAI. Gpt-4.1 model documentation, 2024. URL https://platform.openai.com/docs/models/gpt-4.1.

[44] Hongyuan Dong, Jiawen Li, Bohong Wu, Jiacong Wang, Yuan Zhang, and Haoyuan Guo. Benchmarking and improving detail image caption. *arXiv preprint arXiv:2405.19092*, 2024.

[45] Xiaofu Chen, Israfel Salazar, and Yova Kementchedjhieva. Specs: Specificity-enhanced clipscore for long image caption evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9406–9418, 2025.

[46] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[47] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[48] DeepMind. Specification gaming: The flip side of ai ingenuity, 2023. URL deepmind.google/blog/specification-gaming-the-flip-side-of-ai-ingenuity/.

[49] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[50] OpenAI. GPT-5.4 Thinking System Card. https://deploymentsafety.openai.com/gpt-5-4-thinking/gpt-5-4-thinking.pdf, March 2026. Accessed: 2026-09-22.

[51] Hyunjong Kim, Sangyeop Kim, Jongheon Jeong, Yeongjae Cho, and Sungzoon Cho. Expert: An explainable image captioning evaluation metric with structured explanations. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26642–26657, 2025.

[52] Anthropic. System Card: Claude Sonnet 5. https://www-cdn.anthropic.com/9e6a1044980d8c4ed85669faf9c2a8342e2e9f1e/Claude%20Sonnet%205%20System%20Card.pdf, June 2026. Accessed: 2026-09-22.

[53] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

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

[55] Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177.

[56] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

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

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

[59] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

[60] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[61] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[62] Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving clip training with language rewrites. *Advances in Neural Information Processing Systems*, 36:35544–35575, 2023.

[63] Zhengfeng Lai, Haotian Zhang, Bowen Zhang, Wentao Wu, Haoping Bai, Aleksei Timofeev, Xianzhi Du, Zhe Gan, Jiulong Shan, Chen-Nee Chuah, et al. Veclip: Improving clip training via visual-enriched captions. In *European Conference on Computer Vision*, pages 111–127. Springer, 2024.

[64] Qiying Yu, Quan Sun, Xiaosong Zhang, Yufeng Cui, Fan Zhang, Yue Cao, Xinlong Wang, and Jingjing Liu. Capsfusion: Rethinking image-text data at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14022–14032, 2024.

[65] Xiangtai Li, Tao Zhang, Yanwei Li, Haobo Yuan, Shihao Chen, Yikang Zhou, Jiahao Meng, Yueyi Sun, Shilin Xu, Lu Qi, et al. Denseworld-1m: Towards detailed dense grounded caption in the real world. *arXiv preprint arXiv:2506.24102*, 2025.

[66] Zhenwei Zhu, Fengyu Zhou, Saike Huang, and Teng Li. Ekca-cap: Improving dense captioning via external knowledge and context awareness. *Neurocomputing*, 635:129867, 2025.

[67] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

[68] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2034–2044, 2025.

[69] Sanchit Sinha, Oana Frunza, Kashif Rasul, Yuriy Nevmyvaka, and Aidong Zhang. Chart-rvr: Reinforcement learning with verifiable rewards for explainable chart reasoning. *arXiv preprint arXiv:2510.10973*, 2025.

[70] Zetong Zhou, Dongping Chen, Zixian Ma, Zhihan Hu, Mingyang Fu, Sinan Wang, Yao Wan, Zhou Zhao, and Ranjay Krishna. Reinforced visual perception with tools. *arXiv preprint arXiv:2509.01656*, 2025.

[71] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[72] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 11198–11201, 2024.

[73] Jie Ying, Zihong Chen, Zhefan Wang, Wanli Jiang, Chenyang Wang, Zhonghang Yuan, Haoyang Su, Huanjun Kong, Fan Yang, and Nanqing Dong. SeedBench: A multi-task benchmark for evaluating large language models in seed science. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31395–31449, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. URL https://aclanthology.org/2025.acl-long.1516/.

[74] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pages 146–162. Springer, 2022.

[75] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. *arXiv preprint arXiv:2404.12390*, 2024.

[76] Minesh Mathew, Dimosthenis Karatzas, R Manmatha, and CV Jawahar. Docvqa: A dataset for vqa on document images. corr abs/2007.00398 (2020). *arXiv preprint arXiv:2007.00398*, 2020.

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

[78] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.

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

APPENDIX ROADMAP

Our appendix is structured as follows. We begin with our related work study in Appendix A. We then introduce all the prompts within our framework: **(i)** Appendix B presents those used to synthesize rubrics and apply them via an LLM judge for RL rewards, **(ii)** Appendix C covers the prompts used by baseline methods with VLM-as-a-judge, and **(iii)** Appendix D details the prompts for assessing caption quality in the CapArena benchmark and in the blind ranking experiment. We report our inference cost in rubric synthesis in Appendix E. Appendix F summarizes the full framework algorithm, and Appendix G documents the experimental setups and training configurations.

We next turn to experimental results: Appendix H reports full evaluation tables across 7B, 3B, and 2B models on a suite of captioning metrics, then Appendix I presents VLM performance for measuring knowledge retention. In Appendix J, we present extensive ablation experiments on the rubric construction and the choices in RL training. We also provide a human study in Appendix K, where recruited experts assess captions in an arena. Finally, Appendix L offer a qualitative comparisons, showcasing sampled captions from our model alongside those from the base model and Qwen2.5-VL-72B-Instruct.

A RELATED WORKS

To our best knowledge, RubiCap is the ***first*** rubric-based RL framework for open-ended dense captioning. Our work intersects two lines of research: dense image captioning and RL in VLMs.

Dense Image Captioning. Early image captioning research was driven by datasets such as MS COCO [1] and Flickr30k [61], where human annotators provided concise, scene-level summaries. Recent efforts have shifted toward dense captioning, which requires both global description and the localization of multiple salient regions. However, collecting such fine-grained annotations at scale is substantially more demanding, motivating researchers to explore synthetic alternatives. One line of research reuses web-crawled alt-text, refining it with LLMs for greater clarity and descriptiveness [15, 62, 63]. Other methods prompt strong VLMs directly [13], fuses outputs from multiple VLMs [64], or enrich captions with predictions from external vision experts [14, 65, 66]. Most of these approaches rely on SFT to distill synthetic captions into a specialized captioner. RubiCap takes a different path: instead of imitating a fixed teacher, it trains via RL with rewards derived from ***fine-grained, sample-specific rubrics—letting the model discover better captions rather than merely reproducing existing ones***.

Reinforcement Learning in Visual Language Models. Reinforcement learning has recently driven major advances in LLM capabilities, particularly in domains where correctness can be automatically verified [23, 25]. In vision-language models, RL-based post-training has been applied to improve grounding [67], detection and classification [68], chart-based question answering [69], and tool-augmented visual mathematical reasoning [70]—tasks that all admit deterministic verification via IoU, classification accuracy, or multiple-choice correctness. ***Open-ended dense captioning, where outputs are free-form and no ground-truth verifier exists, presents a fundamentally harder challenge***. The closest concurrent work, CapRL [40], constructs multiple-choice questions from a strong VLM and treats derived accuracy as a proxy reward. However, MCQ options are inherently limited in coverage—any failure mode absent from the option set goes unpenalized, capping evaluation by design. RubiCap sidesteps this limitation. ***By assembling a diverse VLM committee to extract consensus-grounded rubrics, it produces open-ended evaluation criteria capable of surfacing failures that no fixed choice set could anticipate.***

B RUBRIC SYNTHESIS AND JUDGING PROMPTS

We first provide the prompts used to synthesize targeted rubrics from a committee of candidate captions: the system prompt in Prompt 1 and the user prompt in Prompt 2. Both prompts are sent to the rubric writer, which is instructed to return structured JSON outputs. Each rubric item contains the criterion topic, evaluation rule, severity weight, and a supporting justification traced back to teacher consensus.

Prompt 1: System Prompt for Rubric Writer.

```
You are an expert evaluator generating highly discriminative rubrics to assess image caption quality.

## Task
Identify the most discriminative criteria that separate excellent captions from weak or flawed ones. Focus on subtle but decisive quality differences that generic rubrics typically miss, while covering all critical dimensions of captioning performance.

## Rules you MUST follow:
1. Discriminative Power (Highest Priority)
– **Only** include criteria where the student model actually **fails** relative to teacher majority. Do NOT create rubrics for aspects the student model already handles correctly.
– Each rubric MUST meaningfully distinguish the weak student caption from teacher–consensus level performance.

2. Teacher Consensus as Ground Truth
– Ground truth = majority agreement among the five teacher models.
– A visual element, relationship, or interpretation is considered correct only if >= 3 teachers describe it accurately.

3. Weighting by Severity
– 3.0: Critical failures (main subject misidentification, hallucination of major elements, missing essential relationships)
– 2.0: Important but non–critical (secondary objects, spatial/contextual accuracy, attribute precision)
– 1.0: Minor polish (style fluency, phrasing clarity, minor detail richness)

4. Binary & Verifiable
– Every criterion must have a clear, objective pass/fail rule that can be verified by directly comparing the student caption against teacher consensus.

5. Quality over Quantity
– Prefer extremely important and sharp rubrics over many many generic ones.

## Output Requirements:
IMPORTANT: Return valid JSON object only, enclosed in triple backticks (```json). Do not include any additional text, explanations, or comments outside the JSON. Escape all quotes within string values using backslash (\"). Do not use single quotes or unescaped double quotes within JSON string values:
**JSON Structure:**
{
  "rubrics": [
    {
      "criterion": "Clear, specific criterion (e.g., Identifies the red bicycle in foreground)",
      "description": "Detailed explanation of what this measures and why it matters",
      "evaluation_rule": "Concrete rule with clear pass/fail condition",
      "weight": 1.0|2.0|3.0,
      "justification": "Explain the gap: what you see in the image, what teachers captured, what weak model missed/got wrong",
      "student_already_met": "True or False – Whether the weak model already satisfies this criterion",
      "reference_teachers": ["List of teacher model names that correctly satisfied this criterion"],
      "teacher_consensus": "Description of what the majority of reliable teachers agree on for this element"
    }
  ]
}
```

Prompt 2: User Prompt for Rubric Writer.

```
{{IMAGE}}
**Weak Model Output:**
{{WEAK_STUDENT_CAPTION}}

**Teacher Model Outputs:**
1. **Model 1:** {{CANDIDATE_1}}
2. **Model 2:** {{CANDIDATE_2}}
3. **Model 3:** {{CANDIDATE_3}}
```

4. **Model 4:** {{CANDIDATE_4}}
5. **Model 5:** {{CANDIDATE_5}}

Task:
1. Carefully examine the image and identify all important visual elements.
2. Determine teacher consensus (what the majority of the five teacher models describe correctly).
3. Evaluate the weak model's caption across all important dimensions of caption quality: accuracy, completeness, clarity, detail, relationships, and contextual interpretation.
4. For each dimension you choose to include, create one targeted binary rubric item with appropriate weight (1.0–3.0) using the required JSON structure.
5. Do NOT create rubrics for aspects the weak model already handles correctly.

Next, we present the prompt for LLM judge that provides RL rewards. Prompt 3 instructs the judge to evaluate generated caption rollouts against a given rubric criterion and its evaluation rule, assigning a binary satisfaction score. The returned JSON includes a reasoning trace alongside the score.

Prompt 3: User Prompt for RubiCap's Reward Provider.

You are a Quality Assurance Auditor for image captioning. Your task is to validate if a generated caption adheres to a specific criterion based on meaning and intent, not just keyword matching.

CRITERION TO EVALUATE (THE ONLY THING YOU USE TO CONSIDER):
– Criteria name: {criterion}
– Criteria description: {description}
– Evaluation rule: {evaluation_rule}

GENERATED CAPTION TO EVALUATE:
{generated_caption}

EVALUATION INSTRUCTIONS:
1. **Analyze Requirements:** Break down the criteria description and its evaluation rule to understand the core facts or concepts required.
2. **Check Semantic Equivalence:** Analyze the generated caption. Check if the required concepts are present, even if phrased differently.
   – Synonyms are acceptable (e.g., "automobile" instead of "car").
   – Structural changes are acceptable (e.g., "The man holds a cup" vs "A cup is held by the man").
3. **Assign Score:**
   – Score 1 (Pass): The caption conveys the correct *meaning* of the rule. It includes the necessary details, regardless of phrasing.
   – Score 0 (Fail): The caption explicitly misses key information, contradicts the facts, or hallucinates details not found in the rule.

RESPONSE FORMAT:
```json
{{
  "reasoning": "<detailed explanation of the score, why the caption does or does not fully meet the given criterion and its evaluation rule>",
  "score": <0 or 1>
}}
```

IMPORTANT: Return valid JSON object only, enclosed in triple backticks (```json). Do not include any additional text, explanations, or comments outside the JSON. Escape all quotes within string values using backslash (\"). Do not use single quotes or unescaped double quotes within JSON string values.

C Baseline Prompts

We provide the prompts for two baseline reward methods: Direct-Likert and Reference-Likert. Both employ a VLM-as-a-Judge approach, using Qwen2.5-VL-7B-Instruct as the judge. Prompts 4 and 5 instruct the judge to rate caption quality on a scale from 0 to 10. In Prompt 5, a reference caption is additionally provided to guide the evaluation.

Prompt 4: User Prompt for VLM Judge in Direct-Likert Method.

```
You are an expert image caption evaluator.

Generated Caption: {generated_caption}

Evaluate the generated caption on these criteria:
1. Accuracy: Does it correctly describe what's in the image?
2. Completeness: Does it cover the important details?
3. Quality: Is it well−written and coherent?

Provide a score from 0 to 10, where:
– 0–3: Poor quality, major inaccuracies or missing key information
– 4–6: Adequate, captures main elements but lacks detail or has minor issues
– 7–8: Good, accurate and detailed description
– 9–10: Excellent, comprehensive and high−quality caption

Respond with ONLY a single number from 0 to 10, nothing else.
```

Prompt 5: User Prompt for VLM Judge in Reference-Likert Method.

```
You are an expert image caption evaluator. Compare the generated caption with the reference caption for
↪ the given image.

Reference Caption: {reference_caption}
Generated Caption: {generated_caption}

Evaluate the generated caption on these criteria:
1. Accuracy: Does it correctly describe what's in the image?
2. Completeness: Does it cover the important details?
3. Quality: Is it well−written and coherent?

Provide a score from 0 to 10, where:
– 0–3: Poor quality, major inaccuracies or missing key information
– 4–6: Adequate, captures main elements but lacks detail or has minor issues
– 7–8: Good, accurate and detailed description
– 9–10: Excellent, comprehensive and high−quality caption

Respond with ONLY a single number from 0 to 10, nothing else.
```

D EVALUATION PROMPTS

We provide the evaluation prompts used throughout our experiments. Prompt 6, adopted from CapArena [31], is used to compute pairwise win-lose rates as judged by GPT-4.1. Caption assignments are randomized to mitigate position bias, and the judge returns a detailed justification along with the winner decision.

Prompt 7 instructs the evaluator to blindly rank captions and assess quality along four dimensions: *accuracy, completeness, clarity, and hallucination penalty (lower is better)*. We evaluate five captions: those generated by RubiCap-7B, Qwen2.5-VL-72B-Instruct, Qwen2.5-VL-32B-Instruct, Qwen2.5-VL-7B-Instruct (the student base model), and an expert-annotated caption from PIXMOCAP. Here we use GPT-5.4 as evaluator to return per-dimension scores and an overall ranking.

Prompt 6: Used Prompt for Win–Lose Rate in CapArena.

```
Given an image and two candidate captions, you are required to determine which of the two captions is
↪ better.

Below are some guidelines for your reference:

### What to evaluate
```

1. **Precision**: The caption should accurately correspond to the content of the image, providing precise information about it. Common examples of imprecision include errors in color, quantity, spatial relationships, or the posture of people.
2. **Informativeness**: Salient information in the image should be reflected in the caption. Since it is impossible to include every detail, you will need to subjectively judge which aspects of the image are important. For instance, describing an otter as "a small animal" is precise, but it is less informative than specifying "an otter".
3. **Hallucination**: Captions that include descriptions of objects or elements that are clearly absent from the image should be significantly penalized.
4. **Attention to detail**: Annotators should pay close attention to the details in the image to distinguish the quality of the descriptions.
5. **Assistive description**: Imagine a visually impaired person asking you to describe the image for them. How would you convey the image to them?
6. **Reverse thinking**: What image does the caption lead us to imagine? Does the caption effectively lead you to imagine the intended image?
7. **Ties are acceptable**: If you find it genuinely difficult to determine which caption is better (e.g., both captions are excellent), marking a tie is acceptable. While the above guidelines provide a framework, they cannot cover all possible cases. Therefore, we encourage you to make **subjective judgments** based on the specific circumstances and your own reasoning about which caption is better.

What to ignore
When comparing the captions, **do NOT consider differences in**:
– Writing style or phrasing
– **Caption length**
– Grammatical variations

Caption A: {caption_a}
Caption B: {caption_b}

Format your response in JSON format.

RESPONSE FORMAT:
```json
{{
  "reason": "<detailed explanation of the judgment>",
  "judgment": "<A, B, or, Tie>"
}}
```

Prompt 7: User Prompt to rank captions and assess caption quality.

You are an expert image captioning evaluator. Given the image above and the 5 captions below, rigorously assess each caption.

Scoring
Score each caption on four dimensions (integers 0–10):

1. accuracy – Are the described objects, actions, text, colors, and spatial relationships factually correct for this image? Penalize for any wrong attribute, misidentified object, or incorrect action.

2. completeness – Does the caption cover all visually significant elements (main subjects, notable actions, background context, on–screen text if present)? Penalize for missing key details.

3. clarity – Is the caption well–written, specific, grammatically correct, and unambiguous? Penalize for vague language or redundancy.

4. hallucination_penalty – Does the caption assert things NOT visible in the image? 0 = zero hallucination; 10 = pervasive fabrication. Be strict: even plausible but unverifiable claims count as mild hallucination (2–4). This score is applied as a penalty.

Compute: total_score = (accuracy + completeness + clarity) / 3.0 – hallucination_penalty x 1.5

Captions to Evaluate

```
{captions_text}

## Output Format
Respond ONLY with a single valid JSON object – no markdown fences, no extra text.

{
  "assessments": {
    "Caption A": {
      "justification": "<2–3 sentences citing specific visual evidence from the image>",
      "accuracy": <int 0–10>,
      "completeness": <int 0–10>,
      "clarity": <int 0–10>,
      "hallucination_penalty": <int 0–10>,
      "total_score": <float>
    },
    ...
    "Caption E": { ... }
  },
  "ranking": ["Caption X", "Caption X", "Caption X", "Caption X", "Caption X"]
}

The "ranking" list must contain all 5 caption labels ordered from best (index 0) to worst (index 4), sorted
↪ strictly by total_score descending.
```

E SYNTHESIS COST

The teacher committee is queried only once per image during rubric construction, so rubric synthesis is a one-time offline cost rather than an inference cost repeated at every training step; an online setup, in which rubrics are re-synthesized against the current policy, would be *far more expensive and would add substantial latency to every step* [33]. We leave extending RubiCap to online rubric construction as our future work.

Taking PIXMOCAP (50K images) as an example, generating teacher captions with GPT-5 and Gemini-2.5-Pro costs $517.44, and writing rubrics with Gemini-2.5-Pro costs $110.75, for a total of about $628, or $0.013 per image. Once the rubrics are synthesized, RL training requires only a lightweight judge, Qwen2.5-7B-Instruct, run locally, with *no* further reliance on proprietary models. By contrast, commercial human annotation for PixMoCap costs roughly $5–10 per image, about 400–800$\times$ more. The resulting model can then serve as an annotator at no API cost and far lower compute than the 72B and 32B models, while delivering competitive caption quality. ***Overall, RubiCap produces higher-quality captions than human annotation at a tiny fraction of the cost.***

F ALGORITHM OVERVIEW

We summarize the two stages of RubiCap as unified pseudocode in Algorithm 1. The prompt used in Stage 1 (Sec. 2.2) is a general, extensible template that any user can steer the rubric's focus toward specific areas of weakness by design.

G TRAINING DETAILS

We now detail the training configurations and computational resources used in our experiments. We sample 50,000 images from PIXMOCAP[2] and DENSEFUSION[3], holding out 500 for evaluation and using the rest for training. We use Qwen VLMs of different scales as the student models and Qwen2.5-7B-Instruct, a text-only LLM, as the reward provider that scores caption rollouts against the synthesized rubrics. All experiments are conducted on 8 NVIDIA H100 GPUs.

[2] https://huggingface.co/datasets/allenai/pixmo-cap
[3] https://huggingface.co/datasets/BAAI/DenseFusion-1M

Algorithm 1 RubiCap

Require: Image dataset $\mathcal{D}$, student policy π_{θ_s}, teacher committee $\mathcal{T} = \{\mathcal{T}_k\}_{k=1}^{K}$, LLM Rubric Writer, LLM Judge, number of rollouts N, clipping threshold ϵ, severity weights $w \in \{1, 2, 3\}$.

{**Stage 1: Automated Rubric Synthesis (Sec. 2.2)**}
for each image $x \in \mathcal{D}$ **do**
  Generate teacher captions: $\mathcal{C}^{\text{teacher}}(x) = \{c_k^{\text{teacher}}\}_{k=1}^{K}$ where $c_k^{\text{teacher}} \leftarrow \mathcal{T}_k(x)$
  Generate student caption: $c^{\text{student}} \sim \pi_{\theta_s}(\cdot \mid x)$
  Step 1: Extract consensus elements agreed upon by $\geq \lceil K/2 \rceil$ teachers conditioned on x
  Step 2: Diagnose discriminative student deficiencies by comparing c^{student} against teacher consensus; categorize by severity (critical, important, minor)
  Step 3: Formulate targeted rubrics: $\mathcal{R}(x, c^{\text{student}}, \mathcal{C}^{\text{teacher}}(x)) = \{(r_m, w_m)\}_{m=1}^{M}$, where each r_m is a binary criterion with severity weight w_m
end for
{**Stage 2: Rubric-Guided Reinforcement Learning (Sec. 2.3)**}
while not converged **do**
  Sample a batch of images $x \sim \mathcal{D}$
  Sample N rollouts: $\{c_i^{\text{student}}\}_{i=1}^{N} \sim \pi_{\theta_s}(\cdot \mid x)$
  for each rollout $i \in \{1, \ldots, N\}$ **do**
    LLM Judge evaluates each criterion: $\hat{y}_{i,m} \in \{0, 1\}$ for all $r_m \in \mathcal{R}(x)$
    Compute reward: $G_i = \frac{\sum_{m=1}^{M} w_m \cdot \hat{y}_{i,m}}{\sum_{m=1}^{M} w_m}$
  end for
  Estimate advantages: $A_i = \frac{G_i - \text{mean}(\{G_j\}_{j=1}^{N})}{\text{std}(\{G_j\}_{j=1}^{N})}$
  Update θ_s by minimizing $\mathcal{L}_{\text{GRPO}}(\theta_s)$
end while

We optimize the student policy using GRPO [23] with the following hyperparameters: a learning rate of 1×10^{-5} with cosine scheduling and a warmup ratio of 0.01, a maximum completion length of $1{,}024$ tokens, and $N = 4$ generations per image. Training runs for 1 epoch. All the RL baseline methods employ the same training configuration, except for the reward design. During rollout generation, we prompt the student with: *"You are an expert at describing images in detail. Provide comprehensive, accurate captions that describe the main subjects, their actions, the setting, and important visual details in the image."*

For the supervised fine-tuning (SFT) baseline, we use a learning rate of 1×10^{-5} and train for one epoch. We explore several supervision signals for SFT: **(i)** expert-refined annotations from PIXMOCAP, **(ii)** GPT-4V-augmented captions from DENSEFUSION, **(iii)** captions distilled from Qwen2.5-VL-72B-Instruct, and **(iv)** generated captions revised to incorporate our synthesized rubrics, which we refer to as rubric-augmented SFT.

H CAPTIONING RESULTS

We evaluate caption quality with a wide range of metrics spanning several families. The first is LLM-judge assessments: win/loss rates in CapArena [31], MCQ-based scores in CaptionQA [32], and a blind ranking LLM evaluation [51]. We also report scores from models fine-tuned specifically for caption evaluation, such as CAPTURE [44] and SPECS [45], both of which have been shown to correlate well with human judgments. We further include three model-free metrics that measure lexical overlap with reference captions: ROUGE-L [27], METEOR [46], and BLEU [47]. We report these for completeness, though they are known to correlate weakly with human judgment on open-ended captioning. Finally, we present a human study where recruited annotators compare captions in a pairwise arena, from which we compute Elo scores and pairwise win rates (Appendix K). ***This broad coverage ensures that our evaluation is not tied to any single LLM judge and provides a reliable assessment across metric families.***

We describe several metric below and then present the complete numerical results in Table 2 (RubiCap-7B), Table 3 (RubiCap-3B), and Table 4 (RubiCap-2B).

Table 2: Full captioning evaluation results for RubiCap-7B.

	CapArena Win Rate (v.s. Base Model)	CapArena Win Rate (v.s. Expert-Annotated)	CAPTURE	SPECS	ROUGE-L	METEOR	BLEU-4	Average
Base Model (Qwen2.5-VL-7B-Instruct)	50.00	54.00	57.63	69.39	23.18	18.32	8.05	40.08
GRPO Fine-tuning on 50k Images Sampled from PixMoCap Dataset								
NLP Metric-based RL	59.60	56.20	59.66	72.38	24.71	18.71	9.47	42.96
Direct-Likert	66.20	61.00	59.30	75.22	23.60	19.93	7.55	44.69
Reference-Likert	67.20	59.40	59.33	74.54	22.47	20.26	6.56	44.25
RubiCap-7B-PixMoCap	70.80	67.40	59.42	75.47	23.73	19.93	7.53	**46.33**
Performance Gain	*+20.80*	*+13.40*	*+1.79*	*+6.08*	*+0.55*	*+1.61*	*-0.52*	*+6.25*
	CapArena Win Rate (v.s. Base Model)	**CapArena Win Rate (v.s. GPT-4V Augmented)**	**CAPTURE**	**SPECS**	**ROUGE-L**	**METEOR**	**BLEU-4**	**Average**
Base Model (Qwen2.5-VL-7B-Instruct)	50.00	44.80	60.64	69.58	27.25	19.42	11.54	40.46
GRPO Fine-tuning on 50k Images Sampled from DenseFusion Dataset								
NLP Metric-based RL	58.40	45.00	62.35	77.52	28.90	21.50	13.30	43.85
Direct-Likert	59.20	48.20	62.01	74.90	28.38	21.24	12.75	43.81
Reference-Likert	50.60	38.40	62.11	73.43	26.39	23.17	9.89	40.57
RubiCap-7B-DenseFusion	64.40	53.20	62.21	74.51	28.58	21.15	12.80	**45.26**
Performance Gain	*+14.40*	*+8.40*	*+1.57*	*+4.93*	*+1.33*	*+1.73*	*+1.26*	*+4.80*

Table 3: Full captioning evaluation results for RubiCap-3B.

	CapArena Win Rate (v.s. Base Model)	CapArena Win Rate (v.s. Expert-Annotated)	CAPTURE	SPECS	ROUGE-L	METEOR	BLEU-4	Average
Base Model (Qwen2.5-VL-3B-Instruct)	50.00	34.00	56.87	68.83	21.76	17.98	6.50	36.56
CapRL-3B	59.40	34.40	59.07	0.00	18.46	20.75	4.08	28.02
GRPO Fine-tuning on 50k Images Sampled from PixMoCap Dataset								
NLP Metric-based RL	54.00	35.80	58.12	71.22	24.31	17.83	9.14	38.63
Direct-Likert	64.00	37.40	58.35	69.73	23.11	18.36	7.59	39.79
Reference-Likert	7.80	4.00	13.33	16.80	7.27	2.13	0.01	7.33
RubiCap-3B-PixMoCap	68.60	42.80	57.89	69.49	23.30	18.16	8.05	**41.18**
Performance Gain	*+18.60*	*+8.80*	*+1.02*	*+0.66*	*+1.54*	*+0.18*	*+1.55*	*+4.62*
	CapArena Win Rate (v.s. Base Model)	**CapArena Win Rate (v.s. GPT-4V Augmented)**	**CAPTURE**	**SPECS**	**ROUGE-L**	**METEOR**	**BLEU-4**	**Average**
Base Model (Qwen2.5-VL-3B-Instruct)	50.00	20.00	58.68	68.03	25.73	18.84	11.06	36.05
CapRL-3B	45.40	15.20	60.65	0.00	21.73	23.49	6.51	24.71
GRPO Fine-tuning on 50k Images Sampled from DenseFusion Dataset								
NLP Metric-based RL	50.80	19.40	60.91	77.14	28.68	20.54	12.74	38.60
Direct-Likert	53.20	23.80	60.28	69.44	27.27	18.95	11.35	37.76
Reference-Likert	48.60	17.80	60.25	65.66	25.47	21.62	9.59	35.57
RubiCap-3B-DenseFusion	59.20	26.20	59.67	69.33	27.20	18.52	10.97	**38.73**
Performance Gain	*+9.20*	*+6.20*	*+0.99*	*+1.30*	*+1.47*	*-0.32*	*-0.09*	*+2.68*

- **CapArena Win Rate**: A pairwise preference-based metric in which a strong VLM judge compares two captions and selects the better one. The win rate measures how often the model's caption is preferred over a reference across pairwise comparisons. We use GPT-4.1 as the judge.
- **Blind Ranking**: Following the EXPERT metric, we break down captioning assessment into four dimensions: *accuracy, completeness, clarity, and hallucination penalty*. Higher scores are better for the first three dimensions, while a lower score is better for hallucination penalty. As before, we anonymize model names to prevent authority bias. We rank the models on each dimension's scores to obtain the rank distribution. We use GPT-5.4 (shown in the main paper), GPT-4.1, GPT-5, and Claude Sonnet 5 as judges.
- **CAPTURE**: A reference-based metric that uses a T5-based parser [71] to extract structured visual elements (e.g., objects, attributes, and relations) from both the candidate and ground-truth captions, filtering out stop words. Precision and recall are then computed by matching elements via exact, synonym, and soft (cosine similarity-based) methods. CAPTURE has shown higher consistency with expert judgments than other rule-based caption metrics.
- **SPECS**: A reference-free representational similarity metric tailored to long image captioning. SPECS builds on LongCLIP [9] with a training objective that decomposes captions into fine-

Table 4: Full captioning evaluation results for RubiCap-2B.

	CapArena Win Rate (v.s. Base Model)	CapArena Win Rate (v.s. Expert-Annotated)	CAPTURE	SPECS	ROUGE-L	METEOR	BLEU-4	Average
Base Model (Qwen2-VL-2B-Instruct)	50.00	28.80	57.33	67.34	22.14	17.24	6.40	35.61
GRPO Fine-tuning on 50k Images Sampled from PixMoCap Dataset								
NLP Metric-based RL	52.20	27.00	57.63	69.02	23.97	16.56	7.89	36.32
Direct-Likert	57.20	31.40	57.07	64.57	22.87	16.35	7.51	36.71
Reference-Likert	9.00	2.60	25.72	22.43	12.03	4.72	1.06	11.08
RubiCap-2B-PixMoCap	61.60	34.60	57.19	64.40	23.08	16.17	7.75	**37.83**
Performance Gain	*+11.60*	*+5.80*	*-0.14*	*-2.94*	*+0.94*	*-1.07*	*+1.35*	*+2.22*
	CapArena Win Rate (v.s. Base Model)	**CapArena Win Rate (v.s. GPT-4V Augmented)**	**CAPTURE**	**SPECS**	**ROUGE-L**	**METEOR**	**BLEU-4**	**Average**
Base Model (Qwen2-VL-2B-Instruct)	50.00	19.40	59.80	67.49	26.96	19.03	11.60	36.33
GRPO Fine-tuning on 50k Images Sampled from DenseFusion Dataset								
NLP Metric-based RL	40.80	13.20	59.84	72.50	28.06	19.10	11.45	34.99
Direct-Likert	49.40	14.20	58.41	63.88	26.34	17.26	9.41	34.13
Reference-Likert	42.20	13.00	58.34	64.20	25.91	18.05	10.39	33.16
RubiCap-2B-DenseFusion	52.40	22.00	58.14	63.54	26.59	16.91	9.47	**35.58**
Performance Gain	*+2.40*	*+2.60*	*-1.66*	*-3.95*	*-0.37*	*-2.12*	*-2.13*	*-0.75*

grained detail units and applies contrastive, positive, and negative losses to reward accurate visual details while penalizing inaccuracies.

- **ROUGE-L**: A text overlap metric that measures the Longest Common Subsequence (LCS) between the generated and reference captions.
- **METEOR**: A reference-based metric that goes beyond exact word matching by incorporating synonyms, stemming, and paraphrase matching.
- **BLEU-4**: A precision-based metric that counts how many 4-gram sequences in the generated caption also appear in the reference, with a brevity penalty for short outputs.

Results. ***RubiCap consistently outperforms all RL baseline methods across both datasets and all model scales on average (see Tables 2, 3, and 4).*** The largest gains appear at 7B, with average improvements of **+6.25** on PixMoCap and **+4.80** on DenseFusion, and CapArena win rates reaching **70.80%** against the base model (**+20.80**). At 3B, RubiCap maintains clear gains (**+4.62** and **+2.68** on average) while substantially outperforming CapRL-3B, which collapses to a SPECS score of 0.00 on both datasets. Notably, the Reference-Likert baseline fails at 3B and 2B (averaging 7.33 and 11.08), exhibiting severe self-praising behavior and reward hacking—a failure mode that RubiCap avoids entirely. To rule out judge-specific bias, we repeat the blind ranking evaluation with three additional judges from different model families: GPT-4.1, GPT-5, and Claude Sonnet 5 (Fig. 7). ***RubiCap ranks first under all four judges***, confirming that its advantage is not an artifact of any single evaluator.

I Mitigating Catastrophic Forgetting

Setup. Continued supervised fine-tuning can cause the model to forget its pretrained capabilities. To assess this, we evaluate the fine-tuned models on 9 VLM benchmarks using VLMEvalKit [72]: SEEDBench [73], A-OKVQA [74], BLINK [75], ChartQA [55], ScienceQA [60], DocVQA [76], InfoVQA [77], TextVQA [78], and OCRBench [79]. We report the average performance across all benchmarks as a measure of generalization retention.

Results. We first focus on 7B-scale models, results are shown in the rightmost panels of Fig. 2, with 3B- and 2B-scale results provided in Table 5 (3B) and in Table 6 (2B). Across all model scales, RubiCap consistently achieves the highest average performance over the 10 benchmarks, substantially outperforming SFT-based models, which exhibit the largest degradation. These results underscore the limitations of SFT on open-ended tasks, where it often impairs generalization, whereas RubiCap training more effectively mitigates catastrophic forgetting.

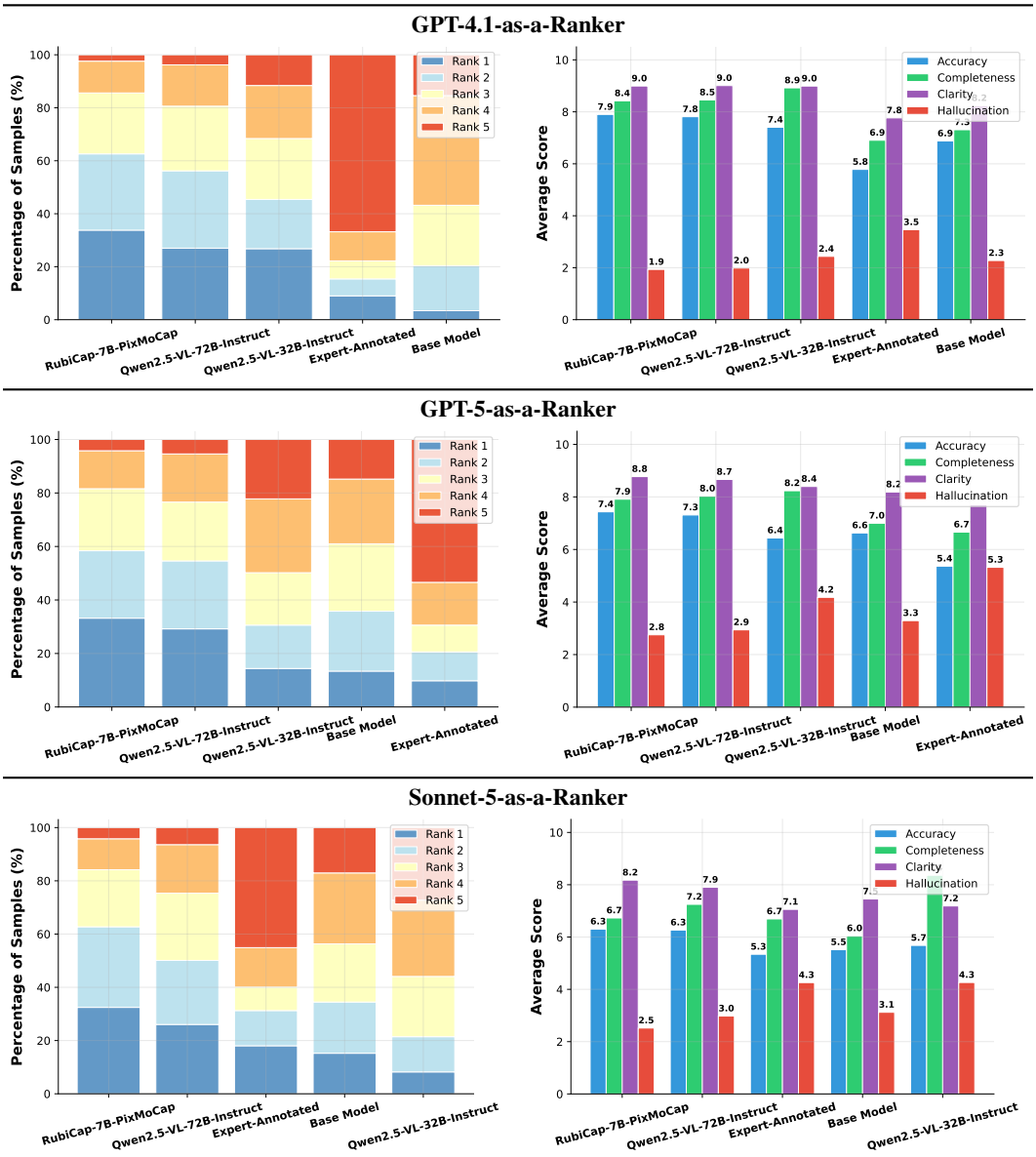


Figure 7: Blind ranking with alternative judges. We repeat the blind ranking of Fig. 4 in the PIXMOCAP setting using GPT-4.1 (top), GPT-5 (middle), and Claude Sonnet 5 (bottom) as judges. **Left:** rank distribution per model. **Right:** sub-metric breakdown per model. Under all three judges, RubiCap again receives the highest proportion of rank-1 assignments, confirming that its advantage holds across judges from different models.

J ABLATION STUDIES

This section presents an extensive ablation study that isolates the contribution of each design choice in RubiCap. We first examine three aspects of per-sample rubric construction: **(i)** the rubric synthesis strategy, **(ii)** the size of the teacher committee; and **(iii)** rubric granularity. We then examine two aspects of RL training: **(iv)** the choice of reward provider; and **(v)** severity-based reweighting versus uniform weighting of rubric items. In all studies, we use Qwen2.5-VL-3B-Instruct as the student base model and fine-tune it on the PixMoCap dataset, which serves as a proxy for the full system.

Rubric Synthesis Strategy. We first ablate how the rubric writer is prompted, comparing three strategies. *Teacher-only*: the rubric writer receives only the candidate captions generated by the teachers and is asked to derive evaluation criteria from them. *Student-aware*: the rubric writer additionally receives the student's caption, but is not explicitly instructed to focus on the student's

Table 5: VLM benchmark results for RubiCap-3B

	SEEDBench	A-OKVQA	BLINK	ChartQA	ScienceQA	DocVQA	InfoVQA	TextVQA	OCRBench	Average
Base Model (Qwen2.5-VL-3B-Instruct)	73.45	85.05	47.32	82.83	80.46	92.60	75.42	78.49	83.00	77.62
GRPO Fine-tuning on 50k Images Sampled from PixMoCap Dataset										
Metric-based RL	72.48	83.22	31.32	75.59	57.14	87.80	70.02	66.35	79.00	69.21
Direct-Likert	73.78	85.05	34.37	79.51	60.96	87.39	70.98	68.35	81.00	71.27
Reference-Likert	20.27	18.36	10.74	68.51	24.70	85.36	65.35	66.00	66.70	47.33
RubiCap-3B-PixMoCap	73.71	84.27	46.26	80.03	79.61	90.05	72.23	75.32	83.30	76.09
GRPO Fine-tuning on 50k Images Sampled from DenseFusion Dataset										
Metric-based RL	68.48	79.55	30.00	73.91	47.72	87.78	69.96	66.12	80.20	67.08
Direct-Likert	73.76	83.92	38.37	78.39	69.10	88.79	71.86	68.83	79.40	72.49
Reference-Likert	39.90	52.54	7.79	70.23	24.01	85.79	68.89	63.17	79.00	54.59
RubiCap-3B-DenseFusion	74.01	84.62	45.37	80.15	80.56	90.36	71.99	75.73	81.50	76.03

Table 6: VLM benchmark results for RubiCap-2B

	SEEDBench	A-OKVQA	BLINK	ChartQA	ScienceQA	DocVQA	InfoVQA	TextVQA	OCRBench	Average
Base Model (Qwen2-VL-2B-Instruct)	71.52	83.83	38.95	70.59	76.24	89.19	63.82	74.80	80.50	72.16
GRPO Fine-tuning on 50k Images Sampled from PixMoCap Dataset										
Metric-based RL	41.03	56.56	16.32	55.38	54.71	80.73	53.39	68.70	51.19	53.11
Direct-Likert	70.81	79.46	29.63	67.47	70.69	84.22	57.03	71.73	71.72	66.97
Reference-Likert	12.23	18.97	3.90	51.50	35.62	79.92	51.65	60.65	29.90	38.26
RubiCap-2B-PixMoCap	71.48	81.91	36.95	70.03	76.09	87.28	61.15	73.68	78.14	70.75
GRPO Fine-tuning on 50k Images Sampled from DenseFusion Dataset										
Metric-based RL	59.31	71.59	24.00	60.58	59.43	82.16	54.80	71.45	61.59	60.55
Direct-Likert	70.90	82.26	28.32	64.47	70.39	83.51	57.62	71.21	67.44	66.24
Reference-Likert	58.41	68.71	23.53	66.99	62.40	83.80	58.04	71.29	68.89	62.45
RubiCap-2B-DenseFusion	71.82	82.08	33.95	69.91	76.39	87.47	60.62	73.46	77.19	70.32

Table 7: Ablation studies on RubiCap-3B. All variants use Qwen2.5-VL-3B-Instruct as the student base model and are fine-tuned on 50k images sampled from the PixMoCap dataset. Rows marked *chosen* denote our final configuration.

	CapArena Win Rate (v.s. Base Model)	CapArena Win Rate (v.s. Expert-Annotated)	CAPTURE	SPECS	ROUGE-L	METEOR	BLEU-4	Average
Base Model (Qwen2.5-VL-3B-Instruct)	50.00	34.00	56.87	68.83	21.76	17.98	6.50	36.56
(i) Rubric Synthesis Strategy								
Teacher-only	65.60	39.20	58.16	69.09	23.53	18.13	8.30	40.29
Student-aware	63.40	41.40	58.04	69.19	23.39	18.02	8.29	40.25
Failure-focused (*chosen*)	69.00	41.80	57.74	69.18	23.25	17.91	8.13	**41.00**
(ii) Teacher Committee Size								
1 Teacher	58.20	37.20	58.30	71.12	24.33	17.99	9.23	39.48
3 Teachers	65.20	40.40	58.25	69.27	23.14	18.14	7.72	40.30
5 Teachers (*chosen*)	67.60	42.00	58.12	69.39	23.52	18.19	8.52	**41.05**
(iii) Rubric Granularity								
Fixed General Criteria	55.60	32.80	58.14	69.43	23.00	17.93	7.76	37.81
Sample-specific Rubrics (*chosen*)	67.60	42.00	58.12	69.39	23.52	18.19	8.52	**41.05**
(iv) Reward Provider								
72B VLM Judge	68.40	43.00	58.38	70.53	23.40	18.22	8.26	**41.46**
7B LLM Judge (*chosen*)	68.60	42.80	57.89	69.49	23.30	18.16	8.05	41.18
(v) Severity Reweighting								
Uniform Weighting	64.00	40.20	57.08	67.63	22.11	17.80	7.41	39.46
Severity Reweighting (*chosen*)	66.40	42.20	57.23	69.19	22.24	18.30	7.42	**40.43**

deficiencies. As a result, the synthesized rubrics may include criteria the student already satisfies. *Failure-focused*: the rubric writer is instructed to identify specifically where the student's caption fails to capture the teachers' consensus.

Results are shown in Table 7. The failure-focused strategy yields the highest win rate against the base model, indicating the strongest self-improvement, as well as the highest win rate against expert-annotated captions. This approach also achieves the best performance on average. In contrast, the

teacher-only and student-aware strategies are suboptimal. Since neither strategy directs the rubric writer toward the student's specific shortcomings, many of the resulting criteria may describe content the student already captures. Such criteria tend to be satisfied by most caption rollouts and therefore contribute less to distinguishing better captions from worse ones, which weakens the reward signal and learn inefficiently.

Teacher Committee Size. RubiCap draws on the collective expertise of multiple teacher models to construct rubrics, with the goal of broadening the coverage of visual details and perspectives that the rubrics capture. However, each additional teacher increases the inference cost of rubric synthesis. To study this trade-off, we vary the committee size among three settings: a single teacher (Gemini 2.5 Pro); three teachers (Gemini 2.5 Pro, GPT-5, and Qwen2.5-VL-72B-Instruct); and five teachers, which add Gemma-3-27B-IT and Qwen3-VL-30B-A3B-Instruct to the three-teacher committee. Among these, only Gemini 2.5 Pro and GPT-5 incur API costs. We report the resulting synthesis cost in Appendix E.

Table 7 shows that average performance improves steadily as more teachers contribute to rubric construction. Notably, adding smaller and weaker models (Gemma-3 and Qwen3-VL in our case) does not introduce noise that degrades performance. Instead, these models contribute additional perspectives that help the rubric writer reach a more reliable consensus. As a result, the rubric writer identifies the student's shortcomings more accurately and produces a more reliable reward signal, improving average performance by +1.57.

Rubric Granularity. Next, we ask whether rubrics must be sample-specific, or whether a fixed set of general criteria suffices as a reward signal. We replace the per-sample rubrics with four criteria shared across all images: **(1)** clarity and fluency, **(2)** conciseness, i.e., no repetition or filler words, **(3)** no self-evaluative language, e.g., "this caption accurately describes…", and **(4)** coherence and informativeness. A VLM judge (Qwen2.5-VL-7B-Instruct) scores each criterion on a 0–5 scale while seeing the image, so any performance gap reflects the granularity of the criteria, not the judge's lack of visual grounding.

Table 7 shows that fixed criteria substantially degrade performance: the resulting model barely beats the base model, with a win rate just above 50%, and underperform the sample-specific rubrics by 3.24 on average. The reason is that general criteria ask whether a caption reads well, not whether it mentions the objects, attributes, and relations in this particular image. Even with the image in view, the judge has no fixed target against which to flag a missing object or a hallucinated detail.

Reward Provider. In the current setting, we use Qwen2.5-7B-Instruct, a text-only LLM, as the judge that reads each rollout and checks it against each rubric criterion. We ask whether a larger, stronger, vision-capable judge yields better rewards. Specifically, we replace it with Qwen2.5-VL-72B-Instruct during RL training, keeping all other settings unchanged.

As shown in Table 7, the 72B VLM judge improves average performance by only $+0.28$. However, this improvement comes at a substantial cost: the larger judge requires far more computation and GPU memory, which slows reward computation and, in turn, overall training. Moreover, the small gap suggests that most of the reward quality comes from the sample-specific rubrics rather than from the judge's capacity or its access to the image: once a rubric specifies exactly what to check, verifying each criterion is a simple task that a 7B text-only model handles well.

Severity Reweighting. Finally, we compare severity-based reweighting with uniform weighting of rubric items. In RubiCap, the rubric writer categorizes each of the student's failures as *minor*, *important*, or *critical*, and the corresponding rubric items are weighted by 1.0, 2.0, and 3.0, respectively. In the uniform setting, all rubric items receive the same weight.

Table 7 shows that uniform weighting leads to lower performance. Under uniform weighting, fixing a minor omission earns the same reward as correcting a critical error, such as a missing main object or a hallucinated detail. As a result, the policy may focus on easy, low-impact items while leaving its most serious failures unaddressed. Severity-based reweighting instead directs the reward toward the failures that matter most for caption quality.

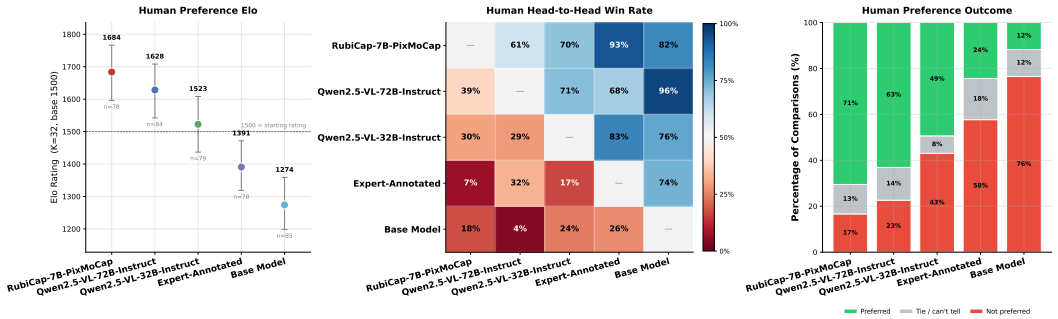


Figure 8: Human annotators prefer RubiCap captions over both larger models and references. **Left**: Elo ratings with 95% bootstrap intervals. **Middle**: pairwise win rates, row over column, ties counted as half a win. **Right**: the win/tie/loss share of each source's comparisons.

K Human Evaluation

In addition to LLM-based and specialized-model evaluation, we conduct a human study on our PixMoCap test split with recruited expert annotators.

Setup. Following the blind ranking protocol, annotators compare five caption sources in pairs: RubiCap-7B-PixMoCap, Qwen2.5-VL-72B-Instruct, Qwen2.5-VL-32B-Instruct, Qwen2.5-VL-7B-Instruct (the base model), and the reference captions from PixMoCap. We built an in-house annotation platform that presents an image alongside two captions, anonymized as Caption A and Caption B; annotators select one or declare a tie. We convert these votes into Elo scores ($K = 32$, base rating 1500, ties counted as 0.5).

Results and Observations. Fig. 8 presents the results. RubiCap-7B receives the highest Elo rating among all sources and is chosen by annotators in 70% of the comparisons it appears in. In head-to-head comparisons, it wins 61% against Qwen2.5-VL-72B-Instruct and 70% against Qwen2.5-VL-32B-Instruct (see middle panel). We show that human annotators prefer RubiCap's captions to those of larger and stronger models, supporting our LLM-judge blind ranking results in Fig. 4.

Annotators also noted several stylistic differences between sources. **First**, the 32B model usually writes long captions with heavy Markdown structure, often nested bullet points, which annotators found hard to read as image descriptions. It tries to cover more details than the 72B model—for example, breaking the scene into individual visual elements and organizing them under headings—but hallucinates more in doing so, and tends to infer context rather than describe what is visible. This results in a lower tie rate when the 32B model is one of the two options. **Second**, the 72B model and RubiCap adapt length to visual complexity, writing more for rich scenes and less for simple ones. Compared to the 32B model, they achieve higher token efficiency when describing an easy image clearly. Moreover, both name specific entities, such as the model of a machine gun or the brand behind a logo—a level of specificity we did not observe in the expert-written annotations. **Third**, human annotations differ from model outputs: annotators write dense caption without Markdown and are inconsistent with capitalization. They also omit several identifiers, labels, and minor watermarks when a string is hard to transcribe or pronounce, such as stylized names like "6ix9ine". However, human annotations are more precise on object counting. **Finally**, the base model usually produces non-English text, including Chinese, Japanese, and Arabic, within English captions we ask for.

L Qualitative Examples

We present qualitative caption comparisons sampled from both datasets, comparing RubiCap-7B against its base model (Qwen2.5-VL-7B-Instruct) in Fig. 9 and the larger Qwen2.5-VL-72B-Instruct in Fig. 10 and Fig. 11. A GPT-4.1 judge selects the preferred caption, with its reasoning shown in each caption block.



Figure 9: Qualitative comparison to base model using RubiCap-7B-DenseFusion.



Figure 10: Qualitative comparison to 72B frontier using RubiCap-7B-PixMoCap.



Figure 11: Qualitative comparison to 72B frontier using RubiCap-7B-DenseFusion.