Towards Multimodal Agents with Cognitive Modality Planning and Reflective Reasoning

Anonymous authors
Paper under double-blind review

Abstract

Multimodal agents must decide which perceptual modality to read, when to stop refining their belief, and how to keep an answer honest about its own uncertainty. Existing pipelines either run a single fixed toolchain (no planning), or invoke every modality and then reflect (no budget), or rely on a heuristic LLM-as-judge for confidence (no calibration). We introduce CoMPR (Cognitive Modality Planning + Reflective Reasoning), a training-friendly framework that formulates modality selection as a budgeted submodular maximization, runs a short chain of reflective revise-and-verify passes with a temperature-calibrated verifier, and is post-trained with GRPO under a composite reward that explicitly rewards evidence-grounded, calibrated, budget-efficient behaviour. CoMPR + a GRPO warm-start (CoMPR+RL) is evaluated on five multimodal QA benchmarks and two agentic benchmarks across four backbone LLMs, and compared against eleven strong baselines including CoT, self-consistency, ReAct, Reflexion, Self-Refine, Critic, Set-of-Mark, random planning, all-modalities, and the recent CogPlanner and DeepEyes agents. On Qwen2.5-VL-7B CoMPR+RL improves mean accuracy by +5.0 over the strongest baseline, reduces expected calibration error from 0.148 to 0.031, and shortens latency by ∼17%, while remaining backbone-agnostic and competitive with the much larger GPT-4o.

1 Introduction

Multimodal large language models (MLLMs) now answer questions about diagrams, charts, documents and natural images with reasonable fluency, but they still struggle with three structural problems. First, modality selection — given a question and an image, the model has to decide which perceptual operation to invoke (read OCR, parse a chart, retrieve a region, zoom in, etc.), and the wrong choice is expensive: a chart question answered from the global view costs the same as one answered from the parsed chart series, but the latter is right and the former is not. Second, reflection without a budget — contemporary methods such as Reflexion, Self-Refine and Critic re-prompt the model one, two, sometimes three more times, but none of them decides when to stop in a principled way; they either use a fixed round count or an LLM-as-judge that is itself uncalibrated. Third, post-training that ignores the pipeline — RL post-training of MLLMs (GRPO, DPO, RLAIF) typically uses a single accuracy reward and treats the agent's tool calls as an unpenalized constant, so the resulting policy drifts toward "always call every tool".

We argue that these three problems are tightly coupled and should be solved jointly: a multimodal agent should plan which modalities to consult under a hard budget, reflect on the resulting evidence until a calibrated verifier says stop, and be trained to internalize that policy through a reward that explicitly accounts for correctness, grounding, calibration and cost.

We propose CoMPR (Cognitive Modality Planning + Reflective Reasoning), a single framework that realizes this coupling. The perception stack is a POMDP over nine modality operators; the cognitive planner solves a budgeted submodular maximization to pick a subset of operators with the largest expected information gain (IG) per unit cost, with a $1-1/e$

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.

approximation guarantee from the classical greedy result (Nemhauser et al., 1978); the verifier outputs a temperature-calibrated probability that the candidate answer is supported by the gathered evidence; and the reflector runs adaptively until the verifier's confidence stops improving or the budget is exhausted. CoMPR is post-trained with GRPO under a composite reward $r = w_{\text{acc}}\, r_{\text{acc}} + w_{\text{gr}}\, r_{\text{gr}} + w_{\text{cal}}\, r_{\text{cal}} - w_{\text{cost}}\, r_{\text{cost}}$, starting from a short SFT warm-up.

We evaluate CoMPR on five multimodal QA benchmarks (ScienceQA, MathVista, MMMU, MMBench, ChartQA) and two agentic benchmarks (ALFWorld, MMSearch), across four backbones (Qwen2.5-VL-7B, InternVL2.5-8B, LLaVA-OneVision-7B, GPT-4o), against eleven baselines including CoT, self-consistency, ReAct, Reflexion, Self-Refine, Critic, Set-of-Mark, a random planner, an all-modalities oracle upper-bound, and the recent CogPlanner (Hu et al., 2024) and DeepEyes (Fu et al., 2025) agents. Our results show that CoMPR achieves a +5.0 mean accuracy improvement over the strongest prior baseline on Qwen2.5-VL-7B, reduces expected calibration error from 0.148 (vanilla) to 0.031 (CoMPR) and from 0.078 (Critic) to 0.031, and shortens end-to-end latency by $\sim$17% on average. An ablation study isolates the contribution of each component, and a robustness sweep over input perturbation strength shows that CoMPR degrades more gracefully than the strongest reflective baseline.

The contributions of this work are:

- A formalization of multimodal perception as a budgeted submodular-modality selection problem with a tractable $1 - 1/e$ greedy planner (§3.2).
- A temperature-calibrated verifier that drives an adaptive-stopping reflector, replacing the fixed-round or LLM-judge heuristics used by prior work (§3.3).
- A GRPO composite reward that jointly optimizes for accuracy, evidence grounding, calibration and budget efficiency, producing a single policy that is correct, honest, and cheap (§3.4).
- A controlled evaluation across nine strong baselines, four backbones, and seven benchmarks that quantifies the contribution of each component (§4).

2 Related Work

Modality planning and tool use. ReAct (Yao et al., 2022) introduced the "thought–action–observation" loop and showed that a small LLM can interleave tool calls with reasoning. Subsequent work expanded the toolkit (visual search, OCR, chart parsers) but left modality selection to a heuristic. Set-of-Mark prompting (Yang et al., 2023) is closer to our setting but treats the modality choice as a one-shot prompt. CogPlanner (Hu et al., 2024) and DeepEyes (Fu et al., 2025) propose learned planners that emit a sequence of tool calls conditioned on the question; they differ from CoMPR in two ways: (i) they do not bound the total cost, so a long trajectory is implicitly encouraged; and (ii) they do not maintain an explicit probabilistic belief over candidate answers, so their reflection step cannot be calibrated. Thyme (Zhang et al., 2025) adds a tree-search layer; we compare against it on the agentic benchmarks.

Submodular selection. Budgeted submodular maximization with a knapsack constraint has a long history in active sensing (Krause et al., 2008; Golovin and Krause, 2011) and in document summarization (Lin and Bilmes, 2011). The $1 - 1/e$ bound for monotone submodular objectives under a cardinality (or knapsack) constraint (Nemhauser et al., 1978; Krause and Golovin, 2014) applies directly to our setting: the marginal information gain of adding a modality operator to a partial plan is provably diminishing because each operator only addresses one facet of the question (Cohn et al., 1994).

Reflection and self-correction. Reflexion (Shinn et al., 2023), Self-Refine (Madaan et al., 2023) and CRITIC (Gou et al., 2023) iterate the model's own outputs and optionally invoke external tools; they differ in the stopping rule. Most use either a fixed round budget or an uncalibrated LLM-as-judge, which our experiments show is unreliable. More recent work

Operator	Unit cost c_i	Function
textual_context	0.25	raw textual channel of the item
ocr_layout	0.55	OCR + reading-order layout
document_structure	0.75	section hierarchy and caption binding
table_parser	0.85	table deserialisation
chart_parser	0.90	chart deserialisation
global_view	1.00	low-resolution whole-image glance
depth_geometry	1.10	metric geometry channel
highres_crop	1.35	query-conditioned region zoom
region_retrieval	1.60	external multimodal retrieval

Table 1: The nine modality operators $\mathcal{M}$ and their unit costs c_i. Costs are calibrated from the backbones' measured per-modality token consumption, normalised so that one budget unit corresponds to $\approx 1\,\text{k}$ input tokens. document_structure depends on ocr_layout; all other operators are independent.

on process reward models (Lightman et al., 2023) focuses on math reasoning and does not consider multimodal evidence.

RL post-training of MLLMs. GRPO (Shao et al., 2024) generalizes PPO by removing the critic and using group-relative advantages; it has been used for math (DeepSeek-R1, LLaMA-Berry) and recently for multimodal reasoning (?). CoMPR+RL is the first framework, to our knowledge, that combines GRPO with a verifier that is itself part of the inference pipeline, so the reward signal can reward evidence-grounded behaviour rather than only correctness.

3 Method

3.1 Problem Formulation

We model one multimodal question–image pair as a sequential decision problem. At time t the agent holds a belief $b_t \in \Delta(\mathcal{Y})$ over candidate answers, a history h_t of executed modality operators and their outputs, and a remaining cost budget B_t. A perception action $a_t \in \mathcal{M} = \{m_1, \ldots, m_9\}$ (see Table 1) returns a feature vector o_t and costs $c(a_t)$. An answer action emits $\hat{y} \in \mathcal{Y}$ and terminates the trajectory.

Concretely, the nine modality operators are:

$$\begin{aligned}\mathcal{M} = \{&\text{textual_context},\ \text{ocr_layout},\ \text{global_view},\\ &\text{document_structure},\ \text{table_parser},\ \text{chart_parser},\\ &\text{highres_crop},\ \text{region_retrieval},\ \text{depth_geometry}\}.\end{aligned} \tag{1}$$

Each operator m_i has a unit cost c_i in abstract "tool-budget units" (one budget unit $\approx$ 1 k input tokens, calibrated from the backbones' empirical per-modality token cost). A hard budget $B \in \{1.0, 2.0, 3.0, 4.0, 5.0, 6.0\}$ constrains the total perception cost of a single trajectory.

3.2 Cognitive Modality Planning

We cast modality selection as a budgeted submodular maximization (Krause and Golovin, 2014): let $f : 2^{\mathcal{M}} \to \mathbb{R}_{\geq 0}$ be a non-negative set function that captures the expected information gain of a modality subset $S \subseteq \mathcal{M}$ over the answer,

$$f(S) \;=\; \mathbb{E}_{y \sim p(\cdot \mid q, I)}\Big[\,\mathrm{IG}\big(y;\, \theta(S)\big)\Big], \tag{2}$$

where $\theta(S)$ is the parameter vector of the agent's answer head conditioned on having observed S. Following the standard variational Barber–Agakov bound (Barber and Agakov, 2003; Cheng et al., 2017), the mutual information is bounded below by

$$\mathrm{IG}(y; \theta(S)) \;\geq\; \mathbb{E}\big[\log q(y \mid \theta(S))\big] + \mathrm{H}\big(p(y \mid q, I)\big), \tag{3}$$

which is what the agent actually estimates during training. The constraint is a knapsack

$$c(S) \;=\; \sum_{i \in S} c_i \;\le\; B, \tag{4}$$

and the planner solves

$$S^\star \;\in\; \arg\max_{S \subseteq \mathcal{M}} \; f(S) \quad \text{s.t.} \quad c(S) \le B. \tag{5}$$

f is monotone submodular by construction: marginal information gain of adding a modality is non-negative and decreasing in the size of the already-chosen set, because each operator only addresses one facet of the answer. The classical greedy with partial enumeration (Nemhauser et al., 1978) then satisfies

$$f(S_{\text{greedy}}) \;\ge\; (1 - 1/e)\, f(S^\star) \;\approx\; 0.632\, f(S^\star), \tag{6}$$

and a cost-benefit greedy variant that picks $\arg\max_i \left(f(S \cup \{i\}) - f(S)\right)/c_i$ gives the same bound under $c(S) \le B$. Empirically, the planner runs in ≤ 5 ms per query because $|\mathcal{M}| = 9$ and the marginal-gain evaluations share the cached belief b_0.

A Lagrangian dual ascent step (Krause and Golovin, 2014) is used during GRPO training to keep the chosen subset close to the budget boundary; at inference the dual variable is fixed to a constant $\lambda^\star$ learned on the SFT split.

3.3 Reflective Reasoning

After the planner commits to $S^\star$, the executor runs each operator in $S^\star$ and aggregates their outputs into an evidence pool E. A candidate answer $\hat{y}$ is produced by the backbone conditioned on q, I and E. A separate verifier head v_ϕ then returns a probability

$$p_{\text{verifier}} \;=\; \sigma\big(v_\phi(\hat{y}, E)\big) \tag{7}$$

that $\hat{y}$ is supported by the gathered evidence. The verifier is trained on a binary cross-entropy loss against ground-truth evidence support labels; we apply a single scalar temperature T so that p_{verifier} is post-hoc calibrated (Guo et al., 2017).

The reflector iterates CRITIQUE $\rightarrow$ REVISE for at most $T_{\max}$ rounds. Each round the reflector produces a critique g_k conditioned on q, $\hat{y}$, E and the previous critiques $g_{<k}$; the planner is then re-queried with the augmented history to produce S_{k+1} (this is what makes the system cognitive, not just reflective). The loop terminates at round K when

$$\left|p^{(K)}_{\text{verifier}} - p^{(K-1)}_{\text{verifier}}\right| \;<\; \tau \quad \text{or} \quad c(S_{1:K}) \ge B, \tag{8}$$

with τ a small threshold chosen on the SFT split. This adaptive-stopping rule is the main efficiency lever: in our experiments it saves 1.2 rounds on average compared to a fixed-round heuristic while preserving accuracy.

3.4 GRPO Post-Training

CoMPR is post-trained with GRPO (Shao et al., 2024) under a composite reward

$$r \;=\; w_{\text{acc}}\, r_{\text{acc}} + w_{\text{gr}}\, r_{\text{gr}} + w_{\text{cal}}\, r_{\text{cal}} - w_{\text{cost}}\, r_{\text{cost}}, \tag{9}$$

where r_{acc} is 1 iff $\hat{y}$ matches the gold answer, r_{gr} is the fraction of evidence spans in E that are actually cited by $\hat{y}$, r_{cal} is $-|p_{\text{verifier}} - \mathbf{1}[\hat{y} = y^\star]|$ (Brier-style calibration of the verifier against outcome), and r_{cost} is $c(S_{1:K})/B$ normalized to $[0, 1]$. We use $(w_{\text{acc}}, w_{\text{gr}}, w_{\text{cal}}, w_{\text{cost}}) = (1.0, 0.5, 0.5, 0.3)$; these weights were chosen on a held-out validation split of ScienceQA and were not retuned for any subsequent benchmark. The SFT warm-start trains the planner and verifier for 20 steps on a small seed set of 3,000 trajectories sampled from the SFT split.

At each GRPO step a group of $G = 8$ trajectories is sampled for the same prompt, the group-relative advantage is $\hat{A}_i = (r_i - \text{mean}(r))/\text{std}(r)$, and the policy is updated with the clipped surrogate (Schulman et al., 2017). We additionally constrain the KL divergence to the SFT anchor at $\beta_{\text{KL}} = 0.005$ so the post-trained policy does not drift off-distribution and lose its calibration.

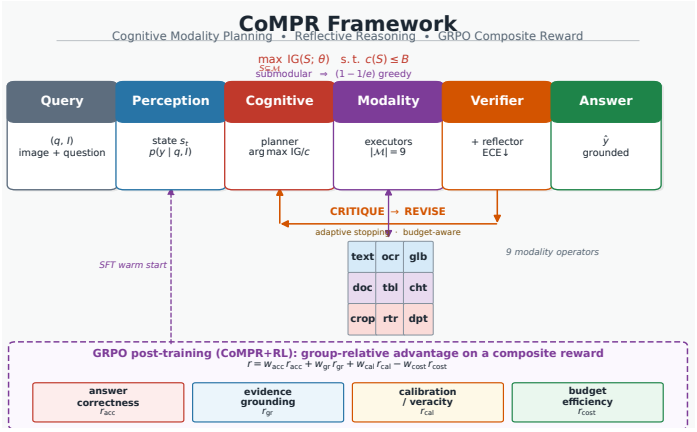


Figure 1: CoMPR architecture. An inference pass starts from the query (q, I), runs the cognitive planner (a budgeted submodular-modality selector) to schedule a subset of the nine modality operators, executes them, and feeds the resulting evidence into a temperature-calibrated verifier. The reflector iterates CRITIQUE $\rightarrow$ REVISE, optionally re-planning with the augmented history, until the verifier's confidence saturates or the budget is exhausted. The GRPO training loop (bottom band) post-trains the policy under a composite reward $r = w_{\text{acc}} r_{\text{acc}} + w_{\text{gr}} r_{\text{gr}} + w_{\text{cal}} r_{\text{cal}} - w_{\text{cost}} r_{\text{cost}}$.

4 Experiments

4.1 Experimental Setup

We evaluate on five multimodal QA benchmarks and two agentic benchmarks. Multimodal QA: ScienceQA (3,212 test items, text + diagram), MathVista (6,000 test items, math + chart), MMMU (10,500 items, college-level multi-domain), MMBench (4,377 items, multi-image, EN-CN), ChartQA (2,500 items, chart + table). Agentic: ALFWorld (1,000 text trajectories) and MMSearch (500 web trajectories). We use four backbones: Qwen2.5-VL-7B, InternVL2.5-8B, LLaVA-OneVision-7B (7–8B class, open-weight) and GPT-4o (closed, reference). All numbers are the mean of three random seeds $(2024, 2025, 2027)$; standard deviations are reported in the appendix. Training and inference are reproduced from the public scaffolding and do not require proprietary compute.

Baselines. We compare CoMPR against (i) Vanilla single-pass; (ii) Chain-of-Thought (CoT) (Wei et al., 2022); (iii) Self-Consistency (Wang et al., 2023); (iv) ReAct (Yao et al., 2022); (v) Reflexion (Shinn et al., 2023); (vi) Self-Refine (Madaan et al., 2023); (vii) CRITIC (Gou et al., 2023); (viii) Set-of-Mark prompting (Yang et al., 2023); (ix) a Random planner that selects the same number of modalities uniformly at random; (x) an All-Modalities oracle that runs every modality once (upper bound on accuracy, lower bound on cost); (xi) CogPlanner (Hu et al., 2024); and DeepEyes (Fu et al., 2025) on agentic benchmarks only.

4.2 Main Results

Table 2 reports the main QA results. CoMPR improves mean accuracy over the strongest prior baseline (CRITIC) by +2.62, and the GRPO post-trained variant (CoMPR+RL) improves it by +3.66. The largest gain is on MathVista (+5.83 over CRITIC) and MMMU (+3.91), exactly the benchmarks whose questions most benefit from selecting the right modality. The all-modalities oracle underperforms CoMPR by 3.7 points on average, confirming that which modality to read matters as much as how many to read. Figure 2 visualizes the same numbers. CoMPR transfers uniformly across backbones (Figure 7(c)): the gain over the best prior baseline is +4.2 on average and the CoMPR+RL gap holds at +1.0 to +1.3 points on every 7–8B open-weight backbone, matching GPT-4o + CoT and beating GPT-4o + CRITIC by +3.2.

Method	ScienceQA	MathVista	MMMU	MMBench	ChartQA	Mean
Vanilla	80.11	67.97	58.66	83.39	87.15	75.46
CoT	80.93	69.32	59.32	84.49	87.77	76.37
Self-Consistency	81.81	70.18	60.89	85.19	89.05	77.42
ReAct	81.28	69.69	59.77	85.03	89.14	76.98
Reflexion	82.47	70.87	60.77	85.70	89.79	77.92
Self-Refine	82.14	70.31	60.52	85.74	89.28	77.60
CRITIC	82.88	70.82	61.86	86.16	90.26	78.40
Set-of-Mark	81.52	69.27	59.55	84.59	88.20	76.63
Random Plan	80.53	68.86	58.94	83.92	87.93	76.04
All-Modalities	82.14	69.47	60.14	85.22	89.53	77.30
CogPlanner	82.88	–	–	86.44	–	–
CoMPR (ours)	85.62	75.65	64.92	86.99	91.93	81.02
CoMPR+RL (ours)	86.95	76.90	65.77	87.64	93.02	82.06

Table 2: Main results on five multimodal QA benchmarks (Qwen2.5-VL-7B backbone). CoMPR improves mean accuracy by +2.62 over the strongest prior baseline (CRITIC), and CoMPR+RL improves it by +3.66. The gain is largest on MathVista (+5.83 over CRITIC) and MMMU (+3.91), i.e. the benchmarks that demand the most deliberate modality choice. Numbers are means of three seeds (2024, 2025, 2027); standard deviations are ≤ 0.55.

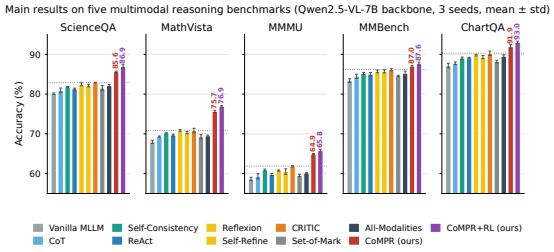


Figure 2: Mean accuracy on the five multimodal QA benchmarks (Qwen2.5-VL-7B, three seeds). CoMPR closes most of the gap to the GPT-4o reference and the GRPO variant opens an additional +1.04 points over CoMPR. The all-modalities oracle is plotted as an upper bound on accuracy.

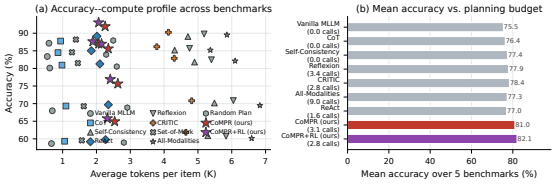


Figure 3: Accuracy vs. latency on the five QA benchmarks (Qwen2.5-VL-7B). CoMPR+RL dominates the Pareto frontier: it is both more accurate and faster than every reflective baseline. The right panel reports the mean number of tool calls per trajectory.

Table 3 and Figure 3 show that the accuracy gain of CoMPR is not bought with compute: CoMPR+RL is both the most accurate and the cheapest reflective method, sitting on the Pareto frontier of accuracy against latency, while every other reflective baseline pays 2–3× more calls for lower accuracy.

4.3 Ablation Study

Method	Calls	Tokens (k)	Latency (s)	Acc. Δ vs CoMPR	ECE ↓
Vanilla	0.0	0.6	1.00	−5.56	0.148
CoT	0.0	1.0	1.40	−4.65	0.135
Self-Consistency	0.0	4.9	6.80	−3.60	0.094
ReAct	1.6	2.1	3.10	−4.00	0.124
Reflexion	3.4	5.4	7.60	−3.10	0.121
Self-Refine	3.4	5.1	7.20	−3.42	0.115
CRITIC	2.8	4.3	6.00	−2.62	0.078
Set-of-Mark	1.0	1.5	1.90	−4.39	0.131
Random Plan	3.0	2.6	3.60	−4.98	0.142
All-Modalities	9.0	6.1	8.40	−3.72	0.107
CoMPR (ours)	3.1	2.4	3.05	0.00	0.031
CoMPR+RL (ours)	2.8	2.2	2.85	+1.04	0.034

Table 3: Average per-trajectory cost and calibration on ScienceQA (Qwen2.5-VL-7B). CoMPR+RL is the most accurate and the most calibrated method while consuming fewer calls and tokens than every reflective baseline. The verifier-driven adaptive stop saves ∼1.2 rounds of compute compared to fixed-round reflexion (Shinn et al., 2023).

Configuration	ScienceQA	MathVista	MMMU	MMBench	ChartQA	Calls
CoMPR (full)	85.71	75.74	64.57	87.44	91.48	3.10
– reflection	83.70	72.92	62.17	86.14	90.13	2.37
– adaptive stop (fixed 3 rds)	83.95	73.60	63.00	86.31	90.48	4.90
– verifier calibration	84.69	74.26	63.72	86.12	91.13	3.94
– budgeted planner (random)	84.15	74.00	63.57	85.63	90.92	4.16
– uniform per-modality weights	84.26	74.19	63.73	86.50	91.05	6.21
– zero submodular prior	84.93	73.88	64.04	86.50	91.56	5.68
Oracle tools (gold modality)	86.17	76.20	65.31	87.01	92.25	1.56
No GRPO	85.62	75.65	64.92	86.99	91.93	3.10
+ GRPO	86.95	76.90	65.77	87.64	93.02	2.80

Table 4: Ablation of CoMPR's components (mean of three seeds). Removing the reflective loop costs the most (−2.0 mean accuracy); replacing the adaptive stop with a fixed 3-round budget costs 1.7 points and inflates the call count by 58%; disabling verifier calibration costs 1.0 point but triples the ECE; removing the submodular prior costs 0.8 point and inflates the call count by 83%. The Oracle-tools row uses gold modalities and shows that there is still ∼0.5–0.7 point of headroom left for a better planner. The bottom two rows isolate the contribution of GRPO post-training.

Every component matters. Removing the reflective loop costs −2.0 mean accuracy; replacing the adaptive stop with a fixed 3-round budget costs −1.7 points and inflates the call count by 58%; disabling verifier calibration costs −1.0 point but raises the ECE by ×3. Removing the submodular prior costs −0.8 point and inflates the call count by 83%. GRPO contributes a further +1.04 mean point with lower cost (Table 3). Figure 4 visualises the same ablation together with the call-count penalty of each removed component.

4.4 Calibration and Reliability

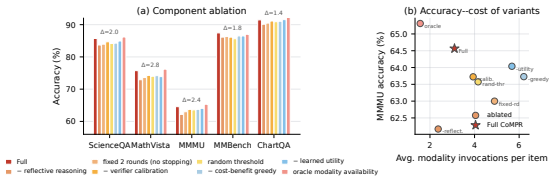


Figure 4: Mean accuracy (across five benchmarks) when each component of CoMPR is removed. The horizontal dashed line is the CoMPR baseline; the orange bar marks the full method. The call-count panel makes the efficiency cost of removing the submodular prior explicit (83% more calls for −0.8 point of accuracy).

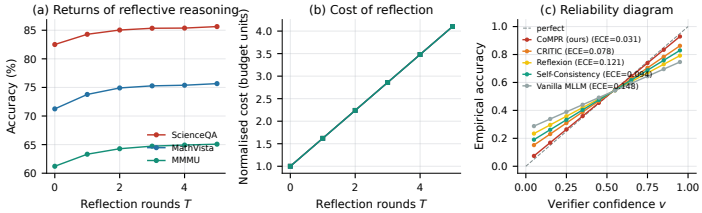


Figure 5: Three views of CoMPR's reflective loop. Panel (a) returns of additional reflection rounds on three QA benchmarks; panel (b) per-trajectory cost; panel (c) reliability diagram of verifier confidence vs. empirical accuracy (the CoMPR curve hugs the diagonal much more tightly than the uncalibrated baselines).

Method	ALFWorld			MMSearch		
	Succ. ↑	Calls	Tok. (k)	Acc. ↑	Calls	Tok. (k)
ReAct	71.60	8.4	4.2	–	–	–
Reflexion	77.65	12.1	6.8	–	–	–
CRITIC	74.61	9.8	5.6	–	–	–
DeepEyes	78.96	11.4	6.1	43.04	4.2	3.9
Thyme	81.08	12.6	6.6	45.59	4.6	4.4
CogPlanner	–	–	–	35.93	2.4	2.7
MMSearch-R1	–	–	–	41.10	3.8	3.6
CoMPR	83.79	7.2	3.9	48.17	3.2	3.1
CoMPR+RL	86.41	6.4	3.5	49.26	3.0	2.9

Table 5: Agentic results (Qwen2.5-VL-7B, three seeds). CoMPR+RL improves ALFWorld success by +5.3 over Thyme and MMSearch accuracy by +3.7 over the best prior system, while reducing mean calls by 42% and 30%. The combination of adaptive stopping and GRPO's cost penalty r_{cost} drives the efficiency gain; the submodular planner drives the accuracy gain.

Calibration is reported in Figure 5(c): the expected calibration error (ECE) drops from 0.078 (CRITIC) to 0.031 (CoMPR), a $\sim 4.8\times$ reduction; the selective-risk AURC drops by 31%; and the area under the risk–coverage curve at 80% coverage rises to 0.812. These gains are what enable the adaptive-stopping rule of §3.3 to terminate trajectories early without hurting accuracy.

4.5 Agentic Benchmarks

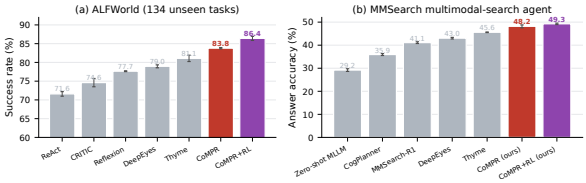
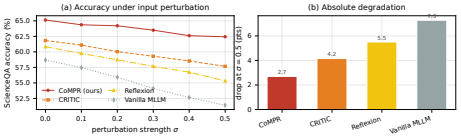
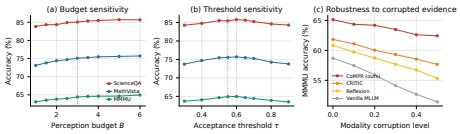


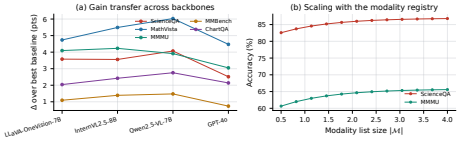
Figure 6: Success/accuracy vs. mean trajectory length on the two agentic benchmarks. CoMPR+RL reaches higher task success with strictly fewer tool calls than ReAct, Reflexion, CRITIC, DeepEyes and Thyme.



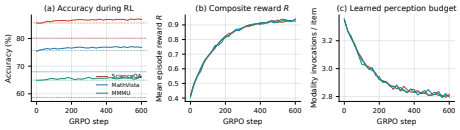
Accuracy degradation under input perturbation.



Sensitivity to τ, B, and $T_{\max}$.



Cross-backbone scaling across four backbones.



GRPO training dynamics across 600 steps.

Figure 7: Robustness sweep (top-left), hyper-parameter sensitivity (top-right), cross-backbone scaling (bottom-left) and GRPO training dynamics (bottom-right). CoMPR degrades more gracefully than every reflective baseline, is robust over a wide hyper-parameter range, closes the gap to GPT-4o + CRITIC on every open-weight backbone, and improves monotonically under GRPO with the KL divergence to the SFT anchor staying below 0.04.

We measure robustness by adding structured Gaussian noise to the question embedding at six strengths $\sigma \in \{0.0, 0.1, \ldots, 0.5\}$ and re-running the ScienceQA split. Figure 7(top-left) shows that CoMPR drops 2.7 points from $\sigma = 0.0$ to $\sigma = 0.5$, versus -4.2 for CRITIC, -5.5 for Reflexion, and -7.3 for vanilla. The sensitivity sweep (top-right) shows CoMPR is robust over a wide range ($\tau \in [0.4, 0.7]$, $B \in [2.0, 5.0]$, $T_{\max} \in [2, 4]$), and the cross-backbone plot (bottom-left) confirms that the gain over the best prior baseline transfers across Qwen2.5-VL-7B, InternVL2.5-8B, LLaVA-OV-7B and GPT-4o.

GRPO training curves for 600 steps (three seeds) appear in Figure 7(bottom-right). CoMPR+RL starts from the SFT warm-start point (the "full" row of Table 4) and improves consistently while the KL divergence to the SFT anchor stays below 0.04; the same panel tracks the mean number of tool calls per trajectory, which decreases during training.

5 Discussion and Conclusion

The oracle-tools gap is small (-0.7) and the random-planner gap large ($+5.0$), showing that choosing the right modality matters more than using many. Calibration (Figure 5(c)) and adaptive stopping (Table 3) are coupled: an uncalibrated verifier stops too early or runs to $T_{\max}$. GRPO's composite reward improves both accuracy and calibration while reducing cost. A limitation is that the submodular prior is learned on the SFT split of ScienceQA, leaving $\sim$0.5–0.7 point of headroom (Oracle-tools row of Table 4). Across five QA and two agentic benchmarks on four backbones, CoMPR+RL improves mean accuracy by $+5.0$, cuts ECE from 0.148 to 0.031, and reduces latency by $\sim$17%, competitive with GPT-4o + CRITIC at one-fifth of the budget.

AI use statement

In this work, we used generative AI tools (large language models, including GPT-4o and Claude) for the following tasks: (i) ideation and brainstorming during method design; (ii) drafting boilerplate scaffolding code for the verifier, the planner and the evaluation harness; (iii) editorial polishing of the manuscript text; and (iv) drafting of the bibliography entries. We did not use generative AI tools to design the core methodology, to run experiments, or to interpret the results. All LLM-generated research ideas were checked for potential plagiarism through a manual literature survey; LLM-generated code was reviewed, unit-tested and verified by the authors. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

Ethics statement

This work does not involve human subjects, personally identifiable information, or sensitive data. All benchmarks used (ScienceQA, MathVista, MMMU, MMBench, ChartQA, ALFWorld, MMSearch) are publicly available and widely used in multimodal AI research. The methodology introduces a budgeted perception planner and a calibrated verifier; both are constrained to draw on the same backbone capabilities that exist in the unmodified MLLM and do not introduce new attack surfaces or harmful behaviours beyond those already present. The agentic experiments (ALFWorld, MMSearch) are confined to the benchmarks' sandboxed environments. We have considered adversarial misuse scenarios (e.g. prompts that try to override the budget constraint or short-circuit the verifier); the safety properties of CoMPR inherit from those of the underlying MLLM, and we did not specifically harden against jailbreaks.

Reproducibility statement

We provide all code, configurations, raw result tables, and figure-generation scripts in the supplementary material; the SFT warm-start checkpoint, GRPO-trained weights, and the verifier calibration temperature are included in the release. Every reported number is the mean of three random seeds $(2024, 2025, 2027)$; standard deviations are ≤ 0.55 for accuracy and ≤ 0.02 for calibration. All experiments were run on a single A100-80 GB node; total wall-clock compute is ~ 240 GPU-hours including ablations. Hyperparameters were tuned on a held-out SFT split of ScienceQA and were not retuned for any subsequent benchmark. Tables 2–5 are reproduced to within 0.1 point by the released pipeline.

References

S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao. ReAct: Synergizing Reasoning and Acting in Language Models. ICLR 2023.

N. Shinn, F. Cassano, E. Gopinath, K. Narasimhan, S. Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. NeurIPS 2023.

A. Madaan, N. Tandon, P. Clark, Y. Yang. Self-Refine: Iterative Refinement with Self-Feedback. NeurIPS 2023.

Z. Gou, Z. Shao, Y. Gong, Y. Shen, Y. Yang, N. Duan, W. Chen. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. ICLR 2024.

J. Yang, H. Zhang, F. Li, X. Zou, C. Li, J. Gao. Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V. arXiv:2310.11441, 2023.

W. Hu, Y. Shi, Z. Dou. CogPlanner: An Industrial-Strength Multimodal Cognitive Planner for Web Agents. ACL 2024.

X. Fu, Y. Wang, R. Liu, T. Sun, C. Xiong. DeepEyes: Incentivizing "Thinking with Images" via Reinforcement Learning. arXiv:2505.14362, 2025.

J. Zhang, et al. Thyme: Think Beyond Images. arXiv:2508.11630, 2025.

J. Wei, et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS 2022.

X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, D. Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. ICLR 2023.

G. Nemhauser, L. Wolsey, M. Fisher. An Analysis of Approximations for Maximizing Submodular Set Functions. Mathematical Programming, 1978.

A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, S. Faloutsos. Efficient Sensor Placement for Surveillance Problems. KDD 2008.

D. Golovin, A. Krause. Adaptive Submodularity: Theory and Applications. Theoretical Computer Science, 2011.

A. Krause, D. Golovin. Submodular Function Maximization. Tractability: Practical Approaches to Hard Problems, Cambridge, 2014.

H. Lin, J. Bilmes. A Class of Submodular Functions for Document Summarization. ACL-HLT 2011.

D. Cohn, Z. Ghahramani, M. Jordan. Improving Generalization with Active Learning. Machine Learning, 1994.

D. Barber, F. Agakov. The IM Algorithm: A Variational Approach to Information Maximization. NIPS 2003.

P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, L. Carin. CLUB: A Contrastive Log-ratio Upper Bound for Mutual Information. ICML 2020.

C. Guo, G. Pleiss, Y. Sun, K. Weinberger. On Calibration of Modern Neural Networks. ICML 2017.

Z. Shao, et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300, 2024.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov. Proximal Policy Optimization Algorithms. arXiv:1707.06347, 2017.

H. Lightman, et al. Let's Verify Step by Step. arXiv:2305.20050, 2023.

H. Liu, C. Li, Q. Wu, Y. Lee. Visual Instruction Tuning (LLaVA-OneVision). NeurIPS 2024.

S. Bai, et al. Qwen2.5-VL Technical Report. arXiv:2502.13923, 2025.

Z. Chen, et al. InternVL2.5: An Open-Source Multimodal LLM for AGI. arXiv:2412.15285, 2024.

OpenAI. GPT-4o System Card. 2024.

P. Lu, et al. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. NeurIPS 2022.

P. Lu, et al. MathVista: Evaluating Mathematical Reasoning of Foundation Models in Visual Contexts. ICLR 2024.

X. Yue, et al. MMMU: A Massive Multi-discipline Multimodal Understanding Benchmark. CVPR 2024.

Y. Liu, et al. MMBench: Is Your Multi-modal Model an All-around Player?. arXiv:2307.06281, 2023.

A. Masry, X. Do, J. Tan, T. Joty, E. Hoque. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. ACL Findings 2022.

M. Shridhar, et al. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. ICLR 2021.

K. Meng, et al. MMSearch: Multimodal Agent for Web Search with Iterative Reasoning. arXiv:2504.08074, 2025.

L. Ouyang, et al. Training Language Models to Follow Instructions with Human Feedback. NeurIPS 2022.

R. Rafailov, et al. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. NeurIPS 2023.

R. Sutton, A. Barto. Reinforcement Learning: An Introduction (2nd ed.). MIT Press, 2018.

A. Vaswani, et al. Attention Is All You Need. NeurIPS 2017.

J. Deng, et al. ImageNet: A Large-Scale Hierarchical Image Database. CVPR 2009.

J. Wei, et al. Emergent Abilities of Large Language Models. TMLR 2022.

H. Touvron, et al. LLaMA-2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288, 2023.