Eliciting Self-Verification in Multimodal Reasoning Agents with Reinforcement Learning

Vishwas Sathish[1⋆†], Viresh Ranjan[2⋆], Xinliang Zhu[2], Arnab Dhua[2], and Douglas Gray[2]

[1] University of Washington, Seattle
vsathish@cs.washington.edu
[2] Amazon, USA
{vireshr,xlzhu,adhua,douggray}@amazon.com

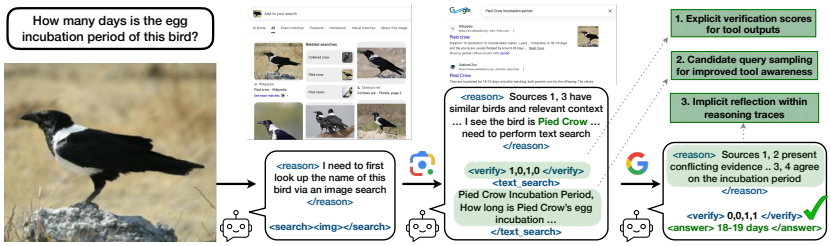
Abstract. Reasoning agents increasingly rely on external tools such as web search to answer complex queries. Reinforcement learning (RL) finetuning algorithms such as GRPO have improved long-form reasoning in text-only language models, particularly for coding and mathematics. Reliable tool use in multimodal agents, however, remains challenging because models must interpret text and images while integrating noisy retrieved evidence, often under sparse outcome-level supervision without explicit verification signals. We present Self-Verification via Reinforcement Learning (SVRL), an RL-only finetuning framework that trains multimodal agents to verify and filter retrieved evidence within their own reasoning traces, reducing reliance on external verifiers at inference time. SVRL also introduces a search-aware penalty that discourages unnecessary tool calls and a query-diversity reward that encourages diverse, well-formed search queries, providing fine-grained feedback on when and what to search. Finetuning Qwen-2.5-VL-7B with SVRL on only 5,000 visual question answering examples yields consistent gains in multi-hop VQA generalization and tool efficiency across benchmarks. Overall, SVRL narrows the gap between compact agents and much larger proprietary models while requiring substantially lower training and inference cost.

Keywords: Large Language Models · Large Multimodal Models · Reinforcement Learning · Visual Question Answering · RL Finetuning · Group Relative Policy Optimization · Test Time Scaling

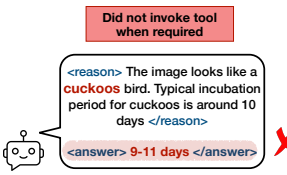
1 Introduction

Multimodal large language models (MLLMs) are evolving from static perception systems into interactive agents that combine vision and text modalities with external tool use. Commercial models such as ChatGPT and Gemini, can interpret images and invoke actions such as web search or code execution [1, 17], but their training data, alignment objectives, and tool-use policies are largely proprietary, limiting scientific control and reproducibility. In open-source MLLMs,

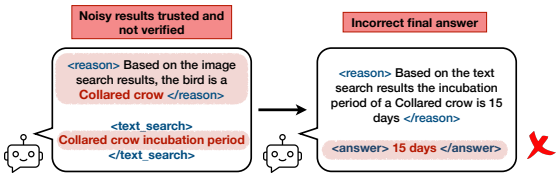
⋆ Equal contribution. † Corresponding author.



(a) Successful multi-turn reasoning and verified tool use with **SVRL**.



(b) No tool calibration in **SFT**/ **Inference** methods.



(c) Unverified results derail the model towards incorrect reasoning, tool use and final answers in **Vanilla GRPO**.

Fig. 1: Multi-hop tool-augmented VQA example and common pitfalls. The question requires identifying the bird and retrieving its incubation period (18–19 days).

capability is commonly built through visual instruction tuning [11, 32], data-driven self-supervision for tool invocation [43], and preference-based optimization [9, 37, 39, 58]. More recently, RL finetuning has re-emerged as a practical route to improving long-horizon reasoning in language models, aided in part by domains with automatic verifiers such as code execution and formal mathematics [5, 12, 28, 46]. Extending these gains to multimodal tool use is substantially more challenging because retrieved tool outputs are unstructured, and thus rarely admit reliable automatic verifiers.

To study this setting, we use multi-hop visual question answering (VQA) as a concrete instantiation of tool-augmented multimodal reasoning, where a model answers a natural-language question grounded in an image. In knowledge intensive benchmarks such as InfoSeek and OK-VQA, the image alone is often insufficient to answer a question: the model must recognize visual entities and retrieve missing facts from external sources [8,16,24,35,45]. This produces multi-hop reasoning over perception, tool calls, and noisy evidence integration, similar to visual information seeking and active inference processes [6,15,22,40]. Figure 1 shows a running example in this regime and highlights representative pitfalls we repeatedly encounter in practice. We discuss them below.

First, agents are often poorly calibrated about *when* search is needed. They may fail to invoke tools when external evidence is necessary or invoke them even when the answer is available from internal knowledge. We analyzed multi-hop reasoning traces produced by MMSearch-R1, an existing GRPO-based finetuning method, on a 2,000-example subset of FVQA [52]. In our analysis, search was required but not invoked in over 10% of cases, while over 30% of cases triggered unnecessary search. Figure 1b illustrates one such failure mode where a tool call

was necessary, but never made. Prior work emphasizes on-demand tool use and the accuracy-cost tradeoff, suggesting that deciding when to search is itself a core part of the problem [24, 25, 52].

Second, even when search is warranted, tool outputs are noisy and require filtering. Retrieved pages can be irrelevant, outdated, or internally inconsistent. Without an explicit verification mechanism, models may copy the first plausible snippet, ignore disconfirming evidence, or fail to reconcile conflicts (Figure 1c). In our analysis of MMSearch-R1, nearly 45% of retrieved results were irrelevant. Even when the agent issued an appropriate query and the returned results contained relevant evidence, it answered incorrectly in over 28% of cases. Prior visual web agents report similar failure patterns under clutter and distractors, often using additional re-ranking or verification stages to mitigate them. These stages add computation, latency, and cost, and can become a deployment bottleneck [18, 24, 36].

Third, learning reliable tool use is difficult under sparse supervision. Many pipelines provide only an outcome-level reward for the final answer, leaving intermediate decisions (what to search, which result to open, what evidence to trust, when to stop) unlabeled. As a result, distinct failure cases can receive the same terminal penalty, making credit assignment brittle when step annotations are scarce. This sparsity also interacts poorly with GRPO: if all rollouts in a group receive the same reward, the within-group variance collapses and the standardized advantages provide little to no learning signal [13]. Consistent with this, 19.5% of MMSearch-R1's text search queries are irrelevant, derailing trajectories early and reducing the information in the final GRPO correctness reward.

Motivated by these issues, we introduce Self-Verification via Reinforcement Learning (SVRL), an RL-only finetuning algorithm for tool-augmented multimodal agents. Section 2 situates SVRL relative to prior work. Section 3 presents the SVRL objective and training procedure, and Section 4 and Section 5 report results on in-distribution and out-of-distribution benchmarks, supported by qualitative examples and tool-use analyses. We also include preliminary test-time scaling experiments that increase the number of candidate search queries during inference, to achieve higher accuracies.

Our main contributions are:

1. **Fine-grained search control.** We augment the GRPO objective with reward terms that discourage unnecessary search and reward informative, diverse query proposals. This provides fine-grained trajectory-level feedback that targets both *when* to search and *what* to search for.
2. **Self-verification for evidence filtering.** We elicit structured verification scores over retrieved items within the agent's reasoning trace and optimize them as part of the RL objective. These scores make evidence selection and filtering explicit and learnable, and are applied at inference without any external verifier.
3. **Stable optimization and test-time query scaling.** We adopt Dr. GRPO to stabilize optimization under trajectory-dependent rewards. We further

show that increasing the number of candidate search queries at inference improves accuracy, indicating that SVRL supports effective test-time scaling of search.

2 Related Work

Visual Question Answering. VQA spans early vision–language architectures and large-scale dataset-driven learning, and has evolved toward settings that require compositional reasoning, reading text, and incorporating external knowledge [2, 19, 23, 26, 47]. Knowledge-intensive and multi-hop benchmarks such as OK-VQA, A-OKVQA, InfoSeek, WebQA, and LiveVQA further stress evidence acquisition and aggregation beyond the image [4, 8, 16, 35, 45]. Our work targets this regime: we train on FVQA (as released in MMSearch-r1) and evaluate both in- and out-of-distribution, emphasizing tool-mediated multi-hop behavior rather than single-pass perception [24, 52].

LLM finetuning for tool use. Modern MLLM pipelines typically start from pretrained transformer language-model backbones [3, 48] and acquire interactive behavior through instruction tuning [11,32,50] and prompting that elicits explicit intermediate reasoning [51]. For knowledge-intensive QA, retrieval-augmented generation (RAG) injects retrieved context into the model input [29]; we report a RAG workflow baseline in Table 2 to separate the benefits of always-on retrieval from selective, agentic search. Tool use itself can be induced from data [43,55] and further aligned with preference-based objectives [37, 39]. Reinforcement learning finetuning has recently shown strong gains on long-horizon reasoning in language models [12, 44, 46], and concurrent multimodal systems apply RL to improve web search behavior [25, 36, 52]. SVRL builds on this direction, but targets calibrated search decisions and explicit evidence filtering under noisy retrieval via fine-grained trajectory rewards that make tool-use errors more distinguishable during optimization.

Verification, reward design, and test-time scaling. Reliability is also improved by spending more compute at inference time (e.g., self-consistency and search branching) [49, 54], or by adding external re-ranking and verification stages in web-based agents [18, 22, 24]. Complementary lines study process-level supervision and verifier training for long-horizon reasoning [10, 31], as well as stabilization variants for RL finetuning [34, 56]. Closest to our emphasis on verification signals are recent works that introduce explicit grounding, but these approaches differ in source of supervision, target task and modality, or where verification is applied in the pipeline [7, 14, 42]. SVRL focuses on making verification and filtering behaviors learnable from compact multimodal rollouts while keeping inference free of external verifiers, and we empirically study how this interacts with GRPO-style learning dynamics under trajectory-level rewards.

3 Self-Verification via Reinforcement Learning

In this section, we present SVRL, a Reinforcement Learning based finetuning framework for multimodal agents. We formalize the setting and objective, introduce two algorithmic modifications for trajectory-level rewards and self-verification, and discuss practical considerations for stable training.

3.1 Preliminaries

Visual Question Answering (VQA). We consider a VQA dataset $\mathcal{D}$ of triples (I, q, y^*), where I is an image, q is a natural-language question about the image, and y^* is the ground-truth answer. While some instances can be answered directly from the image, we focus on multi-hop questions that require external knowledge and tool use (see Figure 1) [2, 8, 35, 45].

Multimodal reasoning as policy optimization. Our primary goal is to improve a multimodal agent's tool-use and reasoning capabilities at inference time. Given an input (I, q), we view the model as a stochastic policy π_θ that generates a trajectory τ consisting of T intermediate reasoning steps $s_{1:T}$ and a final answer y, i.e., $\tau = (s_{1:T}, y)$. The trajectory likelihood factorizes as

$$\pi_\theta(\tau \mid I, q) = \Big(\prod_{t=1}^{T} \pi_\theta(s_t \mid I, q, s_{<t})\Big)\, \pi_\theta(y \mid I, q, s_{\le T}). \tag{1}$$

Each step $s_t = (l_t, a_t, p_t)$ is a structured format and may contain (i) a natural-language reasoning l_t within `<reason>...</reason>` tags and (ii) an action a_t (e.g., a tool invocation or emitting the final answer) with optional tool parameters p_t. A representative multi-step reasoning process can be seen in Figure 1a. We follow the tool set and formatting conventions of MMSearch-R1 throughout this paper for simplicity [52].

Group Relative Policy Optimization (GRPO). We finetune π_θ with RL to maximize a trajectory-level reward $R(\tau)$, regularized by a KL penalty to a fixed reference policy $\pi_{\theta_{\text{ref}}}$:

$$\max_\theta \; \mathbb{E}_{(I,q,y^*)\sim\mathcal{D}}\Big[\mathbb{E}_{\tau\sim\pi_\theta(\cdot|I,q)}[R(\tau)] - \beta\, D_{\text{KL}}\big(\pi_\theta(\cdot \mid I, q) \,\|\, \pi_{\theta_{\text{ref}}}(\cdot \mid I, q)\big)\Big].$$

GRPO is a PPO-style optimizer that estimates advantages from groups of on-policy rollouts without a learned value function [44,46]. For each (I, q), we sample a group $\tau^{1:G}$ using a behavior policy $\pi_{\theta_{\text{old}}}$, compute standardized group advantages $\hat{A}_i$ from $\{R(\tau^{(i)})\}_{i=1}^{G}$, and apply the clipped surrogate $\text{PPOClip}(r, A) = \min(rA, \text{clip}(r, 1-\epsilon, 1+\epsilon)A)$ with an importance ratio $r_i(\theta) = \pi_\theta(\tau^{(i)})/\pi_{\theta_{\text{old}}}(\tau^{(i)})$:

$$\max_\theta \; \mathbb{E}_{\mathcal{D}}\, \mathbb{E}_{\tau^{1:G}} \left[\frac{1}{G}\sum_{i=1}^{G} \text{PPOClip}\big(r_i(\theta), \hat{A}_i\big) - \beta\, D_{\text{KL}}\big(\pi_\theta \,\|\, \pi_{\theta_{\text{ref}}}\big)\right] \tag{2}$$

3.2 When and What to Search?

Humans typically decide whether to search and how to phrase a query by trading off expected information gain against time and effort, consistent with ideas in rational metareasoning and information foraging [38, 41]. We reflect this intuition by augmenting the GRPO maximization objective with search-aware and query-diversity rewards, so trajectory returns depend explicitly on *when* search is invoked and on the quality of the proposed queries. We begin from the RL finetuning setup in MMSearch-R1 [52], which trains a multimodal agent to use text and image search with a reward of the form $r_{\text{base}} = (1-\alpha)\, r_{\text{acc}} \cdot r_{\text{search}} + \alpha\, r_{\text{fmt}}$. Here, r_{acc} is exact-match accuracy against the ground-truth answer, r_{fmt} enforces a valid action format (common in RL finetuning of reasoning models) [12], and r_{search} discourages excessive tool calls to reduce cost and latency.

Tool calibration (when to search). A uniform step-style penalty is well motivated when learning a policy from scratch (RL navigation, manipulation, etc), but it can be sub-optimal when finetuning a pretrained model that already answers a subset of questions without tools. Penalizing search uniformly can also suppress necessary tool use, since it does not distinguish avoidable search from required search. We therefore construct a *search-aware* factor using the pretrained model as a weak competence prior. Concretely, for each training input (I, q) we run a single forward pass on the model without tools to obtain $\hat{y}$; if $\hat{y} = y^*$, we label the instance as `search_free`. During RL rollouts, we downweight trajectories that invoke a search tool on a `search_free` instance:

$$r_{\text{aware}} = \begin{cases} \lambda_{\text{sa}} & \text{if } \texttt{search\_free} \text{ and a search tool is called,} \\ 1 & \text{otherwise.} \end{cases}$$

In our experiments, we set $\lambda_{\text{sa}} = 0.1$ and replace the baseline search penalty with r_{aware}. This targets avoidable tool calls while preserving the incentive to search when needed. Empirically, it improves both the accuracy–search tradeoff and overall tool calibration (Table 1, Figure 5).

Query diversity (what to search). We additionally find that text-search queries produced by the agent are often underspecified, even when search is necessary. To encourage better query formulation, we prompt the model to propose k candidate queries and randomly execute one of the valid, unique queries. Let $\hat{k} \le k$ denote the number of unique queries that are produced in the required format. We define a query-count factor

$$r_{\text{count}} = \begin{cases} 1, & \text{if the rollout makes no text-search call,} \\ \max(\epsilon, \hat{k}/k), & \text{otherwise,} \end{cases}$$

where $\epsilon > 0$ is a small floor to avoid degenerate zero reward when no valid queries are produced. We incorporate r_{count} multiplicatively into the task term of the reward, so rollouts are explicitly differentiated by the number of usable

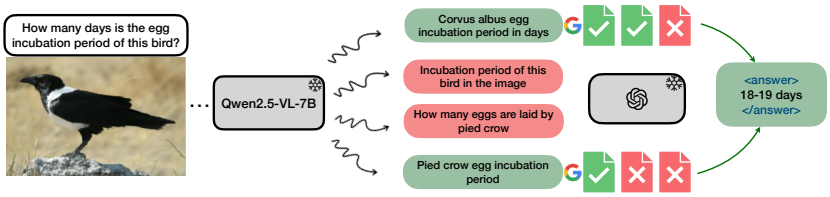


Fig. 2: Test Time Verification. We demonstrate the need for verification of search queries and tool outputs via an external verifier. GPT-5 filters incorrect search queries and tags noisy web page snippets. With this verification process and no additional model finetuning, we observe improvements in multi-hop VQA performance.

query proposals. This encourages the agent to generate multiple well formed, diverse queries, improving the specificity of search intent and downstream evidence retrieval.

3.3 Verifying Search Results

Web retrieval is noisy: search results can be irrelevant, outdated, or internally inconsistent. Drawing on signal-detection and source-monitoring views of validation under uncertainty [20, 27], we make evidence filtering explicit by training the agent to score the usefulness of retrieved items and suppress misleading snippets. This targets a common failure of GRPO-finetuned compact agents, which often over-trust early snippets and aggregate brittle evidence, consistent with prior web-based multimodal agents [18, 24, 36].

Test Time Verification. To quantify the benefit of denoising, we introduce an external verifier at inference time. Specifically, we use a stronger text-only model (GPT-5) to filter both query candidates and retrieved snippets without access to the ground-truth answers. For each input (I, q), we generate multiple candidate text queries from the base agent (via repeated decoding with temperature $= 1$), and ask the verifier to retain only the queries that are likely to retrieve relevant evidence. We then run text search using the retained queries and ask the verifier again to select useful snippets given the original (I, q) and the retrieved context (Figure 2). This test-time filtering consistently improves final accuracy while keeping the answering model fixed, which isolates the error source to noisy tool outputs and weak evidence selection (See Table 1).

Self-Verification. Test-time verification improves accuracy but is computationally expensive and reduces the deployment advantage of compact agents. Our goal is to recover these gains without any external verifier at inference time. We do so by eliciting self-verification inside the agent's reasoning trace (Figure 3) and optimizing it directly during RL finetuning. Concretely, for each input (I, q) the agent (i) proposes multiple candidate text-search queries and (ii) assigns a structured binary usefulness vector to retrieved items, e.g., `<verify> 0,1,0,1 </verify>`, together with a short natural-language justification.

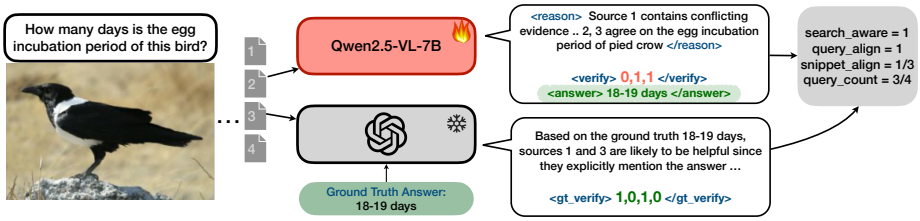


Fig. 3: Self-Verification via oracle feedback during finetuning. GPT-5 is used as the oracle verifier that receives the same web snippets as the model and additionally the ground truth answer. The verifier then annotates each web snippet based on the utility towards inferring the correct answer. This verifier is discarded during inference.

To avoid reward-hacking from additive shaping terms (Section 3.4), we keep verification supervision modular and apply it only when the final answer is correct. Let r_{aware} and r_{count} denote the search-calibration factors from Section 3.2. We introduce two additional alignment factors, both supervised at training time by an external verifier that scores queries and snippets: r_{qalign} measures agreement between the agent's query proposals and verifier labels of which proposed queries are useful, and r_{salign} measures agreement between the agent's snippet-level self-scores and verifier labels of which retrieved items contain answer-supporting evidence. We combine these terms into a single SVRL factor,

$$r_{\text{svrl}} = r_{\text{aware}} \cdot r_{\text{count}} \cdot r_{\text{qalign}} \cdot r_{\text{salign}}. \tag{3}$$

Let $r_{\text{acc}} \in \{0, 1\}$ denote exact-match correctness and $r_{\text{fmt}} \in [0, 1]$ denote the format reward. Our final trajectory reward is

$$r(\tau) = (1 - \alpha)\, r_{\text{acc}}(\tau)\, r_{\text{svrl}}(\tau) + \alpha\, r_{\text{fmt}}(\tau). \tag{4}$$

During training, we obtain the query and snippet labels using a verifier model with access to the ground-truth answer y^*, which is necessary to robustly handle aliases and paraphrases beyond string matching. We then finetune the agent so that its self-verification scores align with these labels. Figure 3 illustrates the SVRL scores for a sample reasoning trace. The verifier is discarded at inference time, where the agent must perform verification and filtering on its own. Finally, we incorporate Equation 4 into the KL-regularized GRPO objective by replacing the trajectory reward $R(\tau)$ with our $r(\tau)$:

$$\max_{\theta} \mathbb{E}_{(I,q,y^*)\sim\mathcal{D}}\Big[\mathbb{E}_{\tau\sim\pi_\theta(\cdot|I,q)}[r(\tau)] - \beta\, D_{\text{KL}}\big(\pi_\theta(\cdot \mid I, q) \,\|\, \pi_{\theta_{\text{ref}}}(\cdot \mid I, q)\big)\Big]. \tag{5}$$

This yields a structured, fine-grained, trajectory-level learning signal that targets query quality and evidence filtering, improving tool integration and accuracy in Table 1 and Table 2. Algorithm 1 summarizes the SVRL finetuning algorithm discussed in this section.

Algorithm 1 SVRL finetuning

Require: Dataset $\mathcal{D}$, system prompt $\mathcal{S}$, train-time verifier $\mathcal{G}$, policy π_θ, search tools $\mathcal{T}$, format weight α, group size G

Self-labeling.
for all $(I, q, y^*) \in \mathcal{D}$ **do**
  $\hat{y} \leftarrow \pi_\theta(\mathcal{S}, I, q)$ ▷ single pass; no tool calls
  $\ell(I, q) \leftarrow \mathbb{I}[\hat{y} = y^*]$ ▷ 1 iff `search_free`
end for
RL finetuning with Dr.GRPO.
while not converged **do**
  Sample minibatch $\mathcal{B} \subset \mathcal{D}$
  for all $(I, q, y^*) \in \mathcal{B}$ **do**
    Sample rollouts $\{\tau^{(i)}\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathcal{S}, I, q; \mathcal{T})$
    for $i = 1$ to G **do**
      Parse $\tau^{(i)} \to (y, p, v)$ ▷ answer, tool params, self-verification
      $v^* \leftarrow \mathcal{G}(I, q, y^*, p)$ ▷ train-time verification labels
      $r_{\texttt{svrl}} \leftarrow \texttt{search\_aware}(\ell(I, q), \tau^{(i)}) \cdot \texttt{query\_count}(\tau^{(i)})$ ▷ Section 3.2
      $r_{\texttt{svrl}} \leftarrow r_{\texttt{svrl}} \cdot \texttt{query\_align}(v, v^*) \cdot \texttt{snippet\_align}(v, v^*)$ ▷ Section 3.3
      $r^{(i)} \leftarrow (1 - \alpha) \cdot \texttt{acc\_score} \cdot r_{\texttt{svrl}} + \alpha \cdot \texttt{format\_score}$
    end for
    Update θ using $\{(\tau^{(i)}, r^{(i)})\}_{i=1}^{G}$ ▷ group advantages + Dr.GRPO objective
  end for
end while

3.4 Practical Considerations

Reward diversity for GRPO. GRPO estimates group-relative advantages by normalizing rewards within a rollout group for each input (I, q) (subtracting the group mean and dividing by the group standard deviation) [46]. If all rollouts in a group receive the same reward, there is no relative preference signal and the resulting update becomes uninformative; if the within-group reward dispersion is very small, the normalization is poorly conditioned and gradients can become noisy. With the baseline reward r_{base} (Section 3.2), we often observe near-tied groups because trajectories share the same correctness and similar tool-use patterns. In contrast, r_{svrl} varies with query formulation, retrieved snippets, and verification outcomes, increasing within-group reward dispersion and yielding more informative advantages. Empirically, this lets us use a smaller group size ($G = 4$) than prior work (MMSearch-R1 uses $G = 8$), improving throughput on a single 8×A100 node.

Dr. GRPO for training stability. In early experiments, vanilla GRPO exhibited unstable optimization, including intermittent spikes in actor gradient norms and drift in response length (Figure 4). We therefore adopt Dr. GRPO, which removes normalization factors in the update (for example, standard-deviation and length normalization) that can amplify gradients under trajectory-dependent rewards [34]. We use Dr. GRPO for all SVRL variants reported in Table 1 and Table 2.

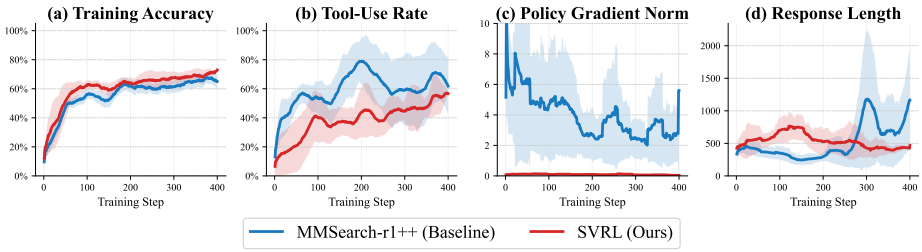


Fig. 4: SVRL yields more stable RL finetuning dynamics. Mean over 3 seeds with ±1 standard deviation shown. All runs use 400 steps, batch size 32, group size 4, and multi-turn tool-use trajectories. SVRL is compared against MMSearch-R1++ (MMSearch-R1 reproduced with ReAct-style prompting [52, 55]). (a) training performance, (b) search ratio, (c) policy gradient norm, and (d) response length.

LLM-judge rewards and reward hacking. We also tried replacing exact-match correctness with an LLM-judge reward during training (Qwen3-32B comparing y to y^*). Although judge rewards could tolerate aliases, they were vulnerable to reward hacking: the policy increased answer verbosity and began to output hedged or multi-answer responses that judges sometimes scored favorably despite the answers themselves being non-committal. This is consistent with known judge failure modes [21,33,37,57]. We therefore use exact-match correctness for training, and reserve LLM judges for evaluation-time scoring (following the benchmark protocol). For reward shaping, we found additive mixtures of shaping terms to be easier to exploit (partial credit despite incorrect answers), so we gate fine-grained signals by correctness via the multiplicative factor $r_{\text{acc}}.r_{\text{svrl}}$. Another instance of reward hacking occurred in the query-alignment ablation without the search-aware factor. Although query alignment is gated by correctness, the model learned to invoke search on nearly every example to maximize the query-alignment reward while still producing correct answers. Adding the search-aware factor reduced unnecessary search, while the full self-verification objective further improved both accuracy and tool efficiency (Table 1).

4 Experimental Setup

Models. Our primary experiments use the base Qwen2.5-VL-7B-Instruct backbone which we finetune with our method and evaluate on all benchmarks. We further re-implement and finetune with the baseline GRPO method presented in MMSearch-R1 and an improved method MMSearch-R1++ that replaces the original system prompt with a ReAct based prompt [52]. Additionally, for strong comparison to modern LLMs, we also report inference-time results for GPT-4o across the datasets used.

Datasets. We train Qwen-2.5-VL-Instruct with our algorithm on the Factual VQA (FVQA) dataset. FVQA contains a mix of direct-answer questions and

Table 1: Ablations on finetuning with GRPO. Acc (%) denotes LLM-as-Judge accuracy. SR (%) denotes search ratio. Details in Section 5.

Model	FVQA-test		InfoSeek	
	Acc	SR	Acc	SR
MMSearch-R1	51.1	64.2	49.8	63.5
MMSearch-R1++	56.4	80.3	55.1	59.7
SVRL-query-align	57.7	97.6	58.2	98.0
SVRL-LLM-judge	58.1	36.7	58.0	50.6
SVRL-search-aware	59.9	80.3	57.0	85.8
SVRL-ttv	60.6	80.3	58.3	85.2
SVRL-self-verification	64.2	61.3	64.7	59.2

multi-hop questions that require tools, which helps RL avoid degenerating into tool use on every example. Following Wu *et al.*, we use 5,000 examples for training and reserve 1,800 for testing. We evaluate on FVQA and InfoSeek as in-distribution benchmarks, since FVQA is sampled from the broader InfoSeek collection and shares the same underlying data and task structure. To assess robustness, we additionally evaluate out-of-distribution on LiveVQA, MMSearch and SimpleVQA datasets, which differ in query style and retrieval conditions and therefore test whether the learned behavior transfers beyond the training distribution [8, 16, 24, 30, 52].

Implementation details. We use an LLM judge (GPT-5) to determine final answer correctness when computing accuracy, following the evaluation protocol and judge prompt presented in [52]. This choice provides a scalable and consistent correctness signal across benchmarks where answers may have minor surface-form variations (e.g., aliases, physical units, date-time formats), while keeping our reported numbers directly comparable to prior work. We train our model with a ReAct-style system prompt to structure tool use and intermediate reasoning, and we include the full prompt templates in the appendix [55]. For tool access, we use Google Image Search and Google text search via a third party API. To reduce redundant and expensive API calls, we implement a caching layer for text and image search results on DynamoDB. All training and inference experiments are run on 8×A100 GPUs using Pytorch, Ray and veRL framework.

5 Results

Ablations. Table 1 isolates the impact of each component. **MMSearch-R1** reproduces the original recipe [52]. **MMSearch-R1++** strengthens this baseline by switching to a ReAct-style prompt and removing the KL penalty and entropy coefficient. **SVRL-LLM-judge** adds DrGRPO but replaces exact-match with an LLM-judge reward. Here, accuracy improves slightly, but answers become longer and less decisive, consistent with reward hacking. **SVRL-query-align**

Table 2: Performance across benchmarks. Acc (%) denotes answer accuracy and SR (%) denotes search ratio. SVRL improves accuracy across all datasets and achieves a better accuracy-search tradeoff, indicating more effective multi-turn reasoning and tool use. Details in text.

Model	FVQA-test		InfoSeek		MMSearch		LiveVQA		SimpleVQA	
	Acc	SR	Acc	SR	Acc	SR	Acc	SR	Acc	SR
Direct Answer										
Qwen-2.5-VL-7B	26.7	0.0	20.1	0.0	12.8	0.0	17.8	0.0	38.4	0.0
Qwen-2.5-VL-32B	24.7	0.0	25.8	0.0	15.7	0.0	18.7	0.0	40.1	0.0
Qwen-2.5-VL-72B	27.1	0.0	28.0	0.0	15.7	0.0	20.1	0.0	42.2	0.0
GPT-4o	41.7	0.0	42.7	0.0	22.2	0.0	26.9	0.0	46.6	0.0
RAG Workflow										
Qwen-2.5-VL-7B	51.6	100	53.7	100	52.2	100	48.0	100	51.6	100
Qwen-2.5-VL-32B	57.0	100	56.8	100	57.9	100	49.6	100	54.5	100
Qwen-2.5-VL-72B	62.2	100	59.4	100	59.6	100	56.0	100	61.0	100
GPT-4o	66.0	100	59.1	100	62.5	100	59.6	100	63.4	100
Adaptive Search										
MMSearch-R1++	56.4	80.3	55.1	59.7	53.8	88.5	48.4	76.2	57.4	42.5
SVRL-full-7B	**65.3**	**61.3**	**64.7**	**59.2**	**60.3**	**82.0**	**57.8**	**48.6**	**58.3**	**27.0**

ablates the need for improving the quality of queries. This comes with a cost of increased search as discussed in Section 3.4. **SVRL-search-aware** adds the calibration term from Section 3.2, improving accuracy by discouraging avoidable search while preserving search when needed. **SVRL-ttv** applies external test-time verification to filter query candidates and snippets, yielding a large gain that isolates evidence selection as a dominant failure mode. **SVRL-self-verification** replaces the external verifier with our train-time self-verification objective (Section 3.3), recovering most of the denoising benefit without verifier dependence at inference.

VQA generalization. Table 2 compares three regimes: *Direct Answer* (single-pass prediction from (I, q)), *RAG Workflow* (retrieval for every instance; SR= 100), and *Adaptive Search* (RL-finetuned agents that decide when to search). SVRL-full-7B consistently improves over the strongest adaptive-search baseline, MMSearch-R1++, by roughly 8 points on FVQA-test and 7 points on InfoSeek while also reducing search ratio, indicating a better accuracy-cost tradeoff. SVRL-full-7B approaches GPT-4o on FVQA-test (65.3 vs. 66.0) despite operating in the 7B regime. On SimpleVQA, gains are smaller, which suggests that search behavior may depend on tool parameters such as query language. We also note that time-sensitive or underspecified questions can introduce irreducible evaluation noise.

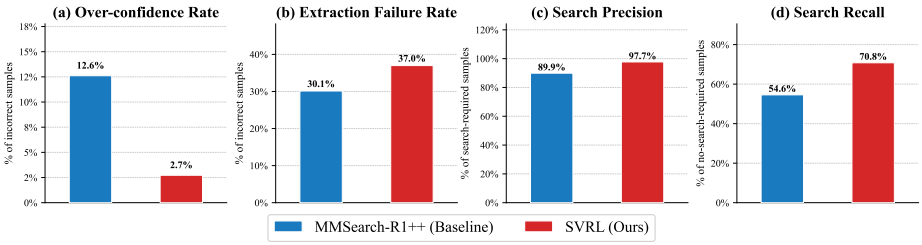


Fig. 5: Tool-use analysis. Aggregated FVQA and InfoSeek statistics.

Analysis. Figure 7 shows qualitative multi-hop examples from the SVRL-finetuned agent. We observe that SVRL induces more robust reasoning traces that compare and reconcile evidence across retrieved results, even though we do not supervise free-form critique text directly. This behavior is consistent with the verification-alignment rewards, which make evidence filtering explicit during optimization. Figure 5 summarizes the corresponding changes in tool behavior: SVRL reduces over-confidence failures where search is omitted despite being required (Figure 5(a)) and improves search precision and recall under the `search_free` annotations (Figures 5(c) and (d)). Finally, Figure 5(b) reports the conditional rate at which relevant evidence appears in retrieved content given an incorrect answer, isolating evidence utilization errors from retrieval quality.

Test-time scaling. A practical benefit of self-verification is improved *test-time scaling*: allocating more inference-time computation (e.g., sampling, search, and verification) can increase accuracy under a larger compute budget, as studied in self-consistency voting, tree-based search over reasoning paths, and self-evaluation guided beam search [49, 53, 54]. We run preliminary scaling experiments by increasing the text search budget at inference. Concretely, we evaluate on a 256-example subset of FVQA-test and allow up to 15 parallel text searches per question, producing multiple candidate answers that we aggregate by simple majority vote[3]. Figure 6 reports accuracy as a function of the allowed number of parallel searches. SVRL continues to benefit from additional search budget, whereas the baseline (MMSearch-R1++) saturates after roughly 5 searches, consistent with weaker query diversity and less reliable evidence filtering. Because SVRL proposes multiple queries within a single reasoning step, it also reduces the need for repeated full trajectory sampling, improving wall-clock ef-

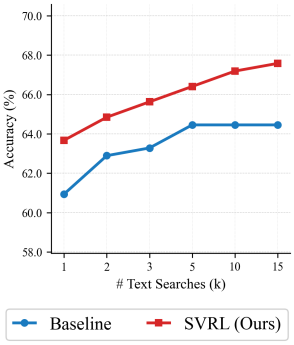


Fig. 6: Test-time scaling. SVRL benefits from additional search budget.

[3] See Supplementary for details

Fig. 7: Qualitative examples. FVQA-test cases for trained SVRL model. Top-left shows an example that did not require search. Top-right shows a sample that required only image search. Bottom two rows shows multi-turn reasoning over multiple tools - one with a correct and the other with an incorrect answer.

ficiency. We leave stronger aggregation rules and more structured search procedures for future work.

6 Conclusion

We presented SVRL, an RL-only finetuning framework that improves tool augmented multi-hop VQA in compact multimodal agents by eliciting self-verification within their reasoning traces. SVRL turns evidence selection into a learnable trajectory signal by coupling calibrated search decisions and query diversity with verifier-aligned supervision for filtering retrieved content during training, while requiring no external verifier at inference. Across FVQA-test and InfoSeek, SVRL consistently improves accuracy and the accuracy–search tradeoff over prior GRPO-based baselines, and narrows the gap to substantially larger proprietary models. Ablations indicate that search calibration and self-verification drive most of the gains, and test-time verification quantifies the benefit of denoising noisy retrieval. Limitations include dependence on web tools and their failure modes, reliance on a training-time judge whose labels may introduce bias, and sensitivity to prompt and tool formatting. Future work will broaden SVRL to richer tool suites and modalities, reduce judge dependence via stronger self-supervised verification signals, and improve robustness under retrieval distribution shift and evolving web content.

Impact Statement

SVRL improves the tool-augmented reasoning and tool-use efficiency of compact multimodal models, narrowing their performance gap with much larger proprietary models. By reducing unnecessary search and avoiding an external verifier at inference time, it can lower deployment cost and resource use. These gains potentially make multimodal agents more accessible and support safer, more private, and more controllable deployment in resource-constrained settings, including future edge-device applications.

Acknowledgements

We thank Rajesh Rao and Rene Vidal for insightful discussions and valuable feedback.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023), https://arxiv.org/abs/2303.08774
2. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: Vqa: Visual question answering. In: ICCV (2015)
3. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)
4. Chang, Y., Narang, M., Suzuki, H., Cao, G., Gao, J., Bisk, Y.: Webqa: Multihop and multimodal qa. In: CVPR (2022)
5. Chen, M., Tworek, J., Jun, H., Yuan, Q., et al.: Evaluating large language models trained on code (2021)
6. Chen, Q., Pitawela, D., Zhao, C., Zhou, G., Chen, H.T., Wu, Q.: Webvln: Vision-and-language navigation on websites. arXiv preprint arXiv:2312.15820 (2023)
7. Chen, X., et al.: Free process rewards without process labels. arXiv preprint arXiv:2412.01981 (2024), https://arxiv.org/abs/2412.01981
8. Chen, Y., Hu, H., Luan, Y., Sun, H., Changpinyo, S., Ritter, A., Chang, M.W.: Can pre-trained vision and language models answer visual information-seeking questions? arXiv preprint arXiv:2302.11713 (2023), https://arxiv.org/abs/2302.11713
9. Christiano, P.F., Leike, J., Brown, T.B., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
10. Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., Schulman, J.: Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168 (2021)
11. Dai, W., Li, J., Li, D., Zhao, Z., Wang, Z., Zhang, Z., et al.: Instructblip: Towards general-purpose vision-language models with instruction tuning. arXiv preprint arXiv:2305.06500 (2023), https://arxiv.org/abs/2305.06500

12. DeepSeek-AI: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025), https://arxiv.org/abs/2501.12948
13. Deng, W., Li, Y., Gong, B., Ren, Y., Thrampoulidis, C., Li, X.: On group relative policy optimization collapse in agent search: The lazy likelihood-displacement (2026), https://arxiv.org/abs/2512.04220
14. Du, J., Chen, Y., Li, Z., Huang, X., et al.: V* guided visual search: Efficient exploration of vqa with dynamic visual feedback. arXiv preprint arXiv:2312.14135 (2023), https://arxiv.org/abs/2312.14135
15. Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., Pezzulo, G.: Active inference: A process theory. Neural Computation **29**(1), 1–49 (2017). https://doi.org/10.1162/NECO_a_00912
16. Fu, M., Peng, Y., Liu, B., Wan, Y., Chen, D.: LiveVQA: Live visual knowledge seeking. arXiv preprint arXiv:2504.05288 (2025)
17. Gemini Team, Anil, R., Borgeaud, S., Alayrac, J.B., Yu, J., Soricut, R., et al.: Gemini: A family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023), https://arxiv.org/abs/2312.11805
18. Geng, X., Xia, P., Zhang, Z., Wang, X., Wang, Q., Ding, R., Wang, C., Wu, J., Zhao, Y., Li, K., Jiang, Y., Xie, P., Huang, F., Zhou, J.: Webwatcher: Breaking new frontier of vision-language deep research agent. arXiv preprint arXiv:2508.05748 (2025), https://arxiv.org/abs/2508.05748
19. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: CVPR (2017)
20. Green, D.M., Swets, J.A.: Signal Detection Theory and Psychophysics. John Wiley & Sons (1966)
21. Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., Wang, S., Zhang, K., Wang, Y., Gao, W., Ni, L., Guo, J.: A survey on llm-as-a-judge. arXiv preprint arXiv:2411.15594 (2024)
22. Hu, Z., Iscen, A., Sun, C., Chang, K.W., Sun, Y., Ross, D.A., Schmid, C., Fathi, A.: Avis: Autonomous visual information seeking with large language model agent. arXiv preprint arXiv:2306.08129 (2023), https://arxiv.org/abs/2306.08129
23. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: CVPR (2019)
24. Jiang, D., Zhang, R., Guo, Z., Wu, Y., Lei, J., Qiu, P., Lu, P., Chen, Z., Fu, C., Song, G., Gao, P., Liu, Y., Li, C., Li, H.: MMSearch: Benchmarking the potential of large models as multi-modal search engines. arXiv preprint arXiv:2409.12959 (2024)
25. Jin, B., Zeng, H., Yue, Z., Yoon, J., Arik, S., Wang, D., Zamani, H., Han, J.: Search-r1: Training llms to reason and leverage search engines with reinforcement learning (2025), https://arxiv.org/abs/2503.09516
26. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C.L., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: CVPR (2017)
27. Johnson, M.K., Hashtroudi, S., Lindsay, D.S.: Source monitoring. Psychological Bulletin **114**(1), 3–28 (1993). https://doi.org/10.1037/0033-2909.114.1.3
28. Le, H., Wang, Y., Gotmare, A.D., Savarese, S., Hoi, S.C.H.: Coderl: Mastering code generation through pretrained models and deep reinforcement learning (2022)
29. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive nlp tasks. NeurIPS (2020), arXiv:2005.11401

30. Li, Y., et al.: SimpleVQA: A factuality-focused visual question answering benchmark. arXiv preprint arXiv:2501.01805 (2025)
31. Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., Cobbe, K.: Let's verify step by step. arXiv preprint arXiv:2305.20050 (2023)
32. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. arXiv preprint arXiv:2304.08485 (2023), https://arxiv.org/abs/2304.08485
33. Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., Zhu, C.: G-eval: Nlg evaluation using gpt-4 with better human alignment. In: EMNLP (2023), arXiv:2303.16634
34. Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W.S., Lin, M.: Understanding r1-zero-like training: A critical perspective. In: COLM (2025), arXiv:2503.20783
35. Marino, K., Rastegari, M., Farhadi, A., Mottaghi, R.: Ok-vqa: A visual question answering benchmark requiring external knowledge. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019), https://arxiv.org/abs/1906.00067
36. Narayan, K., Xu, Y., Cao, T., Nerella, K., Patel, V.M., Shiee, N., Grasch, P., Jia, C., Yang, Y., Gan, Z.: Deepmmsearch-r1: Empowering multimodal llms in multimodal web search. arXiv preprint arXiv:2510.12801 (2025), https://arxiv.org/abs/2510.12801
37. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155 (2022)
38. Pirolli, P., Card, S.: Information foraging. Psychological Review **106**(4), 643–675 (1999). https://doi.org/10.1037/0033-295X.106.4.643
39. Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C.D., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. arXiv preprint arXiv:2305.18290 (2023), https://arxiv.org/abs/2305.18290
40. Rao, R.P.N., Gklezakos, D.C., Sathish, V.: Active predictive coding: A unifying neural model for active perception, compositional learning, and hierarchical planning. Neural Computation **36**(1), 1–32 (2023). https://doi.org/10.1162/neco_a_01627, epub 2023-12-12
41. Russell, S.J., Wefald, E.H.: Do the Right Thing: Studies in Limited Rationality. MIT Press (1991)
42. Sarch, G.H., Saha, S., Khandelwal, N., Jain, A., Tarr, M.J., Kumar, A., Fragkiadaki, K.: Grounded reinforcement learning for visual reasoning. In: Advances in Neural Information Processing Systems (NeurIPS) (2025), https://openreview.net/forum?id=1amnhVRQ3l, poster, OpenReview
43. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., Scialom, T.: Toolformer: Language models can teach themselves to use tools. arXiv preprint arXiv:2302.04761 (2023), https://arxiv.org/abs/2302.04761
44. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
45. Schwenk, D., Khandelwal, A., Clark, C., Marino, K., Mottaghi, R.: A-okvqa: A benchmark for visual question answering using world knowledge. arXiv preprint arXiv:2206.01718 (2022), https://arxiv.org/abs/2206.01718
46. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, S., Zhang, H., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024), https://arxiv.org/abs/2402.03300
47. Singh, A., Natarajan, V., Jiang, Y., Chen, X., Shah, M., Rohrbach, M., Batra, D., Parikh, D.: Towards vqa models that can read. In: CVPR (2019)

48. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2023), https://arxiv.org/abs/1706.03762
49. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., Zhou, D.: Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171 (2022)
50. Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652 (2022)
51. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. NeurIPS (2022), arXiv:2201.11903
52. Wu, J., Deng, Z., Li, W., Liu, Y., You, B., Li, B., Ma, Z., Liu, Z.: Mmsearch-r1: Incentivizing lmms to search. arXiv preprint arXiv:2506.20670 (2025), https://arxiv.org/abs/2506.20670
53. Xie, Q., Narang, S., Jain, G., Balasubramanian, N., Le, Q.V.: Self-evaluation guided beam search for reasoning. In: Advances in Neural Information Processing Systems (NeurIPS) (2023), https://openreview.net/forum?id=Bw82hwg5Q3
54. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y., Narasimhan, K.: Tree of thoughts: Deliberate problem solving with large language models. arXiv preprint arXiv:2305.10601 (2023)
55. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: React: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629 (2022), https://arxiv.org/abs/2210.03629
56. Yu, Q., Zhang, Z., Zhang, Y., et al.: Dapo: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476 (2025)
57. Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E.P., Zhang, H., Gonzalez, J.E., Stoica, I.: Judging llm-as-a-judge with mt-bench and chatbot arena. arXiv preprint arXiv:2306.05685 (2023)
58. Ziegler, D.M., Stiennon, N., Wu, J., Brown, T.B., Radford, A., Amodei, D., Christiano, P.F., Irving, G.: Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593 (2019)