

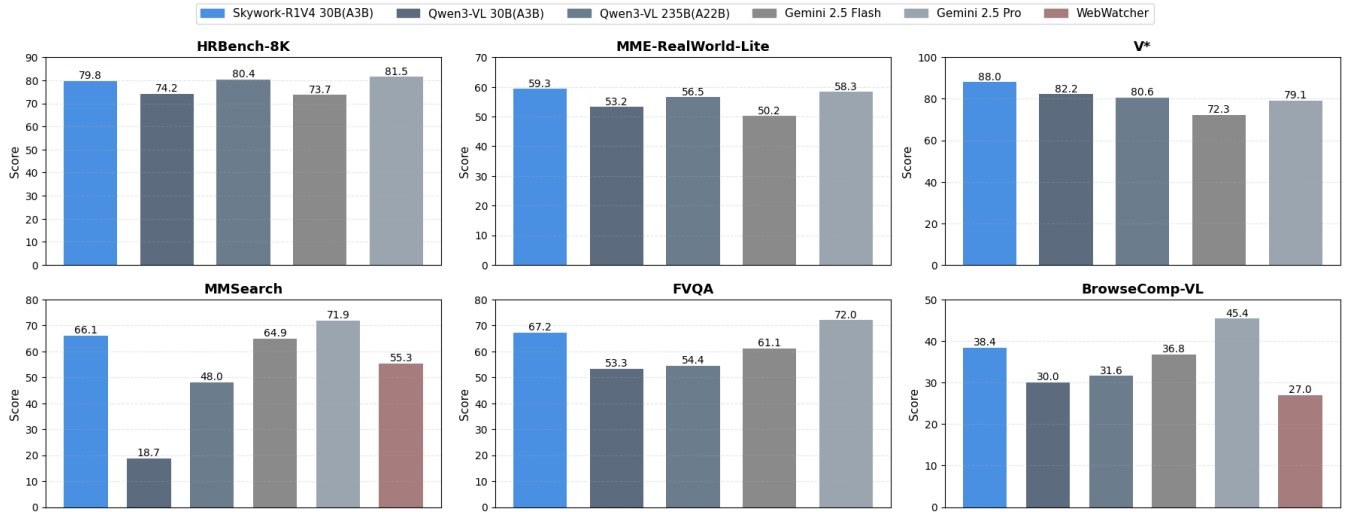
# Skywork-R1V4: Toward Agentic Multimodal Intelligence through Interleaved Thinking with Images and DeepResearch

Multimodality Team, Skywork AI

Despite recent progress in multimodal agentic systems, existing approaches often treat image manipulation and web search as disjoint capabilities, rely heavily on costly reinforcement learning, and lack planning grounded in real tool-execution traces. To address these limitations, we present **Skywork-R1V4**, a 30B (A3B)-parameter multimodal agentic model that unifies *multimodal planning*, *active image manipulation* (“thinking with images”), *deep multimodal search*, and—most critically—*interleaved reasoning* that dynamically alternates between visual operations and external knowledge retrieval. Trained solely via supervised fine-tuning on fewer than 30K high-quality, planning-execution-consistent trajectories and validated through step-wise consistency filtering, Skywork-R1V4 achieves state-of-the-art results across perception and multi-modal search benchmarks: it scores **66.1** on MMSEARCH and **67.2** on FVQA, surpassing GEMINI 2.5 FLASH on 11 out of 11 metrics. Skywork-R1V4 exhibits emergent long-horizon reasoning at inference time, successfully orchestrating up to 10+ tool calls to solve complex, multi-step tasks. Our results establish that sophisticated agentic multimodal intelligence can be achieved through carefully curated supervised learning alone—without any reliance on reinforcement learning.

 **Blog:** <https://skywork-r1v4-lite.netlify.app/>

 **Code:** <https://github.com/SkyworkAI/Skywork-R1V>



**Figure 1:** Skywork-R1V4 30B (A3B) demonstrates exceptional proficiency in code-based image manipulation, text and image search, and web browsing, achieving performance on high-resolution perception benchmarks that rivals or surpasses larger-scale and specialized models, while also showing advantages in multimodal Deepsearch tasks.

## 1. Introduction

Recent years have witnessed significant advances in multimodal large language models (MLLMs), particularly through the integration of reinforcement learning (RL) to enhance reasoning, tool use, and search capabilities. In visual reasoning, works like *PyVISION* (Zhao et al., 2025) and *THYME* (Zhang et al., 2025c) have pioneered “thinking with images” by enabling models to generate and execute code for custom image operations—moving beyond passive perception toward active visual interaction. In parallel, the rise of agentic search frameworks, such as *WEBWATCHER* (Geng et al., 2025) and *MMSEARCH-R1* (Wu et al., 2025), has demonstrated the power of augmenting internal knowledge with live web retrieval, especially for fact-intensive or time-sensitive queries. Furthermore, recent efforts like *COGPLANNER* (Yu et al., 2025) have begun to formalize *multimodal agentic planning*, decomposing complex tasks into sequences of tool calls.

Despite this progress, three critical limitations persist. **First**, most “think-with-image” approaches treat visual manipulation and external knowledge retrieval as **isolated capabilities**—they either operate on images or perform search, but rarely **interleave** the two in a unified reasoning loop. **Second**, state-of-the-art agentic systems overwhelmingly rely on **reinforcement learning**, which incurs prohibitive computational costs, unstable training dynamics, and limited reproducibility—hindering practical deployment. **Third**, planning modules are often trained on synthetic or abstract supervision, lacking grounding in **real tool-execution trajectories**, which reduces their fidelity and generalization in real-world scenarios.

To address these gaps, we present **Skywork-R1V4**, a lightweight yet powerful multimodal agentic model that unifies four synergistic capabilities through supervised fine-tuning alone:

- ◇ **Multimodal Agentic Planning:** Skywork-R1V4 generates structured, executable plans directly grounded in visual input, decomposing complex queries into coherent sequences of tool invocations (e.g., “*first crop the license plate, then run image search, then extract text from the retrieved page*”).
- ◇ **Thinking with Images:** The model actively manipulates images via programmable operations—including cropping, zooming, contrast adjustment, rotation, and pixel-level analysis—to iteratively refine visual understanding and resolve ambiguities.
- ◇ **DeepSearch:** When internal knowledge is insufficient, it performs multi-step, cross-validated web search using three tools: image search (Google Lens), text search, and full webpage retrieval, effectively mitigating hallucination in knowledge-intensive tasks.
- ◇ **Interleaved Image Manipulation and Search:** Crucially, Skywork-R1V4 **dynamically alternates** between image operations and search within a single trajectory—e.g., cropping a region, searching it online, using retrieved context to guide further zooming—enabling truly interactive, perception-grounded reasoning.

Our training data is constructed through a unified, multi-stage pipeline designed to support all four capabilities. For *Thinking with Images*, we generate multi-turn trajectories where models propose and execute Python code for visual operations (e.g., cropping, enhancement), with sandbox execution and step-wise validation of reasoning–output consistency. For *DeepSearch*, we build both basic queries (from FVQA) and complex enhanced-search tasks via constrained random walks over a knowledge graph, followed by two-stage filtering (format and answer consistency via o3-mini) and webpage summarization. To enable *interleaved reasoning*, we collect hybrid trajectories that alternate between image manipulation and search, and apply vision-language models to filter inconsistent or hallucinated samples (e.g., misaligned crops). Finally, for *Multimodal Agentic Planning*, we convert all validated trajectories into structured plans with explicit inter-step dependencies (e.g., “[Result from Step 3]”), ensuring causal coherence. Across all sources, we enforce strict consistency

checks at every reasoning step and action outcome, discarding low-fidelity samples (e.g., those requiring error-prone re-cropping or containing execution failures). This meticulous curation yields a compact yet highly effective dataset of **fewer than 30K SFT samples**, demonstrating that agentic multimodal intelligence can be achieved efficiently through quality, not scale.

Empirically, as shown in Figure 1, Skywork-R1V4 achieves state-of-the-art performance across a wide range of benchmarks. On perception tasks (HRBENCH, MME-REAL, V\*), it sets new records. More notably, on deep multimodal search benchmarks (MMSEARCH, FVQA, BROWSECOMP-VL), it achieves dramatic gains: **66.1** on MMSearch (+47.4 over Qwen3-VL) and **67.2** on FVQA (+13.9). Despite using the same 30B (A3B)-scale architecture as its baseline, Skywork-R1V4 outperforms FLASH on 11 out of 11 reported metrics. These results confirm that our approach not only enhances fine-grained perception but also enables robust, interpretable, and tool-augmented reasoning in complex, open-world scenarios—proving that agentic multimodal intelligence can be achieved efficiently through high-quality supervised learning.

## 2. Method

### 2.1. Think With Image

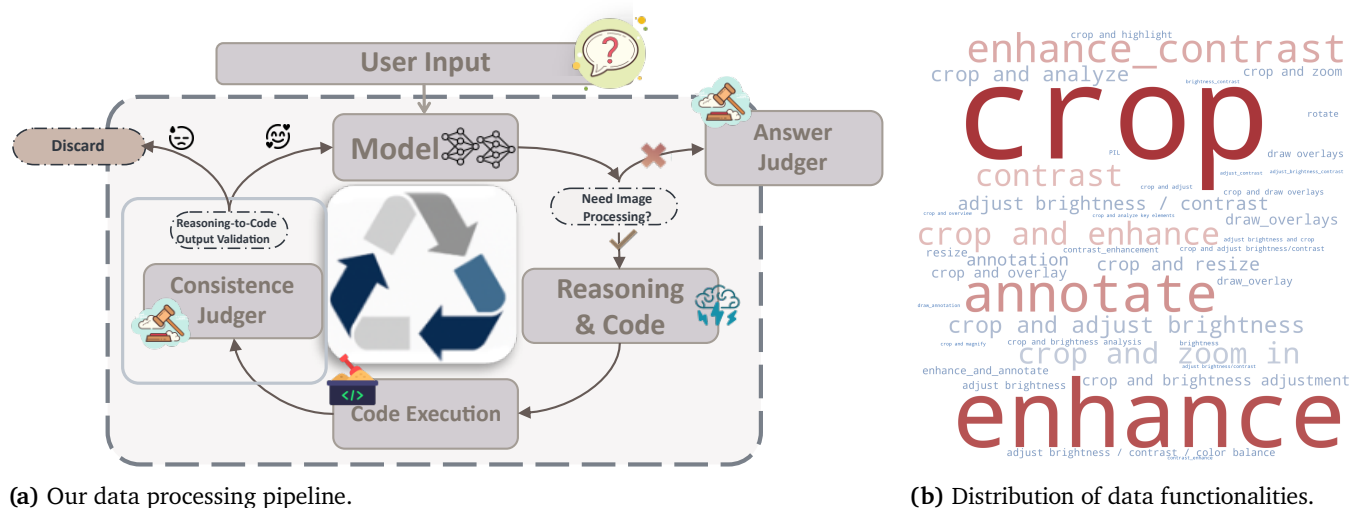
**Data Collection and Formatting.** We select images with resolutions no lower than 1024×1024 from public data sources such as Thyme-RL (Zhang et al., 2025c) and Fine-Vision (Wiedmann et al., 2025) to ensure image quality. We use the questions from these datasets as inputs and employ multiple open-source or proprietary models, such as GLM 4.5 V (Zeng et al., 2025) and Claude 4, for data construction. Our specific pipeline is illustrated in Figure 2a. First, we instruct the model in the system prompt that it can perform a series of operations on the image through coding, including but not limited to cropping, rotation, contrast enhancement, and pixel-level analysis. The model then outputs an initial analysis and code. The corresponding code is executed in our sandbox, and the execution results are returned to the model for the next operation. The model evaluates the results from the previous step; if the current conditions are sufficient to answer the question, the model outputs the final answer; otherwise, it continues planning, generates code, and attempts to answer the question. For each query, we perform 4 rollouts and retain samples where the final answer is consistent with the ground truth answer.

**Outcome and Step-wise Filtering.** After obtaining the preliminary data, we filter the final answers and identify common quality issues, which are mainly divided into two categories: 1. The model’s final answer is inconsistent with the thinking process, which is particularly common in data involving multiple rounds of image operations. The primary reason for this phenomenon is that overly long contexts and numerous images make it difficult for the model to focus on the latest thinking process, sometimes resulting in answers that are completely opposite to the thinking process. Therefore, we compare the model’s last round of thinking process with the answer to ensure consistency. 2. Severe hallucination phenomena, a common scenario being that the model crops out a blank image but claims it can already answer the question and has found the correct object. To mitigate this, we check the consistency between the images produced at each step and the subsequent thinking.

**Dataset Classification.** After completing the screening, we classify the data according to their functions, primarily into categories such as operations containing errors (e.g., code execution errors), single-round image operations, operations including re-cropping, operations including zoom-in, operations involving navigation within the image, and operations including contrast enhancement or other functions. There

are two motivations for this classification: first, it allows us to gain a clear understanding of the functional distribution in the data; second, we observe that the quality of data containing re-cropping or code execution errors is lower compared to other categories. The reason is that model errors tend to accumulate; when the model intends to crop area A but experiences a certain offset, it typically requires many steps to fix this error, leading to excessively long rounds with limited informational value.

**Distribution.** We visualize tools that appear more than 10 times across all data, as shown in Figure 2b. Our tool types are more diverse compared to existing works, not limited to a fixed few tools.



**Figure 2:** (a) Our data processing pipeline. For selected QA pairs, the model first queries whether image operations are needed to enhance perception or if a direct reply is possible. If not, it generates a reasoning process and corresponding code, which is executed in a sandbox environment. The consistency between the reasoning and the sandbox output is then validated. If consistent, the result is fed back for the next iteration until the question can be answered. (b) Distribution of data functionalities, including common operations such as cropping, contrast enhancement, zooming, annotation, and pixel-level analysis.

## 2.2. Multi-modal Search

In the search scenario, we first investigate the nature of search queries in terms of difficulty and categorize them into two types preliminarily. The **first type** consists of images with an outstanding subject and basic query like “Where is the location of it” or “What is the open date of this building?”. Once the main subject in the images is identified through reverse image search, the answer can typically be obtained with several additional rounds of text-based search. We refer to this category as **basic search**. For the **second type**, the complexity and difficulty of the query have been significantly increased, resembling the style of *BrowseComp* tasks, which entail the search agent to exhaustively explore various strategies, cross-validate potentially conflicting results to arrive at the final answer. Solving such queries often requires five rounds or more of search, as only a few rounds are insufficient. We refer to this type as **enhanced search**.

### 2.2.1. Basic Search

For basic search, we primarily utilize open-source datasets, particularly the *FVQA* in MMSearch. Agentic trajectories are generated using claude-4-sonnet. Data quality is all you need. Therefore, we conducted a rigorous two-stage filtering process to ensure the reliability of the collected trajectories.

The first stage is **format filtering**. Some cases do not end with a specific tag or fail to follow the expected *think+action* or *think+answer* pattern. However, such cases are rare, accounting for less than 1% of the total data. The second stage is **answer filtering**. Samples for which the model’s final answer did not agree with the ground truth were removed. Disagreement was determined automatically by an external judge model, o3-mini. We use **Serper** as search provider, and select Image Search(Lens), Search and Webpage for reverse image search, text-based search and web content retrieval, corresponding to three special tags in model output: `<image_search>`, `<text_search>`, and `<web_content>`. When the patterns are detected, the agent use search tool to obtain results as observation and then feed back to the model to continue generation. The entire process follows ReAct (Yao et al., 2022). Lengthy web content outputs are summarized by Qwen3-32B. More details are provided in Appendix.

### 2.2.2. Enhanced Search

**Enhanced Text Query Generation.** To construct verifiable and in-depth query–answer pairs, we operate a constrained random walker over a locally deployed encyclopedia, treating article pages and their internal links as a dense knowledge graph. A walk begins at a seed entity whose introductory and core paragraphs are mined to produce an initial question and a short, uniquely verifiable answer. Then, an LLM-based uniqueness evaluator filters out generic terms, platforms or outlets, abstract concepts, and ambiguous descriptors so that the answer points to a single, concrete referent. From the current page we then harvest internally linked entities (named as target entities), compute mention frequencies, and apply light stochastic shuffling over the top target entities to avoid stereotyped routes, while excluding previously visited nodes and overly pervasive terms to reduce cycles and semantic drift. For each “current-target” pair, we extract a concise relation and a property summary from the text, preserving the most distinctive cues that will later support indirect reference. The question rewriting process is conducted iteratively while keeping the answer constant. During each iteration, the name of the current entity is rewritten to describe the same entity indirectly by mentioning a related target entity and the relationship between them, sometimes adding a short descriptive clue to keep the question specific and unambiguous. If the relation and property do not sufficiently disambiguate, the rewrite is marked invalid and the pipeline resamples or augments evidence. Each query–answer pair is rigorously validated prior to acceptance through consistency checks that block excluded entities or aliases, ensure concise and verifiable answers, and eliminate instances failing uniqueness or interpretability. Therefrom, the resulting corpus comprises natural, executable queries that support multi-step reasoning.

**Text-to-Multimodal Query Reformulation.** For all previously constructed textual query–answer pairs, we further convert the query into a multi-modal form by explicitly grounding the final target entity in an image. Each textual query ends with a clearly defined entity, and the saved random-walk trajectory retains this entity’s name together with a compact description of its distinctive properties. We first locate this final target entity and use its name plus the distilled property description as a structured search condition to retrieve candidate images via a search engine (e.g., serp), thereby biasing results toward visually informative depictions rather than generic or noisy content. After quality filtering, we select a representative image and invoke a large language model to rewrite the original query by replacing the explicit entity name with a

visually anchored referring expression, such as transforming “Loïc Féry” into “the person in this picture”, while keeping the answer unchanged. This systematic conversion ensures that the visual modality becomes an essential part of the reasoning process, rather than a redundant decoration, and enables controlled study of multi-modal planning under identical underlying knowledge requirements.

### 2.3. Interleave Think with Image and Search

So far, we have introduced two independent capabilities: *think with images* and *web search*. Next, we aim to integrate these two abilities, enabling a single trajectory to interleave visual reasoning and information retrieval throughout the process. To achieve the goal, we randomly sample 3k datas from *LiveVQA* (Fu et al., 2025) and generate interleaved trajectories using Claude-4-Sonnet as well. However, unlike in the multi-modal search setting introduced above, we observe that Claude-4-Sonnet’s visual perception capability is relatively limited compared to its planning ability. This often lead to failed code execution or incorrect image manipulation, for example, incorrect image crops which does not align with the intended visual region. In our early experiments, such noisy samples are not filtered, leading to some decrease in model performance. To address the issue, we employ visual-language model to automatically identify the low-quality samples for **consistency filtering**, which preserve the performance and significantly enhance the ability to jointly utilize code and search actions within a single trajectory.

### 2.4. Multi-Modal Agentic Planner

**Planner Query Generation.** To enable the model to learn structured and sequential planning, the planner data is constructed from multi-step action trajectories that simulate how complex objectives are decomposed into executable sub-tasks. Instead of relying on single-turn prompts, each sample is designed from a composed action trajectory (for example, the trajectory obtained in Think With Image and Multi-modal Search) that requires the model to perform stepwise inference and maintain contextual integration. Each action chain reflects a logical progression where the planner must connect entities, events, and visual elements while preserving semantic coherence across different modalities such as text, image, and web retrieval. By training on these multi-step action plans, the planner learns to interpret high-level intent, identify inter-step dependencies, and produce interpretable action plans that are both grounded and executable.

**Data Collection and Formatting.** All generated trajectories are subsequently transformed into a unified structured format that supports interpretable and executable multi-step planning. Each task instance is represented as a sequential list of steps, where every step contains a natural-language description of the sub-task, a specified tool name corresponding to the modality or reasoning function invoked, and a parameter field defining required inputs or contextual dependencies. The detailed plan format can be found in ?? . Inter-step relations are explicitly encoded through symbolic placeholders such as “[Person identified in Step 1]” or “[Result retrieved from Step 4]”, ensuring that the logical continuity of reasoning is preserved across the chain. This formatting strategy captures the causal dependencies among steps, allowing the Planner to perform context-aware execution and verification of intermediate results. After validation and dependency checking, the dataset forms a coherent corpus that couples natural-language reasoning with structured procedural planning, providing a robust foundation for training and evaluating the Multi-Modal Agentic Planner.



**Table 1: System Prompt for Think with Image and Search Tasks.**

Answer the user’s question based on the provided image. Here are some tools you can use if needed:

1. Image search. This will trigger a Google Lens search using the image to retrieve relevant information.
2. Text search. This will trigger a Google Search using a carefully crafted query.
3. Web content. This will fetch the detailed webpage content for you to use as additional information.
4. Code execution. You can write Python code inside `<code>...</code>` to perform image operations (such as cropping, resizing, rotating, color adjustment, denoising, enhancement, etc.) before further processing.

All search results, webpage contents, or code execution outputs will be placed within `<observation>...</observation>` and returned to you.

**Output format options:**

- `<think>...</think> <tool_call>"name": "image_search", "arguments": "image_paths": ["local image path 1", "local image path 2"]</tool_call>`
- `<think>...</think> <tool_call>"name": "text_search", "arguments": "queries": ["your search query 1", "your search query 2"]</tool_call>`
- `<think>...</think> <tool_call>"name": "web_visit", "arguments": "urls": ["target url 1", "target url 2"]</tool_call>`
- `<think>...</think> <tool_call>"name": "code", "arguments": "code": "your python code here"</tool_call>`
- `<think>...</think> <answer>your answer here</answer>`

YOU MUST include your reasoning within `<think>...</think>` before taking any action.

## 2.5. Training Strategy

**Mix-Mode Training.** We mix the aforementioned types of data for supervised fine-tuning (SFT), while incorporating in-house non-think data, primarily including attribute recognition, spatial relations, and VQA data for general scenes. We use different system prompts to distinguish various tasks. During the training process, we observe mutual promotion effects among different task types; for example, training the planner model improves the model’s performance on search tasks, and general VQA data enhances the model’s accuracy in judging basic attributes and relations during the think with image process, leading to consistent improvements on perception benchmarks. The system prompt for planner tasks and think with image/ deep search tasks are shown in Table 3 and Table 1.

**Low-Quality Data Removal.** In the SFT training process, we remove data points from think-with-image that involve sandbox execution errors and those requiring re-cropping, as we observe that these inefficient data cause the model to learn error-fixing patterns, leading to significant performance degradation, with the model attempting to reproduce these erroneous patterns. An obvious concern is that after removing these

data, the model may not be able to correct code errors or cropping errors when encountered. Interestingly, in our experiments, we find that even if the model has not seen such data during training, it can still perform corrections during inference.

### 3. Experiments

#### 3.1. Benchmarks and Results

**Benchmarks and metrics:** We mainly select three categories of benchmarks. The first category focuses on perception tasks because Skywork-R1V4’s image operations mainly aim to enhance perception ability. These benchmarks include the MME-RealWorld (Zhang et al., 2024) series, HR Bench (Wang et al., 2025c), V\* (Wu and Xie, 2024), TreeBench (Wang et al., 2025a) and Visual Probe (Lai et al., 2025), etc. We report results for different splits of each benchmark. For example, for the MME-RealWorld series, we report perception and reasoning accuracy separately. For HR Bench, we report Fine-grained Single-instance Perception (FSP) and Fine-grained Cross-instance Perception (FCP) separately. For V\* (Wu and Xie, 2024), we report recognition and spatial relationship reasoning performance. We use VLMEvalKit (Duan et al., 2024) for high-resolution benchmark evaluation. For deep multimodal search tasks, we adopt three established benchmarks: *MMSearch* (Jiang et al., 2024), *FVQA* (Wang et al., 2017), and *BrowseComp-VL* (Geng et al., 2025), which assess a model’s ability to conduct complex, multi-step information seeking grounded in both visual and textual inputs.

As shown in Table 2, we observe consistent and substantial improvements over the Qwen3-VL 30B(A3B) baseline across all tested settings. On perception tasks, Skywork-R1V4 achieves gains of +1.5 to +14.4 percentage points, with particularly strong results on fine-grained visual understanding: it scores 91.8 on HRBench-4K FSP (+3.3), 90.4 on V\* Attribute (+8.7), and 76.3 on MME-Real-CN Perception (+3.7). In reasoning-intensive perception scenarios, the advantage widens—e.g., +14.4 on MME-Real-CN Reasoning. In deep search tasks—which require integrating visual understanding with external knowledge retrieval—Skywork-R1V4 demonstrates even more pronounced gains: +47.4 on *MMSearch* (66.1 vs. 18.7), +13.9 on *FVQA* (67.2 vs. 53.3), and +8.4 on *BrowseComp-VL* (38.4 vs. 30.0). These results confirm that our approach not only enhances low-level perception but also enables robust, tool-augmented reasoning in complex, open-world search scenarios. Moreover, despite using the same 30B-scale architecture as the baseline, Skywork-R1V4 outperforms Gemini 2.5 Flash on all 11 reported metrics and exceeds Gemini 2.5 Pro on 5 of them, including key perception (e.g., V\* 88.0 vs. 79.1) benchmarks—highlighting the effectiveness of our agentic, image-grounded reasoning framework.

#### 3.2. Visualization of Multimodal Agentic Capabilities

To illustrate the versatile reasoning and tool-integration abilities of Skywork-R1V4, we present a series of qualitative examples in Figures 4–7. These visualizations collectively demonstrate how Skywork-R1V4 dynamically coordinates perception, planning, and external knowledge retrieval across diverse scenarios. In Plan Mode (Figure 4), Skywork-R1V4 generates structured, tool-grounded execution plans for complex open-ended questions. In DeepResearch Mode, it exhibits fine-grained visual reasoning through iterative image manipulation (Figure 5), accurate geolocation via multimodal search (Figure 6), and, most notably, interleaved thinking—seamlessly alternating between image operations and search tools to resolve intricate visual queries (Figure 7). Together, these examples highlight Skywork-R1V4’s capacity for adaptive, grounded,



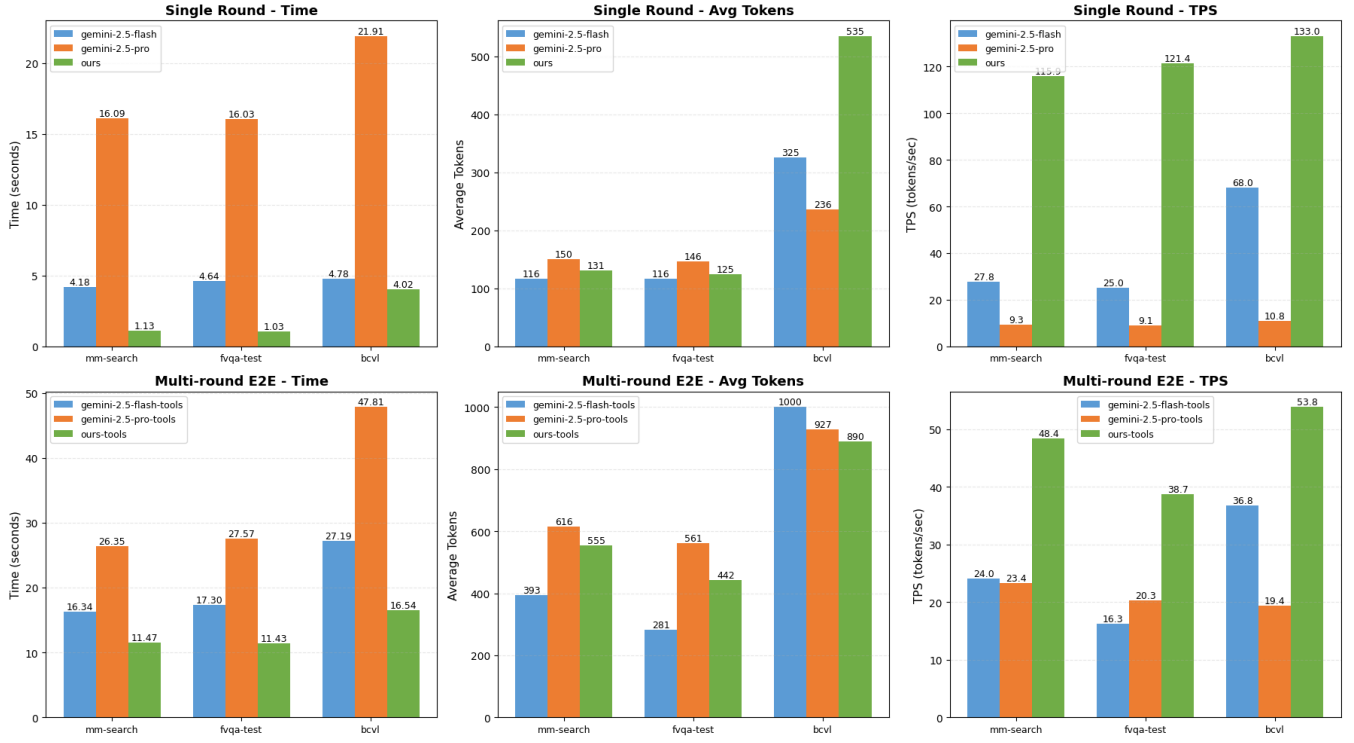
**Table 2: Performance on Perception and Deep Research Tasks.** The best performance for each metric is **bolded**, and the second best is underlined. Gold-colored font indicates improvement over the baseline Qwen3-VL 30B(A3B).

Benchmark	Split	Skywork-R1V4 30B(A3B)	Qwen3-VL 30B(A3B)	Qwen3-VL 235B(A22B)	Gemini 2.5 Flash	Gemini 2.5 Pro
<i>Perception</i>						
HRbench-4K	FSP	<b>91.8</b> <sup>+3.3</sup>	88.5	<u>89.0</u>	81.5	85.5
	FCP	73.8 <sup>+5.3</sup>	68.5	<u>77.0</u>	74.0	<b>82.3</b>
	Overall	82.8 <sup>+4.3</sup>	78.5	<u>83.0</u>	77.5	<b>83.9</b>
HRbench-8K	FSP	<b>88.8</b> <sup>+8.5</sup>	80.3	83.0	75.8	<u>83.0</u>
	FCP	70.8 <sup>+2.5</sup>	68.3	<u>77.3</u>	71.8	<b>80.0</b>
	Overall	79.8 <sup>+5.5</sup>	74.2	<u>80.4</u>	73.7	<b>81.5</b>
MME-Real	Perception	<u>73.4</u> <sup>+3.0</sup>	70.4	<b>74.3</b>	62.3	73.1
	Reasoning	<u>56.4</u> <sup>+8.7</sup>	47.7	52.5	51.0	<b>58.2</b>
	Overall	<u>71.4</u> <sup>+3.7</sup>	67.7	<b>71.6</b>	60.9	71.3
MME-Real-CN	Perception	<b>76.3</b> <sup>+3.7</sup>	72.6	<u>76.0</u>	65.8	74.5
	Reasoning	<b>59.4</b> <sup>+14.4</sup>	45.0	53.8	51.3	<u>58.3</u>
	Overall	<b>70.8</b> <sup>+7.1</sup>	63.7	68.8	61.2	<u>69.3</u>
MME-Real-Lite	Perception	<b>63.2</b> <sup>+5.2</sup>	58.0	<u>60.2</u>	50.4	59.9
	Reasoning	<u>53.2</u> <sup>+6.9</sup>	46.3	50.7	49.9	<b>55.1</b>
	Overall	<b>59.3</b> <sup>+6.1</sup>	53.2	56.5	50.2	<u>58.3</u>
V*	Attribute	<b>90.4</b> <sup>+8.7</sup>	81.7	79.1	77.3	<u>86.8</u>
	Spatial	<b>84.2</b> <sup>+1.3</sup>	<u>82.9</u>	<u>82.9</u>	64.4	68.4
	Overall	<b>88.0</b> <sup>+5.8</sup>	<u>82.2</u>	80.6	72.3	79.1
TreeBench	Overall	48.4 <sup>+5.7</sup>	42.7	<u>49.6</u>	45.9	<b>54.6</b>
Visual Probe	Hard	<b>42.4</b> <sup>+12.3</sup>	30.1	<b>42.4</b>	28.3	<u>33.9</u>
	Medium	<b>42.9</b> <sup>+7.1</sup>	35.8	<u>39.1</u>	31.3	35.4
	Easy	<b>66.7</b> <sup>+1.5</sup>	65.2	<u>65.9</u>	45.3	49.6
<i>Deep Research</i>						
MMSearch	Overall	<u>66.1</u> <sup>+47.4</sup>	18.7	48.0	64.9	<b>71.9</b>
FVQA	Overall	<u>67.2</u> <sup>+13.9</sup>	53.3	54.4	60.7	<b>72.0</b>
BrowseComp-VL	Overall	38.4 <sup>+8.4</sup>	30.0	31.6	<u>40.8</u>	<b>45.4</b>

and tool-augmented multimodal reasoning.

### 3.3. Efficiency

Based on **3B** activation parameters, we have a significant advantage in inference speed compared to other models. Specifically, we deploy our model using vllm and test Gemini 2.5 Pro and Gemini 2.5 Flash via API requests, based on benchmarks such as MMSearch, FVQA-Test, and Browsecomp-VL. Each query is processed sequentially. We record the start and end timestamps for each benchmark and calculate the average time for comparison in two scenarios. One is **Direct Mode**, where the model can only output based on its internal knowledge, yielding a one-turn result. The other is **Search Mode**, where the model can search the web to answer the question and decide when to stop, resulting in a multi-turn answer in a ReAct style, as opposed to Direct Mode. Furthermore, we record the turn count for the benchmarks mentioned above and the token count for the answers, thereby comparing tokens per second across the models. We choose OpenRouter for the Gemini API.



**Figure 3:** Comparison of model efficiency. The first row presents the results from single-round inference without tool usage. The reported time, average tokens, and tokens per second (TPS) are averaged across samples within each benchmark. The second row shows the results from multi-round inference with code and search tools enabled.

As illustrated in Figure 3, our model achieves substantially faster inference compared to Gemini-2.5-Flash and Gemini-2.5-Pro. Specifically, in single-round evaluation on the MM-Search and FVQA-Test benchmarks, our model is approximately **4× faster than Gemini-2.5-Flash** and **15× faster than Gemini-2.5-Pro**. For BCVL, the speedup is around 5×, mainly because this benchmark is significantly more complex than the others, and we observe that for certain questions in the BCVL benchmark, our model tends to produce repetitive outputs in the single-round mode, often reaching the maximum generation limit. This behavior partially accounts for the relatively higher number of generated tokens compared to the other two benchmarks. In the **multi-round end-to-end (E2E)** setting with code and search tools enabled, our model’s average inference time and tokens-per-second (TPS) remain about **2× higher** than other models. Notably, we include the time spent on tool in the TPS calculation, however, this portion does not correspond to the model’s intrinsic output process. If the tool execution time were excluded, the actual model-side TPS would be even higher.

Note that the full **FVQA-Test** benchmark contains **1,800 questions**, and evaluating the entire set would incur considerable time and computational cost. Therefore, we randomly sampled **180 questions** (10%) as a representative subset for evaluation. Finally, the inference efficiency depends not only on the number of model parameters, but also on factors such as **quantization methods**, **hardware configurations**, and **network latency** during HTTP requests. Hence, the results reported here are based on three benchmarks and **may not fully represent the general performance across all conditions**.

## 4. Related Work

**Multimodal Reasoning and Think with Images.** Improving the reasoning abilities of multimodal large language models (MLLMs) remains a central challenge. A series of prior works, including the R1V, R1V2, and R1V3 frameworks (Peng et al., 2025a, Wang et al., 2025b, Shen et al., 2025b), have highlighted the importance of complex multimodal reasoning and established strong baselines for interleaved vision–language problem solving. Building on this line of research, subsequent studies have increasingly leveraged reinforcement learning (RL) during post-training to boost performance across vision tasks (Liu et al., 2025, Shen et al., 2025a), complex multimodal reasoning (Huang et al., 2025, Peng et al., 2025b, Meng et al., 2025), and even the design of reward models themselves (Zhang et al., 2025b, Wang et al., 2025e,d). However, a common limitation in many of these approaches is their emphasis on enriching the textual reasoning trace, while treating visual inputs as passive context rather than an interactive element within the reasoning loop. To address this, a few recent methods have explored “thinking with images” by enabling models to actively manipulate visual content—e.g., through dynamic cropping (Zheng et al., 2025, Lai et al., 2025) or generating auxiliary visual aids (Chern et al., 2025, Zhang et al., 2025a). More powerful paradigms, such as Pyvision (Zhao et al., 2025) and Thyme (Zhang et al., 2025c), go further by allowing models to synthesize and execute code to perform custom image operations. Our work builds on this direction but significantly extends the toolkit: we equip the agent not only with programmatic image manipulation capabilities but also with access to multimodal retrieval (text and image search) and external web resources, enabling truly interactive, tool-augmented reasoning grounded in both perception and world knowledge.

**Multi-Modal Search.** Since the release of DeepSeek R1 (DeepSeek-AI, 2025) in early 2025 and the subsequent rise in the popularity of reinforcement learning (RL) methods, several studies (Li et al., 2025a, Jin et al., 2025, Feng et al., 2025, Li et al., 2025b) have begun to explore integrating RL with search-based tools to enhance problem-solving. These approaches enable models to retrieve and utilize external knowledge when their internal knowledge is insufficient to answer a question. However, these studies primarily focus on the text-based only domain. Subsequently, a series of works have emerged that explore multimodal reinforcement and the integration of search tools. To the best of our knowledge, MMSearch-R1 (Wu et al., 2025) is the first work to employ reinforcement learning to enhance the search capability of multimodal large language models, and it introduced the FVQA dataset, which is highly valuable for basic multimodal search tasks. Subsequently, WebWatcher (Geng et al., 2025) also made attempts in this direction and proposed the BrowseComp-VL benchmark. Nevertheless, reinforcement learning entails considerable training overhead, and existing agentic RL frameworks are still in their early stages. In contrast, our approach is grounded in supervised fine-tuning (SFT) and provides strong empirical evidence that a small quantity of high-quality SFT data can outperform reinforcement-based methods by a large margin, thereby substantiating the notion that “less is more.” Furthermore, we introduce a novel integration of “thinking with images” and search, which markedly improves the interpretability of multimodal reasoning.

**Multimodal Agentic Planning.** A dedicated planning module helps decompose complex goals into coherent sequences of tool calls, improves long-horizon task assignment, and makes the agent’s decision process more interpretable. Recent work has started to treat planning as an explicit object inside tool-augmented agents. DeepPlanner (Fan et al., 2025) introduces an end-to-end RL framework with advantage shaping that amplifies gradients on high-entropy planning tokens and complex rollouts. This substantially improves planning quality for deep research agents operating over web search and browsing, although the setting remains purely textual. Moving to multimodal retrieval-augmented generation, CogPlanner (Yu et al., 2025) formulates MRAG Planning and equips an MLLM “planning expert” that iteratively decides whether to

call text search, image search, or no retrieval, and introduces CogBench as a dedicated benchmark for evaluating such multimodal planning strategies. However, the aforementioned reinforcement learning-based planners entail considerable computational and engineering overhead, and existing multimodal agentic RL frameworks are still far from practical for large-scale deployment. In contrast, our approach relies solely on SFT, demonstrating that a relatively small amount of carefully curated SFT data is sufficient to learn strong multimodal planning behaviors without the cost and instability of RL. Moreover, instead of fabricating abstract supervision, we extract plans directly from realistic trajectories collected in earlier search-augmented and think-with-images agents, so that the supervision reflects how agents actually interact with tools at inference time. This grounding in real execution traces leads to more faithful plan distributions and substantially higher downstream success rates in multimodal agentic planning.

## 5. Conclusion and Future Directions

Skywork-R1V4 shows that supervised fine-tuning with high-quality, tool-grounded trajectories enables strong multimodal agentic capabilities—spanning executable planning, active image manipulation, deep multimodal search, and interleaved reasoning—without requiring reinforcement learning. By enforcing strict consistency between reasoning steps, tool executions, and final answers across fewer than 30K curated samples, the model achieves state-of-the-art performance on perception and complex search benchmarks, even outperforming larger proprietary systems. This demonstrates that carefully designed SFT data, grounded in real tool interactions, provides a highly effective and efficient path toward agentic multimodal intelligence.

Looking ahead, several promising directions emerge. The framework can incorporate richer visual and web-interaction tools, such as segmentation, depth estimation, or structured DOM navigation, to support more sophisticated tasks. Planning can be enhanced with memory mechanisms or predictive modules to improve long-horizon coherence. Notably, while this work relies solely on SFT, future efforts can explore hybrid paradigms that combine the stability of supervised imitation learning with the adaptability of multimodal agentic reinforcement learning—potentially enabling agents to refine their strategies through environmental feedback. Such integrations may further advance the robustness and autonomy of MLLMs in open-world settings. Skywork-R1V4 establishes a strong foundation for these developments by proving that high-fidelity supervision, even at modest scale, unlocks compelling agentic behavior.

## 6. Contributions

**Core Contributors:** Yifan Zhang\*, Liang Hu\*, Haofeng Sun, Peiyu Wang, Yichen Wei, Yang Liu, Xuchen Song†

**Contributors:** Shukang Yin, Jiangbo Pei, Wei Shen, Peng Xia, Yi Peng, Tianyidan Xie, Eric Li, James Zhou

\* Equal contribution

† Project leader

## References

- Ethan Chern, Zhulin Hu, Steffi Chern, Siqi Kou, Jiadi Su, Yan Ma, Zhijie Deng, and Pengfei Liu. Thinking with generated images. *arXiv preprint arXiv:2505.22525*, 2025.
- DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv*, 2025.
- 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.
- Wei Fan, Wenlin Yao, Zheng Li, Feng Yao, Xin Liu, Liang Qiu, Qingyu Yin, Yangqiu Song, and Bing Yin. Deepplanner: Scaling planning capability for deep research agents via advantage shaping, 2025. URL <https://arxiv.org/abs/2510.12979>.
- Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjuan Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025.
- Mingyang Fu, Yuyang Peng, Benlin Liu, Yao Wan, and Dongping Chen. Livevqa: Live visual knowledge seeking. *arXiv preprint arXiv:2504.05288*, 2025.
- Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, et al. Webwatcher: Breaking new frontier of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*, 2025.
- Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv*, 2025.
- Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Chaoyou Fu, Guanglu Song, et al. Mmsearch: Benchmarking the potential of large models as multi-modal search engines. *arXiv preprint arXiv:2409.12959*, 2024.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025. URL <https://arxiv.org/abs/2503.09516>.
- Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. *arXiv preprint arXiv:2509.07969*, 2025.
- Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models, 2025a. URL <https://arxiv.org/abs/2501.05366>.
- Xuefeng Li, Haoyang Zou, and Pengfei Liu. Torl: Scaling tool-integrated rl, 2025b. URL <https://arxiv.org/abs/2503.23383>.
- Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv*, 2025.

- Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, et al. Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. *arXiv*, 2025.
- Yi Peng, Peiyu Wang, Xiaokun Wang, Yichen Wei, Jiangbo Pei, Weijie Qiu, Ai Jian, Yunzhuo Hao, Jiachun Pan, Tianyidan Xie, Li Ge, Rongxian Zhuang, Xuchen Song, Yang Liu, and Yahui Zhou. Skywork r1v: Pioneering multimodal reasoning with chain-of-thought, 2025a. URL <https://arxiv.org/abs/2504.05599>.
- Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. *arXiv*, 2025b.
- 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*, 2025a.
- Wei Shen, Jiangbo Pei, Yi Peng, Xuchen Song, Yang Liu, Jian Peng, Haofeng Sun, Yunzhuo Hao, Peiyu Wang, Jianhao Zhang, and Yahui Zhou. Skywork-r1v3 technical report, 2025b. URL <https://arxiv.org/abs/2507.06167>.
- Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Jiani Zheng, Sule Bai, Zijian Kang, Jiashi Feng, et al. Traceable evidence enhanced visual grounded reasoning: Evaluation and methodology. *arXiv preprint arXiv:2507.07999*, 2025a.
- Peiyu Wang, Yichen Wei, Yi Peng, Xiaokun Wang, Weijie Qiu, Wei Shen, Tianyidan Xie, Jiangbo Pei, Jianhao Zhang, Yunzhuo Hao, Xuchen Song, Yang Liu, and Yahui Zhou. Skywork r1v2: Multimodal hybrid reinforcement learning for reasoning, 2025b. URL <https://arxiv.org/abs/2504.16656>.
- Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2413–2427, 2017.
- Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7907–7915, 2025c.
- Xiaokun Wang, Peiyu Wang, Jiangbo Pei, Wei Shen, Yi Peng, Yunzhuo Hao, Weijie Qiu, Ai Jian, Tianyidan Xie, Xuchen Song, Yang Liu, and Yahui Zhou. Skywork-vl reward: An effective reward model for multimodal understanding and reasoning, 2025d. URL <https://arxiv.org/abs/2505.07263>.
- Xiyao Wang, Chunyuan Li, Jianwei Yang, Kai Zhang, Bo Liu, Tianyi Xiong, and Furong Huang. Llava-critic-r1: Your critic model is secretly a strong policy model. *arXiv preprint arXiv:2509.00676*, 2025e.
- Luis Wiedmann, Orr Zohar, Amir Mahla, Xiaohan Wang, Rui Li, Thibaud Frere, Leandro von Werra, Aritra Roy Gosthipaty, and Andrés Marafioti. Finevision: Open data is all you need. *arXiv preprint arXiv:2510.17269*, 2025.
- Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. Mmsearch-r1: Incentivizing lmms to search, 2025. URL <https://arxiv.org/abs/2506.20670>.



- Penghao Wu and Saining Xie. V?: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13084–13094, 2024.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. *CoRR*, abs/2210.03629, 2022.
- Xiaohan Yu, Zhihan Yang, and Chong Chen. Cogplanner: Unveiling the potential of agentic multimodal retrieval augmented generation with planning, 2025. URL <https://arxiv.org/abs/2501.15470>.
- Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.
- Huanyu Zhang, Wenshan Wu, Chengzu Li, Ning Shang, Yan Xia, Yangyu Huang, Yifan Zhang, Li Dong, Zhang Zhang, Liang Wang, et al. Latent sketchpad: Sketching visual thoughts to elicit multimodal reasoning in mllms. *arXiv preprint arXiv:2510.24514*, 2025a.
- Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint arXiv:2408.13257*, 2024.
- Yi-Fan Zhang, Xingyu Lu, Xiao Hu, Chaoyou Fu, Bin Wen, Tianke Zhang, Changyi Liu, Kaiyu Jiang, Kaibing Chen, Kaiyu Tang, et al. R1-reward: Training multimodal reward model through stable reinforcement learning. *arXiv preprint arXiv:2505.02835*, 2025b.
- Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, et al. Thyme: Think beyond images. *arXiv preprint arXiv:2508.11630*, 2025c.
- Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Ming Li, Qilong Wu, Kaipeng Zhang, and Chen Wei. Pyvision: Agentic vision with dynamic tooling. *arXiv preprint arXiv:2507.07998*, 2025.
- Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deep-eyes: Incentivizing "thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

## Appendix

### A. Dialogue Modes

In this section, we present the dialogue formats supported by our model. Our system provides **two distinct dialogue modes**, each designed for different levels of reasoning, tool usage, and performance requirements.

**General Mode** In this default mode, the model relies primarily on its internal knowledge and code tool to enhance image understanding capability. It avoids heavy external tools, enabling fast responses while maintaining strong multimodal reasoning performance.

**DeepResearch Mode** In this mode, the model performs autonomous iterative reasoning and external tool use. It interleaves image analysis, image manipulation, external search, and other tool-augmented operations to progressively gather evidence and refine its understanding until reaching a final, well-supported answer.

**Plan Mode** The model outputs a sequence of execution steps and high-level strategies, enabling structured reasoning and task planning grounded in visual context.

Examples of dialogues under each mode are shown below.

Table 3: System Prompt for Planner Tasks.

You are a planning and tool-orchestration assistant.

**Goal:** Given a user’s question (and optionally an image), produce a **step-by-step plan** that solves the task. The plan must be returned **exactly** in the following assistant content format (a single JSON array string), where **each step is an object with three keys**: "description": A precise natural-language instruction describing the reasoning or operation. "tool\_name": One of "image\_search", "text\_search", "web\_visit", or "none" (use "none" when no tool is needed). "parameters": A JSON object containing structured parameters for the tool call (or {} if none).

**Tool semantics:** image\_search: Identify people/objects/scenes from an image. Parameters: {"image\_path": "<path>"}. text\_search: Query a search engine for facts. Parameters: {"query": "<query>"}. web\_visit: Extract/verify details from a webpage. Parameters: {"url": "<URL>"}. none: For reasoning/summarization. Parameters: {}. Use tools **only when necessary**, but be concrete and complete when they help.

**Dependencies & placeholders:** When a step depends on prior results, use explicit placeholders like [Person from Step 1] in both description and parameters.

**Output format (critical):** Return **only** a single string containing a JSON array of step objects—no extra prose, markdown, or explanations. Example:

```
[{
  "description": "Identify the person shown in the image...",
  "tool_name": "image_search",
  "parameters": {"image_path": "/data/images/person_1.png"}
}, ...]
```

**Writing style:** Steps must be **concise, actionable, and specific**. Prefer 2–10 steps. Include a final reasoning/verification step with "tool\_name": "none".

**Allowed tool names (exact match required):** "image\_search", "text\_search", "web\_visit", "none".

**Example shaping:**

1. Analyze the question and identify key entities. ("none")
2. If an image is provided, analyze it. Use image\_search if external knowledge is needed.
3. Use text\_search for background facts.
4. Use web\_visit to verify on authoritative pages.
5. Use "none" to reason over findings.
6. Conclude with a verification step ("none").

**Now produce the plan:** Given the user’s input, return **only** the assistant content string as specified—no other text.



**Question:** Whats this? explain it in detail

**Model Responses:** <think> The user is asking about a person's team win rate in the 2025 League of Legends World Championship (S Series). The image shows a person in a gaming setting wearing a Telcom-sponsored hoodie. Let me break down what needs to be done:

1. First, I need to identify who this person is using image search
2. Then find out which team they play for
3. Search for information about that team's performance in the 2025 League of Legends World Championship
4. Find their specific win rate

Let me create a step-by-step plan:

Step 1: Use image\_search to identify the person in the image

Step 2: Use text\_search to find out which team this person plays for

Step 3: Use text\_search to find information about that team's performance in the 2025 LoL World Championship

Step 4: Use reasoning to determine the win rate based on the search results

Let me format this properly as a JSON array: </think>

1. **Step 1** (image\_search): Identify the person shown in the image using reverse image search. image\_path = "image path"
2. **Step 2** (text\_search): Search for information about which team the identified person plays for in League of Legends. query = "[Person identified in Step 1] team League of Legends current"
3. **Step 3** (text\_search): Search for the team performance and win rate of the identified team in the 2025 League of Legends World Championship (S Series). query = "[Team from Step 2] 2025 League of Legends World Championship S Series win rate performance"
4. **Step 4** (web\_visit): Visit a reliable esports statistics website or official LoL Championship page to verify the team's win rate in the 2025 World Championship. url = "[URL from Step 3 containing team statistics]"
5. **Step 5** (none): Analyze the gathered information to determine the specific win rate of the team in the 2025 League of Legends World Championship.

Figure 4: Plan Mode.



**Question:** What is written in the image? Please try to answer the question with short words or phrases if possible.

**Model Responses:** <think> I need to find a dog in this image to determine its color. The image shows a panoramic view of Paris with many buildings, streets, and people, but I don't immediately see a dog. I should crop the image to focus on areas where people are gathered, as dogs might be with their owners in these areas. I'll start by cropping the central area where there's a park with many people, as this seems like a likely place to find a dog.</think>

```
import os
...
# Calculate crop coordinates for the central park area
left = int(width * 0.4)
top = int(height * 0.6)
right = int(width * 0.7)
bottom = int(height * 0.9)
...
# Print the path of the saved enhanced image
print(processed_path)
```

<observation> <sub-image 1> </observation> <think> The current crop shows the park area with many people, but I still don't see a clear dog. I need to look more closely at the areas where people are gathered, particularly around the edges of the park where dogs might be with their owners. I'll crop a smaller area focusing on the left side of the park where there are more people walking along the path. </think>

```
import os
...
# Calculate crop coordinates for the left side of the park
where people are walking
...
```

<observation> <sub-image 2> </observation> <think> The current crop shows a street scene with many cars and people, but I still don't see a clear dog. I need to look more carefully at the areas where people are walking, particularly around the edges of the image where dogs might be with their owners. I'll crop a smaller area focusing on the right side of the image where there are more people walking along the sidewalk. </think>

```
import os
...
# Calculate crop coordinates for the right side of the
image where people are walking
...
```

<observation> <sub-image 3> </observation> <think> In the current crop, I can now see a small white dog walking on the sidewalk near the people. The dog appears to be white in color. Since I can now clearly see the dog and determine its color, I can provide the answer to the question. The dog is white, which corresponds to option D (Black and white) if we consider white as one of the colors in the black and white combination. </think>

<answer> D. Black and white </answer>

**Figure 5:** Skywork-R1V4 enables dynamic visual exploration by iteratively cropping and querying different regions of an image to locate target objects. Starting from a panoramic view of Paris, the model strategically zooms into high-activity zones (e.g., parks and sidewalks), progressively refining its focus until it successfully identifies a small white dog — demonstrating adaptive reasoning and spatial navigation for fine-grained visual understanding.



**Question:** where the picture was taken?

**Model Responses:** **<think>** I need to analyze this image to identify the location. The image shows traditional Chinese architecture with distinctive features like curved tile roofs, white walls with dark wooden beams, and a sign that reads Shengjia Hotel. The background shows modern high-rise buildings, indicating this is likely in a Chinese city where traditional and modern architecture coexist. The architectural style and the hotel sign suggest this could be in a historic district or tourist area. I should use image search to get more specific information about this location. **</think>**

**<tool\_call>** {"name": "image\_search", "arguments": {"image\_paths": [<image>] } **</tool\_call>**

**<observation>** 1. JINC SAINT Hotel (Jiefangbei Huguang Assembly Hall) link: [link 1] image: [image 1]

2. Yangmeiling Scenic Spot (2025) - All You Need to Know BEFORE ... link: [link 2] image: [image 2] ...

**</observation>**

**<think>** Based on the image search results, I can see multiple references to Chongqing, China, and specifically mentions of JINC Saint Hotel in Chongqing. The search results show this is located in Chongqing's Yuzhong District, and there are references to it being near Jiefangbei and Huguang Assembly Hall. The architectural style and the hotel sign (Shengjia Hotel) confirm this is in China. Let me search for more specific information about this location. **</think>**

**<tool\_call>** {"name": "text\_search", "arguments": {"queries": ['Chongqing JINC Saint Hotel location']} **</tool\_call>** }

**<observation>** 1. Jinke Saint Hotel link: [link 1] text: Jinke Saint Hotel). 6th Floor, Tower A, Caihui Plaza, No. 99 Minzu Road, Yuzhong District, Chongqing ...

**</observation>**

**</think>** Based on the search results, I now have clear information about the location. The image shows the JINC Saint Hotel in Chongqing, China. The search results consistently indicate this hotel is located in Chongqing's Yuzhong District, specifically:

The hotel is located at No. 99 Minzu Road, Caihui Plaza, Yuzhong District, Chongqing It's near Jiefangbei (Liberation Monument area) It's close to Hongyadong , which is a famous tourist attraction The area is known for its traditional Chinese architecture with modern high-rises in the background The hotel is described as being in the heart of Chongqing's tourist and commercial district The architectural style and the hotel sign visible in the image match this location description perfectly. **</think>**

**<answer>**

This image was taken in Chongqing, China, specifically in the Yuzhong District near the Hongyadong area. The photo shows the JINC Saint Hotel located at No. 99 Minzu Road, Caihui Plaza, with the distinctive traditional Chinese architecture of the Hongyadong area visible in the foreground and modern high-rise buildings of Chongqing in the background.

**</answer>**

**Figure 6:** Skywork-R1V4 demonstrates flexible integration of image search and text search to accurately determine the geographic location of a photograph — here, identifying the JINC SAINT Hotel in Chongqing's Yuzhong District by cross-referencing visual architectural cues with textual and geospatial data.



**Question:** What is the specific event that occurs when the crown on the main smartwatch in the center is pressed five times rapidly?

**Model Responses:** **<think>** I need to analyze this image to understand what happens when the crown on the main smartwatch is pressed five times rapidly. The image shows a smartwatch with a "Fall Detection" settings screen displayed. I can see options for "Always detect" and "During exercise" with the "Always detect" option selected. However, I need to focus on the crown mechanism of the watch to understand what happens when it's pressed five times rapidly.

Let me crop the main smartwatch to get a clearer view of its crown and any indicators that might show what happens when it's pressed rapidly.

**</think>**

```
import os
...
# Crop the main smartwatch in the center to focus on the crown
area
...
```



**<observation>** **<sub-image 1>** **</observation>**

**<think>** Now I have a cropped image focusing on the main smartwatch. I can see the watch face displaying "Fall Detection" settings with options for "Always detect" and "During exercise". However, I need to look more closely at the crown mechanism to understand what happens when it's pressed five times rapidly. Let me search for this specific image to get more information about the watch model and its features. **</think>**

**<tool\_call>** {"name": "image\_search", "arguments": {"image\_paths": [**<sub-image 1>**]}} **</tool\_call>**

**<observation>** 1. Amazon.com: Ticwatch Atlas - Reloj inteligente para hombre, Android Wear OS, reloj inteligente al aire libre, batería de 90 horas, más de 110 modos de entrenamiento, mapa de calor, detección de caídas, link: [link 1] image: [image 1]  
2. Amazon.com: Ticwatch link: [link 2] image: [image 2] ...

**</observation>**

**<think>** Based on the image search results, this appears to be a Ticwatch Atlas smartwatch. The search results show it's an Android Wear OS smartwatch with fall detection features. However, I need to find specific information about what happens when the crown is pressed five times rapidly. This is likely a specific feature or emergency function. Let me search for more detailed information about Ticwatch Atlas's crown press functionality.

**</think>**

**<tool\_call>** {"name": "text\_search", "arguments": {"queries": ["Ticwatch Atlas crown press five times"]}} **</tool\_call>**

**<observation>** 1. TicWatch Atlas: How to use Fall Detection & Emergency SOS link: [link 1] text: You can also press the rotating crown five times to instantly initiate an SOS call. \*Fall Detection functionality depends on network ... **</observation>**

**</think>** Perfect! Now I have clear information about what happens when the crown on the Ticwatch Atlas is pressed five times rapidly. Based on multiple search results, I can see that pressing the rotating crown five times instantly initiates an Emergency SOS call. This is a safety feature that alerts emergency contacts when the user is in distress. Let me provide a clear and concise answer about this specific event. **</think>**

**<answer>**

When the crown on the main Ticwatch Atlas smartwatch is pressed five times rapidly, it instantly initiates an Emergency SOS call. This safety feature alerts emergency contacts when the user is in distress or needs immediate assistance. **</answer>**

**Figure 7:** Skywork-R1V4 demonstrates interleaved thinking by dynamically alternating between image operations (e.g., cropping to focus on the crown) and multimodal search tools (image + text search) to identify key visual elements, and retrieve contextual knowledge — enabling precise, grounded answers to complex visual queries.