Self-Evolutionary Reinforced Knowledge Distillation for Multi-Modal Tool-Use Agents

Lei Shen*
Xi'an Jiaotong University
Xi'an, China
shenlei@stu.xjtu.edu.cn

Chengyu Wang†
Alibaba Group
Hangzhou, China
chengyu.wcy@alibaba-inc.com

Yuanjie Lyu
Alibaba Group
Hangzhou, China
s1583050085@gmail.com

Yuanhao Yue
Alibaba Group
Hangzhou, China
yueyuanhao.yyh@alibaba-inc.com

Jun Huang
Alibaba Group
Hangzhou, China
huangjun.hj@alibaba-inc.com

Zhongmin Cai†
Xi'an Jiaotong University
Xi'an, China
zmcai@mail.xjtu.edu.cn

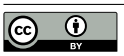
Abstract

Autonomous multi-modal agents are increasingly important in real-world applications due to their ability to reason about complex environments and orchestrate tool use. However, deploying multi-modal large language models (MLLMs) for tool use is often constrained by computational cost and inference latency, creating a pressing need for compact models that retain strong agentic capabilities. Training small multi-modal agents remains difficult: limited backbone capacity weakens multi-step reasoning, reward signals for tool use are often sparse and brittle, and naive distillation can fail to transfer the procedural knowledge required for reliable tool invocation and grounding. In this paper, we propose a two-stage self-evolutionary knowledge distillation framework that equips small MLLMs with robust and adaptive tool-use behaviors. Our method combines (i) mutual information-guided trajectory distillation, which selectively transfers high-utility segments of agentic trajectories from a larger teacher, and (ii) reinforcement-driven policy evolution with iterative teacher feedback. To stabilize learning and prevent semantic collapse, we introduce weighted semantic objectives and iteratively expand competence through error-driven optimization, hybrid experience replay, and group-relative policy refinement with multi-dimensional rewards over answer correctness, invocation validity, and tool effectiveness. Integrated with interactive tool modules, our approach enables small models to achieve strong performance across diverse tool-use benchmarks. Comprehensive experiments show consistent improvements over single-pass distillation and RL baselines. Overall, our framework provides a practical path to deploy efficient multi-modal agents without sacrificing tool-use reliability.

CCS Concepts

• **Computing methodologies** → **Intelligent agents**.

*Work was done when interned at Alibaba Group.
†Corresponding authors.



This work is licensed under a Creative Commons Attribution 4.0 International License.
KDD '26, Jeju Island, Republic of Korea
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2259-2/2026/08
https://doi.org/10.1145/3770855.3817780

Keywords

Multi-modal tool use; knowledge distillation; reinforcement learning

ACM Reference Format:
Lei Shen, Chengyu Wang, Yuanjie Lyu, Yuanhao Yue, Jun Huang, and Zhongmin Cai. 2026. Self-Evolutionary Reinforced Knowledge Distillation for Multi-Modal Tool-Use Agents. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '26), August 09–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3770855.3817780

1 Introduction

Multi-modal large language models (MLLMs) [1, 35, 41, 45, 60] have become central to recent progress in artificial intelligence, enabling agents to perceive, reason, and act in complex real-world environments by integrating diverse modalities such as vision, text, and structured data [12, 13, 24]. The capacity of such agents to orchestrate tool use, namely to dynamically invoke external functions and resources for solving complex compositional tasks, has been a key driver of recent advances in embodied AI [33, 42, 52], interactive reasoning systems [5, 8, 21], and task-oriented automation [27, 49]. Figure 1 illustrates a representative agentic trajectory in which the model interleaves intermediate reasoning with tool calls and tool outputs over multiple steps. Representative applications include visual question answering [4, 15, 53, 57], robotic manipulation [7, 18], and web-based information retrieval [10, 19, 37, 40], all of which require agents to interpret heterogeneous inputs and make effective decisions through adaptive tool use.

Despite their promise, state-of-the-art agentic MLLMs remain prohibitively large and computationally intensive, restricting deployment in resource-constrained settings such as edge devices, embedded systems, and real-time applications [20, 26]. High memory demands and slow inference latency significantly limit the practicality of large multi-modal agents for scalable and responsive tool use in the physical world. These constraints create an urgent need for compact and efficient multi-modal agents that retain advanced reasoning and tool-use capabilities while operating at a fraction of the computational cost [28].

However, distilling multi-modal agents introduces challenges in both learning and generalization. Smaller models suffer from

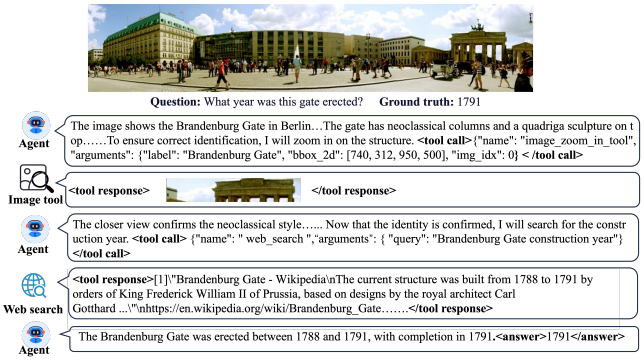


Figure 1: An agentic trajectory generated by an MLLM agent. The agent interleaves intermediate reasoning with tool invocations and tool outputs to solve a multi-modal task.

reduced capacity, hindering their ability to capture complex dependencies and the nuanced reasoning required for sophisticated tool use [2, 3, 23]. The multi-modal nature of tool-use tasks further exacerbates optimization difficulty: reward signals are often sparse or brittle, which can destabilize policy learning and impede effective exploration [25, 29]. Moreover, although knowledge distillation from large teacher models promises efficient transfer, it can lead to semantic collapse or incomplete transfer of reasoning procedures, particularly for agentic behaviors involving multi-step tool integration [36]. As a result, standard reinforcement learning or distillation pipelines often fail to produce well-structured tool invocations and semantically coherent decisions across diverse scenarios.

To address these bottlenecks, in this paper, we propose a systematic self-evolutionary reinforced knowledge distillation framework, REvoKD[1], tailored to multi-modal tool-use agents under model-size constraints. Our approach follows a two-stage paradigm: (i) mutual information–guided trajectory distillation and (ii) multi-round reinforced self-evolution. In the first stage, we employ an information-theoretic distillation procedure that selectively transfers high-utility segments of agentic trajectories from a large teacher agent to a small student model, ensuring that the student is initialized with the knowledge essential for effective tool-use execution.

In the second stage, we conduct a self-evolutionary reinforcement learning cycle. The student agent interacts with multi-modal tool-use environments, receiving targeted feedback from the teacher agent and rewards from a multi-dimensional evaluator that assesses answer correctness, output-format validity, and tool-invocation effectiveness. Our framework incorporates hybrid experience replay and group-relative policy optimization (GRPO) [32] to improve training stability and behavioral diversity, while refining the policy in response to unresolved errors. Interactive tool modules are integrated directly into the training loop, enabling environment exploration and access to up-to-date knowledge.

Comprehensive empirical evaluation on diverse tool-use benchmarks (SimpleVQA, Visual Probe, MME-Real-Lite, FVQA-Test, and MMSearch) shows substantial gains over conventional reinforcement learning and single-pass distillation approaches. In particular, on Qwen3-VL-4B-Instruct, REvoKD improves the Vision Perception & Reasoning average from 33.5 (SFT) / 34.2 (GRPO) to 42.8, and the Deep Research average from 37.6 (SFT) / 34.7 (GRPO) to 39.7. We observe similar trends on the smaller 2B student model.

In summary, our main contributions are as follows:

- We introduce REvoKD, a self-evolutionary reinforced knowledge distillation framework that enables small multi-modal agents to acquire robust and adaptive tool-use capabilities from large teacher models.
- We develop mutual information–guided trajectory distillation and a multi-round reinforced self-evolution mechanism that together enable effective knowledge transfer and continual policy improvement in complex tool-use environments.
- Extensive experiments on diverse tool-use benchmarks demonstrate that REvoKD consistently outperforms reinforcement learning and distillation baselines in tool-use accuracy and task performance under limited computational resources.

2 Related Work

Multi-Modal Tool-Use Agents. Multi-modal large language models (MLLMs) have made substantial progress in recent years on vision–language understanding, cross-modal reasoning, and task planning (e.g., GPT-4V[2] and the Qwen3-VL series [1]). These models can operate in agentic tool-use settings by selecting and invoking external tools (e.g., search engines and image-processing APIs) according to task requirements, enabling automated execution of complex multi-step tasks.

Recent work has largely focused on integrating tool use capabilities into conversational agents. For instance, ReAct [46] couples environment perception with action generation by interleaving tool calls with chain-of-thought reasoning. Toolformer [30] learns, via self-supervision, when to insert API calls within reasoning traces to improve task completion. Visual ChatGPT [39] connects vision models to chat environments to enable invocation of image-editing tools. While these methods demonstrate strong performance for large models, they often become unstable when transferred to smaller backbones due to reduced representational capacity and weaker multi-modal fusion. Our work targets efficient transfer of tool-use capabilities to small multi-modal agents, with particular emphasis on improving tool-invocation accuracy and policy stability in realistic interactive environments.

Knowledge Distillation for Multi-Modal Trajectory Transfer. Knowledge distillation (KD) compresses knowledge from a large teacher into a smaller student model [11]. For multi-modal tool-use agents, effective transfer often requires distilling not only the final output but also the tool-use trajectory, *i.e.*, the sequence and structure of tool invocations as well as the quality of generated tool parameters, both of which directly affect task success. Conventional KD pipelines typically rely on single-stage supervised fine-tuning (SFT). For example, MiniGPT-4 [59] distills vision-language capabilities from large image-text corpora, enabling smaller models to exhibit strong multi-modal conversational behavior. However, single-stage KD faces two key limitations: (i) distilled data quality depends heavily on the teacher, so teacher noise can induce student

[1]REvoKD stands for Reinforced self-Evolutionary Knowledge Distillation.

[2]https://openai.com/index/gpt-4v-system-card/

miscalibration or policy drift; and (ii) the lack of online interaction restricts targeted adaptation to failure modes that only appear during tool execution.

Several recent studies have explored interactive alignment and data augmentation, such as LLaVA-RLHF [34], which improves model outputs via feedback. However, these methods primarily focus on static generation tasks and typically do not explicitly distill or optimize realistic tool-use trajectories in interactive environments.

Agentic Reinforcement Learning. For multi-step planning and tool use, reinforcement learning (RL) provides a principled framework for exploration and policy optimization in interactive and dynamic environments [16, 51]. Unlike single-turn tasks, agentic RL emphasizes a closed-loop interaction cycle in which an agent perceives, plans, decides, and acts; tool invocation becomes an action type within the policy space [50]. This perspective is increasingly adopted in MLLM-based agents, where effective learning requires credit assignment over long trajectories and robustness to non-stationary tool outputs.

Classic policy optimization methods such as proximal policy optimization (PPO) [31] have been widely adopted for sequential decision-making, including dialogue generation and robotic control, and improve training stability by constraining the magnitude of policy updates. Nevertheless, in multi-modal agent settings, the tool-use action space is large and trajectory diversity is high, which can slow convergence and limit exploration when PPO is trained with a single reward objective [56].

Recently, group relative policy optimization (GRPO) [32] has been proposed to update policies based on relative advantage comparisons within a batch, prioritizing higher-value trajectories and often improving robustness [9, 22]. Although GRPO has been explored for reward optimization in multi-turn dialogue, several challenges remain [47]. Existing reward signals are often one-dimensional, focusing primarily on task success while neglecting output-format validity and tool-use intent. Moreover, limited integration with genuinely interactive environments can lead to performance degradation at deployment time, and the lack of explicit error-repair mechanisms hinders continual improvement of tool-use policies.

3 Methodology

3.1 Framework Overview

As shown in Figure 2, REvoKD follows a sequential two-stage training framework. In Stage 1, the student agent performs trajectory distillation on curated multi-modal tool-use exemplars, using a weighted semantic objective to emphasize high-utility components of teacher trajectories. This stage initializes the student with reliable tool-use behaviors and structured trajectory patterns. In Stage 2, training starts from the distilled student obtained in Stage 1 and further improves the policy through iterative reinforced self-evolution in dynamic multi-modal environments. In this stage, the agent interacts with tools, receives teacher feedback on failed rollouts, replays successful experiences, and is optimized with multi-dimensional rewards under GRPO. The two stages are executed sequentially: the distillation stage provides initialization, and the RL stage performs subsequent policy refinement. This training recipe supports continual improvement and better generalization of tool-use behaviors.

3.2 Dataset Construction

We curate specialized datasets to support both trajectory distillation and reinforcement-driven evolution, balancing broad task coverage with high-quality multi-step tool trajectories.

Trajectory Distillation Stage. The dataset for trajectory distillation is primarily constructed from Thyme-SFT [53]. We first filter approximately 180,000 raw samples and then classify them using Qwen3-VL-Plus [1]. The classifier assigns each sample to one of thirteen tool-use task categories: attribute recognition, quantity recognition, location recognition, object recognition, scene recognition, map understanding, action recognition, textual reasoning, event understanding, compositional reasoning, temporal analysis, change detection, and quality analysis. This pipeline yields 45,000 candidate samples. To extract effective trajectories for multi-turn tool-use tasks, we use multi-dimensional rejection sampling on sequences from the teacher agent Qwen3-VL-Plus, enforcing semantic correctness and logical compositionality. We further validate candidate trajectories using Qwen3-Max [43] as an LLM-as-a-judge. The final distillation corpus contains 14,000 high-quality trajectories, which serve as seeds for knowledge transfer.

Reinforcement Evolution Stage. For policy evolution, the training corpus integrates samples from Thyme-RL [53] and Visual Probe [21]. In addition to maintaining the same category taxonomy for balanced coverage, we emphasize instances that require robust grounding of tool results and multi-step error correction. Following the same pipeline, we select 5,000 categorized samples from Thyme-RL with the same set of task types. Additionally, we incorporate 2,000 targeted samples from Visual Probe to strengthen visual grounding. This combined dataset supports iterative optimization and continual expansion of tool-use strategies.

3.3 Mutual Information-Guided Trajectory Distillation

Rather than relying solely on standard supervised learning, we formulate trajectory distillation as a selective process guided by the mutual information (MI) between the generated trajectory and the input context. In our setting, a *trajectory* is serialized as an output sequence $Y = (y_1, \ldots, y_T)$ that interleaves (i) intermediate natural-language reasoning, (ii) tool-call *actions* (including the tool name and its arguments), (iii) tool *observations* (the tool outputs returned to the agent), and (iv) the final answer. Formally, given a multi-modal input x and the output sequence Y, MI admits the chain-rule decomposition

$$I(Y; x) = \sum_{t=1}^{T} I(y_t; x \mid y_{<t}), \tag{1}$$

where $I(y_t; x \mid y_{<t})$ measures how strongly token y_t depends on the input x beyond what is predictable from the prefix $y_{<t}$. To concentrate learning on high-utility components, we define a weighted mutual information objective:

$$\tilde{I}(Y; x) = \sum_{t=1}^{T} w_t \, I(y_t; x \mid y_{<t}), \tag{2}$$

where w_t modulates the relative emphasis across token positions.

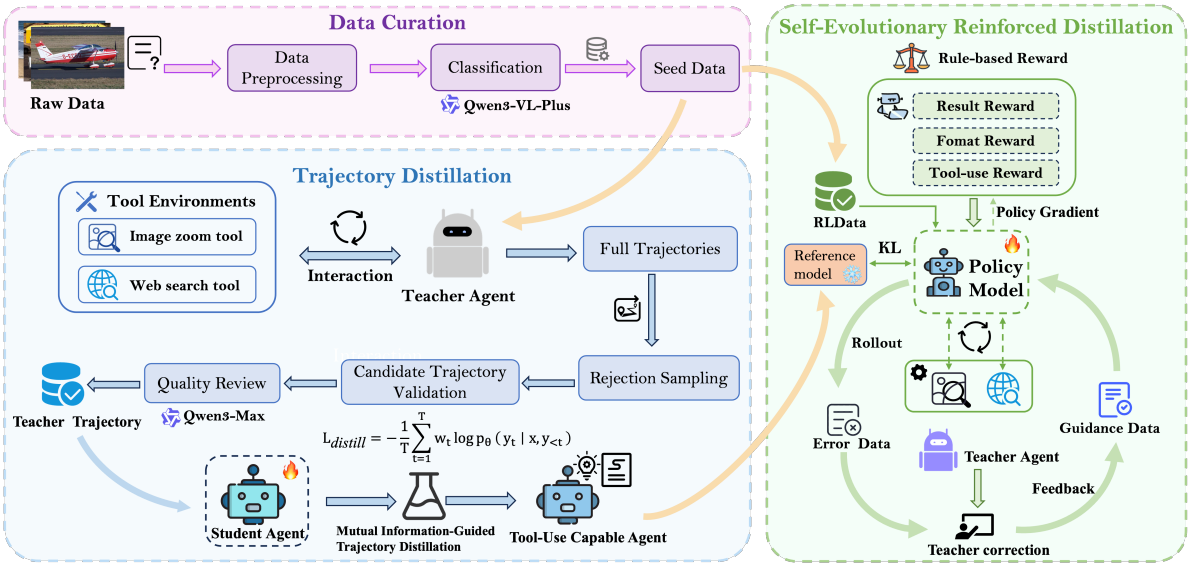


Figure 2: Overview of REvoKD for knowledge distillation in multi-modal tool-use agents.

Directly computing or optimizing $\tilde{I}(Y;x)$ is intractable for large sequence models; therefore, we optimize a weighted negative log-likelihood surrogate that emphasizes information-rich tokens.

$$\mathcal{L}_{\text{distill}} = -\frac{1}{T}\sum_{t=1}^{T} w_t \log p_\theta(y_t \mid x, y_{<t}), \tag{3}$$

with

$$w_t = \begin{cases} \lambda_{\text{tool}}, & \text{if } y_t \text{ is tool-related} \\ \lambda_{\text{other}}, & \text{otherwise.} \end{cases} \tag{4}$$

Note that we refer to tokens associated with tool names, arguments, and tool outputs as *tool-related* tokens. For implementation, trajectory distillation is performed using low-rank adaptation (LoRA) [14], which constrains updates to a low-dimensional subspace, thereby preserving the model's priors while allowing efficient adaptation to task-specific semantics. From an information-theoretic perspective, LoRA can be interpreted as a regularized projection in parameter space, reducing the risk of semantic collapse [44].

3.4 Multi-Round Reinforced Self-Evolution

3.4.1 Multi-Modal Tools. Our framework incorporates two interactive modules: an *Image Zoom* tool and a *Web Search* tool. These modules provide external information during policy evolution, broadening the agent's effective perceptual resolution and improving access to up-to-date facts. The Image Zoom tool enables fine-grained manipulation of visual inputs via zoom-in, zoom-out, and cropping operations, thereby improving local feature extraction and spatial attention. This functionality allows the agent to focus on salient image regions, which is essential for context-aware object recognition and compositional reasoning. The Web Search tool provides real-time access to information from the Internet, enriching the agent's internal representations with external factual evidence. This capability is particularly important for queries requiring factual precision and temporal relevance. Each tool interfaces with the agent through structured parameters and returns outputs in a unified format, allowing both trajectory distillation and reinforcement learning to operate on unified text-action sequences rather than tool-specific semantics. For efficiency, tool outputs are cached in an experience replay buffer implemented with SQLite[3], which reduces redundant tool calls and improves training efficiency.

3.4.2 Multi-Round Reinforced Self-Evolution. To robustly improve tool-use behavior in multi-modal environments, we propose a multi-round reinforced self-evolution paradigm as the second stage of REvoKD. Conventional single-pass RL approaches [58] often suffer from limited adaptivity to errors, policy instability, and inefficient exploration in high-dimensional action spaces. REvoKD addresses these limitations via iterative, error-driven learning cycles that couple targeted teacher feedback with group-based policy refinement.

We formalize multi-round distillation as a closed-loop optimization process over the agent's policy. Let $\mathcal{D}$ denote a set of multi-modal tool-use tasks. For each $x \in \mathcal{D}$, a rollout produces a tool-use trajectory y. Let $\pi_k(\cdot \mid x, h_x)$ denote the student policy at round k, conditioned on the input x and its associated context h_x.

Stage 1: Error-Guided Teacher Feedback. At round k, the agent performs policy rollouts over $\mathcal{D}$ and collects input–trajectory pairs:

$$\mathcal{Y}_k = \{(x, y) \mid x \in \mathcal{D},\ y \sim \pi_k(\cdot \mid x, h_x)\}, \tag{5}$$

where h_x denotes the context information for input x.

The semantic evaluation operator Eval maps a pair (x, y) to a scalar score $s(x, y) \in [0, 1]$. We define the error set as

$$\mathcal{E}_k = \{(x, y, s) \mid (x, y) \in \mathcal{Y}_k,\ s = \text{Eval}(x, y),\ s < \tau_{\text{err}}\}, \tag{6}$$

[3] https://sqlite.org/

where τ_{err} is a tunable error threshold. We further define the aggregate semantic error as

$$\text{Err}_k = \frac{1}{|\mathcal{D}|} \sum_{(x,y,s)\in\mathcal{E}_k} (1-s). \tag{7}$$

Failed predictions are forwarded to the teacher agent $\mathcal{T}$, which returns a feedback tuple for each error: a diagnostic explanation $d(x,y)$, an augmented context $\hat{x}$, and a corrected trajectory y^*. The set of teacher-guided refinements at round k is

$$\mathcal{G}_k = \{(\hat{x}, y^*, d) \mid (x,y,s)\in\mathcal{E}_k,\ (\hat{x}, y^*, d) = \mathcal{T}(x,y,s)\}. \tag{8}$$

Claim 1 (Error Reduction via Structured Feedback). Assume that teacher corrections satisfy $\text{Eval}(\hat{x}, y^*) \geq \tau_{\text{succ}}$ for all $(\hat{x}, y^*, d) \in \mathcal{G}_k$, and that the policy update is KL-regularized to stay close to π_k. Then incorporating $\mathcal{G}_k$ into subsequent updates decreases the measured error on the corrected samples and typically reduces the aggregate error over rounds.

Rationale. The teacher provides high-scoring corrections for previously failed cases; KL-regularized updates encourage the updated policy to increase the likelihood of these corrections without inducing large distribution shifts. Empirically, this yields monotonic improvements on the corrected subset and reduces overall error as correction coverage grows.

Stage 2: Policy Refinement via Experience Replay and GRPO. The policy π_k is updated using an experience replay buffer $\mathcal{M}_k = \mathcal{G}_k \cup C_k$, where $\mathcal{G}_k$ contains teacher-corrected samples and C_k stores previously successful cases.

Let g_i denote a group of trajectories that share the same input x_i but differ in sampled candidates, constructed following GRPO [32]. For each group, the normalized comparative advantage is

$$A_{g_i}(y) = \frac{R(y) - \mu_{g_i}}{\sigma_{g_i} + \epsilon}, \tag{9}$$

where $R(y)$ denotes the scalar trajectory-level reward, μ_{g_i} and σ_{g_i} denote the group mean and standard deviation of rewards, and ϵ is a small constant.

Agent parameters are updated by maximizing the expected group advantage subject to KL regularization:

$$\theta_{k+1} = \arg\max_{\theta} \left[\mathbb{E}_{(x,y)\sim\pi_\theta}[A_{g(x)}(y)] - \beta\,\text{KL}(\pi_\theta\|\pi_k)\right], \tag{10}$$

where $g(x)$ denotes the group associated with input x.

After the update, mastery is consolidated by adding sufficiently good samples to the success set:

$$C_{k+1} = C_k \cup \{(x,y) \mid (x,y)\in\mathcal{Y}_k,\ \text{Eval}(x,y) \geq \tau_{\text{succ}}\}. \tag{11}$$

Stage 3: Global Self-Evolution and Mastery. The correction and refinement process is repeated for K rounds, or until $\text{Err}_k < \varepsilon$. The success set is updated as above, while the error set is recomputed at each round from fresh rollouts:

$$\mathcal{E}_{k+1} = \{(x,y,s) \mid (x,y)\in\mathcal{Y}_{k+1},\ s = \text{Eval}(x,y),\ s < \tau_{\text{err}}\}. \tag{12}$$

Define mastery ratio $\rho_k = |C_k|/|\mathcal{D}|$ and error entropy as

$$H_{\text{err}}^{(k)} = -\sum_{x\in\mathcal{D}} p_{\text{fail}}^{(k)}(x) \log p_{\text{fail}}^{(k)}(x), \tag{13}$$

where $p_{\text{fail}}^{(k)}(x)$ denotes the empirical failure probability of input x estimated from rollouts at round k.

Claim 2 (Practical Convergence with Iterative Correction). As correction coverage expands and successful cases are retained in C_k, the empirical mastery ratio ρ_k typically increases over rounds and $H_{\text{err}}^{(k)}$ decreases, reflecting fewer recurring failure modes and more concentrated error mass.

3.4.3 Reward Modeling. To guide policy optimization, we design a reward model that jointly evaluates task completion accuracy, output-format validity, and tool-invocation effectiveness. This multi-dimensional feedback mitigates common failure modes of sparse or one-dimensional rewards by encouraging policies to generate responses that are not only correct but also well-structured and appropriate for tool-use scenarios.

(1) Result Reward (R_{result}). This component evaluates whether the generated answer is semantically equivalent to the ground-truth reference. Let $\delta_{\text{success}} \in \{0,1\}$ denote the success indicator, as determined by an external evaluation model under strict equivalence criteria:

$$R_{\text{result}} = \begin{cases} 1.0, & \text{if } \delta_{\text{success}} = 1 \\ 0.0, & \text{otherwise} \end{cases} \tag{14}$$

Responses that are excessively long, vacuous, or fail to meet task-specific informational requirements are marked as incorrect to improve robustness against generic or irrelevant outputs.

(2) Format Reward (R_{format}). This component enforces syntactic compliance with output requirements, such as correct use of `<answer>` tags and non-empty, well-formed content:

$$R_{\text{format}} = \begin{cases} 0.5, & \text{if the format is valid and non-empty} \\ -0.5, & \text{otherwise} \end{cases} \tag{15}$$

Format validity is determined via rule-based parsing to ensure that responses are machine-processable.

(3) Tool-Use Reward (R_{tool}). This component quantifies tool invocation and its contribution to task resolution. Define $u_{\text{used}} \in \{0,1\}$ as the indicator of tool usage, identified by the presence of `<tool_call>` or `<tool_response>` markers in the output. The reward is defined as:

$$R_{\text{tool}} = \begin{cases} 1.0, & \text{if } \delta_{\text{success}} = 1 \text{ and } u_{\text{used}} = 1 \\ 0.4, & \text{if } \delta_{\text{success}} = 0 \text{ and } u_{\text{used}} = 1 \\ 0.2, & \text{if } \delta_{\text{success}} = 1 \text{ and } u_{\text{used}} = 0 \\ 0.0, & \text{if } \delta_{\text{success}} = 0 \text{ and } u_{\text{used}} = 0 \end{cases} \tag{16}$$

This formulation rewards tool usage when it leads to correct outcomes while still assigning partial credit to correct answers obtained without explicit tool invocation.

The overall scalar reward for RL-based optimization is computed as a weighted sum:

$$R_{\text{total}} = \alpha R_{\text{result}} + \mu R_{\text{format}} + \gamma R_{\text{tool}}, \tag{17}$$

where α, μ, and γ are hyperparameters controlling the relative emphasis of each component. In our experiments, we set $\alpha = \mu = \gamma = 1.0$, assigning equal weight to semantic correctness, syntactic integrity, and tool-use behavior.

3.4.4 Algorithmic Flow. In summary, the REvoKD framework employs a closed-loop multi-stage optimization procedure to distill robust tool-use policies for multi-modal agents. Each round consists of (i) error-driven teacher feedback, (ii) policy refinement with experience replay and group-based optimization, and (iii) consolidation of successful cases. This iterative protocol reduces semantic errors on $\mathcal{D}$ and maintains behavioral diversity via regularization. A detailed algorithm description is provided in the appendix.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate REvoKD on two benchmark categories, Vision Perception & Reasoning and Deep Research, covering five publicly available datasets. *Vision Perception & Reasoning*: SimpleVQA [6], MME-Real-Lite [54], and Visual Probe [21] assess visual perception and multi-modal reasoning, including tool-assisted inference. *Deep Research*: FVQA-Test [40] and MMSearch [17] focus on knowledge-intensive and information-seeking visual question answering scenarios.

For preprocessing, we retain only QA pairs from MMSearch that correspond to end-to-end multi-modal search tasks, ensuring evaluation captures the full pipeline of multi-modal reasoning. From SimpleVQA, we select examples from the basic categories to maintain consistent and controllable difficulty. Together, this benchmark suite spans capabilities from perception and reasoning to knowledge retrieval, supporting evaluation under diverse scenarios.

Baselines. We consider four categories of baselines: *Direct Answer*: The model generates answers directly to input questions without using external tools. We report results for the closed-source Qwen3-VL-Plus and open-source Qwen3-VL models of various scales, as well as Kimi-VL-A3B-Instruct [35] and InternVL3.5-38B [38]. *Tool Use*: The model is permitted to invoke multi-modal tools during reasoning and to integrate tool outputs into its response. This setting evaluates the impact of external information integration for both closed- and open-source models with tool-calling capabilities. *Knowledge Distillation*: We compare three related settings. (1) SFT denotes standard supervised fine-tuning on the same teacher-trajectory corpus used in Stage 1, with a conventional token-level objective and uniform token weights. (2) REvoKD (Trajectory Distill.) corresponds to Stage 1 only, which uses our weighted trajectory objective to place greater emphasis on tool-related and other high-utility tokens. (3) REvoKD further applies multi-round reinforced self-evolution after Stage 1. This comparison disentangles the effects of standard supervision, our weighted distillation objective, and the subsequent RL-based policy refinement. *Single Reinforcement learning*: This category optimizes the model solely through reinforcement learning, without incorporating any SFT or knowledge distillation stages. We select GRPO and its variant DAPO [48] as representative methods. This baseline is designed to isolate the effect of reinforcement learning as the sole training signal, providing a reference for evaluating the incremental benefits of subsequent knowledge distillation and other training paradigms.

Evaluation. Following the LLM-as-a-Judge protocol [55], we use Qwen3-Max as an automated judge model to verify output correctness. The judge compares each prediction to the ground truth semantically, mitigating string-level bias and improving robustness for multi-modal evaluation.

Implementation Details. *Trajectory Distillation Stage*: Using the ms-swift framework[4] with LoRA-based parameter-efficient fine-tuning, we train for 2 epochs with a learning rate of 1×10^{-4}. By default, we set $\lambda_{\text{tool}} = 2\lambda_{\text{other}}$. *Reinforced self-evolution Stage*: Using Verl[5], each training step samples 64 instances with 16 rollouts per instance, supporting up to 5 conversation turns per rollout. Tool calls (e.g., Image Zoom and Web Search) are handled asynchronously, with results dynamically integrated into the multi-modal context. Both semantic evaluation and the teacher agent $\mathcal{T}$ are implemented using Qwen3-VL-235B-A22B, with $\tau_{\text{succ}} = \tau_{\text{err}} = 0.8$. *Hardware*: All experiments are conducted on an NVIDIA H20 GPU cluster, with 96 GB of memory per GPU, ensuring stable multi-round multi-modal training.

4.2 Main Results

Table 1 summarizes performance on the Vision Perception & Reasoning and Deep Research benchmarks for direct-answer models, tool-augmented agents, and compact students. We draw four observations. **(1) Tool use matters.** Across model sizes, tool-augmented agents consistently outperform direct-answer baselines, indicating that many instances require explicit interaction (e.g., search or verification) and multi-step execution rather than closed-book generation. **(2) SFT vs. RL.** Supervised trajectory distillation helps students learn the *structure* of agent trajectories (when and how to call tools) but can be brittle when contexts or tool outputs shift. RL training can encourage interaction, yet it is sensitive to sparse or brittle rewards and may yield unstable tool-use behaviors (e.g., incorrect tool selection or invalid calls). **(3) REvoKD benefits small models most.** Combining MI-guided trajectory distillation with multi-round reinforced self-evolution yields more reliable tool selection and tool-result grounding than either SFT or RL alone, leading to the strongest results for 2B/4B students. In vision perception & reasoning, REvoKD-4B achieves an average score of 42.8 (+12.0 over base model, +9.3 over SFT, and +8.6 over GRPO), surpassing the tool usage performance of the larger 30B model and approaching that of the 32B model. In Deep Research, it attains an average score of 39.7 (+10.1 over base model, +2.1 over SFT, and +5.0 over GRPO), matching the performance of the 8B model and demonstrating its ability to close the gap with substantially larger counterparts. **(4) Gains concentrate in harder settings.** The largest improvements appear on challenging perception/reasoning subsets and research-style tasks, suggesting REvoKD improves the agentic process (planning, correct tool invocation, and effective use of tool feedback), not only final-answer fluency.

Efficiency–performance trade-off. We additionally report efficiency and cost statistics in Appendix C. REvoKD-4B maintains a similar memory footprint to Qwen3-VL-4B while substantially improving performance and using fewer tool calls on average. Its additional cost mainly arises during training, while deployment remains efficient for the distilled student. We further examine cross-family transferability in Appendix D, where GPT-4o-generated trajectories still provide substantial gains for a Qwen3-VL-4B student.

[4] https://github.com/modelscope/ms-swift
[5] https://github.com/verl-project/verl

Table 1: Performance comparison across benchmarks with average scores for Vision Perception and Deep Research groups.

Method	Vision Perception & Reasoning								Deep Research		
	SimpleVQA	Visual Probe			MME-Real-Lite			Avg.	FVQA-Test	MMSearch	Avg.
		Hard	Medium	Easy	Perception	Reasoning	Overall				
Direct Answer											
Qwen3-VL-Plus	53.0	6.6	16.8	27.0	48.3	48.3	48.3	30.3	35.0	27.3	31.2
Qwen3-VL-235B-A22B	53.9	9.4	16.0	24.1	47.6	48.1	47.8	30.2	34.9	27.0	31.0
InternVL3.5-38B	38.0	12.3	15.7	48.2	45.9	45.5	45.8	32.0	21.6	12.7	17.2
Qwen3-VL-32B	45.8	6.6	11.6	40.4	44.1	42.0	43.3	29.5	29.5	17.7	23.6
Qwen3-VL-30B-A3B	46.1	4.7	12.7	36.9	42.7	43.6	43.0	28.7	27.1	17.0	22.1
Kimi-VL-A3B-Instruct	26.0	16.0	6.7	28.4	26.5	18.8	23.5	20.1	17.4	7.3	12.4
Qwen3-VL-8B	40.6	12.3	14.6	41.1	38.2	28.3	34.3	28.6	23.2	14.0	18.6
Qwen3-VL-4B	38.3	8.5	13.8	40.4	37.2	35.5	35.5	27.3	21.0	13.3	17.2
Tool-use Reasoning											
Qwen3-VL-Plus	52.6	23.6	41.8	53.2	52.7	50.4	51.8	44.6	42.4	46.7	44.6
Qwen3-VL-235B-A22B	53.1	23.6	44.8	57.4	54.1	53.5	53.8	46.5	47.2	50.3	48.8
Qwen3-VL-32B	49.6	22.6	41.0	52.5	58.3	52.4	56.0	44.3	42.1	49.3	45.7
Qwen3-VL-30B-A3B	42.4	22.6	30.2	53.9	50.6	44.0	48.1	39.4	41.5	45.3	43.4
Qwen3-VL-8B	46.2	20.8	36.2	56.7	48.7	44.1	46.9	41.4	40.7	43.0	41.9
Qwen3-VL-4B	40.9	9.4	26.9	35.5	43.6	37.2	41.1	30.8	32.8	26.3	29.6
Qwen3-VL-2B	12.7	7.5	12.7	19.8	23.4	16.3	20.6	14.7	16.8	16.0	16.4
Student: Qwen3-VL-4B-Instruct											
Qwen3-VL-4B-SFT	38.8	14.2	24.6	44.7	45.4	44.7	45.1	33.5	35.4	39.7	37.6
Qwen3-VL-4B-RL (GRPO)	40.8	11.3	34.0	48.9	39.0	32.4	36.4	34.2	33.7	35.7	34.7
Qwen3-VL-4B-RL (DAPO)	41.5	15.1	34.7	49.6	49.1	39.9	45.5	37.3	34.9	39.0	37.0
REvoKD-4B	44.5	21.7	35.4	58.9	57.2	47.9	53.6	42.8	38.7	40.7	39.7
Student: Qwen3-VL-2B-Instruct											
Qwen3-VL-2B-SFT	31.6	11.3	18.3	26.2	34.7	28.7	32.4	24.0	31.4	36.0	33.7
Qwen3-VL-2B-RL (GRPO)	34.8	19.8	20.9	46.8	42.7	36.1	40.1	32.5	31.0	36.7	33.9
Qwen3-VL-2B-RL (DAPO)	28.3	11.3	26.9	48.9	44.3	34.1	40.3	31.1	25.6	30.0	27.8
REvoKD-2B	35.6	27.4	24.3	42.6	48.3	41.2	45.5	35.1	31.7	37.0	34.4

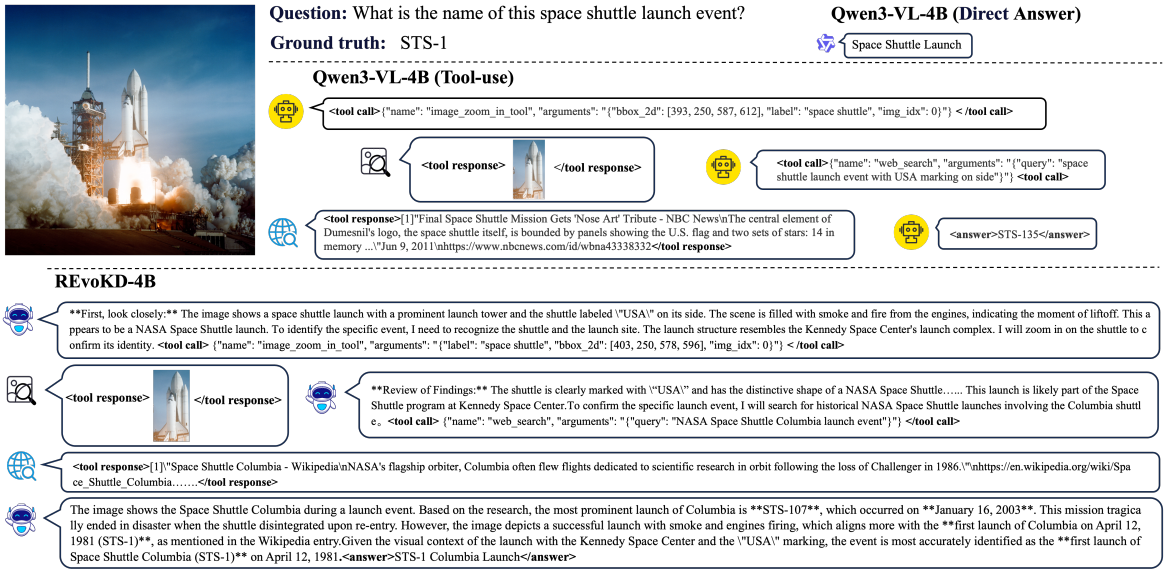
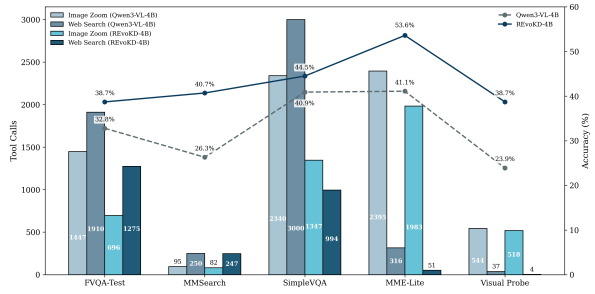


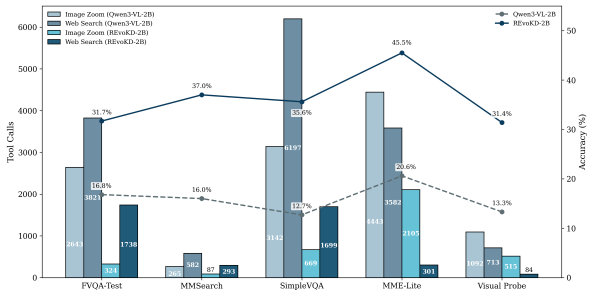
Figure 3: Case study comparison between REvoKD-4B and original Qwen3-VL-4B (in both direct answer and tool-use modes).

Table 2: Ablation results of our method using Qwen3-VL-4B-Instruct as the student backbone.

Method	Vision Perception & Reasoning								Deep Research		
	SimpleVQA	Visual Probe			MME-Real-Lite			Avg.	FVQA-Test	MMSearch	Avg.
		Hard	Medium	Easy	Perception	Reasoning	Overall				
Base Model	40.9	9.4	26.9	35.5	43.6	37.2	41.1	30.8	32.8	26.3	29.6
REvoKD (Trajectory Distill.)	40.4	17.0	26.1	48.2	48.4	41.2	45.6	35.5	37.3	39.3	38.3
REvoKD (Self-Evolution Round 1)	42.0	17.0	31.0	52.5	52.4	44.4	49.2	38.3	37.9	39.0	38.5
REvoKD (Self-Evolution Round 2)	43.7	17.9	35.1	58.2	55.6	47.1	52.3	41.4	37.9	39.7	38.8
REvoKD (Self-Evolution Round 3)	44.5	21.7	35.4	58.9	57.2	47.9	53.6	42.8	38.7	40.7	39.7



(a) REvoKD-4B before/after distillation



(b) REvoKD-2B before/after distillation

Figure 4: Comparison of REvoKD tool usage frequency and accuracy before and after distillation.

4.3 Ablation Study

Table 2 analyzes REvoKD using Qwen3-VL-4B-Instruct as the student. We observe a clear stage-wise improvement. **(1) Trajectory distillation is a strong starting point.** Compared to the base model, the trajectory distillation step yields a substantial boost on both benchmark groups, indicating that selectively transferring key agentic steps (tool calls and tool-conditioned responses) is crucial for initializing usable tool-use behavior. **(2) Reinforced self-evolution provides consistent additional gains.** Each reinforced self-evolution round further improves performance, with the largest increases appearing on harder perception/reasoning subsets (e.g., Visual Probe and MME-Lite), suggesting the student learns to better *adapt* its decisions to diverse tool outputs and multi-step dependencies rather than merely imitating demonstrations. **(3) Later rounds show diminishing but steady returns.** While Round 1 already improves noticeably over pure trajectory distillation, Rounds 2-3 continue to refine the policy and yield the best overall results, implying that repeated interaction plus teacher/evaluator feedback helps correct persistent failure modes that are difficult to eliminate in a single pass.

We further provide additional ablations in Appendix E, including a comparison between GRPO and PPO and a component analysis of teacher feedback and tool-use reward. The results show that GRPO is more effective than PPO for Stage-2 optimization, and that both teacher feedback and the tool-use reward contribute to the final performance gains of REvoKD.

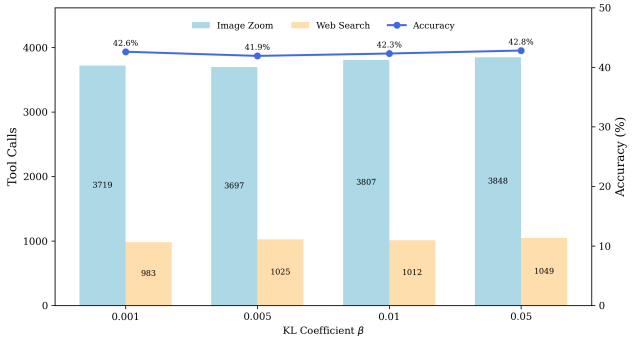
4.4 Detailed Analysis

We analyze the accuracy and tool-call patterns before and after distillation for both 4B and 2B students, as shown in Figure 4. Distillation consistently improves accuracy across all benchmarks, with larger gains on tool-intensive tasks (e.g., SimpleVQA and MME-Lite), indicating stronger multi-step evidence acquisition. In terms of behavior, REvoKD shifts tool use toward *more appropriate* calls: Image Zoom calls are more concentrated on visual reasoning benchmarks (MME-Lite/Visual Probe), while Web Search is predominantly used in research-style tasks (FVQA-Test/MMSearch). Tool-call counts decrease across all tasks, yet task effectiveness improves, demonstrating more precise invocations and reduced unnecessary operations. Overall, the result suggests that REvoKD improves performance by learning tool selection and tool-result grounding, rather than simply increasing the number of tool calls.

In addition, we investigate the effect of different KL coefficients in multi-round reinforced self-evolution training. Figure 5 shows their impact on REvoKD-4B's performance across two tasks. In Vision Perception & Reasoning, variations in β have little effect on tool usage or accuracy; the highest accuracy is achieved at $\beta = 0.05$. In Deep Research, Web Search is used markedly more than Image Zoom, and accuracy peaks at $\beta = 0.001$ and $\beta = 0.05$, suggesting that moderate to large β values may help improve performance over multiple rounds.

4.5 Case Study

Figure 3 shows a representative multi-step trajectory where the agent answers a visual question by interleaving reasoning with tool calls. Instead of answering directly, it forms an explicit plan, invokes a vision tool to extract image evidence, and then uses search to verify key facts before producing the final response. This example highlights two practical benefits encouraged by REvoKD:



(a) Effect of varying β on tool use and average accuracy in vision perception & reasoning.

(b) Effect of varying β on tool use and average accuracy in deep research.

Figure 5: Effect of different KL coefficients β on REvoKD-4B.

(i) *well-formed and purposeful tool invocations*, and (ii) *evidence-grounded answers* that integrate tool outputs instead of relying on hallucinated details. More cases are provided in the appendix.

5 Conclusion

We presented REvoKD, a self-evolutionary knowledge distillation framework that equips small MLLMs with strong tool-use capabilities. REvoKD integrates mutual information-guided trajectory distillation with multi-round, reinforcement-driven self-evolution, supported by multi-dimensional reward modeling and iterative teacher feedback. Experiments across vision perception/reasoning and deep research benchmarks show consistent improvements over single-pass distillation and RL training.

Limitations. Our current setup assumes access to a capable teacher model and a fixed set of tools, and training requires repeated environment interaction, which can be costly when tools are slow or rate-limited. In addition, our evaluation focuses on a set of representative benchmarks; generalization to more open-ended tasks with long horizons and noisy tool outputs remains challenging.

Future work. Promising directions include scaling REvoKD to open-world environments with larger and evolving tool libraries, learning finer-grained and cheaper feedback signals (e.g., verifier distillation), and improving long-horizon credit assignment and safety constraints for tool invocation in real deployments.

Acknowledgments

This work was supported by the Science and Technology Foundation of the State Grid Corporation of China (No. 5700-202440332A-2-1-ZX) and the Alibaba Research Intern Program.

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631* (2025).

[2] Wenrui Cai, Chengyu Wang, Junbing Yan, Jun Huang, and Xiangzhong Fang. 2025. Enhancing Reasoning Abilities of Small LLMs with Cognitive Alignment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. 7434–7449.

[3] Wenrui Cai, Chengyu Wang, Junbing Yan, Jun Huang, and Xiangzhong Fang. 2025. Reasoning with omnithought: A large cot dataset with verbosity and cognitive difficulty annotations. *arXiv preprint arXiv:2505.10937* (2025).

[4] Meng Cao, Haoze Zhao, Can Zhang, Xiaojun Chang, Ian Reid, and Xiaodan Liang. 2025. Ground-R1: Incentivizing Grounded Visual Reasoning via Reinforcement Learning. *arXiv preprint arXiv:2505.20272* (2025).

[5] Jiawei Chen, Xintian Shen, Lihao Zheng, Zhenwei Shao, Hongyuan Zhang, Pengfei Yu, Xudong Rao, Ning Mao, Xiaobo Liu, Lian Wen, et al. 2025. MindWatcher: Toward Smarter Multimodal Tool-Integrated Reasoning. *arXiv preprint arXiv:2512.23412* (2025).

[6] Xianfu Cheng, Wei Zhang, Shiwei Zhang, Jian Yang, Xiangyuan Guan, Xianjie Wu, Xiang Li, Ge Zhang, Jiaheng Liu, Yuying Mai, et al. 2025. Simplevqa: Multimodal factuality evaluation for multimodal large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4637–4646.

[7] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. 2023. Palm-e: An embodied multimodal language model. (2023).

[8] Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Sravana Jyothi Narayanaraju, Xinze Guan, and Xin Eric Wang. 2025. GRIT: Teaching MLLMs to Think with Images. *arXiv preprint arXiv:2505.15879* (2025).

[9] Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2025. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536* (2025).

[10] Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, et al. 2025. Webwatcher: Breaking new frontier of vision-language deep research agent. *arXiv preprint arXiv:2508.05748* (2025).

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[12] Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. 2024. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500* (2024).

[13] Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. 2025. GLM-4.1 V-Thinking: Towards Versatile Multimodal Reasoning with Scalable Reinforcement Learning. *arXiv preprint arXiv:2507.01006* (2025).

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.

[15] Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and Ariel Fuxman. 2024. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9590–9601.

[16] Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, et al. 2025. Verltool: Towards holistic agentic reinforcement learning with tool use. *arXiv preprint arXiv:2509.01055* (2025).

[17] Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Chaoyou Fu, Guanglu Song, et al. 2024. Mmsearch: Benchmarking the potential of large models as multi-modal search engines. *arXiv preprint arXiv:2409.12959* (2024).

[18] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. 2022. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094* 2, 3 (2022), 6.

[19] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516* (2025).

[20] Minki Kang, Jongwon Jeong, Seanie Lee, Jaewoong Cho, and Sung Ju Hwang. 2025. Distilling llm agent into small models with retrieval and code tools. *arXiv preprint arXiv:2505.17612* (2025).

[21] Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. 2025. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. *arXiv preprint arXiv:2509.07969* (2025).
[22] Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025. Torl: Scaling tool-integrated rl. *arXiv preprint arXiv:2503.23383* (2025).
[23] Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. 2025. Small models struggle to learn from strong reasoners. *arXiv preprint arXiv:2502.12143* (2025).
[24] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. 2024. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 26689–26699.
[25] Jiacai Liu, Chaojie Wang, Chris Yuhao Liu, Liang Zeng, Rui Yan, Yiwen Sun, Yang Liu, and Yahui Zhou. 2024. Improving multi-step reasoning abilities of large language models with direct advantage policy optimization. *arXiv preprint arXiv:2412.18279* (2024).
[26] Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, et al. 2024. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. In *Forty-first International Conference on Machine Learning*.
[27] Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Guanjing Xiong, and Hongsheng Li. 2025. UI-R1: Enhancing Efficient Action Prediction of GUI Agents by Reinforcement Learning. *arXiv preprint arXiv:2503.21620* (2025).
[28] Zhenyu Ning, Jieru Zhao, Qihao Jin, Wenchao Ding, and Minyi Guo. 2024. Inf-MLLM: Efficient streaming inference of multimodal large language models on a single GPU. *arXiv preprint arXiv:2409.09086* (2024).
[29] Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958* (2025).
[30] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems* 36 (2023), 68539–68551.
[31] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
[32] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* (2024).
[33] Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, et al. 2025. Openthinkimg: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617* (2025).
[34] Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, et al. 2024. Aligning large multimodal models with factually augmented rlhf. In *Findings of the Association for Computational Linguistics: ACL 2024*. 13088–13110.
[35] Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. 2025. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491* (2025).
[36] Yijun Tian, Yikun Han, Xiusi Chen, Wei Wang, and Nitesh V Chawla. 2025. Beyond answers: Transferring reasoning capabilities to smaller llms using multi-teacher knowledge distillation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*. 251–260.
[37] Qiuchen Wang, Ruixue Ding, Yu Zeng, Zehui Chen, Lin Chen, Shihang Wang, Pengjun Xie, Fei Huang, and Feng Zhao. 2025. VRAG-RL: Empower Vision-Perception-Based RAG for Visually Rich Information Understanding via Iterative Reasoning with Reinforcement Learning. *arXiv preprint arXiv:2505.22019* (2025).
[38] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. 2025. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265* (2025).
[39] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671* (2023).
[40] Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. 2025. MMSearch-R1: Incentivizing LMMs to Search. *arXiv preprint arXiv:2506.20670* (2025).
[41] Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S Yu. 2023. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 2247–2256.
[42] Yi Xu, Chengzu Li, Han Zhou, Xingchen Wan, Caiqi Zhang, Anna Korhonen, and Ivan Vulić. 2025. Visual Planning: Let's Think Only with Images. *arXiv preprint arXiv:2505.11409* (2025).
[43] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388* (2025).
[44] Menglin Yang, Jialin Chen, Jinkai Tao, Yifei Zhang, Jiahong Liu, Jiasheng Zhang, Qiyao Ma, Harshit Verma, Regina Zhang, Min Zhou, et al. 2024. Low-rank adaptation for foundation models: A comprehensive review. *arXiv preprint arXiv:2501.00365* (2024).
[45] Huanjin Yao, Ruifei Zhang, Jiaxing Huang, Jingyi Zhang, Yibo Wang, Bo Fang, Ruolin Zhu, Yongcheng Jing, Shunyu Liu, Guanbin Li, et al. 2025. A survey on agentic multimodal large language models. *arXiv preprint arXiv:2510.10991* (2025).
[46] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.
[47] Deng Yixuan and Ji Xiaoqiang. 2025. Multi-Reward GRPO Fine-Tuning for De-biasing Large Language Models: A Study Based on Chinese-Context Discrimination Data. *arXiv preprint arXiv:2511.06023* (2025).
[48] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476* (2025).
[49] Xinbin Yuan, Jian Zhang, Kaixin Li, Zhuoxuan Cai, Lujian Yao, Jie Chen, Enguang Wang, Qibin Hou, Jinwei Chen, Peng-Tao Jiang, et al. 2025. Enhancing Visual Grounding for GUI Agents via Self-Evolutionary Reinforcement Learning. *arXiv preprint arXiv:2505.12370* (2025).
[50] Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. 2025. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547* (2025).
[51] Hanchen Zhang, Xiao Liu, Bowen Lv, Xueqiao Sun, Bohao Jing, Iat Long Iong, Zhenyu Hou, Zehan Qi, Hanyu Lai, Yifan Xu, et al. 2025. AgentRL: Scaling Agentic Reinforcement Learning with a Multi-Turn, Multi-Task Framework. *arXiv preprint arXiv:2510.04206* (2025).
[52] Yifan Zhang, Liang Hu, Haofeng Sun, Peiyu Wang, Yichen Wei, Shukang Yin, Jiangbo Pei, Wei Shen, Peng Xia, Yi Peng, et al. 2025. Skywork-R1V4: Toward Agentic Multimodal Intelligence through Interleaved Thinking with Images and DeepResearch. *arXiv preprint arXiv:2512.02395* (2025).
[53] Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, et al. 2025. Thyme: Think beyond images. *arXiv preprint arXiv:2508.11630* (2025).
[54] Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. 2024. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint arXiv:2408.13257* (2024).
[55] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems* 36 (2023), 46595–46623.
[56] Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, et al. 2023. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964* (2023).
[57] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. 2025. DeepEyes: Incentivizing" Thinking with Images" via Reinforcement Learning. *arXiv preprint arXiv:2505.14362* (2025).
[58] Guanghao Zhou, Panjia Qiu, Cen Chen, Jie Wang, Zheming Yang, Jian Xu, and Minghui Qiu. 2025. Reinforced mllm: A survey on rl-based reasoning in multimodal large language models. *arXiv preprint arXiv:2504.21277* (2025).
[59] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592* (2023).
[60] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479* (2025).

A Algorithm Flow

Algorithm 1 summarizes the multi-round self-evolutionary distillation stage of REvoKD. In each round, we sample multiple candidate trajectories per input to construct GRPO groups.

B Evaluation Benchmarks

We evaluate on two benchmark groups. *Vision Perception & Reasoning* includes SimpleVQA (2025 QA pairs), MME–Lite (1919 QA pairs), and Visual Probe (515 QA pairs), covering factual visual QA, simplified real-world multi-modal reasoning, and core visual

Algorithm 1 Multi-Round Self-Evolutionary Distillation Stage

Input: multi-modal task set $\mathcal{D}$, initial policy π_0, teacher agent $\mathcal{T}$, success threshold τ_{succ}, error threshold τ_{err}, max rounds K, candidates per input N
Initialize: mastery buffer $C_0 \leftarrow \emptyset$
for $k = 0$ to $K-1$ **do**
  Initialize rollout set $\mathcal{Y}_k \leftarrow \emptyset$, correction set $\mathcal{G}_k \leftarrow \emptyset$
  for each $x \in \mathcal{D}$ **do**
    for $j = 1$ to N **do**
      Sample trajectory $y^{(j)} \sim \pi_k(\cdot \mid x, h_x)$
      Add $(x, y^{(j)})$ to $\mathcal{Y}_k$
      $s^{(j)} \leftarrow \text{Eval}(x, y^{(j)})$
      if $s^{(j)} < \tau_{\text{err}}$ **then**
        $(\hat{x}^{(j)}, (y^*)^{(j)}, d^{(j)}) \leftarrow \mathcal{T}(x, y^{(j)}, s^{(j)})$
        Add $(\hat{x}^{(j)}, (y^*)^{(j)}, d^{(j)})$ to $\mathcal{G}_k$
      end if
    end for
  end for
  Compute $\text{Err}_k \leftarrow \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{Y}_k,\ s=\text{Eval}(x,y),\ s<\tau_{\text{err}}} (1-s)$
  Experience replay buffer $\mathcal{M}_k \leftarrow \mathcal{G}_k \cup C_k$
  Update policy π_{k+1} on $\mathcal{M}_k$ using GRPO with KL regularization, grouping candidates by input x as in [32]
  Initialize $C_{k+1} \leftarrow C_k$
  for each $(\hat{x}, y^*, d) \in \mathcal{G}_k$ **do**
    if $\text{Eval}(\hat{x}, y^*) \geq \tau_{\text{succ}}$ **then**
      Add $(\hat{x}, y^*)$ to C_{k+1}
    end if
  end for
  if $\text{Err}_k < \varepsilon$ **then**
    break
  end if
end for
Return: distilled policy π_k, mastery buffer C_k

perception across multiple difficulty levels. *Deep Research* includes FVQA–Test (1800 QA pairs) and MMSearch (300 QA pairs), focusing on knowledge-intensive visual question answering and multi-modal search scenarios that require external information retrieval.

C Efficiency and Cost Analysis

Our efficiency claim mainly concerns deployment efficiency of the distilled 2B/4B students, rather than low training cost. REvoKD uses a strong teacher, repeated environment interaction, and tool calls during training to obtain a more compact student for deployment. We therefore report both deployment efficiency and training/inference cost statistics.

Table 3 shows that REvoKD-4B keeps nearly the same memory footprint as Qwen3-VL-4B while improving both benchmark groups and reducing tool calls. Compared with Qwen3-VL-8B, it offers a better efficiency–performance trade-off. REvoKD-2B further reduces memory and latency.

Table 4 shows that REvoKD incurs extra cost mainly during training from tool interaction and teacher feedback, while deployment remains efficient for the distilled student. All statistics are measured under the same hardware and decoding setup.

Table 3: Efficiency–performance comparison. "VP&R Avg" and "DR Avg" denote the average scores on the Vision Perception & Reasoning and Deep Research benchmark groups. Latency includes tool invocation time.

Model	Mem. (GB)	Latency (s)	Tool Calls	VP&R Avg	DR Avg
Qwen3-VL-8B	32.57	28.38	2.80	41.4	41.9
Qwen3-VL-4B	18.75	7.92	1.88	30.8	29.6
REvoKD-4B	18.78	8.66	1.10	42.8	39.7
REvoKD-2B	11.60	6.72	1.19	35.1	34.4

Table 4: Training and inference cost statistics. "Tool / Sample" and "Teacher / Sample" denote the average number of tool calls and teacher calls per sample, respectively. Teacher calls are only used during REvoKD training.

Method	Stage	Total Tools	Tool / Sample	Teacher / Sample
Base	Inference	12334	1.88	–
SFT	Inference	22759	3.47	–
RL	Inference	7149	1.09	–
REvoKD-4B	Inference	7197	1.10	–
REvoKD-4B	Training	–	1.49	0.38

D Cross-Family Distillation

To assess cross-family transferability, we replace the original Qwen3-VL-Plus teacher with GPT-4o to generate agentic trajectories for a representative training subset, and then perform Stage-1 distillation on Qwen3-VL-4B under the same unified text-tool format. Table 5 shows that GPT-4o-generated trajectories still provide clear gains over the zero-shot baseline, especially on MMSearch. This suggests that REvoKD can transfer useful tool-use knowledge across model families and remains robust to heterogeneous teachers.

Table 5: Cross-family distillation from GPT-4o to Qwen3-VL-4B.

Student	Teacher	Setting	MME-Real-Lite	MMSearch
Qwen3-VL-4B	–	Zero-shot	41.1	26.3
Qwen3-VL-4B	Qwen3-VL-Plus	Stage-1 (SFT)	45.6	39.3
Qwen3-VL-4B	GPT-4o	Stage-1 (SFT)	43.7	36.7

E Additional Ablation Results

GRPO vs. PPO. We use GRPO because Stage 2 samples multiple trajectories per input, making group-wise comparison more suitable than relying only on absolute rewards. We compare GRPO and PPO under the same Stage-2 setup on the same trajectory-distilled 4B model, using MME-Real-Lite and MMSearch.

Table 6 shows that GRPO outperforms PPO and improves over trajectory distillation alone, supporting its use in Stage 2.

Component ablation. We further conduct a one-round ablation on the same 4B model and benchmarks to isolate the contributions of key Stage-2 components. As shown in Table 7, removing either teacher feedback or the tool-use reward degrades performance. Note that "w/o Teacher Feedback" corresponds to the first Stage-2 round, where teacher feedback is not yet available.

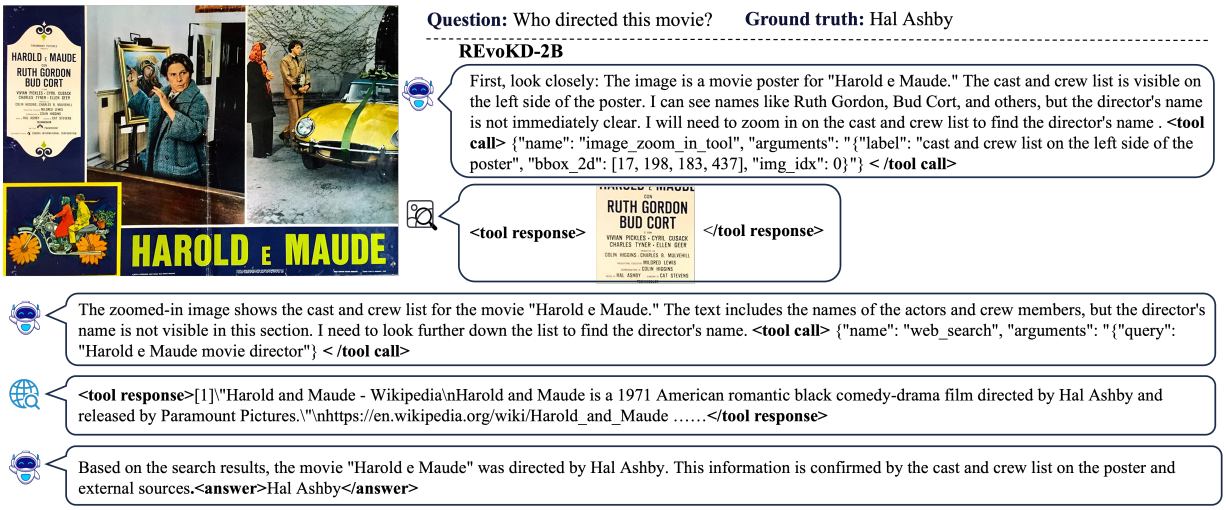


Figure 6: Case study of REvoKD-2B.

Tool-Use Integrated Reasoning Prompt Template

SYSTEM MESSAGE
Your role is that of a research assistant specializing in visual information. Answer questions about images by looking at them closely and then using research tools. Please follow this structured thinking process and show your work.
Start an iterative loop for each question:
- **First, look closely:** Begin with a detailed description of the image, paying attention to the user's question. List what you can tell just by looking, and what you'll need to look up.
- **Next, find information:** Use a tool to research the things you need to find out.
- **Then, review the findings:** Carefully analyze what the tool tells you and decide on your next action.
Continue this loop until your research is complete.
To finish, bring everything together in a clear, synthesized answer that fully responds to the user's question.

USER PROMPT
Question: {question}
Multiple tool invocations are permitted. Prior to each tool invocation, you must articulate your analytical reasoning.
Critical reminder: Tools operate in a stateless manner and will not preserve any variables, functions, or definitions from prior executions.
Reasoning step by step, write ONLY the final answer enclosed in <answer> and </answer> tags. Do not include any extra text outside the tags after the final answer.

Figure 7: Prompt template for tool-integrated reasoning.

Tool : Image Zoom Tool

Description: Zoom in on a specific region of an image by cropping it based on a bounding box (bbox) and an optional object label.
Input: Image, bounding box and object label.
Output: Cropped and zoomed-in image.
Arguments:
bbox_2d: [x1, y1, x2, y2] — Bounding box in relative coordinates (0~1000), where (x1, y1) is the top-left corner and (x2, y2) is the bottom-right corner.
label: *string* — Optional label for the object inside the bounding box.
img_idx: *integer* — Index of the image to zoom in (starting from 0).

(a) Image zoom tool

Tool : Web Search

Description: Retrieve information from the internet using a fully-formed semantic search query.
Input: A search query string.
Output: Search results from the retrieval service.
Arguments:
query: *string* — Fully-formed semantic search query describing the needed information.

(b) Web search tool

Figure 8: Tool descriptions.

Table 6: Comparison of PPO and GRPO as Stage-2 optimizers on top of trajectory distillation for the 4B model.

Variant	MME-Real-Lite	MMSearch
REvoKD (Trajectory Distill.)	45.6	39.3
w/ PPO	44.7	38.3
w/ GRPO	49.2	39.0

Table 7: Component ablation of REvoKD-4B on MME-Real-Lite and MMSearch.

Variant	MME-Real-Lite	MMSearch
Full REvoKD	53.6	40.7
w/o Teacher Feedback	49.2	39.0
w/o Tool-use Reward	51.5	35.3

F Prompt Templates and Tool Descriptions

We provide the tool-integrated reasoning prompt templates (Fig. 7) and tool descriptions (Fig. 8) used in this paper below.

G Additional Case Study

Figure 6 shows an example of REvoKD-2B answering a question that requires grounding on small text and coordinating multiple tools. Given a movie-poster image and the question "Who directed this movie?", the agent first uses image zoom to inspect the credits region. When the zoomed evidence is insufficient to identify the director, it switches to web search to recover the missing information, and then combines the tool outputs to produce the correct answer. This example highlights two capabilities encouraged by our training pipeline: *grounded tool invocation* and *robust recovery* when the initial evidence is insufficient.