# SEE LESS, SHOP BETTER: EFFICIENT VISUAL EVIDENCE ACQUISITION FOR E-COMMERCE SEARCH AGENTS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

E-commerce search agents are increasingly used to find and recommend products based on complex user requirements. However, many decision-critical product attributes are available only in image-heavy detail pages. Directly exposing all visual content from a large set of products to an agent is both computationally expensive and difficult to scale. To address this challenge, we study how e-commerce search agents can effectively and efficiently utilize visual product information. We first introduce two complementary evaluation tasks: *Fixed-Item Question Answering*, which isolates and directly evaluates an agent's ability to comprehend multimodal product details, and *Open-Ended Product Recommendation*, which holistically assesses its end-to-end performance across the full search-and-recommendation pipeline. We then decouple product visual understanding from the main agent through a visual evidence tool, allowing the agent to selectively acquire visual information on demand. Building on this framework, we develop a post-training pipeline combining a staged Supervised Fine-Tuning (SFT) curriculum with Reinforcement Learning (RL) under constrained tool-turn budgets. Experiments show that our staged SFT substantially boosts both multimodal product-detail understanding and open-ended recommendation, while budget-constrained RL further improves the efficiency–effectiveness trade-off—enabling our 9B agent to achieve a 46.8% recommendation quality rate under a strict 4-turn budget, surpassing GPT-5.6 Sol's 45.6% with unconstrained tool use.

## 1 INTRODUCTION

Large Language Model (LLM)-based shopping agents enable users to discover products through natural-language interaction (Yao et al., 2022; Wang et al., 2026). However, identifying products that genuinely satisfy user needs requires more than matching catalog text, as product information is distributed across textual metadata and diverse visual content (Ling et al., 2025). Many decision-critical details—such as internal structures, dimension diagrams, and interface layouts—are available only in product-detail images. Effective e-commerce search therefore requires agents to actively acquire and interpret product-side visual information rather than rely exclusively on textual attributes.

As illustrated in Figure 1, this requirement creates a severe scalability challenge. A single query may retrieve many candidate products, each accompanied by numerous detail images; indiscriminately processing all of them inflates the visual context and increases computation and latency (Ling et al., 2025; Chen et al., 2024). System effectiveness and user effort are also established factors influencing satisfaction in information retrieval (Al-Maskari & Sanderson, 2010). Yet exhaustive visual inspection is unnecessary: textual metadata can already confirm some candidates and rule out others, leaving only promising products for visual verification. The key challenge is therefore not simply whether an agent can understand images, but deciding *where* to acquire visual evidence and *how* to utilize limited interaction opportunities effectively.

This challenge also complicates evaluation. Existing benchmarks study multimodal product understanding (Ling et al., 2025), complex shopping intents (Wang et al., 2026), or multimodal shopping interactions (Hao et al., 2026), but end-to-end recommendation performance alone cannot reveal whether a failure originates from candidate retrieval or fine-grained visual comprehension (Zhao et al., 2026). We therefore introduce two complementary, modality-aware tasks grounded in real product catalogs. *Fixed-Item Question Answering* directly evaluates multimodal understanding over



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

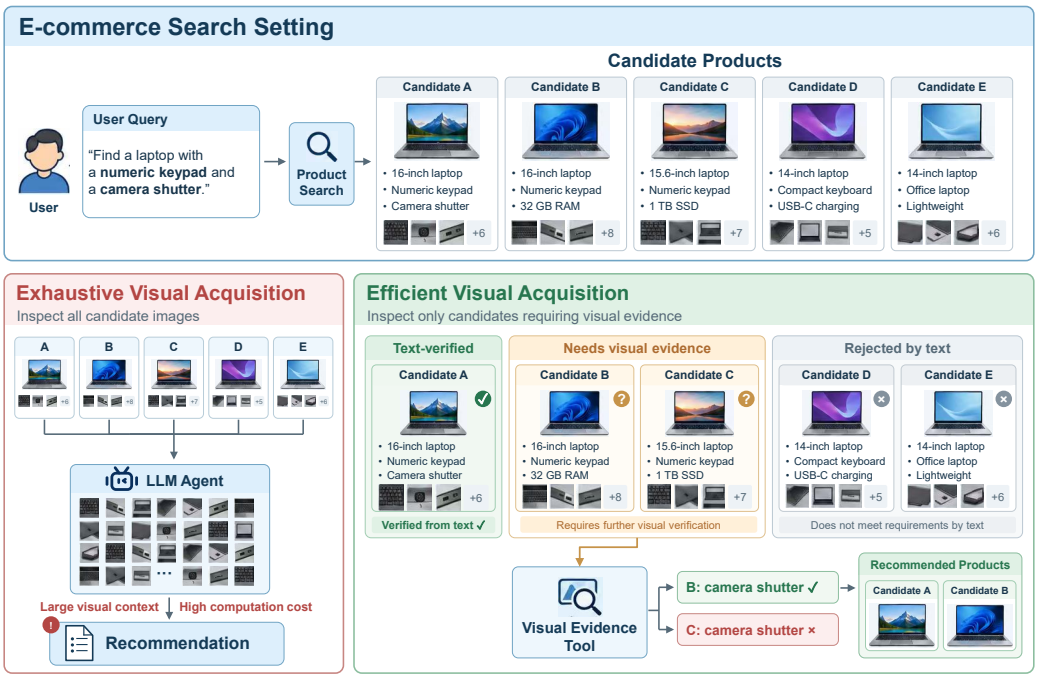


Figure 1: Overview of visual evidence acquisition for e-commerce search. Rather than exhaustively inspecting all candidate images, the agent selectively invokes the visual evidence tool for candidates whose query-relevant attributes cannot be verified from text alone.

designated products, while *Open-Ended Product Recommendation* evaluates the complete search-and-recommendation process. Together, they provide both a controlled view of product-level multimodal capability and a holistic evaluation of agentic search.

Building on these tasks, we develop a framework for training efficient multimodal e-commerce search agents (Figure 2). A decoupled *Visual Evidence Tool* allows the main agent to selectively access image-heavy product details and receive compact textual evidence, avoiding direct exposure to the full visual context. We further construct evidence-grounded trajectories and organize SFT as a staged curriculum that progresses from product-level visual grounding to end-to-end search. Finally, RL under constrained tool-turn budgets trains the agent to make more effective use of limited interactions, improving recommendation quality through more efficient visual evidence acquisition.

Our contributions are summarized as follows:

- We introduce two complementary, modality-aware evaluation tasks that disentangle product-level multimodal understanding from end-to-end e-commerce search and recommendation, enabling targeted diagnosis of visual information needs while retaining realistic system-level evaluation.
- We develop a decoupled multimodal search framework and a staged post-training pipeline that combines evidence-grounded SFT with RL under constrained tool-turn budgets, enabling deployment-scale agents to coordinate search and visual verification efficiently.
- Extensive experiments demonstrate substantial improvements in both recommendation quality and interaction efficiency. Our post-training pipeline substantially improves the base model's unconstrained recommendation score; under a strict four-turn budget, the RL-trained agent achieves **46.8%**, surpassing a strong proprietary model with unconstrained tool use (**45.6%**) while exhibiting a stronger quality–efficiency trade-off.

## 2 Related Work

### 2.1 E-commerce Agents

E-commerce agents have evolved from text-centric interaction toward richer multimodal workflows. Early systems such as WebShop (Yao et al., 2022) formulate shopping as text-centric interaction over product catalogs. In parallel, VisualWebArena (Koh et al., 2024) show that realistic web interaction often requires visual grounding. More recently, MindFlow (Gong et al., 2025) brings multimodal

capability into e-commerce customer service through an MLLM-as-Tool architecture that decouples high-level planning from visual perception.

### 2.2 E-COMMERCE BENCHMARKS

Existing benchmarks cover complementary dimensions of e-commerce intelligence. EcomMMMU (Ling et al., 2025) and OxyEcomBench (Liu et al., 2026b) focus on multimodal product understanding, while ECom-Bench (Wang et al., 2025) evaluates multimodal agents in customer-service scenarios. ShoppingBench (Wang et al., 2026) evaluates grounded shopping intents in an interactive product environment, whereas DeepShop (Lyu et al., 2025) and Shopping Reasoning Bench (Fan et al., 2026) emphasize complex product search and higher-level shopping reasoning, respectively. EComStage (Zhao et al., 2026) diagnoses perception, planning, and action stages separately. More recently, MMShopBench (Hao et al., 2026) evaluates multimodal shopping agents that infer user needs from uploaded images and multi-turn dialogues to retrieve and verify products.

Despite these advances, existing benchmarks rarely treat acquiring product-side visual evidence as an active decision during search. Our multimodal evaluation instead disentangles text-answerable requirements from visually-dependent ones, jointly evaluating isolated product understanding and open-ended product search and recommendation.

### 2.3 EFFICIENT WEB SEARCH AND TOOL-USE AGENTS

A parallel line of research investigates how web search agents learn to invoke tools and search efficiently. Search-R1 (Jin et al., 2025) and DeepResearcher (Zheng et al., 2025) train LLMs for multi-step search with RL. To mitigate the high latency and optimize search efficiency, SlimSearcher (Xie et al., 2026) compresses redundant reasoning and tool calls via SFT and RL, ParallelSearch (Zhao et al., 2025) improves concurrency across independent search calls, and BATS (Liu et al., 2026a) studies test-time scaling under explicit budget awareness. Concurrently, in multimodal reasoning, DeepEyes (Zheng et al., 2026), Pixel Reasoner (Su et al., 2025), and AdaTurn (Liang et al., 2026) train agents for active visual perception under interaction constraints.

Our work targets a more distinct challenge in e-commerce search: the agent must operate over both a large-scale product candidate space and the dense internal visual space of each item. Given practical deployment overhead, limited interaction opportunities must be allocated across candidate discovery, cross-product verification, and selective visual evidence acquisition. Moreover, the final objective is to identify a set of qualified products rather than a single ground-truth answer typical of web search.

## 3 BENCHMARKING MULTIMODAL E-COMMERCE SEARCH AGENTS

To comprehensively evaluate multimodal e-commerce search agents at both the product and system levels, we introduce two complementary tasks: *Fixed-Item Question Answering* evaluates product-level multimodal understanding, while *Open-Ended Product Recommendation* examines the complete search-and-recommendation process.

### 3.1 FIXED-ITEM QUESTION ANSWERING: ISOLATING MULTIMODAL COMPREHENSION

Evaluating LLM agents in an open-ended product search pipeline presents a significant credit-assignment challenge: when an agent makes an incorrect recommendation, it is difficult to determine whether the failure originates from flawed query formulation, inaccurate candidate retrieval, or an inability to comprehend multimodal product details (Zhao et al., 2026). To disentangle these factors, we introduce Fixed-Item Question Answering (QA) as a controlled evaluation setting. Given a specified product or a pair of products, the expected answer is determined solely by the facts associated with these designated items, removing product discovery as a confounding factor.

### 3.2 OPEN-ENDED PRODUCT RECOMMENDATION: END-TO-END SYSTEM EVALUATION

While Fixed-Item QA isolates product understanding, it bypasses the end-to-end complexity of practical e-commerce search. A real-world e-commerce agent must autonomously navigate the complete pipeline: retrieving candidate products, acquiring necessary multimodal evidence, verifying candidate alignment, and making final recommendations. Moreover, unlike fixed-item QA, product recommendation generally admits multiple valid answers, making evaluation against a single predefined target product insufficient. We therefore introduce Open-Ended Product Recommendation,

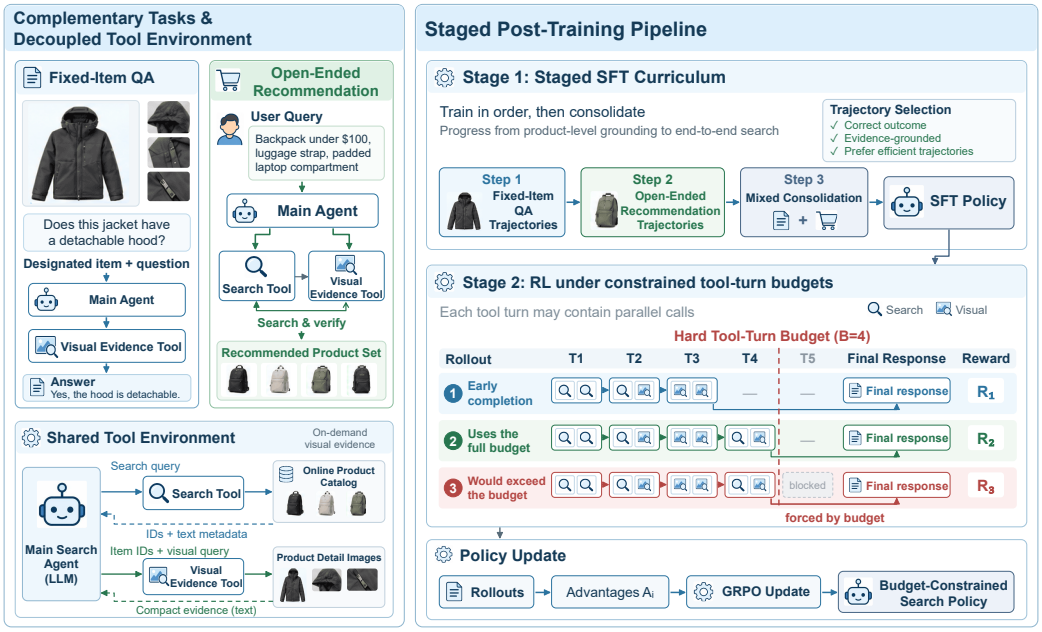


Figure 2: Overview of our evaluation tasks, decoupled tool environment, and staged post-training pipeline. The agent learns product-level visual grounding and end-to-end recommendation through a staged SFT curriculum, followed by RL under constrained tool-turn budgets.

where the agent is instead required to explore the open catalog and recommend products that satisfy either explicit user constraints or high-level goals.

### 3.3 Dataset Construction

We design the construction pipeline around three principles:

1. **Grounded:** Queries are derived from verified facts on real product pages, ensuring realistic and answerable tasks.
2. **Evaluable:** Each query has explicit, bounded criteria—minimum facts for Fixed-Item QA and hard constraints for Open-Ended Recommendation—for objective assessment.
3. **Modality-Aware:** Image-derived facts are aligned with textual metadata to distinguish text-answerable requirements from those that genuinely require visual evidence.

**For Fixed-Item QA**, a *Product Anchor* built from the main image and basic metadata prevents attributes of accessories, bundled items, or marketing elements from being assigned to the target product. Facts extracted across images are consolidated and used to create queries for direct attributes, explicit constraints, and bounded usage scenarios. Each query is paired with explicit *minimum facts*: the indispensable factual units that a response must cover without contradiction, enabling objective evaluation without requiring a fixed surface-form answer.

**For Open-Ended Recommendation**, queries are derived from visual facts of a reference product, which serves only to ensure that at least one feasible solution exists. These facts are converted into product-independent *hard constraints*, allowing any retrieved item that satisfies the same requirements to qualify. We further validate each query against candidates from the real search environment and discard cases whose constraints can be verified from candidate text alone, thereby preserving genuine visual-evidence requirements. Appendix A provides the complete construction pipeline, annotation procedure, and quality-control criteria.

## 4 Training Multimodal E-commerce Search Agents

We next describe how we train multimodal e-commerce search agents to acquire visual evidence efficiently during product search. As illustrated in Figure 2, our framework combines a decoupled search-and-visual tool environment with staged SFT and RL under constrained tool-turn budgets.

### 4.1 Agent Architecture with On-Demand Visual Evidence

**Agent Environment and Action Space.** Operating within a real-world e-commerce environment, our agent tackles a user query $q$ through a sequential reasoning-and-acting loop, following the general interleaved reasoning and tool-interaction paradigm (Yao et al., 2023). At each step $t$, the agent updates its internal state $s_t = (q, a_{<t}, o_{\leq t})$, where $a$ represents past actions and $o$ represents tool observations. Based on $s_t$, the core LLM acts as the central planner, generating either a final response or a set of tool actions $A_t \subseteq \mathcal{A}$. The available action space $\mathcal{A}$ comprises three specialized tools: (1) **Product Search** for retrieving candidate items and their textual metadata, (2) a **Visual Evidence Tool** for on-demand image inspection, and (3) a **Calculator** for deterministic numerical operations. Notably, the agent is designed to emit multiple independent tool calls in parallel within a single turn (i.e., $|A_t| \geq 1$), allowing for simultaneous broad exploration and verification.

**Coarse-to-Fine Visual Acquisition.** One significant challenge in multimodal e-commerce search is the visual context explosion: a single search often yields dozens of candidate products, each containing numerous high-resolution images. Directly inputting this catalog into the agent's context window is computationally prohibitive because visual tokens impose substantial inference cost (Chen et al., 2024). To address this, our Visual Evidence Tool abstracts visual processing into an on-demand, two-stage pipeline:

1. **Coarse-Grained Indexing:** Before the agent commits to reading an image, candidate images are rapidly scanned at a low resolution to generate lightweight textual indices that serve directly as routing signals. For images with simple layouts, this step directly extracts all visible text; for complex images (e.g., those containing dense specification tables or multiple structural diagrams), it summarizes the main visual topics without exhaustive transcription.
2. **Fine-Grained Evidence Reading:** Guided by the agent's specific sub-query and the coarse indices, a Selector module isolates a small subset of the most relevant images. A high-resolution Reader then examines this targeted subset, extracting precise, visible facts and returning only the required evidence in a concise textual format.

This decoupled architecture ensures that the agent acquires multimodal evidence strictly when needed, balancing comprehensive information acquisition with the finite context and reasoning capacity of the LLM planner.

### 4.2 Supervised Fine-Tuning

Deploying a highly capable, large-scale model for production e-commerce search can be computationally expensive, particularly for long-context serving (Kwon et al., 2023). To enable efficient production deployment, we use supervised fine-tuning (SFT), a standard approach for adapting pre-trained language models to desired behaviors (Ouyang et al., 2022), to train a deployment-scale base model.

**Evidence-Grounded Trajectory Selection.** A critical risk when curating training trajectories is the *parametric knowledge shortcut*: an LLM may correctly answer a product-specific query using memorized knowledge rather than actively acquiring the required information through tools, reflecting the distinction between parametric memory and retrieved evidence (Lewis et al., 2020). Training on such data may encourage the model to memorize task-specific answers instead of learning generalizable, evidence-grounded tool behaviors. To mitigate this issue, we filter trajectories based strictly on evidence reliability. A trajectory is deemed valid only if the final answer is correct and the minimum facts or hard constraints are explicitly supported by the observations retrieved via tool calls. Correct answers produced without sufficient acquired evidence are discarded. Among the valid trajectories, we further prioritize those requiring the fewest tool invocation turns and actions, introducing an early bias toward efficient information acquisition.

**Task Curriculum.** Following the general principle of curriculum learning (Bengio et al., 2009), rather than mixing all demonstrations from the beginning, we organize SFT as a progression from controlled product understanding to full search and recommendation. The agent first learns from Fixed-Item QA trajectories, where the target product is known and the main challenge is to acquire the necessary product information and answer from visual evidence. It then learns from Open-Ended Product Recommendation trajectories, which introduce product retrieval, candidate verification, and multi-product decision making. Finally, both trajectory types are mixed to consolidate these localized perception and global planning capabilities into a unified search & verify policy.

### 4.3 Reinforcement Learning for Effective and Efficient Search

While SFT equips the agent with effective evidence-grounded tool behaviors, it does not explicitly optimize resource allocation during open-ended search. We therefore further train the agent using Group Relative Policy Optimization (GRPO) (Shao et al., 2024) under constrained tool-use budgets, aiming to explore the Pareto optimal frontier between recommendation quality and execution efficiency.

**Budget-Constrained Interaction and Outcome Reward.** We introduce a hard tool-use budget $B$, defining the maximum number of tool turns the agent is permitted to execute before formulating a final response. To maximize information gain within this budget, the agent is encouraged to emit multiple independent tool calls in parallel during a single turn. Once the budget $B$ is exhausted, the environment forces the agent to terminate exploration and yield a final recommendation based on the acquired context.

The objective is to maximize the quality of the final recommendation. We define the reward based on the number of unique recommended products that fully satisfy the user's hard constraints; identical or duplicate items are merged during evaluation to prevent reward hacking. Let $N_{\text{correct}}$ denote the number of uniquely qualified products in the agent's final answer. The trajectory reward is:

$$R(\tau) = \frac{\min(N_{\text{correct}}, K)}{K} \tag{1}$$

where $K$ is the target number of recommended items (e.g., $K = 4$). Rather than adding a per-turn penalty to the reward, which can induce a brevity bias and introduces an additional hyperparameter (Xie et al., 2026), we impose efficiency through the hard budget. This formulation incentivizes the agent to learn optimal resource allocation for any given budget, ensuring robust performance even when execution must be abruptly truncated in production.

**Group-Relative Policy Optimization.** For each query, we sample a group of $M$ trajectories and normalize their rewards to obtain the relative advantage

$$A_i = \frac{R_i - \mu_R}{\sigma_R + \epsilon}. \tag{2}$$

where $\mu_R$ and $\sigma_R$ denote the mean and standard deviation of rewards within the group. We then optimize the policy using the clipped GRPO objective, which follows the clipped probability-ratio construction of PPO (Schulman et al., 2017; Shao et al., 2024):

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\mathbb{E}_{i,t}\left[\min\left(r_{i,t}(\theta)A_i, \text{clip}\left(r_{i,t}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\right)A_i\right)\right]. \tag{3}$$

where $r_{i,t}(\theta) = \frac{\pi_\theta(a_{i,t}|s_{i,t})}{\pi_{\theta_{\text{old}}}(a_{i,t}|s_{i,t})}$ is the probability ratio.

During preliminary training, we observed that trajectories could suffer from premature entropy collapse, occasionally leading to degenerate repetitive tool-call generation. We therefore include a small entropy bonus, a standard policy-optimization regularizer for encouraging exploration (Mnih et al., 2016), to maintain policy diversity without introducing additional task-specific reward heuristics:

$$\mathcal{L}_{\text{entropy}}(\theta) = -\lambda_H \mathbb{E}_{i,t}\left[H\left(\pi_\theta(\cdot \mid s_{i,t})\right)\right], \tag{4}$$

where $H$ is the token distribution entropy, and $\lambda_H$ is the entropy coefficient.

## 5 Experiments

### 5.1 Experimental Setup

**Datasets and Evaluation Metrics.** We evaluate our agent using the multimodal search dataset curated in Section 3, comprising over 2,210 queries across six diverse e-commerce categories. To avoid product-level information leakage across training and evaluation, we split the dataset strictly by product rather than by individual queries. The unified test set consists of 346 queries spanning both Fixed-Item QA and Open-Ended Recommendation tasks. We evaluate task success and efficiency using three primary metrics:

- **Accuracy:** Used for Fixed-Item QA, representing the percentage of queries where the final response correctly aligns with the ground truth.
- **Qualified@4:** Used for Open-Ended Recommendation, this metric calculates the proportion of the top four unique products in the agent's final recommendation that explicitly satisfy all user constraints.

Table 1: Main results on Fixed-Item QA and Open-Ended Product Recommendation.

| Model | Fixed-Item QA | | Open-Ended Product Recommendation | | | | |
|---|---|---|---|---|---|---|---|
| | | | *Unconstrained* | | *Budget-Constrained (Qualified@4)* | | |
| | **Acc.** ↑ | **Turns** ↓ | **Qualified@4** ↑ | **Turns** ↓ | $B = 2$ ↑ | $B = 3$ ↑ | $B = 4$ ↑ |
| Kimi-K3 | 83.7 | 0.84 | 47.3 | 4.07 | 27.59 | 31.57 | 28.13 |
| GLM-5.3-High | 81.9 | 0.80 | 35.2 | 3.35 | 20.69 | 26.72 | 26.51 |
| GLM-5.3-Max | 83.7 | 0.93 | 44.5 | 4.00 | 30.50 | 37.18 | 37.39 |
| GPT-5.6-Sol | 83.3 | 0.88 | 45.6 | 4.03 | 14.66 | 30.06 | 30.06 |
| Qwen3.6-27B | 74.0 | 0.86 | 19.3 | 2.62 | 14.87 | 16.38 | 18.10 |
| *Qwen3.5-4B* | 68.1 | 1.04 | 22.1 | 7.20 | 8.19 | 12.07 | 14.66 |
| + SFT | 81.3 | 0.88 | 32.9 | 5.13 | 20.58 | 24.68 | 28.77 |
| + SFT + RL | 80.2 | 0.88 | **36.9** | 5.21 | **24.03** | **27.16** | **35.34** |
| *Qwen3.5-9B* | 70.8 | 0.92 | 16.9 | 6.40 | 11.53 | 10.78 | 12.28 |
| + SFT | 80.9 | 0.83 | 38.7 | 4.41 | 24.57 | 31.90 | 34.16 |
| + SFT + RL | **81.6** | 1.05 | **49.5** | 5.59 | **32.00** | **40.09** | **46.77** |

- **Tool-Call Turns:** To quantify efficiency, we track the average number of tool-use turns per query. Notably, multiple independent tool calls executed in parallel are counted as a single turn.

For budgeted evaluations, agents operate under a hard tool-use turn limit ($B$). Once the budget is exhausted, the environment terminates further exploration, requiring the agent to generate a final recommendation based on the acquired context.

**Baselines and Environment.** We evaluate our approach against strong closed-source and open-source models, including GPT-5.6 Sol (OpenAI, 2026), GLM-5.3 (GLM-5-Team, 2026), Kimi K3 (Kimi Team, 2026), and Qwen3.6-27B (Qwen Team, 2026b). For our method, our backbone policies build on deployment-scale models, specifically Qwen3.5-4B and Qwen3.5-9B (Qwen Team, 2026a). Throughout the evaluation, the agent interacts with a real-world e-commerce search environment, ensuring that the task complexities reflect actual production scenarios.

**Implementation Details.** During inference, agent rollouts use a temperature of 1.0 with reasoning enabled. To improve statistical reliability, open-ended queries are evaluated via four independent rollouts per model, and we report the average scores. All training experiments are conducted on 8 NVIDIA H200 GPUs. For trajectory generation, we utilize GLM-5.3 (high) as the teacher model to curate the initial SFT data. Detailed SFT/RL hyperparameters and additional implementation details are provided in Appendix B.

### 5.2 Main Results

Table 1 summarizes outcomes for Fixed-Item QA, unconstrained Open-Ended Recommendation, and budgeted Open-Ended Recommendation ($B = 2, 3, 4$).

**Performance Gap of Lightweight Models.** The Qwen3.5-9B base model achieves only 70.8% Accuracy on Fixed-Item QA and struggles on unconstrained Open-Ended Recommendation with a Qualified@4 of merely 16.9%. This highlights the inherent difficulty of the task for lightweight pre-trained models.

**Bootstrapping via Evidence-Grounded SFT.** Applying our evidence-grounded SFT curriculum substantially improves both product-level understanding and end-to-end recommendation. Qwen3.5-9B improves from 70.8% to 80.9% on Fixed-Item QA and from 16.9% to 38.7% Qualified@4 on unconstrained Open-Ended Recommendation. Its performance thereby becomes comparable to GLM-5.3-High on both tasks (81.9% and 35.2%, respectively), validating the efficacy of our staged evidence-grounded SFT pipeline.

**Effectiveness of Budgeted RL.** Budget-constrained RL substantially enhances decision efficiency under strict interaction limits. At $B = 4$, Qwen3.5-9B reaches 46.77% Qualified@4, improving substantially over its SFT initialization at the same budget (34.16%). This result exceeds all compared baselines under the same four-turn constraint, even surpassing unconstrained GPT-5.6-Sol (45.6%).

**Generalization Across Scales.** These gains consistently transfer to the smaller Qwen3.5-4B backbone, where staged SFT and RL boost Fixed-Item QA from 68.06% to 80.21% and unconstrained

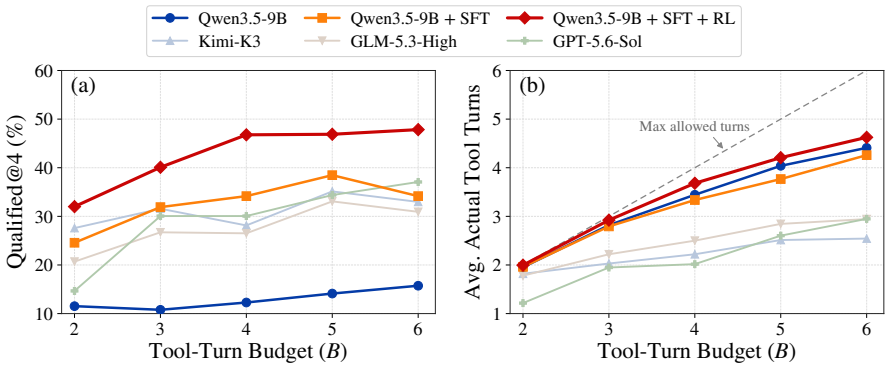


Figure 3: Quality–efficiency trade-off under different tool-turn budgets. (a) Qualified@4 as the budget increases. (b) Average actual tool turns consumed under each budget; the dashed line denotes the maximum allowed turns.

Table 2: Ablation on SFT data composition. We compare the impact of different trajectory types on the Fixed-Item QA Accuracy (%) and Open-Ended Recommendation Qualified@4 (%).

| **SFT Data Composition** | **Qwen3.5-4B** | | **Qwen3.5-9B** | |
|---|---|---|---|---|
| | Accuracy ↑ | Qualified@4 ↑ | Accuracy ↑ | Qualified@4 ↑ |
| Base (Zero-Shot) | 68.1 | 22.1 | 70.8 | 16.9 |
| Fixed-Item only | 75.7 | 8.51 | 78.8 | 10.8 |
| Open-Ended only | 64.2 | 23.4 | 73.6 | 30.8 |
| Mixed Jointly (Fixed + Open) | 76.7 | 32.2 | 75.7 | 34.8 |
| Staged Curriculum (Fixed → Open) | **81.3** | **32.9** | **80.9** | **38.7** |

Qualified@4 from 22.09% to 36.85%, suggesting the robustness and generalizability of our proposed training pipeline.

### 5.3 Quality–Efficiency Trade-off under Tool Budgets

**Pareto Frontier and Budget Extrapolation.** As shown in Figure 3, our SFT+RL agent establishes the strongest quality–efficiency frontier across $B \in \{2, \ldots, 6\}$. Beyond its rapid gains within the training range ($B = 2$ to $B = 4$), it generalizes smoothly to unseen budgets of $B = 5$ and $B = 6$, with Qualified@4 continuing to improve from 46.77% to 47.84%. In contrast, several baselines fluctuate or even degrade with additional budget: SFT drops from 38.47% at $B = 5$ to 34.16% at $B = 6$, while Kimi-K3 drops from 35.13% to 32.97%. This suggests that additional interaction opportunities do not automatically improve recommendations without an effective search policy.

**Active Budget Utilization vs. Early Termination.** Actual executed turns further explain this difference. Strong baseline agents often terminate well below the available cap, consuming only 1.22–1.82 turns at $B = 2$ and plateauing at 2.54–2.95 turns even at $B = 6$. While the Base model uses more turns (4.41 at $B = 6$), its recommendation quality remains low (15.73%). In contrast, our SFT+RL policy increasingly utilizes the available budget as it grows, reaching 4.63 turns at $B = 6$ while maintaining monotonic quality improvements.

### 5.4 Ablation Studies

#### 5.4.1 Contribution of SFT Data Composition

To isolate the contribution of different supervision sources, we compare several SFT data compositions in Table 2. We evaluate single-task variants trained solely on Fixed-Item QA or Open-Ended Recommendation trajectories, standard joint training (mixing both throughout), and our staged curriculum. Across both 4B and 9B scales, single-source training leads to notable task bias: training only on Fixed-Item QA hurts Open-Ended Qualified@4 (8.51% on 4B and 10.78% on 9B), whereas training only on Open-Ended trajectories limits fine-grained QA performance. While naive mixing balances both tasks, our staged curriculum consistently performs best, demonstrating the benefit of progressively learning product-level visual grounding before integrating end-to-end search planning.

Table 3: Ablation of RL tool-budget training strategies. Starting from the same 9B SFT checkpoint, we compare the checkpoint without RL against variants trained for 80 updates under different budget schedules. We report Qualified@4 on Open-Ended Product Recommendation.

| Training Strategy | Training Budget | $B = 2\uparrow$ | $B = 3\uparrow$ | $B = 4\uparrow$ |
|---|---|---|---|---|
| No-RL | – | 24.6 | 31.9 | 34.2 |
| No-Constraint | – | 27.2 | 32.0 | 36.2 |
| Fixed | 2 | 30.3 | 34.8 | 37.8 |
| Mixed | $\{2, 3, 4\}$ | 28.3 | 35.2 | 40.7 |
| Curriculum | $4 \to 3 \to 2$ | **32.0** | **40.1** | **46.8** |

Table 4: Impact of visual tool access across different task subsets. We evaluate multiple agents with and without the Visual Evidence Tool on the Text-Answerable and Visually-Dependent subsets of Fixed-Item QA (Accuracy), as well as Open-Ended Recommendation (Qualified@4).

| **Model** | Text-Answerable QA | | Visually-Dependent QA | | Open-Ended Recommendation | |
|---|---|---|---|---|---|---|
| | w/o Visual | w/ Visual | w/o Visual | w/ Visual | w/o Visual | w/ Visual |
| Kimi-K3 | 98.7 | 100.0 | 17.8 | 77.9 | 11.5 | 47.3 |
| GLM-5.3-High | 100.0 | 100.0 | 13.6 | 75.6 | 9.6 | 35.2 |
| GLM-5.3-Max | 100.0 | 98.7 | 14.6 | 78.4 | 11.6 | 44.5 |
| GPT-5.6-Sol | 100.0 | 100.0 | 16.4 | 77.5 | 12.6 | 45.6 |
| Qwen3.6-27B | 100.0 | 98.7 | 15.5 | 65.3 | 8.2 | 19.3 |
| Qwen3.5-9B Base | 96.0 | 97.3 | 9.9 | 61.5 | 4.9 | 16.9 |

#### 5.4.2 Effect of Tool-Budget Schedules in RL

To study how tool-budget exposure during RL affects the learned policy, we compare the SFT initialization (*No-RL*) against four RL schedules over 80 updates. As shown in Table 3, training without an explicit budget (*No-Constraint*) provides only modest improvements over the SFT initialization, whereas exposing the policy to constrained rollouts consistently improves performance under limited tool budgets. Among the compared strategies, the $4 \to 3 \to 2$ curriculum performs best across all evaluation budgets, suggesting that gradually tightening interaction constraints provides a better balance between exploration and compact execution.

#### 5.4.3 Visual Dependency and the Value of Visual Access

Table 4 reveals two clear patterns. First, our modality-aware partitioning is highly reliable: on the Text-Answerable subset, all models achieve near-ceiling accuracy ($96.0\%$–$100\%$) regardless of visual access, whereas on the Visually-Dependent subset, removing the visual tool causes accuracy to plummet from $61.5\%$–$78.4\%$ to $9.86\%$–$17.84\%$. This sharp contrast confirms that our benchmark effectively isolates genuine visual information needs from text shortcuts. Second, visual access is indispensable for end-to-end product search, improving Open-Ended Qualified@4 by $2.4\times$ to $4.1\times$ across all evaluated models.

## 6 Conclusion

In this work, we study multimodal e-commerce search agents from the perspective of selective visual evidence acquisition. To evaluate both fine-grained perception and system-level execution, we introduce two complementary, modality-aware tasks: *Fixed-Item Question Answering* and *Open-Ended Product Recommendation*. Building on a decoupled tool environment that isolates raw product images from the main agent's context, we develop a staged post-training pipeline that combines an evidence-grounded SFT curriculum with reinforcement learning under progressively constrained tool-turn budgets. Experiments across 4B and 9B backbones demonstrate that our approach substantially advances the quality–efficiency Pareto frontier, enabling a 9B agent under a strict four-turn budget to surpass strong unconstrained proprietary models while generalizing across tool budgets.

### AI Use Statement

In this work, we used AI to aid writing and to generate synthetic datasets. The manuscript was primarily written by the authors; AI assistance was limited to language polishing and formatting

author-provided experimental results into table templates. Large language models were also used to generate synthetic data during dataset construction. All generated data were manually reviewed: 2.37% were corrected and 2.33% were removed as unqualified, corresponding to a total manual handling rate of 4.69%. All AI-assisted table formatting was checked twice by the authors. We take full responsibility for the final manuscript and dataset.

### Ethics Statement

This work utilizes only publicly available e-commerce product information and synthetic user queries, involving no private data, or broader ethical concerns.

### Reproducibility Statement

The submitted supplementary archive contains the key artifacts required to reproduce our data, training, and evaluation pipelines, including the code and prompts for dataset construction, the Visual Evidence Tool implementation, the judge prompts and evaluation code, and the core SFT and RL training code. The corresponding sections and appendices of the paper provide detailed descriptions of the dataset-construction and quality-control procedures (Appendix A), tool interfaces and training configurations (Appendix B), and evaluation criteria and settings (Appendix C). Together, the paper and supplementary archive document the complete workflow from data construction and manual verification to post-training and final evaluation.

## References

Azzah Al-Maskari and Mark Sanderson. A review of factors influencing user satisfaction in information retrieval. *Journal of the American Society for Information Science and Technology*, 61(5): 859–868, 2010. doi: 10.1002/asi.21300.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48. Association for Computing Machinery, 2009. doi: 10.1145/1553374.1553380.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Computer Vision – ECCV 2024*, volume 15139 of *Lecture Notes in Computer Science*, pp. 19–35. Springer, 2024. doi: 10.1007/978-3-031-73004-7_2.

Shuxian Fan, Seonwoo Min, Youna Hu, Botao Xia, Jayakrishnan Unnikrishnan, Rowan Musselmann, Yifan Gao, Qingyu Yin, Priyanka Nigam, and Bing Yin. Shopping reasoning bench: An expert-authored benchmark for multi-turn conversational shopping assistants, 2026. arXiv preprint arXiv:2606.12608.

GLM-5-Team. GLM-5: From vibe coding to agentic engineering, 2026. arXiv preprint arXiv:2602.15763.

Ming Gong, Xucheng Huang, Chenghan Yang, Xianhan Peng, Haoxin Wang, Yang Liu, and Ling Jiang. MindFlow: Revolutionizing e-commerce customer support with multimodal LLM agents, 2025. arXiv preprint arXiv:2507.05330.

Google DeepMind. Gemini 3.1 Pro Model Card, February 2026. Model card.

Zeying Hao, Hao Guo, Mengtao Xu, Yimin Hu, Yuheng Song, Zesheng Zhou, Jinsong Lan, and Xiaoyong Zhu. MMShopBench: A real-log benchmark for multimodal, multi-turn shopping agents, 2026. arXiv preprint arXiv:2607.29002.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O. Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025.

Kimi Team. Kimi K3: Open frontier intelligence, 2026. arXiv preprint arXiv:2607.24653.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.50.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626. Association for Computing Machinery, 2023. doi: 10.1145/3600006.3613165.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Susan Liang, Chao Huang, Filippos Bellos, Jing Bi, Jason J. Corso, and Chenliang Xu. AdaTurn: Budget-aware test-time scaling for active visual perception agents, 2026. arXiv preprint arXiv:2607.14547.

Xinyi Ling, Hanwen Du, Zhihui Zhu, and Xia Ning. EcomMMMU: Strategic utilization of visuals for robust multimodal e-commerce models. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 769–790, Mumbai, India, 2025. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. doi: 10.18653/v1/2025.ijcnlp-long.43.

Tengxiao Liu, Zifeng Wang, Jin Miao, I-Hung Hsu, Jun Yan, Jiefeng Chen, Rujun Han, Fangyuan Xu, Yanfei Chen, Ke Jiang, Samira Daruki, Yi Liang, William Yang Wang, Tomas Pfister, and Chen-Yu Lee. Budget-aware tool-use enables effective agent scaling. In *Third Conference on Language Modeling*, 2026a.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153.

Yong Liu, Ximan Liu, Guoqing Yang, Bing Bai, Xiaoqiang Xu, Zhen Chen, Ke Zhang, and Yan Li. OxyEcomBench: Benchmarking multimodal foundation models across e-commerce ecosystems, 2026b. arXiv preprint arXiv:2605.13173.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Yougang Lyu, Xiaoyu Zhang, Lingyong Yan, Maarten de Rijke, Zhaochun Ren, and Xiuying Chen. DeepShop: A benchmark for deep research shopping agents, 2025. arXiv preprint arXiv:2506.02839.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1928–1937. PMLR, 2016.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition, July 2026. Model release.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. Model release.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026b. Model release.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. arXiv preprint arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. arXiv preprint arXiv:2402.03300.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0280.

Haoxin Wang, Xianhan Peng, Xucheng Huang, Yizhe Huang, Ming Gong, Chenghan Yang, Yang Liu, and Ling Jiang. ECom-Bench: Can LLM agent resolve real-world e-commerce customer support issues? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 276–284, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-industry.19.

Jiangyuan Wang, Kejun Xiao, Qi Sun, Huaipeng Zhao, Tao Luo, Jian Dong Zhang, and Xiaoyi Zeng. ShoppingBench: A real-world intent-grounded shopping benchmark for LLM-based agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(39):33521–33529, 2026. doi: 10.1609/aaai.v40i39.40640.

Zequn Xie, Junjie Wang, Dan Yang, Jie Feng, Yue Shen, Jian Wang, and Jinjie Gu. SlimSearcher: Training efficiency-aware web agents via adaptive reward gating, 2026. arXiv preprint arXiv:2606.07074.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20744–20757, 2022. doi: 10.52202/068431-1508.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Kaiyan Zhao, Zijie Meng, Zheyong Xie, Jin Duan, Yao Hu, Zuozhu Liu, and Shaosheng Cao. EComStage: Stage-wise and orientation-specific benchmarking for large language models in e-commerce, 2026. arXiv preprint arXiv:2601.02752.

Shu Zhao, Tan Yu, Anbang Xu, Japinder Singh, Aaditya Shukla, and Rama Akkiraju. ParallelSearch: Train your LLMs to decompose query and search sub-queries in parallel with reinforcement learning, 2025. arXiv preprint arXiv:2508.09303.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. DeepResearcher: Scaling deep research via reinforcement learning in real-world environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 414–431, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.22.

Ziwei Zheng, Minghao Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: Incentivizing "thinking with images" via reinforcement learning. In *International Conference on Learning Representations*, 2026.

# A Dataset and Annotation Details

## A.1 Construction Pipeline

Our dataset is constructed from real product pages covering six categories: dresses, folding beds, skincare serums, cat food, air fryers, and laptops. The construction pipeline consists of four stages.

**Product-page snapshot.** For each product, we collect its title, structured textual attributes, product images, and detail-page images. All subsequent annotations operate on this frozen snapshot, ensuring that the evidence remains stable throughout dataset construction.

**Checklist-guided fact extraction.** We define a category-specific checklist that organizes product information into first-level attribute groups, such as appearance, structure, specifications, functions, usage instructions, and associated items. A vision-language model extracts candidate facts from each image following this checklist. To avoid assigning information to the wrong object, each product is accompanied by a *product anchor* that identifies the primary product and distinguishes it from gifts, accessories, bundled items, and related products. Every extracted fact retains its source image, product subject, and checklist category.

**Fact consolidation and text–visual alignment.** Candidate facts extracted from different images are consolidated within the same product and checklist category. Semantically equivalent facts are deduplicated, while complementary descriptions of the same attribute are merged. Facts with mutually inconsistent evidence are isolated and excluded from query construction.

Each consolidated visual fact is then compared against the product's complete textual snapshot, including its title, brand, structured attributes, and tags. We distinguish four relations:

- **Text-sufficient:** the textual metadata fully expresses the fact;
- **Visual-more-specific:** text covers the basic attribute, while images provide necessary additional details;
- **Visual-only:** the core attribute or conclusion is absent from text;
- **Conflict:** text and images provide incompatible conclusions about the same attribute.

After aligning all visual facts, we perform a second pass over the complete textual snapshot to recover objective text-only facts that are not represented by any visual fact. Conflict records are supplied during this pass to prevent contradictory attributes from being reintroduced as text-only facts.

**Task construction.** The aligned facts are used to construct two complementary task families.

For *Fixed-Item QA*, the product identity is given. Queries are written as direct attribute questions, checks against explicit user constraints, or judgments about bounded usage scenarios. Each independent answer point is paired with one *minimum fact*, which records the least factual information required for a correct answer.

Fixed-item questions are further divided by evidence requirement:

- **Text-sufficient:** the answer is fully available from textual metadata;
- **Visual-salient:** essential answer information is available only from images;
- **Visual-helpful:** a text-answerable basic question is paired with a more detailed question that requires additional visual evidence.

For *Open-Ended Product Recommendation*, a reference product is used only to establish that the task has at least one feasible solution. Explicit product requirements produce *Constraint* tasks, whereas bounded user objectives that permit multiple implementations produce *Goal* tasks. Their qualification criteria are stored as *hard constraints*. For Goal tasks, an implementation observed in the reference product may be recorded as a feasible example, but it is never treated as the only acceptable solution.

Open-ended queries are validated through real product search. Retrieved products with the same model and core configuration are grouped to avoid counting duplicate listings. We then determine whether each candidate can be confirmed using textual metadata alone. Queries with sufficient text-confirmed candidates are removed from the visual benchmark; the retained tasks either require visual verification or can benefit from it.

Table 5: Dataset statistics under the product-disjoint split.

| Split | Products | Fixed-Item | Open-Ended | Total |
|---|---|---|---|---|
| Train | 57 | 1,435 | 270 | 1,705 |
| Validation | 6 | 128 | 31 | 159 |
| Test | 12 | 288 | 58 | 346 |
| Total | 75 | 1,851 | 359 | 2,210 |

### A.2 Dataset Statistics

We split the dataset by product rather than by query, preventing information from the same product from appearing across training and evaluation sets. Multi-product questions are retained only when all referenced products belong to the same split.

The Fixed-Item QA set contains 1,059 Direct, 436 Constraint, and 356 Scenario questions. By evidence requirement, it contains 305 text-sufficient questions, 1,138 visual-salient questions, and 408 questions forming 204 visual-helpful pairs. The Open-Ended Recommendation set contains 300 Constraint and 59 Goal tasks, of which 294 are visual-required and 65 are visual-optional.

### A.3 Quality-Control Rubrics

We apply the following criteria during fact consolidation, query revision, evidence construction, retrieval validation, and final item-level inspection.

Table 6: Quality-control criteria used throughout dataset construction.

| Criterion | Requirement |
|---|---|
| Factual grounding | Every factual conclusion must be directly supported by the cited product text or image. External product knowledge and unsupported causal inference are not allowed. |
| Subject consistency | Evidence must describe the correct product or associated item. Attributes of gifts, accessories, bundled products, or related models cannot be assigned to the primary product. |
| Semantic fidelity | The condition, subject, scope, and strength of a query or answer must not exceed the source fact. Parameters, structures, and ingredients cannot be expanded into unsupported comfort, safety, compatibility, or efficacy claims. |
| Evaluability | Every part of a Fixed-Item query must have a corresponding minimum fact. Every Open-Ended query must have explicit qualification criteria that can be applied consistently to arbitrary candidate products. |
| Task validity | Direct, Constraint, and Scenario questions must reflect their actual user formulation. Open-ended Goals must describe bounded outcomes rather than broad professional suitability or subjective experience. |
| Modality validity | A visually dependent question must not be directly answerable from the available textual metadata. Open-ended tasks are additionally checked against products retrieved from the real search environment. |
| Transferability | Open-ended qualification criteria must apply to any candidate product and must not encode the reference product's exact implementation as the only valid answer. |
| Non-redundancy | Questions with the same user decision, required information, and evaluation criteria are deduplicated, while questions requiring genuinely different judgments are retained. |

Malformed outputs, unresolved conflicts, unsupported questions, incomplete evidence, and unstable evaluation targets are removed. The remaining queries undergo a final item-level review covering query naturalness, task type, answer completeness, evidence modality, and consistency between the query and its evaluation criteria.

### A.4 Representative Examples

#### A.4.1 Fixed-Item Question Answering

Table 7 illustrates the three user formulations and the three evidence settings. Only the query and minimum required facts are shown.

Table 7: Representative Fixed-Item QA examples. A visual-helpful pair contains two independently answerable questions about the same underlying user concern.

| Intent | Evidence | Query | Minimum fact(s) |
|---|---|---|---|
| Direct | Text-sufficient | Does this laptop have a backlit keyboard? | The laptop is equipped with a monochrome backlit keyboard. |
| Constraint | Visual-salient | My automatic feeder requires kibble smaller than 12 mm in diameter. Does this cat food satisfy that limit? | The kibble is approximately 1 cm in diameter. |
| Scenario | Visual-salient | During lunch breaks, I never have a place for my phone and glasses. Can this folding bed help? | The folding bed has a side storage pocket. |
| Direct | Visual-helpful | **Basic:** Does this laptop screen have an eye-care certification?<br>**Detailed:** Which eye-care certifications has the screen passed, and does it support low blue light? | **Basic:** The screen has an eye-care certification.<br>**Detailed:** The screen has passed TÜV Hardware Low Blue Light and TÜV Eyesafe certifications and supports low-blue-light display. |

#### A.4.2 Open-Ended Product Recommendation

Table 8 contrasts an explicit Constraint with a Goal that permits alternative implementations. The reference product and source identifiers are omitted because they are not part of the user-facing task or the qualification decision.

Table 8: Representative Open-Ended Recommendation examples. Constraint tasks specify the required implementation, whereas Goal tasks preserve the user objective and allow alternative valid solutions.

| Type | Visual status | Query | Qualification criterion |
|---|---|---|---|
| Constraint | Visual-required | I need a folding bed with a dedicated strap that keeps it secured when folded. | The product must include a strap specifically designed to secure the folded state. |
| Goal | Visual-optional | I need an office folding bed that stays secured after folding, so it will not unfold when leaned against a wall or carried. What would you recommend? | The product must provide a mechanism that prevents it from unfolding after storage. A securing strap is one feasible realization, but other directly supported mechanisms are also valid. |

## B Tool and Training Details

### B.1 Visual Tool Interfaces

**Visual Evidence Tool.** The visual tool takes a product-specific sub-query and a set of candidate image identifiers. Coarse-grained indexing processes each image at low resolution, extracting visible text for simple layouts and summarizing visual topics for complex layouts. The Selector combines these indices with the sub-query and selects up to four images. The Reader examines the selected images at higher resolution and returns a concise answer, visible facts linked to image identifiers, and uncertainty when the evidence is insufficient. Image indices guide retrieval; final facts are extracted from the selected image pixels.

Table 9: Auxiliary image-retrieval comparison, scored by Gemini 3.1 Pro Preview. The final column requires both a correct answer and evidence consistent with all minimum answer facts.

| Retrieval | Accuracy (%) | Accuracy + evidence (%) |
|---|---|---|
| Embedding | 55.00 | 48.64 |
| Ours | 67.27 | 60.91 |

Gallery images are resized to a long side of 400 pixels for indexing and 800 pixels for reading; detail images use the corresponding target widths to preserve their elongated layout. This separates inexpensive image indexing from detailed evidence extraction.

**Embedding-Based Retrieval.** Following common contrastive vision–language retrieval practice (Radford et al., 2021), the alternative interface encodes the sub-query and candidate images into a shared multimodal embedding space. Images are ranked by cosine similarity,

$$s(q, I) = \frac{e_q^\top e_I}{\|e_q\|_2 \|e_I\|_2}, \tag{5}$$

and the top four are passed to the same Reader. Both interfaces therefore share the image budget and evidence-reading stage; they differ in how relevant images are identified.

Table 9 reports an auxiliary comparison on 55 visually dependent fixed-item questions, with four rollouts per questions. Outputs are scored by Gemini 3.1 Pro Preview (Google DeepMind, 2026). Our coarse-to-fine visual acquisition improves answer accuracy and the proportion of correct answers supported by the minimum required evidence. These results motivate its use for acquiring detailed product information.

**Illustrative Interaction.** For a request such as "Recommend foldable beds with removable covers," the planner first searches for candidate products, then issues product-specific visual queries about the folding structure and cover removability. Independent queries can be executed in parallel within one tool-use turn. The Reader returns the observed attributes with image references, and the planner uses these facts to produce the final recommendations.

### B.2 Supervised Fine-Tuning

The SFT corpus contains 164 fixed-item and 163 open-ended demonstrations. We train for two epochs: the first presents shuffled fixed-item demonstrations followed by shuffled open-ended demonstrations, and the second shuffles their union. Complete conversations are retained as context. Following standard supervised adaptation of language models (Ouyang et al., 2022), supervision applies to assistant tokens following the last user request, including reasoning, formal tool calls, and the final response. For supervised positions $\mathcal{A}(\tau)$, the objective is

$$\mathcal{L}_{\mathrm{SFT}} = -\frac{\sum_\tau \sum_{t \in \mathcal{A}(\tau)} \log \pi_\theta(y_t \mid x, y_{<t})}{\sum_\tau |\mathcal{A}(\tau)|}. \tag{6}$$

We update the language-model parameters and freeze the visual parameters. The main SFT configuration uses a constant learning rate of $5 \times 10^{-6}$.

### B.3 Budget-Constrained Reinforcement Learning

**Sampling and Reward.** RL uses 80 open-ended training queries selected to exhibit both successful and unsuccessful sampled solutions. Each update samples eight queries and initially generates four trajectories per query. Optimization uses query groups containing at least two valid trajectories. The tool budget is supplied explicitly during interaction, and its exhaustion requires a final answer from the acquired context. The curriculum reduces the budget from four to three to two. Controlled budget-schedule comparisons allocate equal update intervals to these stages.

The reward is the recommendation's Qualified@4 score defined in Appendix C. Thus, reward measures product qualification, while evidence consistency guides SFT demonstration selection. Within each query group $V_q$, the normalized advantage is

$$A_i = \frac{R_i - \mathrm{mean}_{j \in V_q} R_j}{\mathrm{std}_{j \in V_q} R_j + 10^{-6}}, \tag{7}$$

where the standard deviation uses the sample estimate.

Table 10: Optimization settings for SFT and the controlled RL budget-schedule comparison.

| **Parameter** | **SFT** | **RL comparison** |
|---|---|---|
| Optimizer | AdamW | AdamW |
| Optimizer betas | $(0.9, 0.98)$ | $(0.9, 0.98)$ |
| Weight decay | 0.1 | 0.1 |
| Gradient clipping | 1.0 | 1.0 |
| Dropout | 0 | 0 |
| Learning rate | $5 \times 10^{-6}$ | $10^{-6} \rightarrow 3 \times 10^{-7}$ |
| Schedule | Constant | Linear |
| Batch size | 3 trajectories | 8 queries |
| Initial samples/query | — | 4 |
| Context limit | 32,768 | 49,152 |
| Generation limit | — | 8,192 |
| Sampling temperature | — | 1.0 |
| Entropy coefficient | — | 0.001 |

Table 11: Behavioral shifts from SFT to RL on Open-Ended Product Recommendation. Visual Parallelism denotes the average number of visual calls within each visual-tool turn.

| Model | Qualified@4 ↑ | Avg. Tool Turns | Avg. Visual Calls | Visual Parallelism |
|---|---|---|---|---|
| Base | 16.9 | 6.4 | 6.3 | 1.7 |
| SFT | 38.7 | 4.4 | 6.4 | 2.4 |
| SFT+RL | **49.5** | 5.6 | 12.8 | 3.7 |

**Policy Objective.** Let $\rho_{i,t} = \pi_\theta(y_{i,t} \mid h_{i,t}) / \pi_{\text{old}}(y_{i,t} \mid h_{i,t})$ denote the token-probability ratio between the updated and sampling policies, and let $\langle \cdot \rangle_{\mathcal{A}}$ average over generated assistant tokens. We use the clipped GRPO objective (Schulman et al., 2017; Shao et al., 2024) with entropy regularization:

$$\mathcal{L}_{\text{RL}} = -\Big\langle \min\left(\rho_{i,t} A_i,\ \text{clip}(\rho_{i,t}, 0.8, 1.28) A_i\right) + \beta \mathcal{H}(\pi_\theta(\cdot \mid h_{i,t})) \Big\rangle_{\mathcal{A}}. \tag{8}$$

Here, $\mathcal{H}(\pi) = -\sum_v \pi(v) \log \pi(v)$ is the token-distribution entropy. Tool outputs provide conditioning context. The entropy term acts on the policy loss and encourages sampling diversity (Mnih et al., 2016).

**Optimization Settings.** Table 10 gives the SFT settings and the controlled RL comparison configuration. Both stages use AdamW (Loshchilov & Hutter, 2019). The latter uses a 90-update schedule with linear learning-rate decay from $10^{-6}$ to $3 \times 10^{-7}$ and $\beta = 0.001$. Language-model parameters are updated in BF16, with tool-acquisition components fixed during policy training.

### B.4 Post-Training Tool-Use Behavior

Table 11 reveals how post-training reshapes the tool-use behavior of Qwen3.5-9B. SFT reduces tool turns from 6.40 to 4.41 while keeping visual calls nearly unchanged (6.30 vs. 6.40), increasing visual parallelism from 1.69 to 2.44 and Qualified@4 from 16.90% to 38.69%. RL further raises Qualified@4 to 49.46%, with more visual calls (12.82) and higher parallelism (3.66), suggesting that the policy learns to use the available budget for broader, more parallel visual verification.

## C Evaluation Details

**Evaluation Judge.** We use GLM-5.3-Flash (GLM-5-Team, 2026) as a shared judge across evaluated policies, following established LLM-as-a-judge evaluation practice (Zheng et al., 2023; Liu et al., 2023). The judge produces structured, item-level decisions with supporting reasons. We assess task correctness and evidence consistency separately: correctness measures whether the answer fulfills the user's request, while evidence consistency measures whether its product claims are supported by information acquired during the interaction.

**Fixed-Item QA.** Each query is associated with a set of minimum answer facts derived from its designated products. The judge checks each fact as correct, missing, or incorrect, accounting for product identity, numerical values, conditions, negation, and comparison relations. Semantically equivalent expressions are accepted. A response receives an accuracy score of one only when it correctly covers all minimum facts and completely answers the query. Optional details are recorded separately. This criterion evaluates factual completeness while allowing variation in wording.

**Open-Ended Recommendation.** The judge evaluates each of the first four recommended products against the requested product category and all annotated requirements, using search evidence and product images. For explicit constraints, every necessary condition must be confirmed. For goal-oriented requests, a product qualifies when its documented capabilities directly achieve the stated goal; the reference product provides one feasible solution, and alternative implementations are equally valid. Missing evidence for a required condition results in a non-qualifying product. Recommendations sharing the same product identity and core configuration are counted once. If $N_{\text{qualified}}$ unique products qualify, the rollout score is

$$\text{Qualified@4} = \frac{N_{\text{qualified}}}{4}. \tag{9}$$

The denominator remains four when fewer products are recommended. This design rewards complete, distinct, and verifiably suitable recommendations.

**Evidence Consistency.** We check product claims against the preceding context and acquired search and product-detail results, including visual-reader outputs. For fixed-item queries, we verify the supporting evidence for correctly expressed minimum facts and inspect the full response for unsupported claims. For open-ended queries, we inspect claims separately for each recommended product. Evidence must match the product identity and relevant configuration, preserving values, conditions, and certainty. This assessment distinguishes a correct answer supported by retrieved evidence from a correct but unsupported assertion.

**Use in SFT Data Selection.** These assessments guide demonstration selection. Fixed-item trajectories must answer correctly and support the minimum facts with acquired evidence; visually dependent questions also require actual visual-tool use. For open-ended queries, we prioritize trajectories with more qualified and evidence-supported recommendations, retaining those with at least two qualified products. Among trajectories with comparable answer quality and grounding, we prefer fewer visual-tool turns and calls. Selecting one trajectory per query balances coverage with evidence-grounded and efficient demonstrations