FASHIONARENA: A LARGE-SCALE, PRODUCTION-GROUNDED ENVIRONMENT FOR TRAINING AND EVALUATING FASHION SHOPPING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fashion shopping agents are increasingly built to identify user intent and find products with multimodal inputs. Existing fashion shopping environments mainly face two limitations: (i) insufficient product scale and (ii) reliance on catalog-based retrieval, which can produce discrepancies between offline and online tool feedback. These limitations may interfere with agents' learning and exploration. We introduce FashionArena, a large-scale, production-grounded environment and benchmark for training and evaluating fashion shopping agents. It provides approximately 1.6 million real products and six executable shopping tools, with online-concordant feedback grounded in cached responses from online services. Our tasks are constructed from real user intents captured in online behavior logs, covering text-only and image–text requests. We categorize these intents by their shopping needs and constraints, then collect candidate products and agent trajectories through online tools. Human annotation assesses both product suitability and the quality of tool use throughout agent trajectories. The resulting dataset contains 20K training tasks and 2K held-out evaluation tasks. Evaluation follows human-defined criteria and accepts suitable products beyond the reference answers. Experiments show that offline task performance closely matches online task performance. Supervised fine-tuning followed by reinforcement learning significantly improves recommendation performance on our benchmark. These gains persist when the trained policies interact directly with online tools. We release the environment, datasets, and training code to support reproducible research on fashion shopping agents.

1 INTRODUCTION

Fashion shopping agents translate multimodal user intent into recommendations satisfying constraints on style, occasion, material, and budget (Hao et al., 2026). Beyond retrieval, they must explore alternatives, verify attributes, and refine recommendations using gathered evidence. Training and evaluation therefore require environments with broad fashion coverage and tools that support product search and verification.

Existing shopping environments advance interactive learning but remain limited in fashion coverage and online-feedback fidelity. WebShop (Yao et al., 2022), ShoppingBench (Wang et al., 2026a), and ShopSimulator (Wang et al., 2026b) support interactive shopping and policy learning; MMShopBench (Hao et al., 2026) introduces real multimodal requests. The latter three cover general shopping with relatively small fashion catalogs, leaving fewer suitable alternatives for multi-constraint requests and potentially restricting exploration. ShoppingBench and ShopSimulator use BM25 (Wang et al., 2026a;b); MMShopBench combines BM25 text retrieval with embedding-based image retrieval (Hao et al., 2026). This catalog-based search can produce feedback different from online services.

Higher offline scores do not necessarily indicate closer agreement with online behavior. Our controlled comparison shows that BM25 and product-dense retrieval return substantially different candidates from the online tool (Table 1(a)). Changing only the retrieval backend also changes agents' search behavior and task performance, with the model, product pool, and other tools fixed (Ta-

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.

Table 1: Search backend comparison: (a) candidate metrics on 1,200 unseen queries; (b) agent results on 375 text-only medium tasks. Setup, metric definitions, and result analysis are provided in Appendix B.3.

(a) Candidate metrics

Backend	R@6	J@6	NDCG@6	Item-fit
BM25	.0169	.0095	.0136	.2844
Product-dense	.0403	.0225	.0339	.4690
Response replay	.3936	.2746	.4130	.2980
Online tool	–	–	–	.3291

(b) Agent results (GPT-5.5 / Kimi-K3)

Backend	Searches/task	Calls/task	C
BM25	8.04 / 5.43	14.07 / 11.99	.822 / .537
Product-dense	4.43 / 2.90	9.99 / 8.31	.825 / .552
Response replay	7.10 / 4.04	12.98 / 9.78	.752 / .484
Replay-LOO	7.07 / 4.17	13.14 / 10.13	.730 / .487
Online tool	7.11 / 4.25	13.01 / 10.10	.744 / .456

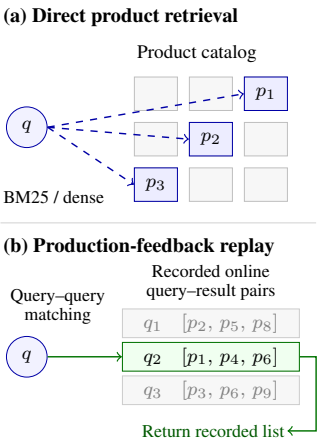


Figure 1: Catalog retrieval versus FashionArena's replay of online search responses.

ble 1(b)). Because agents learn exploration and recommendation strategies from tool feedback, offline evaluation should examine both task performance and agreement with online responses. This motivates an environment grounded in online responses, with paired evaluations of feedback fidelity and direct tests of learning transfer.

We introduce **FashionArena**, a public environment and benchmark combining million-scale fashion coverage, real user intent, and annotated trajectories. Approximately 1.6 million real products, six executable tools, and historical online responses support offline exploration, verification, and recommendation. Unlike catalog-based retrieval, FashionArena matches new queries to historical queries and returns their recorded online product lists (Figure 1). Tasks derive from text-only and multimodal user behavior logs, categorized by shopping needs and constraints. Humans annotate product suitability and individual tool calls; SFT demonstrations require every call to pass review. Open-ended evaluation accepts suitable products beyond the references, measuring requirement satisfaction rather than fixed-answer recovery. We benchmark proprietary and open-weight models, assess scoring reliability, analyze constraint-satisfaction failures, and test online–offline fidelity and the transfer of training gains. Our contributions are summarized as follows:

1. **Large-scale environment and annotated data.** We provide approximately 1.6 million real products, six executable tools, and log-grounded multimodal tasks, with both outcome-level and tool-call-level human annotations for training and evaluation.
2. **Production-grounded shopping interactions.** Controlled comparisons distinguish online-feedback fidelity from offline task performance. Paired evaluations show close agreement in aggregate online and offline model performance under the evaluated protocol.
3. **Model evaluation and learning baselines.** We benchmark proprietary and open-weight models and analyze their strengths and failures. We train shopping agents with supervised fine-tuning followed by reinforcement learning. This training improves task qualification rates in both offline and online evaluations.

2 Related Work

Tool use and interactive learning. Agent benchmarks evaluate both individual tool calls and complete tasks. BFCL (Patil et al., 2025) tests whether models call functions correctly. Interactive benchmarks test how agents combine tools, use earlier results, and respond to users over multiple turns (Yao et al., 2025; Zhou et al., 2025; He et al., 2026). For example, VitaBench includes tasks across real-world services. WebArena and AppWorld evaluate agents that act on websites and applications (Zhou et al., 2024; Trivedi et al., 2024). AgentGym extends this setting from evaluation to training, with environments where agents learn through interaction (Xi et al., 2025).

Table 2: Comparison of existing shopping environments across different dimensions. 1M+ fashion denotes million-scale fashion coverage, not total catalog size; MM: Multi-Modal queries. –: not established.

Environment	Tool observations	Product retrieval	Task construction	1M+ fashion	MM	RL	Domain
WebShop	Catalog	BM25	Product-first	–	–	✓	General
ShoppingBench	Catalog	BM25	Product-first	–	–	✓	General
ShopSimulator	Catalog	BM25	Product-first	–	–	✓	General
MMShopBench	Catalog	BM25 + dense	Real user dialogues	–	✓	–	General
FashionArena	Cached responses	Response replay	Real user logs	✓	✓	✓	Fashion

Shopping environments. Shopping benchmarks test whether agents can find products that meet user requirements (Table 2). WebShop (Yao et al., 2022) studies search and purchase in a real product catalog. ShoppingBench adds complex purchase requirements (Wang et al., 2026a). ShopSimulator focuses on personalized dialogue over multiple turns (Wang et al., 2026b). These three construct requests from preselected products: humans write the requests in WebShop and ShopSimulator, while an LLM generates them in ShoppingBench. Shopping Companion studies user preferences across sessions and rewards for tool use (Yu et al., 2026). ShoppingBench and ShopSimulator also report SFT and RL results. Both search local product catalogs with Lucene/BM25 (Robertson & Zaragoza, 2009).

Fashion and multimodal shopping. Users express shopping needs through both words and images. Fashion datasets cover language-guided image search (Wu et al., 2021), multimodal dialogue (Saha et al., 2018; Kottur et al., 2021), and outfit generation (Chen et al., 2019). MMShopBench (Hao et al., 2026) evaluates shopping agents using real multimodal conversations and human-annotated requirements. Its tools search by keywords or image similarity and allow agents to check product details. It also supports SFT.

3 FashionArena: Production-Grounded Fashion Shopping

Figure 2 connects log-grounded task construction, online collection, and policy learning. Real user intents guide exploration; human review assesses both product suitability and individual tool calls. The annotated products and trajectories support separate training and held-out evaluation sets, together with the offline tool environment described below.

3.1 Tasks and Tools

A task provides a single user request $x = (q, I)$ with text q and an optional reference image I. The agent must recommend products that satisfy the user's style, occasion, material, and budget requirements. It receives neither the full catalog nor suitability labels. Following ReAct (Yao et al., 2023), it performs multiple reasoning and tool-use steps under fixed user requirements and a finite step budget. Multiple product lists may satisfy the request.

Six tools let agents search for candidates, inspect reference images, verify product details, and submit recommendations without making a purchase. Product specifications come from price and SKU records, while web search supplies background information. Appendix B.2 (Table 6) defines the six tool capabilities and their offline implementations.

An agent can search for candidates, check their attributes, revise its query when the results are unsuitable, and submit a final list $\mathcal{L}_\tau$. Search results alone do not establish suitability: the agent must combine the returned evidence to assess the user's requirements. Appendix B.1 formalizes this interaction.

3.2 Data Construction

Log-grounded user intent. We construct tasks from real multi-constraint shopping requests in online logs and their intent-preserving rewrites, covering text-only and multimodal inputs. We categorize the observed intents by occasion, product category, body shape, budget, style, and product

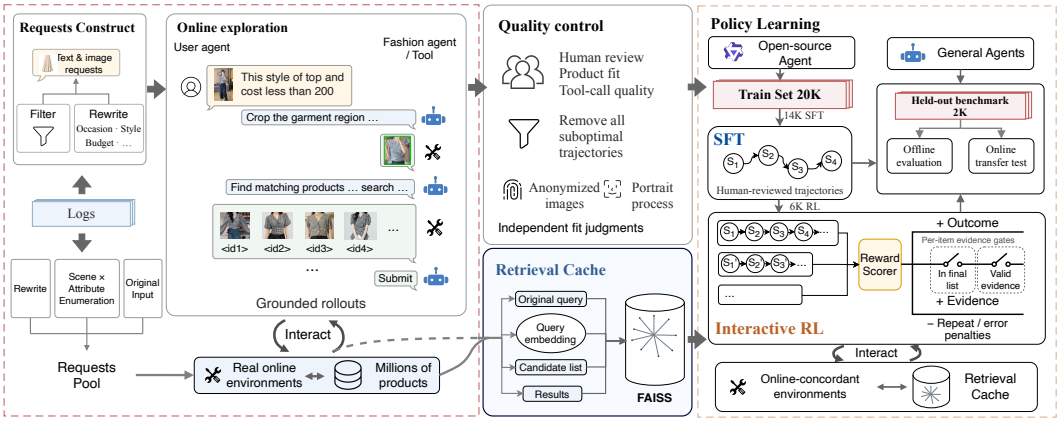


Figure 2: Overview of FashionArena. Left: log-grounded requests guide online product and trajectory collection. Center: human review assesses product suitability and tool-call quality. The cache supports text-query replay and embedding-based retrieval over cached images; agents can also express visual intent as text queries. Right: fully approved demonstrations initialize SFT, followed by RL with cached tools and offline/online evaluation.

attributes, organizing the needs and constraints expressed by users. Image–text requests include product photographs or user-taken images; outfit-matching tasks also include human-verified reference garment pairings. This grounds the task distribution in real shopping needs rather than requirements inferred from preselected target products.

Online collection at product scale. Shopping agents explore candidates through online tools for these requests. We retain request text and images, candidate IDs, tool arguments and responses, and final recommendations. The associated environment contains approximately 1.6 million deduplicated real product records, supporting exploration across a much larger product space than the collected reference answers (Figure 3).

Outcome and trajectory annotation. Human annotators assess both product suitability and each tool call in the agent trajectory. Tool-level review checks whether a call is appropriate given the request and prior observations, repeats an earlier call unnecessarily, or misuses a tool. Initial human annotations undergo quality-control review to check their correctness; structural checks and LLM-assisted filtering support data cleaning. The resulting annotations capture both recommendation outcomes and the quality of the interaction process. SFT demonstrations are retained only when every tool call passes human review. Product-level positives identify suitable candidates found during collection; evaluation also accepts other suitable products. *Collection and annotation details and a cross-family reference-coverage comparison are provided in Appendices A.1 and A.2.*

Training and test sets. We provide approximately 20K training tasks: 14K easier demonstration trajectories for SFT and 6K harder tasks with fewer than three collected reference positives for RL. Reference counts describe discovered positives, not all suitable products in the catalog. The held-out evaluation set contains 1,500 text-only and 500 image–text tasks, selected for requirement coverage. These are different requests, not paired versions. Training and test requests, trajectories, and labels are disjoint, although they may share the product catalog.

3.3 Offline Environment

The online experiments directly access a non-public production shopping service. The offline environment is constructed from cached responses collected from this service. Experiments show that the offline environment closely matches the online service in aggregate task performance and model rankings under the evaluated protocol (Section 5.1). The environment covers approximately 1.6 million deduplicated product records. During online data collection, multiple models expand search queries and retrieve candidate products through an online search service trained on production queries and product data. We cache query–product-list pairs together with product wiki, price, and SKU responses.

Product pool composition. Seven fashion categories account for 93.5% of the pool, each contributing 10.9–17.4% (Figure 3). The core fashion subset is predominantly women's products (74.2%), with men's and neutral products accounting for 17.4% and 8.4%. The pool therefore spans major fashion categories.

Semantic routing and response replay. For a new text query, Qwen3-VL-Embedding-2B (Li et al., 2026) computes its embedding, and FAISS (Douze et al., 2024) finds similar cached queries. The first nonempty neighboring record supplies the product list previously returned by the online tool, containing at most six candidates. Routing therefore matches queries to historical queries rather than ranking products directly (Figure 1).

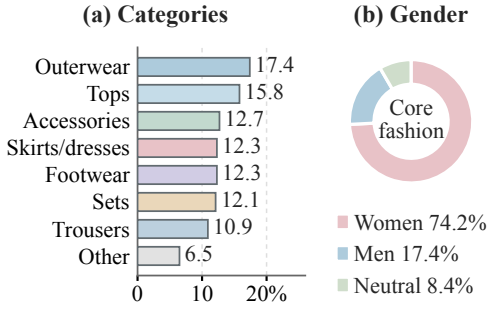


Figure 3: Product-pool composition. (a) All records. (b) Core fashion, excluding Other (6.5%).

Image retrieval and product evidence. Image retrieval uses embedding similarity and is limited to cached images. For visual similarity or outfit-matching requests, agents can also express visual intent as text queries and use text retrieval (Figure 2). Cropping runs locally; user photographs are de-identified without altering garments. Price and SKU lookups take product IDs and return cached records. Product wiki queries instead take product images; the offline tool preserves this image-based interface and returns cached responses from the non-public online wiki service. Cached records preserve missing fields and collection-time values rather than live prices or inventory. External knowledge search requires a user-configured API and network access.

3.4 Training

To evaluate FashionArena's utility for policy training, we initialize Qwen3.5-9B through SFT on human-verified demonstrations, then apply GRPO (Shao et al., 2024) on the RL task split. GRPO compares rewards among new trajectories generated through the cached tools for each task, updating search, verification, and recommendation decisions.

Recommendation outcome. For request x, let τ be a tool-use trajectory and $\mathcal{L}_\tau = (p_1, \ldots, p_n)$ its submitted product list. The length n includes duplicates, while m counts distinct submitted IDs matching training references ($0 \le m \le n$). A match requires a reference ID or maximum similarity to a reference of at least 0.8. Similarity combines SKU-token and title-bigram Jaccard scores with weights 0.6 and 0.4, renormalized over available fields. The outcome score is

$$S_x(\mathcal{L}_\tau) = \begin{cases} 1, & m \ge 3 \text{ or } (n > 0 \text{ and } m = n), \\ m/3, & \text{otherwise.} \end{cases} \tag{1}$$

Full credit requires at least three distinct matches or a nonempty all-match list; otherwise the score is $m/3$. This objective follows the two qualification paths rather than measuring list precision.

Evidence and call costs. For each evidence type $k \in \{\text{search}, \text{price}, \text{SKU}, \text{wiki}\}$, define $b_{k,i} = 1$ if trajectory τ supplies valid evidence of type k for entry p_i, and 0 otherwise. For $n > 0$, the coverage $E_k = \frac{1}{n}\sum_{i=1}^{n} b_{k,i}$ is the fraction of submitted entries with that evidence. Search evidence requires a successful search returning the product. Price, SKU, and wiki evidence require successful lookups with valid corresponding fields after search discovery. Evidence credit does not require a training match, but calls without valid evidence receive no credit.

Let $z_x = 1$ if task x requires SKU verification and 0 otherwise. H is the environment-step budget. N_{rep} counts successful calls repeating identical tools and arguments without new information; N_{err} counts unknown-tool, invalid-argument, and tool-constraint errors, excluding service failures. For $n > 0$, the total trajectory reward is

$$R_x(\tau) = \text{clip}_{[0,1]}\left[0.75 S_x(\mathcal{L}_\tau) + R_{\text{evidence}} - C_{\text{calls}}\right]. \tag{2}$$

For the same trajectory, the evidence reward R_{evidence} and call cost C_{calls} are

$$R_{\text{evidence}} = 0.05E_{\text{search}} + (0.15 - 0.05z_x)E_{\text{price}} + 0.05z_xE_{\text{SKU}} + 0.05E_{\text{wiki}},$$
$$C_{\text{calls}} = 0.10\frac{N_{\text{rep}}}{H} + 0.05\min(N_{\text{err}}, 2). \tag{3}$$

Empty lists receive $R_x(\tau) = 0$. When $z_x = 0$, the SKU weight shifts to price evidence. Appendix D.2 gives field checks, step-level credit assignment, and examples.

Tool-evidence credit is gated by successful calls and valid returned fields, rewarding information acquisition rather than checking whether returned values satisfy the request. Reference-based matching provides an approximate suitability signal for RL, while final performance is assessed separately using LLM judges (Section 4.1).

4 EXPERIMENTS

We evaluate model performance in the released offline environment, test its fidelity against online tools, and examine whether training gains transfer online.

Table 3: Model performance, averaged over three judges. All results use the offline environment except the online transfer block. Qualified ($C^\dagger$) covers 1,500 text-only tasks, 500 image–text tasks, and all 2,000 tasks. Other metrics use all tasks.

Model	Qualified ($C^\dagger$, %)			Overall diagnostics						
	Text	Image–text	All	Fit-3 (%)	All-fit (%)	Item-fit (%)	Mean items	Tool calls	Finish (%)	E2E (%)
Proprietary models										
Qwen3.8-Max	**73.8**	82.8	**76.1**	**75.6**	52.2	78.2	4.18	15.7	21.0	16.0
Claude Opus 5	67.0	**86.7**	71.9	71.2	47.3	78.3	4.15	10.8	44.1	33.0
GPT-5.5	65.6	77.9	68.6	57.9	59.1	77.9	3.32	9.4	**99.9**	**68.6**
Gemini 3.5 Flash	64.6	71.8	66.4	59.3	61.9	80.5	3.03	14.2	96.0	66.1
Claude Sonnet 5	55.1	69.5	58.7	57.5	51.5	81.5	2.75	19.0	61.7	43.1
GPT-5-mini	47.1	64.5	51.5	47.6	36.9	69.1	3.24	6.9	81.5	49.1
Open-weight models										
Qwen3.8-27B	68.1	74.3	69.7	59.2	**62.9**	**82.2**	3.15	14.5	42.6	29.6
Kimi-K3	58.5	73.9	62.3	50.3	57.0	75.7	3.11	9.8	74.3	47.0
Qwen3.5-397B-A17B	49.4	57.0	51.3	42.2	46.3	69.2	3.01	9.4	33.1	15.6
Qwen3.5-35B-A3B	49.8	48.1	49.4	35.7	43.6	65.1	2.96	7.2	8.6	3.9
Qwen3.5-27B	44.8	61.1	48.8	38.2	43.1	65.7	3.06	8.0	33.0	17.6
Qwen3.5-9B learning baselines										
Base	40.1	55.0	43.8	43.8	24.9	76.4	2.91	21.4	47.9	39.7
SFT	59.9	83.3	65.7	65.4	39.5	68.0	4.70	9.3	94.0	64.5
SFT+RL	61.8	84.3	67.4	67.1	39.4	68.5	4.75	9.1	96.2	66.6
Qwen3.5-9B transfer evaluation (Online)										
Base	43.1	56.0	46.3	46.1	26.7	75.5	3.09	20.7	50.6	41.9
SFT	60.1	82.3	65.7	65.5	39.3	68.1	4.68	9.1	95.5	64.7
SFT+RL	62.9	83.1	68.0	67.7	40.2	68.7	4.74	9.1	96.9	67.1

4.1 EXPERIMENTAL SETUP

Models and settings. The main leaderboard evaluates 11 proprietary and open-weight models in the released offline environment, so researchers can reproduce the comparison without access to the source product services. All models use the same system prompt and 2,000 held-out tasks. Temperature is 1.0; the interaction limit is 30 steps. We score one rollout per task. We compare Base, SFT, and SFT+RL checkpoints of Qwen3.5-9B in both environments. SFT uses approved demonstrations; RL interacts with the offline environment.

Evaluation protocol. We use LLM judges to assess whether each recommended product satisfies the user's request (Zheng et al., 2023). GPT-5, Claude Opus 4.8, and Gemini 3.5 Flash assess each

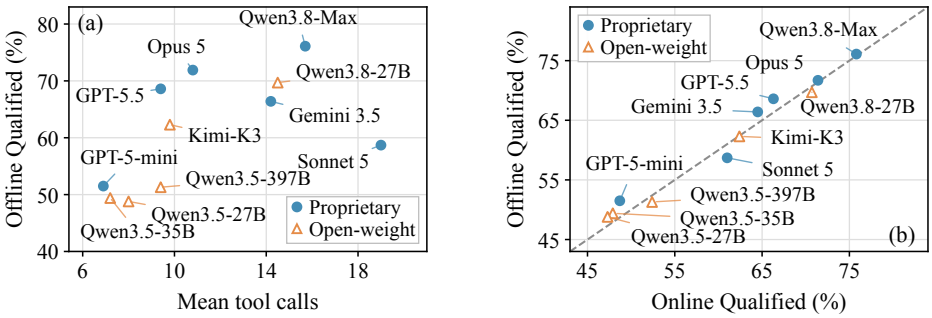


Figure 4: (a) Model performance versus tool calls. (b) Online–offline agreement. Both use 2,000 tasks and three judges. Dashed line: equal scores.

recommendation against the user's request using its product title, image, price, and recommendation rationale. They apply human-defined guidelines to judge suitability in context, each producing its own product-level labels. SKU records are available to human annotators and quality-control reviewers but are not included in the LLM-judge input. We evaluate the structured recommendation submission, or extract recommendations from the final answer when no structured list is submitted. Before judging, recommendations are deduplicated by product ID within each task, so each judge assesses each distinct product only once. Any suitable product is accepted, regardless of whether its ID appears in the reference set. Section 5.2 evaluates agreement with human labels and consistency across judges.

Evaluation metrics. We compute recommendation-quality metrics from these suitability labels. **Fit-3** and **Fit-4** require at least three and four suitable recommendations, respectively; **All-fit** requires a nonempty list with every product suitable. Our primary metric, **Qualified** ($C^{\dagger}$), accepts either Fit-3 or All-fit. Section 5.2 discusses the metric rationale and sensitivity checks. **Item-fit** is the fraction of recommended products judged suitable, pooled across tasks. **Mean items** and **Tool calls** are the average list length and tool-call count per task. **Finish** measures the task rate of submission calls; **E2E** requires both submission and Qualified. For baselines and trained models alike, quality metrics are computed separately for each judge and averaged across the three judges. Task-level rates weight all tasks equally; missing or None judgments count as failures. We also report text-only and image–text results separately. Appendix C.1 provides formal definitions.

4.2 Model Performance

Model rankings depend on request type (Table 3): Qwen3.8-Max leads overall (76.1%) and on text-only requests, while Claude Opus 5 leads on image–text requests (86.7%). Qwen3.8-27B leads the open-weight baselines (69.7%). Aggregate scores alone obscure these modality-specific strengths.

Recommendation quality and execution differ. Qwen3.8-Max offers more options (4.18 items; 75.6% Fit-3), whereas Qwen3.8-27B returns shorter lists (3.15 items) with higher All-fit (62.9%) and Item-fit (82.2%). Qualification and precision thus capture different strengths. More calls need not improve outcomes: Sonnet 5 makes twice as many as GPT-5.5 (19.0 versus 9.4) but scores lower (Figure 4(a)). Execution also changes the ranking: Qwen3.8-Max has only 21.0% Finish and 16.0% E2E, versus GPT-5.5's 99.9% and 68.6%. Qualified alone misses failures to submit otherwise suitable recommendations.

4.3 Training and Online Transfer

Training improves task qualification, with gains retained online (Table 3). For Qwen3.5-9B, offline Qualified rises from 43.8% for the base model to 65.7% after SFT and 67.4% after subsequent RL (Section 3.4). Evaluating the same checkpoints directly with online tools gives 46.3%, 65.7%, and 68.0%, respectively, without additional online training. SFT accounts for most of the gain, followed by a smaller increase after RL.

Both SFT and SFT+RL improve tool-use efficiency and submission completion. Offline mean calls fall from 21.4 to 9.3 and 9.1, while Finish rises from 47.9% to 94.0% and 96.2%, respectively. E2E correspondingly increases from 39.7% to 64.5% and 66.6%; online evaluation shows the same directions of change. Both trained checkpoints produce longer lists (4.70–4.75 items versus 2.91), but have lower Item-fit (68.0–68.5% versus 76.4%). Thus, training improves task qualification and submission completion with fewer calls, while trading off product-level precision for broader recommendation lists. *Per-judge scores, precision analysis, and bootstrap confidence intervals are reported in Appendix D.4.*

5 Analysis

5.1 Fidelity to the Online Environment

We assess candidate recovery, agent behavior, and model performance, holding policies and evaluation settings fixed across tool backends (Appendix B.3.1).

Partial candidate recovery, similar aggregate behavior. Replay achieves Recall@6 of 0.394 on unseen queries, exceeding both direct-retrieval backends but recovering only part of the online lists (Table 1(a)). Its mean searches, total calls, and task scores nevertheless closely track online values (Table 1(b)). For both agents, product-dense reduces search and total calls, whereas BM25 increases them relative to online. Product-dense also leads in candidate Item-fit and GPT-5.5 task score. These results motivate evaluating task performance and fidelity to online feedback as distinct properties. Appendix B.3.4 gives additional retrieval diagnostics.

Task-level cache exclusion. Replay-LOO excludes each task's cached records, including its online trajectory. Qualified changes by -2.2 points for GPT-5.5 and $+0.3$ for Kimi-K3; search and call counts remain close to online (Table 1(b)). This tests task-level reuse beyond unseen queries. *Exclusion rules and results appear in Appendix B.3.*

Aligned model rankings and scores. Across 11 models and 2,000 tasks, rankings closely align (Spearman $\rho = 0.991$, Kendall $\tau = 0.964$), with only GPT-5-mini and Qwen3.5-397B-A17B exchanging adjacent positions. The mean absolute Qualified gap is 1.37 points and the maximum is 2.80 (Figure 4(b)). Agreement in both ordering and scores supports aggregate comparison of the tested models despite partial candidate recovery.

5.2 Evaluation Reliability

LLM-judge validity. On 300 negative-enriched tasks excluded from rubric tuning, three-judge majority voting achieves 80.8% item accuracy against quality-controlled human labels ($\kappa = 0.526$). GPT-5, Claude Opus 4.8, and Gemini 3.5 Flash (GPT/Opus/Gemini in Table 4) give aligned rankings of the 11 offline models ($\rho = 0.925$–0.980), despite task-level disagreements. GPT-5 scores fixed GPT-5.5 trajectories on 500 online tasks three times: Qualified is 77.4–79.2%, with 91.0–94.0% pairwise agreement on task qualification. These checks support aggregate model comparison through consistent rankings and limited scoring variability. *Comparison with human labels, per-judge scores, and repeated-scoring details appear in Appendices C.3 and C.4.*

Table 4: Evaluation reliability.

Human-label agreement (1,435 items)		
Accuracy		80.8%
κ		0.526
Judge pair	Rank ρ	Agree.
GPT / Opus	0.980	81.9%
GPT / Gemini	0.925	85.2%
Opus / Gemini	0.964	83.0%
Repeated scoring: Qualified		
Run 1		77.4%
Run 2		78.6%
Run 3		79.2%
Agreement		91.0–94.0%

Metric rationale and sensitivity. Qualified combines two recommendation goals: Fit-3 rewards multiple suitable options, while All-fit accepts shorter, fully suitable lists when strict requests leave few matching products. This avoids requiring agents to add unsuitable items merely to reach a minimum count. We report Item-fit alongside Qualified because Fit-3 can also accept lists containing unsuitable products. In the separate twelve-baseline online analysis in Appendix C.2, requiring at least two items for All-fit changes overall Qualified by at most 5.1 percentage points, with a rank correlation of $\rho = 0.958$ against the original rule. Fit-3 and Qualified rankings have $\rho = 0.923$.

Thus, Qualified credits both multiple options and short, fully suitable lists while preserving broadly consistent model rankings under the tested rule changes.

5.3 Task Difficulty and Failure Modes

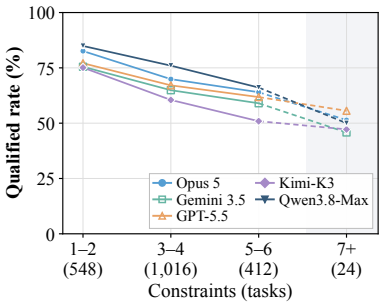


Figure 5: Qualified rates by constraint count for five models. Shaded: 7+ ($n = 24$).

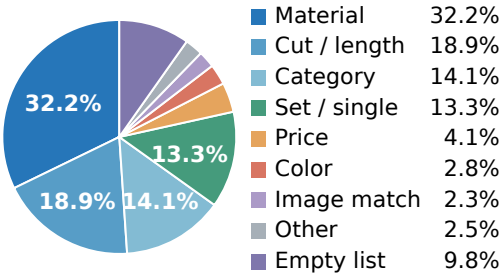


Figure 6: Qwen3.8-27B offline failures

Joint constraints are challenging. Requests with more explicitly expressed constraints are associated with lower qualification rates across all five models (Figure 5). We count request constraints using keyword matching, including cues for body shape and budget. Across the four buckets, Qwen3.8-Max declines from 84.9% to 50.0% and GPT-5.5 from 77.1% to 55.6%. All five models score only 45.8–55.6% in the highest-count bucket ($n = 24$). This shared pattern identifies joint constraint satisfaction as a challenge across model families.

Failure patterns. Finding some suitable products is often insufficient for a qualified recommendation list. Of Qwen3.8-27B's 562 failed tasks under majority voting, 79.4% contain suitable products but remain unqualified. Keyword summaries of judge explanations most often concern material and cut/neckline/length, followed by category and set-versus-single-item issues (Figure 6). A separate execution problem appears in empty outputs: 54 of 55 cases exhaust the budget despite nonempty search results. Empty lists account for 27.8% of Claude Opus 5's failures, versus 9.8% for Qwen3.8-27B, showing that execution failures vary substantially across models. Together, these observations motivate better final-list verification and search termination. *The two-model comparison, empty-output analysis, and diagnostic protocol, including its distinction from mean-over-judges scoring, are given in Appendix E.1.*

6 Conclusion

FashionArena provides a million-scale fashion shopping environment with multimodal tasks grounded in real user logs, human annotations of recommendation outcomes and individual tool calls, and open-ended evaluation. Controlled comparisons distinguish task performance from fidelity to online feedback. Under the evaluated protocol, paired evaluations show close agreement in aggregate online and offline performance. SFT followed by RL improves task qualification, and the same trained checkpoint retains this improvement when evaluated directly with online tools. These results support the environment's utility for policy training and online transfer. More constrained requests have lower qualification rates, while failure analyses reveal partly suitable lists and budget-exhausted empty outputs. The environment, annotated data, and baseline code support further research on fashion shopping agents.

Limitations. Our RL experiments focus on GRPO following SFT. Exploring alternative RL algorithms and training strategies may further improve policy learning. Open-ended fashion evaluation also remains challenging: style, outfit compatibility, and user preferences admit multiple reasonable judgments, which can lead to disagreement even under human-defined criteria. Finally, our data contain single-turn user requests with multi-step tool interactions; extending the benchmark to multi-turn user–agent dialogues remains future work.

AI USE STATEMENT

AI-based tools assisted manuscript drafting and language polishing, data processing, and LLM-based evaluation as described in the paper. The authors reviewed the experimental results and claims and take full responsibility for the final content.

ETHICS STATEMENT

FashionArena is constructed from real user shopping requests collected from online logs. Both product images and images included in user requests undergo processing to remove or mask sensitive information and protect privacy.

REPRODUCIBILITY STATEMENT

We describe the environment construction, data annotation, training procedure, hyperparameter settings, and evaluation protocol to facilitate reproducibility. We will make our environment, datasets, and training and evaluation code publicly available to support further research on fashion shopping agents.

REFERENCES

Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. POG: Personalized outfit generation for fashion recommendation at alibaba ifashion. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The Faiss library. *arXiv preprint arXiv:2401.08281*, 2024. URL https://arxiv.org/abs/2401.08281.

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. *arXiv preprint arXiv:2607.29002*, 2026.

Wei He, Yueqing Sun, Hongyan Hao, Xueyuan Hao, Zhikang Xia, Qi Gu, Hui Su, and Xunliang Cai. VitaBench: Benchmarking LLM agents with versatile interactive tasks in real-world applications. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10007108.

Satwik Kottur, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi. SIMMC 2.0: A task-oriented dialog dataset for immersive multimodal conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026. URL https://arxiv.org/abs/2601.04720.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019.

Amrita Saha, Mitesh Khapra, and Karthik Sankaranarayanan. Towards building large scale multimodal domain-aware conversation systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

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. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33521–33529, 2026a. doi: 10.1609/aaai.v40i39.40640. URL https://ojs.aaai.org/index.php/AAAI/article/view/40640.

Pei Wang, Yanan Wu, Xiaoshuai Song, Weixun Wang, Gengru Chen, Zhongwen Li, Kezhong Yan, Qi Liu, Ken Deng, Shuaibing Zhao, Shaopan Xiong, Xuepeng Liu, Xuefeng Chen, Wanxi Deng, Wenbo Su, and Bo Zheng. ShopSimulator: Evaluating and exploring RL-driven LLM agent for shopping assistants. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35600–35619. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.acl-long.1647. URL https://aclanthology.org/2026.acl-long.1647/.

Hui Wu, Yupeng Gao, Xiaoxiao Guo, Ziad Al-Halah, Steven Rennie, Kristen Grauman, and Rogerio Feris. Fashion IQ: A new dataset towards retrieving images by natural language feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Xin Guo, Dingwen Yang, Chenyang Liao, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. AgentGym: Evaluating and training large language model-based agents across diverse environments. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27914–27961. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1355. URL https://aclanthology.org/2025.acl-long.1355/.

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, 2022.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b126cc38b8638e07bef37e7b2bb72bf-Abstract-Conference.html.

Zijian Yu, Kejun Xiao, Huaipeng Zhao, Tao Luo, and Xiaoyi Zeng. Shopping Companion: Benchmarking and training LLM agents for long-horizon preference-grounded e-commerce tasks. *arXiv preprint arXiv:2603.14864*, 2026. URL https://arxiv.org/abs/2603.14864.

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*, 2023.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

Yang Zhou, Mingyu Zhao, Zhenting Wang, Difei Gu, Bangwei Guo, Ruosong Ye, Ligong Han, Can Jin, and Dimitris N. Metaxas. M^3-Bench: Multi-modal, multi-hop, multi-threaded tool-using MLLM agent benchmark. *arXiv preprint arXiv:2511.17729*, 2025. URL https://arxiv.org/abs/2511.17729.

A Data Construction and Coverage

Reference products are recovered by both Qwen and non-Qwen models at similar average rates (Table 5), indicating that they are not accessible only to the collection family.

A.1 Collection and Annotation

Sources and collection. The source data are real user behavior logs containing text-only and multimodal shopping requests. Tasks comprise these requests and rewrites that preserve their original shopping intent and constraints. Categorization organizes expressed needs by occasion, category, body shape, budget, style, and product attributes. Qwen3.7-plus is the primary trajectory collector; multiple models expand search queries and explore products through online tools for these log-grounded requests. Records preserve request text and images, candidate IDs, tool arguments and responses, and final recommendations.

Two levels of human annotation. At the outcome level, annotators determine whether each candidate product satisfies the request. At the trajectory level, annotators assess each tool call in context for appropriateness, unnecessary repetition, and misuse. Initial annotations undergo human quality-control review to check their correctness and correct errors. For product judgments, humans can inspect titles, images, prices, recommendation rationales, and SKU information. Structural validation and LLM-assisted screening support cleaning. The SFT selection criterion requires every tool call in a demonstration to pass human review, so a suitable final list alone does not qualify a trajectory for SFT. These complementary annotations support learning both what to recommend and how to use the tools. Qualified accepts suitable products beyond the annotated references; only auxiliary reference-recovery metrics require ID matches.

A.2 Cross-Family Coverage

We compare six Qwen and seven non-Qwen models, excluding SFT models. The records also include Qwen3.6-27B, Claude Opus 4.8, and Qwen3.7-plus, which are absent from the main leaderboard. Coverage is the fraction of annotated task–positive pairs recommended. We report the mean across models and the union within each family, noting their unequal sizes.

Table 5: Reference-positive coverage (%) across model families. Q/N denote Qwen/non-Qwen. Both mean and union coverage use task–positive pairs as the denominator.

Subset	Positive pairs	Q mean	N mean	Q union	N union
Overall	6,816	24.3	24.2	57.3	59.0
Text	4,434	29.6	28.2	66.5	66.8
Image–text	2,382	14.5	16.9	40.3	44.6

Per-model coverage is nearly equal overall (24.3% versus 24.2%). Non-Qwen models have slightly higher union coverage, including on image–text tasks. These results support cross-family discoverability of the collected positives, without treating the reference set as exhaustive.

B Tasks, Tools, and Environment Fidelity

B.1 Task Formulation

User requests and recommendation objectives. Using the task $x = (q, I)$ and recommendation list $\mathcal{L}_\tau$ defined in Section 3.1, let $y_x(p) \in \{0, 1\}$ denote whether product p satisfies the request. Suitability labels are hidden from the agent; $I = \varnothing$ for text-only tasks.

Partially observable multi-step decisions. The model cannot directly inspect the entire catalog and must obtain information through tools. We formulate the task as a request-conditioned, finite-horizon partially observable decision process:

$$\mathcal{M}_x^e = (\mathcal{S}, \mathcal{A}, \mathcal{O}, P_e, \Omega_e, R_x, H), \qquad e \in \{\text{on}, \text{off}\}. \tag{4}$$

The latent state $\mathcal{S}$ includes product/service states, remaining budget, and reward-relevant interaction records. P_e and Ω_e define transitions and observations in $\mathcal{O}$; H is the step budget and R_x the reward in Equation 10. Each action $a_t = (k_t, u_t)$ specifies a tool and arguments (Table 6).

Following ReAct (Yao et al., 2023), the agent selects actions from its observable history rather than the full state:

$$\begin{aligned} h_t &= (x, a_0, o_1, \ldots, a_{t-1}, o_t), \qquad a_t \sim \pi_\theta(\cdot \mid h_t), \\ s_{t+1} &\sim P_e(\cdot \mid s_t, a_t), \qquad o_{t+1} \sim \Omega_e(\cdot \mid s_{t+1}, a_t). \end{aligned} \tag{5}$$

Here $h_0 = x$. The episode ends on termination or budget exhaustion ($T \leq H$), yielding τ and $\mathcal{L}_\tau$. Multi-step interaction refers to agent–tool calls, not changing user requirements. Online and offline share task/action interfaces but differ in P_e and Ω_e, allowing paired evaluation of their feedback.

B.2 Tool Interfaces

Table 6: Six tools and their offline implementations. Online and offline environments share the same interfaces; cached product records do not provide live inventory or pricing.

Tool	Function	Public implementation and scope
`similar_search`	Text/image candidate search	Text queries replay historical responses; image retrieval uses embedding similarity over cached images. Agents can also express visual intent as text queries.
`item_wiki`	Product descriptions and attributes	Accepts product images, matching the online interface, and returns cached wiki responses.
`query_price_sku`	Price/SKU verification	Accepts product IDs and returns frozen price/SKU records, without live inventory changes.
`web_search`	Background information retrieval	User-configured external API; requires network access.
`image_crop`	Image-region selection	Deterministic local image cropping.
`finish`	Structured output recording	Parses and records recommendations; no real purchase.

B.3 Search Backend Comparison

Response replay better approximates the tested online feedback than direct catalog retrieval, whereas product-dense retrieval yields higher candidate suitability and task scores. This distinction, summarized in Section 5.1, motivates evaluating environment fidelity separately from offline task performance. We provide the controls and supplementary tests below.

B.3.1 Experimental Setup

Table 1 uses 1,200 text queries absent from the replay cache for candidate evaluation, and 375 medium-difficulty text-only tasks for GPT-5.5 and Kimi-K3 agent evaluation. Agents are not retrained.

Within each model, tasks, prompts, non-search tools, decoding settings, budgets, and the three-judge rubric are fixed. Offline backends share a product snapshot; every search returns six products in the same schema. The two models have different interaction budgets, so backend comparisons are made within each model.

Search backends. BM25 and product-dense rank catalog products by lexical and embedding similarity, respectively. Response replay returns a historical query's recorded product list; Replay-LOO additionally excludes the current task's records. Online tool calls the source retrieval service directly. Thus, the comparison concerns candidate-generation mechanisms, not indexing algorithms alone.

Response replay and task-level exclusion. A FAISS IVF index (`nprobe=64`) retrieves ten neighboring queries; the first nonempty record in similarity order supplies the complete product list. Replay-LOO excludes all records associated with the current task, including its online trajectory. This is stricter than the query-only exclusion in Appendix B.3.4. It applies only to the agent comparison: the unseen-query test has empty task-exclusion sets.

B.3.2 Metric Definitions

Candidate metrics and Qualified use a 0–1 scale. Slash-separated agent results refer to GPT-5.5 / Kimi-K3.

Online-candidate recovery. R@6 measures recovery of online top-six product IDs; J@6 divides shared IDs by the union of the two lists. NDCG@6 additionally rewards recovering online candidates near the top of the returned list. These measure output fidelity, not suitability. Online self-overlap is omitted (shown as –).

Candidate suitability (Item-fit). Blind three-judge evaluation measures whether returned candidates satisfy the request, independently of online-list membership. Here Item-fit evaluates search candidates, whereas the leaderboard evaluates final recommendations.

Agent outcomes. Searches/task and Calls/task average search calls and all tool calls over all 375 tasks. Lower counts are not inherently better. Qualified (C) follows Equation 6, with three judges and missing judgments counted as failures.

B.3.3 Results and Interpretation

Fidelity and task scores favor different backends. Replay improves all three online-list recovery metrics over both direct-retrieval backends (Table 1(a)). Its Recall@6 corresponds to about 2.36 of six online products, so recovery remains partial. Replay also gives search/call means close to online, while product-dense reduces tool use and BM25 increases it (Table 1(b)). Product-dense has the highest candidate Item-fit. For GPT-5.5, both direct-retrieval backends exceed replay's Qualified by 7.0–7.3 percentage points. Kimi-K3 has the same numerical ordering. These fixed-policy comparisons show that backend choice affects both behavior and measured performance; higher offline scores alone do not establish greater online fidelity.

Task-level cache exclusion. Excluding each task's own records changes Qualified by -2.2 percentage points for GPT-5.5 and $+0.3$ for Kimi-K3 (Table 1(b)).

Scope of the conclusions. The backend comparison evaluates two fixed agents on text-only tasks. Backend-specific training and multimodal transfer remain outside this experiment's scope. Online returns include 1.44% of items outside the offline snapshot; two missing judge outputs count as failures.

B.3.4 Additional Retrieval Diagnostics

Supplementary tests also show partial recovery of online candidates (Table 7). Leave-one-query-out (LOO) excludes a query's own records; Unseen uses queries absent from the cache. These tests are separate from Table 1(a). Pooled top-30 recall is substantially higher than actual top-6 recall, but describes a diagnostic candidate pool rather than the list returned to the agent.

Table 7: Supplementary online-list recovery. Top-6 is actual output; pooled top-30 is diagnostic only. Rewrite recall uses known suitable products, not online lists. –: not reported.

Metric	LOO	Unseen
Recall@6 (%)	45.83	38.80
NDCG@6	0.48	–
Jaccard@6	–	0.273
Pooled Recall@30 (%)	≈ 82	75.83
Rewrite recall	0.9204	

Separately, Qwen3.5-Flash generates 544 query variants, including originals, for 120 tasks. Recall against known suitable products is 0.9204, averaged within tasks and then across tasks. This evaluates recovery of annotated suitable products under query reformulation, not overlap with online search results.

C Evaluation Protocol and Reliability

C.1 Metrics and Aggregation

For evaluation, $\mathcal{L}_\tau$ is the recommendation list after deduplication by product ID within each task. Deduplication precedes LLM judging, so a product already judged is not judged again within the same task–judge evaluation. Let n be the number of distinct evaluated products, $m_j = \sum_{p \in \mathcal{L}_\tau} y_x^{(j)}(p)$ the number judged suitable by judge j, v_x indicate a recorded recommendation-submission action, and u_x count recorded tool calls. Table 8 separates content quality, list length, and interface execution. Scores are recomputed from archived product-level verdicts; record handling is described in Appendix C.4.

Table 8: Metric definitions. Conditions apply to one task and one judge; baseline and training task scores average over three judges and fixed task denominators. Item-fit is aggregated at product level.

Metric	Rule	Interpretation
Fit-3	$\mathbf{1}[m_j \geq 3]$	At least three suitable options
Fit-4	$\mathbf{1}[m_j \geq 4]$	At least four suitable options; other items may be unsuitable
All-fit	$\mathbf{1}[n > 0 \wedge m_j = n]$	Every item in a nonempty list is suitable
Qualified ($C^\dagger$)	Fit-3 or All-fit	Union of two qualification paths
Item-fit	Fraction of suitable recommended items	Product-level precision
Mean items	Task mean of n	Recommendation-list length
Tool calls	Task mean of u_x	Tool-use volume
Finish	$v_x = 1$: recorded submission call	Auxiliary interface metric
E2E	$v_x Q_x^{(j)}$	Submission and qualified content

Task-level qualification and aggregation are

$$Q_x^{(j)}(\mathcal{L}_\tau) = \mathbf{1}[m_j \geq 3 \ \vee \ (n > 0 \ \wedge \ m_j = n)]\,,$$
$$C^\dagger = \frac{1}{JN} \sum_{j=1}^{J} \sum_{x=1}^{N} Q_x^{(j)}. \tag{6}$$

Here N is the fixed subset size, and $J = 3$ for the main leaderboard, learning comparisons, and paired online–offline comparison. Missing judgments count as failures rather than reducing the denominator. Each judge first determines task outcomes, which are then averaged; product labels are not majority-voted first. Empty recommendations satisfy neither channel. Extraction prioritizes the submitted tool payload and falls back to the final answer when it is absent. E2E intersects conditions on the same task rather than multiplying aggregate rates. Image–text membership follows the image supplied in the evaluation input, including reference-product images used by matching tasks: 500 image–text and 1,500 text-only tasks.

C.2 METRIC SENSITIVITY

The tested metric changes largely preserve model ordering (Table 9). This model-level analysis uses earlier online records from twelve baselines. R@GT measures recovery of annotated positives; its denominator excludes one task without a reference positive.

Table 9: Metric sensitivity and reference-set metrics on complete twelve-baseline online records. Rank comparisons use model-level Spearman correlations.

Check	Result
All-fit minimum list length: $1 \rightarrow 2$	Maximum Qualified change: 5.1 pp overall; 6.6 pp text
Variant versus original Qualified ranking	$\rho = 0.958$
Fit-3 versus Qualified ranking	$\rho = 0.923$
R@GT versus Qualified ranking	$\rho = 0.853$
Reference-set F1 versus Qualified ranking	$\rho = 0.881$

C.3 AGREEMENT WITH HUMAN ANNOTATIONS

We compare LLM-judge decisions with human annotations finalized through quality-control review. Annotators first label product suitability, and quality-control reviewers check and correct those labels. The resulting labels serve as the reference for this comparison.

Comparison setup. The comparison covers 1,435 product judgments from 300 tasks excluded from rubric tuning: 200 tasks contain negative products and 100 contain only positives; 68 are image–text tasks. The sample is enriched for negative products and is not representative of customer traffic. LLM judges receive product titles, images, prices, and recommendation rationales in the context of the request; humans can additionally consult SKU information. Agreement is measured under these respective information settings. We use three-judge majority voting for product-level agreement, whereas leaderboard scores average the judges' task outcomes.

C.3.1 AGREEMENT AND RELABELING SUBSETS

Table 10: Three-judge majority accuracy against quality-controlled human labels. Subsets distinguish labels unchanged or corrected during quality-control review.

Subset	Items	Accuracy
All	1,435	80.8% ($\kappa = 0.526$)
Image–text labels unchanged	326	96.3%
Image–text labels corrected	56	46.4%

Errors concentrate in items corrected during quality-control review: they account for 30 of the 42 image–text errors (Table 10). The high accuracy on unchanged labels reflects class imbalance: 319 of 326 items are positive, so an always-positive rule reaches 97.9%, and the judge detects only one of seven negatives. Negative-product recognition, rather than accuracy on mostly positive items, is therefore the priority for judge improvement.

C.4 CROSS-JUDGE CONSISTENCY AND REPEATED SCORING

Judges largely preserve model ordering despite differences in absolute scores (Table 11). All judge scores use the same outputs and a fixed denominator of 2,000 tasks per model.

Table 4 reports pairwise rank correlation and task agreement. The corresponding task-level Cohen's κ values are 0.606 (GPT/Opus), 0.672 (GPT/Gemini), and 0.627 (Opus/Gemini), averaged across models on common records. All three judges agree on 75.1% of task outcomes. The largest within-model score range is 9.0 percentage points for Gemini 3.5 Flash, motivating multi-judge reporting rather than reliance on one judge's absolute score.

Table 11: Offline Qualified by judge (%).

Model	GPT-5	Opus 4.8	Gemini 3.5	Mean
Qwen3.8-Max	75.5	74.2	78.5	76.1
Claude Opus 5	71.1	72.4	72.3	71.9
Qwen3.8-27B	69.0	69.1	71.0	69.7
GPT-5.5	69.5	66.5	69.9	68.6
Gemini 3.5 Flash	66.3	61.9	70.9	66.4
Kimi-K3	63.2	59.0	64.8	62.3
Claude Sonnet 5	57.4	58.7	60.1	58.7
GPT-5-mini	53.1	50.5	50.7	51.5
Qwen3.5-397B-A17B	51.7	51.1	51.0	51.3
Qwen3.5-35B-A3B	50.4	46.4	51.3	49.4
Qwen3.5-27B	50.4	45.8	50.2	48.8

Gemini 3.5 denotes Gemini 3.5 Flash. Components and means are rounded independently.

C.4.1 Repeated Scoring of Fixed Outputs

The repeated-scoring results in Table 4 hold 500 online GPT-5.5 trajectories fixed and rerun GPT-5 judging with final-answer extraction. Pairwise task agreement is 94.0%, 91.0%, and 92.6%. This measures scoring variability with agent outputs held fixed.

D Policy Training and Results

D.1 Training Suitability and Policy Objective

During training, reference products and similarity rules approximate $y_x(p)$. Let G_x be the training task's reference set and $\sigma_x(p)$ its maximum weighted product similarity to a reference. Define

$$\widehat{y}_x(p) = \mathbf{1}[p \in G_x \ \vee \ \sigma_x(p) \geq \delta], \qquad \delta = 0.8. \tag{7}$$

For a nonreference product, $\sigma_x(p) = \max_{g \in G_x} s(p, g)$, where

$$s(p, g) = \frac{\sum_{f \in A(p,g)} w_f J_f(p, g)}{\sum_{f \in A(p,g)} w_f}, \qquad w_{\mathrm{SKU}} = 0.6, \quad w_{\mathrm{title}} = 0.4. \tag{8}$$

Here J_f is the Jaccard similarity of the corresponding token sets, and $A(p, g)$ includes fields that are nonempty for at least one of the two products. If both products lack both fields, the score is zero. Strings are Unicode-normalized and case-folded. SKU tokens are attribute–value pairs; title tokens are character bigrams after removing whitespace and punctuation. Exact reference-ID matches receive similarity 1. This score compares product records, not a query embedding with a product embedding.

Policy objective. On the independent RL task distribution $\mathcal{D}_{\mathrm{RL}}$, we optimize

$$\max_{\theta} \ \mathbb{E}_{x \sim \mathcal{D}_{\mathrm{RL}},\, \tau \sim (\pi_\theta, \mathcal{M}_x^{\mathrm{off}})} [R_x(\tau)], \qquad T \leq H. \tag{9}$$

GRPO updates the policy using relative rewards among trajectories for the same task. Training reference sets G_x exclude test supervision. Evaluation uses independent tasks and LLM judges, not the training similarity rule in Equation 7.

D.2 Reward Implementation

The reward in Equations 1–3 combines reference-based outcome credit, tool-evidence coverage, and call costs. The list length n counts nonempty submitted ID entries, including duplicates, while the match count m counts distinct matching IDs. Repeating one matching ID three times therefore gives $n = 3$, $m = 1$, and $S = 1/3$.

Valid evidence. Evidence is checked for every submitted entry, including nonmatches. In addition to the ordering rules in Section 3.4, valid evidence requires a nonempty, nonzero price and nonempty text for SKU and wiki fields. Failed or empty responses earn no credit. A single price-and-specification lookup may supply both price and SKU evidence. The task-level flag z_x specifies whether SKU verification is required, independent of the agent's tool choice. Price evidence contributes to the reward regardless of whether the request specifies a budget.

Step-level credit and costs. Search credit goes to the latest step returning the submitted ID; price, SKU, and wiki credit go to the first qualifying lookup. Outcome credit goes to the terminal step. Costs are charged to repeated-call steps and the first two model-error steps. Repeat detection uses tool names and canonicalized arguments. There is no general length penalty, so additional nonrepeated exploration is not penalized merely for taking more steps.

Writing $\widetilde{r}_t$ for these pre-clipping step credits and T for the number of trajectory steps, the implementation preserves the clipped trajectory total by adjusting the terminal credit:

$$\begin{aligned} \widetilde{R}_x &= \sum_{t=1}^{T} \widetilde{r}_t, \\ R_x &= \begin{cases} \mathrm{clip}_{[0,1]}(\widetilde{R}_x), & n > 0, \\ 0, & n = 0, \end{cases} \\ r_t &= \widetilde{r}_t \quad (t < T), \qquad r_T = \widetilde{r}_T + R_x - \widetilde{R}_x. \end{aligned} \tag{10}$$

The terminal adjustment applies to empty lists as well, so the undiscounted step credits sum to the clipped trajectory reward. Rewards and step credits are rounded to four decimal places, with a final-step correction for rounding.

D.3 GRPO Configuration

Table 12 gives the offline GRPO configuration. The 16-step training episode budget is separate from the optimization-step limit and the 30-step evaluation budget.

Table 12: GRPO training settings.

Setting	Value
Policy initialization	Qwen3.5-9B SFT checkpoint
Policy optimization	GRPO
Trajectories per task	8
Trajectories per batch	128
Maximum environment steps per episode	16
Learning rate	10^{-6}
Warmup steps / schedule	10 / cosine decay
Configured optimization-step limit	401

D.4 Learning Results and Uncertainty

The additional SFT+RL gains are positive under every judge in both environments, and the task-paired bootstrap intervals for overall Qualified and E2E exclude zero. This subsection supplements Table 3 with judge-specific scores and uncertainty. SFT+RL denotes the final training checkpoint (step 400), initialized from the reported Qwen3.5-9B SFT checkpoint and further optimized through RL in the offline environment. We use the final checkpoint without selecting among checkpoints based on evaluation performance; tasks and scoring follow Section 4.1.

Precision and list length. SFT increases offline mean list length from 2.91 to 4.70 and Fit-3 from 43.8% to 65.4%, while Item-fit decreases from 76.4% to 68.0% (Table 3). Longer lists contain more recommendations, enlarging the denominator of Item-fit; the lower suitable-item fraction indicates that suitable recommendations do not grow proportionally with total recommendations. These aggregate results show a trade-off between providing multiple suitable options and product-level

precision, rather than establishing list length as the sole cause of the decline. The online results show the same pattern: mean list length rises from 3.09 to 4.68, while Item-fit falls from 75.5% to 68.1%.

Table 3 shows small increases in Item-fit and mean list length after RL, while overall offline All-fit changes from 39.5% to 39.4%. Thus, higher overall qualification does not mean improvement on every recommendation criterion.

D.4.1 JUDGE BREAKDOWN

Table 13: Overall Qualified (%) by judge on 2,000 tasks. Components and means are rounded independently.

Model	Environment	GPT-5	Opus 4.8	Gemini 3.5	Mean
Base	Online	45.1	46.7	47.0	46.3
SFT	Online	66.5	64.7	65.8	65.7
SFT+RL	Online	68.6	67.5	67.7	68.0
Base	Offline	42.5	44.2	44.7	43.8
SFT	Offline	66.6	65.0	65.6	65.7
SFT+RL	Offline	68.4	66.2	67.7	67.4

D.4.2 TASK-PAIRED UNCERTAINTY

We resample the 2,000 tasks with replacement for 10,000 bootstrap replicates, retaining paired checkpoint outputs and three-judge task scores. All four intervals exclude zero, although offline Qualified is close to the boundary (Table 14). This supports an additional gain for the evaluated checkpoint pair across tasks.

Table 14: Task-paired bootstrap differences: SFT+RL minus SFT, in percentage points.

Environment	Metric	Difference	95% CI	p
Online	Qualified	$+2.27$	$[+0.65, +3.88]$	0.006
Online	E2E	$+2.43$	$[+0.78, +4.12]$	0.004
Offline	Qualified	$+1.70$	$[+0.08, +3.33]$	0.039
Offline	E2E	$+2.17$	$[+0.52, +3.88]$	0.012

Judge-specific exact McNemar tests give mixed results; the judges share tasks and trajectories and are not independent replications. The bootstrap therefore uses aggregate three-judge task scores. It captures task-sampling uncertainty, not rollout or training-seed variation, and the checkpoint comparison does not isolate individual reward components.

E AGENT BEHAVIOR ANALYSIS

E.1 OFFLINE FAILURE DIAGNOSTICS

Both models usually fail with partly suitable lists, but empty recommendations occupy a larger share of Opus's failures (Figure 7). This extends the Qwen analysis in Section 5.3 to Claude Opus 5 on the same 2,000 tasks.

Diagnostic protocol. A task enters the analysis when at least two judges mark it unqualified. We assign its most frequent judge-level pattern: empty recommendations (M1), no suitable items in a nonempty list (M2), or some suitable items but an unqualified list (M3). Counts are 55/61/446 for Qwen and 145/44/332 for Opus. These majority-vote diagnostics are separate from the leaderboard's mean-over-judges scores.

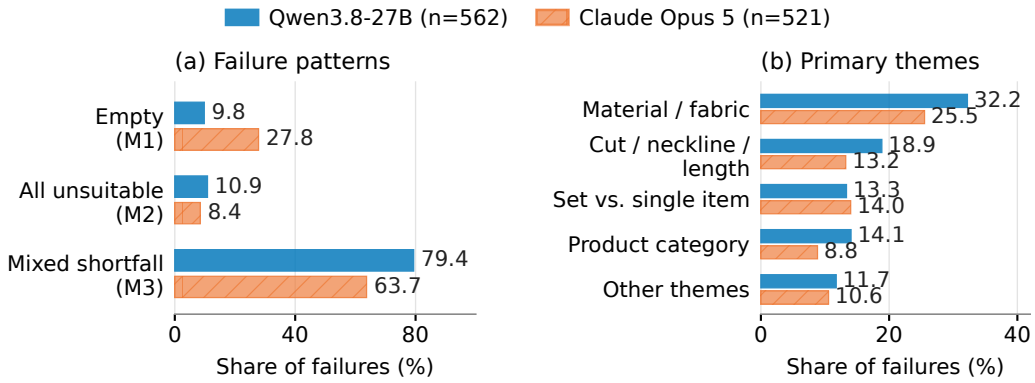


Figure 7: Two-model offline failure analysis. Both panels use all failed tasks as denominators (Qwen: 562; Opus: 521). (a) Recommendation patterns. (b) Primary keyword-derived themes from judge explanations, excluding empty recommendations.

Partly suitable lists account for 79.4% of Qwen failures and 63.7% of Opus failures; empty lists account for 9.8% and 27.8%. Material/fabric is the leading theme for both models. The shared pattern points to final-list verification as a challenge beyond finding any suitable product.

Keyword rules assign each nonempty failure its most frequent theme in judge explanations, breaking ties by a fixed order. Empty-list tasks are excluded from theme assignment. Themes are descriptive rather than independently annotated causes: a keyword can mention an attribute without identifying a violation. Both panels use all failed tasks as denominators (562/521).

In the main-text pie chart, the finer Qwen categories separate price, color, and image matching; Other combines gender-related and residual cases. Together with empty lists, the sectors partition all 562 failures.

Empty outputs have different execution patterns. Qwen exhausts its budget in 54 of 55 empty-list cases despite nonempty search results, averaging 42.8 search calls; a step may contain multiple calls. Its remaining case submits no extractable product. Opus repeatedly supplies an incorrectly named text-query argument in 140 cases and exhausts the budget in five. These observations identify stopping and argument recovery as distinct improvement targets.

F Prompts and Interaction Examples

F.1 Rollout Prompt

English translation of the original Chinese rollout prompt. The actual prompt requires Chinese responses. The quoted prompt is preserved as a historical experimental artifact. Its references to live prices, current availability, and asking the user do not describe capabilities of the frozen offline evaluation: product facts are collection-time snapshots, and no additional user reply is provided in the fixed-request task. These prompt–environment mismatches are limitations of the reported setup, not evidence that the offline tools supply live facts.

Rollout Prompt

Task Definition

You are a professional AI fashion styling assistant. Your goal is to find and recommend real, currently available fashion products that satisfy the user's shopping request by progressively calling available tools in an interactive e-commerce environment.

- At each step, you receive a user message or tool response; adjust your plan promptly based on environment feedback.
- If necessary information is missing and cannot be obtained with available tools, ask the user. Do not fabricate unknown parameter values.
- All output must be in Chinese, except tool names, parameter names, and English proper nouns or brands appearing in original product information.

Environment Information

Overview

The fashion shopping system is a backend environment built around real products currently for sale. It is the sole trusted data source for this task and includes:

- Fashion catalog: a large collection of real products, searchable by text descriptions and/or images, returning product IDs, titles, and main images.
- Price and SKU information table: indexed by product ID, recording live prices and specifications such as material, fabric, fit, size, and color.
- Product wiki: introductory information such as selling points, styles, and styling suggestions.
- Internet information: time-sensitive content such as current fashion trends and brand information.

Environment Rules and Constraints

- Each product has a globally unique ID. Its factual information can only be obtained by querying the environment.
- Price and SKU information may be absent and therefore unavailable. Each product has at most one price/SKU information record, not multiple records.
- Catalog search returns several candidates ranked by query relevance; not every candidate necessarily satisfies all requirements.
- Wiki content is introductory and does not contain specifications such as price, material, or size. The price/SKU table is authoritative for specifications.
- Internet information is time-sensitive and is only for trends and background, not as a source of product information.

Final Response Format

The final response has three parts:

1. Opening (3–5 sentences): Give professional styling advice grounded in the user's situation and naturally introduce the product list.

2. Product list: Use a separate entry for each product, with a blank line between entries. Include, in order: number, product ID, title, Markdown image, price/specification verification results, and recommendation rationale.

3. Closing: End with one natural follow-up invitation, without pressuring the user to buy.

F.2 Interaction Example and Tool Observations

Excerpts from the English translation of `travel_med_2856`, retaining all nine rounds, twelve tool calls, and message order. Execution used Chinese. Ellipses mark omitted text; image URLs and repeated product payloads are omitted. Reasoning and call identifiers are not shown. The complete source record is retained with the example. Recorded recommendations are illustrative, not independently validated here.

User

[1] I'm traveling to Vietnam next week. Please find a women's vacation set. I have an apple-shaped body and want to conceal my stomach; it should be breathable, preferably a currently popular, photogenic style, under RMB 500.

Assistant

[2] [. . .] Since you want a "currently popular, photogenic style," I'll first check this summer's vacation trends, then find suitable products.

Tool Call: `web_search`

Arguments: {"query": "2026 summer Vietnam vacation outfits photogenic styles women"}

Tool Call/Response

[3] Vietnam's summer (May–October) is approximately 28–35°C, hot and humid with strong UV. Favor breathable, lightweight, sun-protective cotton/linen, silk, or chiffon. [. . .]

Sources and timing: summarized from 2024–2025 fashion-week Resort collections and Vietnam travel-outfit guides (latest data from 2024), for reference for summer 2026 trips.

Assistant

[4] [. . .] For an apple-shaped figure, I'll focus on high-waisted, loose, breathable sets that conceal the stomach and photograph well.

Tool Call: `similar_search`

Arguments: {"query_text": "women's vacation set conceal stomach breathable high-waisted wide-leg trousers photogenic", "image_path": ""}

Tool Call/Response

[5] success: true; error: null.

Item ID: `942544321347`

Title: New Chinese-style women's summer 2026 fashionable casual refined premium short-sleeve and wide-leg trouser two-piece set

[. . . Five other products and image URLs omitted.]

Assistant

[6]

Tool Call: `query_price_sku`

Arguments: {"item_ids": "942544321347,948094732848,936848030509,917568073306,941557047173,917712623369", "query_type": "price"}

Tool Call/Response

[7] success: true; query type: price.

`942544321347`: 248.0; `948094732848`: 238.0; `936848030509`: 0.0; `917568073306`: 238.0; `941557047173`: 112.4; `917712623369`: 79.8. All records report success.

Assistant

[8] These results are mostly sporty and casual, rather than ideal for vacation photographs. I'll search for more vacation-oriented styles.

Tool Call: `similar_search`

Arguments: {"query_text": "women's vacation printed chiffon set loose slimming high-waisted wide-leg trousers two-piece set", "image_path": ""}

Tool Call/Response

[9] success: true; error: null.
Item ID: `796700065039`
Title: 2026 new summer cotton-linen set for middle-aged mothers, premium stylish loose top and wide-leg trousers two-piece set
[... Five other products and image URLs omitted.]

Assistant

[10] These lean too much toward Chinese-inspired or middle-aged styles, rather than photogenic vacation outfits for younger women. I'll search for more fashionable vacation options.
Tool Call: `similar_search`
Arguments: {"query_text": "women's seaside vacation beach set camisole wide-leg trousers tropical print slimming photogenic", "image_path": ""}

Tool Call/Response

[11] success: true; error: null.
Item ID: `916165515872`
Title: Summer 2026 fashionable Korean-inspired outfit: ethnic-pattern cover-up paired with floral camisole dress set
[... Five other products and image URLs omitted.]

Assistant

[12] These vacation dresses with cover-ups look good for photographs. I'll check their prices and search more precisely.
Tool Calls (2):
Tool Call: `query_price_sku`
Arguments: {"item_ids": "916165515872,932835144985,1035359410143,1013833790937,917360863329,762658656208", "query_type": "price"}
Tool Call: `similar_search`
Arguments: {"query_text": "women's vacation openwork crochet set loose figure-concealing wide-leg trouser two-piece set seaside photos", "image_path": ""}

Tool Call/Response

[13] success: true; query type: price.
`916165515872`: 129.9; `932835144985`: 89.0; `1035359410143`: 119.9; `1013833790937`: 289.0; `917360863329`: 88.0; `762658656208`: 88.0. All records report success.

Tool Call/Response

[14] success: true; error: null.
Item ID: `909403954726`
Title: Yunnan Dali vacation photo outfit: openwork crochet lace layered camisole dress with floral-embroidered cover-up
[... Five other products and image URLs omitted.]

Assistant

[15] I've found several good vacation styles. I'll check these new candidates' prices before filtering them.
Tool Call: `query_price_sku`
Arguments: {"item_ids": "909403954726,716188808044,907052740427", "query_type": "price"}

Tool Call/Response

[16] success: true; query type: price.
`909403954726`: 69.98; `716188808044`: 99.0; `907052740427`: 99.9. All records report success.

Assistant

[17] [...] I'll learn more about the products.
Tool Calls (3); image URLs abbreviated:
Tool Call: `item_wiki`
Arguments: {"image_url": "...", "item_id": "917360863329"}
Tool Call: `item_wiki`
Arguments: {"image_url": "...", "item_id": "916165515872"}
Tool Call: `item_wiki`
Arguments: {"image_url": "...", "item_id": "909403954726"}

Tool Call/Response

[18] item_id: 917360863329; success: true.
This dress has a distinctive design, combining camisole straps and fringe for a Bohemian style. [...]

Tool Call/Response

[19] item_id: 916165515872; success: true.
[...] Its loose silhouette suits different body shapes, while comfortable, breathable fabric is suitable for summer. [...]

Tool Call/Response

[20] item_id: 909403954726; success: true.
This white maxi dress has a sleeveless design and a relaxed vacation style. Suitable for casual beaches and resorts, its fabric looks lightweight and breathable for summer.

Assistant

[21] [...] All are under RMB 500 and offer good value.
Tool Call: `finish`
Arguments: `items`: [...]
[The five submitted product records are omitted here; the same IDs and prices appear in message 23.]

Tool Call/Response

[22] success: true; acknowledged: true.
[Repeated product records omitted.]
content: Issue Resolved

Assistant

[23] [...] I've chosen five sets, all under RMB 500. No need to plan separate pieces—just wear them in Vietnam!
Item ID: `916165515872`
Title: 1. Ethnic-pattern cover-up with floral camisole dress set
Price: RMB 129.9. [...] The loose outer layer conceals the stomach and feels breathable and comfortable. [...]
Item ID: `917360863329`
Title: 2. Vacation-style fringed layered openwork cover-up and camisole dress set
Price: RMB 88. [...] Refreshing blue works for beach and old-town photographs; the loose layered silhouette is friendly to apple-shaped figures, at excellent value.
Item ID: `909403954726`
Title: 3. Openwork crochet lace camisole dress and floral-embroidered cover-up set
Price: RMB 69.98. [...] Floral embroidery adds refinement; the loose silhouette is slimming, and the price is under RMB 70.
Item ID: `716188808044`
Title: 4. Openwork fringed camisole, long-sleeve sun-protection top and skirt three-piece set
Price: RMB 99. [...] Long sleeves are practical, while openwork and fringe create an effortless vacation look with good value.
Item ID: `907052740427`

Assistant (continued)

Title: 5. Bohemian camisole and Thai-inspired dopamine-color travel/seaside top set

Price: RMB 99.9. [. . .] Bright Bohemian dopamine colors suit photographs against colorful Vietnamese houses or at the beach, especially if you enjoy saturated, vivid images.

[Styling tips and all image URLs omitted.]

If you would like size recommendations, fabric composition, or available colors, tell me and I'll check the specifications for you!