FORT-SEARCHER: SYNTHESIZING SHORTCUT-RESISTANT SEARCH TASKS FOR TRAINING DEEP SEARCH AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Training deep-search agents requires verifiable questions that demand substantial evidence acquisition, yet structurally complex tasks may still admit cheap identifying routes when clues are overly selective, co-covered by the same evidence, directly searchable, or resolved from solver priors. We formalize this gap with a shortcut-aware difficulty framework that distinguishes intended construction complexity from identifying routes available through retrieval. Guided by this framework, we introduce FORT, a **F**ramework **o**f Shortcut-**R**esistant **T**raining-Data Synthesis, which targets four shortcut risks across entity selection, evidence graph construction, question formulation, and adversarial refinement. We evaluate realized search behavior with solver-conditioned trajectory diagnostics rather than graph complexity or search length alone. Compared with existing open-source deep-search datasets, FORT shows later answer exposure, higher retrieval effort, lower single-clue selectivity, greater evidence dispersion, and higher realized pre-answer dependency costs. Using the resulting trajectories, we train FORT-Searcher with supervised fine-tuning alone. Across model scales, FORT-Searcher consistently improves over its backbones, achieving the strongest overall performance among similarly sized agents.

1 INTRODUCTION

Large language models (LLMs) increasingly serve as tool-empowered search agents (Chen et al., 2025a; Hu et al., 2026a). Effective open-web search requires more than retrieval: agents must iteratively refine queries, locate hard-to-find information, assess evidence reliability, and connect scattered clues (Chen et al., 2025b; Qiao et al., 2025; Hu et al., 2026b; Zeng et al., 2026b). Training these agents with supervised fine-tuning (SFT) or reinforcement learning requires verifiable questions that demand search (Song et al., 2025b;a; Mei et al., 2026; Thakur et al., 2026). Yet traditional retrieval datasets often do not require long-horizon evidence acquisition (Yang et al., 2018).

To address this limitation, recent deep search task synthesis methods often increase apparent difficulty by enriching the intended solving structure, including hop count and graph shape (Sun et al., 2025; Trivedi et al., 2022; Tao et al., 2025; Team et al., 2026b), hierarchical constraints (Xia et al., 2025), and evidence dispersion or treewidth (Chu et al., 2026). However, the underlying structural priors used to generate the question-to-answer chain do not necessarily translate into the realized search process when the task is actually executed by the search agent. The agent may still bypass much of the intended evidence acquisition by leveraging evidence from the environment or its own prior knowledge. Such shortcuts may limit the training signal for long-horizon search, planning, and information synthesis.

To characterize this mismatch, we develop a shortcut-aware difficulty framework for multi-constraint agentic retrieval. It links difficulty to intended task complexity and how the question is realized by the retrieval environment and solver. Many clues and a long evidence graph need not prevent a cheaper identifying route. We call these route-level collapses, together with solver-level prior binding, *shortcuts*. Data synthesis should therefore assess whether questions with complex structures admit cheap bypasses in real search trajectories.

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.

This framework leads to four actionable shortcut risks that should be controlled during data synthesis: *evidence co-coverage*, where multiple clues can be verified from the same evidence item; *single-clue selectivity*, where one clue or a small clue subset is already enough to identify the answer; *exposed constants*, where names, strings, or numbers on the question surface make downstream queries directly executable; and *prior-knowledge binding*, where the solver names the answer before retrieval anchors it. Since these risks are only partially visible from the final question alone, we monitor their realized effects through trajectory signatures, including solving cost, answer hit time, and answer-before-evidence rate.

Guided by this framework, we introduce FORT, a **F**ramework **o**f Shortcut-**R**esistant **T**raining-Data Synthesis. FORT targets shortcut risks across the full data construction process. It selects long-tail root entities to reduce prior-knowledge binding, initializes and expands evidence graphs to avoid linearly exposed structures, collects facts from heterogeneous sources to reduce evidence co-coverage, constructs derived facts that are less likely to appear verbatim in a single source, and selects facts that are weak in isolation but identifying in combination. During question formulation, it withholds intermediate entity names and fuzzes exact constants into truthful ranges, categories, or indirect descriptions. Finally, adversarial refinement runs a strong search agent against each draft question and repairs shortcut-prone, ambiguous, or over-obfuscated cases.

Compared with existing open-source deep-search datasets, FORT exhibits later answer exposure and stronger shortcut-related factor diagnostics under the evaluated search settings. Using these data, we train FORT-Searcher with supervised fine-tuning alone, with its 35B variant achieving the strongest overall score among similarly sized agents.

Our contributions are:

- We develop a framework for multi-constraint agentic retrieval that relates realized difficulty to the cheapest identifying route and identifies four actionable shortcut risks.
- We propose FORT, which targets shortcut risks across entity selection, evidence graph construction, question formulation, and adversarial refinement.
- Experiments show stronger shortcut-related diagnostics and improved downstream performance from FORT training data.

2 Search Tasks and Shortcut Risks

2.1 Task and scope

We study *multi-constraint agentic retrieval*: an agent iteratively queries a retrieval interface (e.g., a web search engine), examines evidence, and decides when to answer. We use the shorthand

$$q = (\mathcal{X}, \mathcal{C}_q, \Sigma), \tag{1}$$

where $\mathcal{X}$ is the task-relevant space of candidate entities or determinate values, $\mathcal{C}_q$ is the set of factual constraints expressed by the question, and Σ is the retrieval interface (Wang et al., 2026a). Each constraint $c : \mathcal{X} \to \{0, 1\}$ indicates whether a candidate satisfies it. The question wording is implicit in q and determines initially available query constants. For any subset $\mathcal{P} \subseteq \mathcal{C}_q$, define

$$\mathrm{Ans}(\mathcal{P}) = \{x \in \mathcal{X} : \ c(x) = 1 \text{ for all } c \in \mathcal{P}\}. \tag{2}$$

For the formal analysis, we assume $\mathrm{Ans}(\mathcal{C}_q) = \{y^\star\}$, so the full constraint set uniquely identifies the target. The solver observes the question and has access to Σ, but not the construction graph or supporting evidence. A subset $\mathcal{P} \subseteq \mathcal{C}_q$ is *identifying* if $\mathrm{Ans}(\mathcal{P}) = \{y^\star\}$. Hence, a complex graph may still admit a short solution route if an identifying subset is cheaply retrievable.

2.2 Evidence-based difficulty and identifying routes

A complex construction graph does not by itself guarantee costly evidence acquisition. To characterize the search a task requires, we consider an *evidence-only reference solver* that issues only executable queries, using constants available from the question or previously retrieved evidence; returns an answer only after retrieved evidence verifies an identifying subset; and uses no question-specific prior knowledge. This idealized class defines a structural reference cost, rather than an implementable

Table 1: Shortcut mechanisms and representative FORT controls. Details are shown in Appendix B.

Shortcut risk	Shortcut mechanism	FORT control
Evidence co-coverage	One result from Σ supports several constraints in an identifying subset $\mathcal{P}$, allowing joint evidence acquisition.	Multi-source enrichment; derived fact construction.
Single-clue selectivity	For some $c \in \mathcal{C}_q$, $\mathrm{Ans}(\{c\})$ contains only $y^\star$ or a few candidates.	Generic fact selection; low-specificity clue formulation.
Exposed constants	Surface constants make downstream queries directly executable, bypassing their discovery.	Cycle-based initialization; name withholding; exact-value fuzzing.
Prior-knowledge binding	The solver links clues in $\mathcal{C}_q$ to $y^\star$ from prior knowledge before retrieving supporting evidence.	Long-tail root selection; adversarial refinement.

open-web policy. Let $D_{\mathrm{post}}(q)$ (the pure-posterior cost of Appendix A) denote the infimum expected number of retrieval calls, conditional on success, over reference policies with nonzero success probability. This success-conditioned cost permits rare successful runs and need not reflect the effort of reliable route discovery. Each retrieval invocation is counted separately.

Let $\mathcal{I}_q$ denote the family of identifying subsets defined by $\mathrm{Ans}(\mathcal{P}) = \{y^\star\}$. For $\mathcal{P} \in \mathcal{I}_q$, a *valid identifying route* is a sequence of executable queries whose accumulated evidence verifies $\mathcal{P}$ for $y^\star$. Write $Q_\Sigma(\mathcal{P})$ for its minimum number of retrieval calls. Two quantities further constrain this cost: $M_{\mathrm{ev}}(\mathcal{P})$ is the minimum number of calls needed to obtain evidence verifying $\mathcal{P}$ when query executability is ignored; $\mathrm{dep}(\mathcal{P})$ is the minimum dependency depth over valid routes, where a route's depth is the number of queries in its longest chain linked by required constants obtained from earlier retrievals. Formal definitions are given in Appendix A.

Proposition 2.1 (Evidence-acquisition lower bound)**.** *Under the reference-solver restrictions above,*

$$\begin{aligned} D_{\mathrm{post}}(q) \geq Q^\star_\Sigma &:= \min_{\mathcal{P}\in\mathcal{I}_q} Q_\Sigma(\mathcal{P}) \\ &\geq \min_{\mathcal{P}\in\mathcal{I}_q} \max\{M_{\mathrm{ev}}(\mathcal{P}), \mathrm{dep}(\mathcal{P})\}. \end{aligned} \tag{3}$$

If an eligible reference policy can identify and follow a cheapest valid route without exploratory retrieval, then $D_{\mathrm{post}}(q) = Q^\star_\Sigma$.

Every successful reference trajectory is a valid route for an identifying subset, costing at least $Q^\star_\Sigma$. Every valid route also incurs at least the unconstrained evidence-acquisition cost and contains its dependency chains, yielding both inequalities; Appendix A gives full proofs.

The minimum covers all identifying subsets and valid routes, including unintended ones. If verification requires at least one retrieval, an initially executable query verifying an identifying subset gives $Q^\star_\Sigma = 1$, regardless of the construction graph's size or depth (Corollary A.3). This establishes cheap-route existence; a particular solver may need additional search to discover it. Conversely, a large $Q^\star_\Sigma$ forces substantial retrieval cost on every successful reference trajectory.

2.3 Four shortcut risks

Section 2.2 motivates four actionable shortcut risks. *Single-clue selectivity* can make small subsets identifying (enlarging $\mathcal{I}_q$); *evidence co-coverage* can reduce verification calls (lowering M_{ev}); and *exposed constants* can make downstream queries executable earlier (lowering dep). These mechanisms can enable cheaper identifying routes. *Prior-knowledge binding* acts at the solver level: prior knowledge may identify the answer before the solver acquires reference-solver evidence.

Risks can overlap: an exact quotation may be both an exposed constant and a highly selective clue. Table 1 maps risks to construction controls; Appendix B gives analysis and diagnostic examples.

2.4 OBSERVABLE TRAJECTORY DIAGNOSTICS

To examine how the shortcut risks in Section 2.3 manifest during search, we define three observable diagnostics for a successful trajectory τ under a fixed solver, retrieval interface Σ, and search budget.

Solving cost. Solving cost counts retrieval invocations separately, including multiple calls within one assistant interaction:

$$\Omega(\tau) = |\tau|. \tag{4}$$

Large $\Omega(\tau)$ indicates substantial retrieval effort, but an answer may appear early, followed by verification or detours. We therefore also measure when the answer first appears.

Answer hit time. Let $T_{\text{tool}}(\tau)$ and $T_{\text{model}}(\tau)$ denote the first assistant round in which the gold answer y^* or a normalized alias appears in retrieved evidence and visible model text, respectively, with behavior before the first retrieval round assigned index 0. Absent occurrences have time and event-order position $+\infty$. We define

$$T_{\text{hit}}(\tau) = \min\{T_{\text{tool}}(\tau), T_{\text{model}}(\tau)\}. \tag{5}$$

Unlike Ω, which counts retrieval calls, T_{hit} measures earliest answer exposure in assistant rounds, not completed verification or sufficient evidence. Evidence co-coverage, single-clue selectivity, and exposed constants can expose answers earlier through cheaper identifying routes; I_{prior} separately captures model-side answer-before-evidence behavior. The positive correlation between T_{hit} and realized pre-answer dependency cost (Appendix D.4.1) indicates that later exposure tends to accompany costlier dependencies. We therefore use T_{hit} as an observable diagnostic, not a direct estimate of minimum route cost.

Answer-before-evidence indicator. To capture observable answer-before-evidence behavior associated with solver-side prior binding, we record whether the model names the gold answer before it first appears in retrieved evidence:

$$I_{\text{prior}}(\tau) = \mathbf{1}\left[\text{pos}_{\text{model}}(y^*) < \text{pos}_{\text{tool}}(y^*)\right], \tag{6}$$

where pos follows trajectory event order. This behavioral proxy does not directly reveal internal prior knowledge; human annotation validates its correspondence to shortcut-like answer-before-evidence behavior (Appendix D.4.2). Successful trajectories that name the answer before retrieving supporting evidence may cost less than the no-prior reference solver.

For dataset-level comparisons, we report solver accuracy over all evaluated questions and compute $\widehat{\Omega}$, T_{hit}, and $\widehat{p}_{\text{prior}}$ over all successful trajectories. Accuracy measures overall solvability, while the trajectory diagnostics characterize search behavior conditional on success.

To compare realized answer exposure across open-source deep-search datasets, we evaluate all datasets under the same solver and retrieval budget. Figure 3a shows that high retrieval cost can still coexist with early answer exposure. For example, OpenSeeker has $\widehat{\Omega} = 50.2$, but $T_{\text{hit}} = 2.5$ and $\widehat{p}_{\text{prior}} = 20.0\%$. This motivates shortcut-aware controls beyond retrieval effort alone.

3 FORT: SHORTCUT-RESISTANT TASK SYNTHESIS

FORT synthesizes search tasks through four stages targeting the shortcut risks in Section 2.3: entity selection and graph initialization, evidence graph construction, question formulation, and adversarial refinement, as shown in Figure 1. The construction graph $\mathcal{G} = (V, E)$ represents entities as nodes and verified relations as edges, providing sourced evidence for question formulation (Wu et al., 2025; Sun et al., 2025). Adversarial search then evaluates each draft using the diagnostics in Section 2.4.

3.1 ENTITY SELECTION AND GRAPH INITIALIZATION

We select a root entity r from broad Wikidata semantic categories, excluding abstract concepts and preferring long-tail entities, especially those without an English Wikipedia page. A lightweight pre-search removes entities with insufficient external evidence.

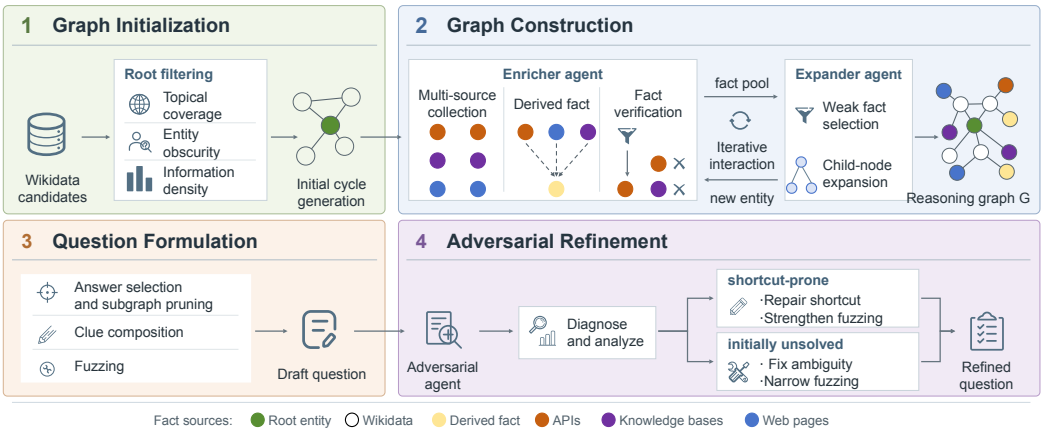


Figure 1: Overview of FORT. A selected entity seeds an evidence graph, which is expanded through fact enrichment and selection. A formulation agent turns a selected subgraph into a question, and adversarial refinement revises shortcuts or overly broad clues.

When available, a pre-mined Wikidata cycle containing r initializes $\mathcal{G}_0$; otherwise, we use $(\{r\}, \varnothing)$. Cycle initialization supports indirect relational descriptions while hiding intermediate entity names, reducing exposed-constant shortcuts. Appendix C.1 details the cycle library.

3.2 Evidence Graph Construction

Starting from $\mathcal{G}_0$, the enricher and expander construct the evidence graph under depth limit D and node budget B. The enricher collects, derives, and verifies facts, while the expander selects facts and promotes referenced entities for further expansion. Algorithm 1 in Appendix C.2 gives the procedure.

Multi-source enrichment and derived facts. To reduce evidence co-coverage, the enricher collects facts from heterogeneous sources, including Wikidata, open web pages, structured databases, Google Scholar, and Google Maps, and distributes root facts across sources where possible. Non-root nodes sample sources for efficiency, since only one or two facts are typically selected per node.

The enricher also constructs derived facts from multiple atomic facts or source texts, making the resulting relations less likely to appear verbatim in a single retrieved item (Shen et al., 2025). For example, separately retrieved records that artists A and B performed at venue V yield the relation that they performed at the same venue. The four constructors are coincidence bridging, count aggregation, numerical relation, and meta-fact extraction; Appendix C.3 gives their definitions and examples.

Fact verification and selection. Candidate facts are checked for source and entity consistency (Cui et al., 2025), rejecting conflicts with retrieved evidence, accepted graph facts, or mismatched entities. From the verified pool, an LLM-based selector favors facts that are weakly identifying in isolation but useful for further expansion. Their combination remains informative while reducing single-clue shortcuts. Appendix C.3 provides details and examples.

3.3 Question Formulation

The formulation agent selects an answer node $y^\star$ and subgraph $\mathcal{G}^* \subseteq \mathcal{G}$, removes redundant facts, and verbalizes clues that are jointly identifying but individually generic. Intermediate entity names are replaced with referring expressions such as "the artist," "the institution," or "the work," limiting direct access to queries that require those names.

For remaining names and literal values, *exact-value fuzzing* replaces an explicit constant with a truthful, less directly searchable description (Zhang et al., 2025). For example, 1863 can become "the second half of the nineteenth century." The agent chooses among category generalization, range relaxation, meta-attribute description, arithmetic encoding, and contrastive exclusion according to

the value's semantic type, distinctiveness, and role in the question. The complete question should preserve the same gold answer and remain verifiable. Appendix C.4 details the five transformations.

Following PostTrainBench (Rank et al., 2026), we apply n-gram filtering and remove flagged examples, totaling approximately 0.01% of the data.

3.4 Adversarial Refinement

We evaluate each draft using Section 2.4. Drafts below the retrieval-cost threshold τ_{th} are refined as shortcut-prone, while unsolved drafts are repaired for ambiguity or excessive fuzziness. The threshold triggers refinement but does not filter the final dataset.

For shortcut-prone drafts, we replace co-covered evidence, remove overly selective facts, or further withhold or fuzz exposed constants (Anantha et al., 2025; Kazoom et al., 2025). Answer-before-evidence cases trigger root or evidence-path revision. For unsolved drafts, we narrow over-fuzzed clues, remove ambiguity, or restore necessary constraints. Revised drafts are reevaluated under the same search setting. Appendix C.5 provides complete examples.

4 Experiments

4.1 Experimental setup

We train FORT-Searcher with SFT on search trajectories generated from FORT tasks using DeepSeek-V4-Pro-0813 as the teacher. Qwen3.6-35B-A3B (Qwen Team, 2026b) is the primary backbone, with Qwen3.5-9B and Qwen3.5-4B (Qwen Team, 2026a) used for scaling analysis.

We evaluate on BrowseComp (Wei et al., 2025), BrowseComp-ZH (Zhou et al., 2025), xbench-DeepSearch (Chen et al., 2025c), and WideSearch (Wong et al., 2026). We compare against large-scale and 30B-scale search agents, following official evaluation protocols when available and otherwise using the same evaluation harness and budget. Full training, evaluation, and context-management settings are provided in Appendix D.1.

4.2 Main Results

Table 2 compares FORT-Searcher with existing search agents on four benchmarks. FORT-Searcher-35B-A3B achieves the highest overall score among the 30B-scale agents, while the 9B and 4B variants consistently outperform their corresponding Qwen3.5 backbones, showing that FORT-generated trajectories provide effective supervision.

For Table 2, context management is enabled for BrowseComp and BrowseComp-ZH and disabled for xbench-DeepSearch and WideSearch. Figure 2a further disentangles training from context management by evaluating both the original backbones and FORT-Searcher without context management. Even without context management, FORT training improves BrowseComp by 5.3–18.9 points and BrowseComp-ZH by 16.4–25.8 points across model scales. Context management provides additional gains to FORT-Searcher at all three scales (Tang et al., 2025; Yen et al., 2025).

Since most baselines do not fully disclose their training data, dataset sizes, or training procedures, a fully controlled system-level comparison is not feasible. To improve comparability, we compare models under matched inference settings and budgets in Figure 2b, and additionally train FORT-Searcher on the same backbones as corresponding baselines in Appendix D.2. Under matched inference budgets, FORT-Searcher achieves the strongest performance on BrowseComp at medium-to-high retrieval budgets. Under same-backbone comparisons, its advantage also persists across backbone variants. These complementary controls support the robustness of the observed gains.

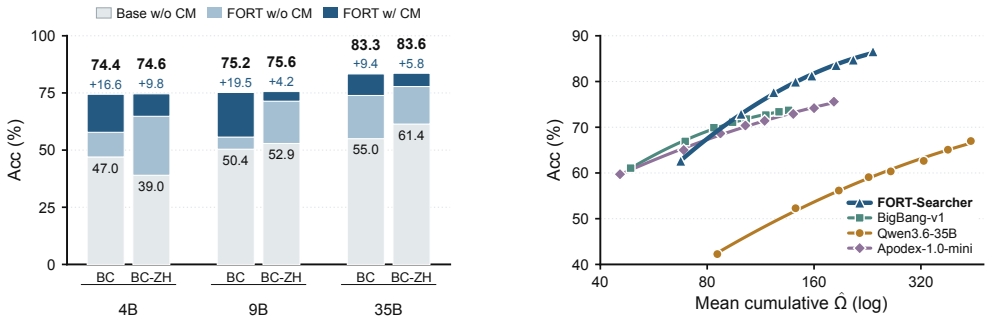
4.3 Further Analysis

4.3.1 Dataset Difficulty Comparison

Dataset-level difficulty. To compare shortcut-related realized search behavior across deep-search datasets, we compare FORT with InfoSeek (Xia et al., 2025), DeepResearch-9K (Wu et al., 2026b),

Table 2: Performance comparison on search benchmarks. Overall denotes the average score across available benchmarks. Bold values indicate the best results within each group.

Model	BrowseComp	BC-ZH	xbench	WideSearch	Overall
Large-scale Agents					
DeepSeek-V4-Pro-0813	84.4	**85.3**	84.1	79.1	83.2
DeepSeek-V4-Flash-0731	80.6	81.0	**89.0**	80.6	82.8
GLM-5.3	75.3	74.1	83.3	61.9	73.7
Kimi-K3	**91.2**	81.7	86.3	78.3	**84.4**
Qwen3.8-Max	80.5	82.1	84.9	**81.6**	82.3
30B-scale Agents					
REDSearcher	57.4	58.2	-	-	-
Apodex-1.0-mini	71.5	80.6	80.0	67.3	74.9
BigBang-v1	76.5	69.0	86.3	50.2	70.5
Qwen3.6-35B-A3B	62.9	63.2	82.7	62.3	67.8
FORT-Searcher-35B-A3B	**83.3**	**83.6**	**87.0**	**74.1**	**82.0**
9B-scale Agents					
Qwen3.5-9B	57.4	55.4	78.9	58.4	62.5
FORT-Searcher-9B	**75.2**	**75.6**	**82.7**	**63.5**	**74.3**
4B-scale Agents					
Qwen3.5-4B	53.2	40.3	78.0	53.3	56.2
FORT-Searcher-4B	**74.4**	**74.6**	**81.0**	**57.9**	**72.0**



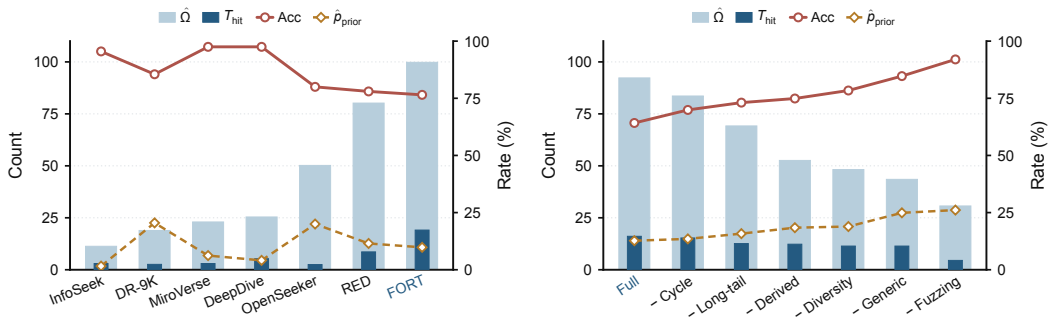
(a) Training and context management.

(b) BrowseComp accuracy versus $\widehat{\Omega}$.

Figure 2: FORT training and context-management effects, and BrowseComp retrieval-budget scaling.

MiroVerse-Voyager (Team et al., 2025), DeepDive (Lu et al., 2025), OpenSeeker (Du et al., 2026), and REDSearcher (Chu et al., 2026). For each dataset, we randomly sample 200 questions and evaluate them with DeepSeek-V4-Pro-0813 under the same retrieval settings and budget. Accuracy is computed over all 200 questions, while trajectory diagnostics are computed over successful runs. These diagnostics measure realized search behavior under the evaluated solver and retrieval interface rather than the oracle minimum identifying-route cost. To assess solver dependence, we repeat the analysis with the weaker Qwen3.5-397B-A17B (Qwen Team, 2026a), with the same overall pattern reported in Appendix D.3.

Figure 3a shows that FORT yields the highest solving cost and latest answer exposure under the same diagnostic setting. Compared with REDSearcher, FORT maintains comparable solver accuracy while exhibiting both higher $\widehat{\Omega}$ and later T_{hit}, showing that higher retrieval effort is accompanied by later answer exposure rather than simply lower solvability. Meanwhile, $\widehat{p}_{\text{prior}}$ remains comparable.



(a) Dataset comparison.

(b) Ablation on shortcut-resistant components.

Figure 3: Solver accuracy and trajectory diagnostics for dataset comparison and cumulative ablation. For dataset comparison, accuracy is computed on 200 randomly sampled questions per source and trajectory diagnostics on the successful subset. For ablation, accuracy uses all questions and trajectory diagnostics use successful trajectories. Each ablation removes one additional component.

Table 3: Training results under different settings.

Data settings			Result
$\widehat{\Omega}$	T_{hit}	$\widehat{p}_{\text{prior}}(\%)$	BrowseComp
30.0	4.3	8.7	55.7
65.0	7.6	13.4	59.7
100.0	10.7	16.1	60.7
100.0	**18.9**	**9.2**	**63.9**

Table 4: Effect of adversarial refinement.

Draft settings		Results		
Draft type	Version	$\widehat{\Omega}$	$\overline{T}_{\text{hit}}$	$\widehat{p}_{\text{prior}}(\%)$
Shortcut-prone	Original	28.7	5.4	15.7
	Refined	**55.6**	**12.1**	**12.4**
Initially unsolved	Original	–	–	–
	Refined	87.2	21.2	1.7

Since T_{hit} and $\widehat{p}_{\text{prior}}$ are success-conditioned, we additionally examine $\widehat{\Omega}$ on failed runs. Compared with REDSearcher, the strongest baseline under the same diagnostic setting, FORT remains more costly on failed trajectories, averaging 169.5 versus 135.2 retrieval calls. Thus, the higher retrieval effort is not confined to successful trajectories. Appendix D.4.1 further shows lower single-clue selectivity, more dispersed evidence, and higher realized pre-answer dependency costs.

Training-data difficulty. To examine whether FORT's training value can be explained by retrieval cost alone, we construct four 3K-trajectory training sets. Because matching distributions across synthesis pipelines is infeasible, we improve comparability by keeping the teacher, search environment, trajectory count, distillation pipeline, backbone, and SFT recipe fixed. Three sets are sampled from the pooled collection of six open-source datasets above, with average solving costs of 30, 65, and 100 retrieval calls. The fourth uses FORT data with a matched $\widehat{\Omega} = 100.0$. As shown in Table 3, increasing the average retrieval cost of open-source training data from 30.0 to 100.0 improves BrowseComp from 55.7 to 60.7. At the matched retrieval cost, training on FORT further improves performance to 63.9, showing that retrieval cost alone does not explain the observed gap.

4.3.2 Adversarial Refinement

To evaluate whether adversarial refinement calibrates question difficulty, we consider two failure modes: *shortcut-prone* drafts with early identifying routes or prior binding, and *initially unsolved* drafts affected by excessive fuzziness or ambiguity. We refine both groups and rerun the adversary under the same search setting. Refinement resolves 70% of shortcut-prone drafts and yields correct solutions for 40% of initially unsolved drafts, measured over all submitted drafts in each group.

As shown in Table 4, shortcut-prone drafts increase from $\widehat{\Omega} = 28.7$ to 55.6 and from $T_{\text{hit}} = 5.4$ to 12.1, while $\widehat{p}_{\text{prior}}$ decreases from 15.7 to 12.4. Refined initially unsolved drafts achieve $\widehat{\Omega} = 87.2$

and $T_{\text{hit}} = 21.2$, indicating harder realized search behavior under the evaluated adversarial search setting. To assess whether increased difficulty comes at the expense of task validity, we manually audit 150 FORT questions. All final groups retain at least 94% answer correctness and 92% source support. Questions accepted without refinement achieve 94% uniqueness and 92% unambiguous formulation, while refinement of initially unsolved drafts improves these rates from 72% to 90% and from 74% to 88%, respectively. Appendix D.5 provides the full protocol and results.

4.3.3 Shortcut-Resistance Ablation

To examine the effect of progressively removing shortcut-resistant controls, we conduct a cumulative ablation on 3000 questions. Each configuration removes one additional component from the preceding one under the same evaluation setting.

Figure 3b shows consistent shifts toward easier realized search behavior as controls are removed: accuracy rises from 64.2 to 92.0, while $\widehat{\Omega}$ decreases from 92.3 to 30.7, T_{hit} from 16.1 to 4.5, and $\widehat{p}_{\text{prior}}$ increases from 12.7 to 26.1. Under matched training settings, BrowseComp performance also decreases from 61.9 for the full configuration to 58.1 and 54.9 for two representative cumulative ablations, indicating that configurations with fewer shortcut-resistant controls also provide less effective training data. Full downstream results are reported in Appendix D.6.

Because the ablation is cumulative, these results characterize the joint effect of the controls. We therefore additionally evaluate isolated removals of long-tail selection, derived facts, and fuzzing in Appendix D.6, providing component-level evidence for three representative controls.

5 Related Work

Deep-search task synthesis. Recent work synthesizes multi-step search tasks from knowledge graphs, web evidence, and agent exploration (Lu et al., 2025; Xia et al., 2025; Du et al., 2026; Wu et al., 2026a; Li et al., 2025). QUEST, SearchArt, and ORBIT further improve the scale and verification of synthetic search data (Xie et al., 2026; Mei et al., 2026; Thakur et al., 2026). Recent methods also place more emphasis on controlling shortcuts during construction. LoHoSearch explicitly controls candidate-space size and structural complexity (Zhao et al., 2026), while Evo-BrowseComp removes redundant paths from the constructed reasoning graph (Wang et al., 2026b). SearchArt further reduces direct retrieval and parametric shortcuts through long-tail entities and dispersed evidence (Mei et al., 2026). These approaches make the intended reasoning structure harder to bypass, but they mainly characterize difficulty from properties of the constructed task. In contrast, FORT asks whether this intended difficulty survives once the task is exposed through a concrete retrieval interface. It explicitly targets low-cost identifying routes caused by evidence co-coverage, selective clues, exposed constants, and model priors.

Search difficulty and execution. Observed search difficulty depends on the environment, retrieval interface, and agent execution. SearchGym and DeepSearch-World study training in verifiable search environments (Zhang et al., 2026a; Geng et al., 2026). SIEVE shows that interface expressiveness changes evidence access (Wang et al., 2026a), while dynamic context management affects performance over long search trajectories (Tang et al., 2025). Trajectory analyses show that search length alone reveals neither when useful evidence is acquired nor whether it is effectively used (Liu et al., 2026). These works primarily study system or agent effects on performance. Under a concrete solver and retrieval setting, FORT uses solving cost, answer-exposure timing, and prior-answer behavior to characterize realized search and identify and repair residual shortcuts in synthesized questions.

6 Conclusion

We introduce a shortcut-aware difficulty framework and FORT, targeting four shortcut risks throughout synthesis and adversarial refinement. Under the evaluated settings, FORT shows later answer exposure, lower single-clue selectivity, greater evidence dispersion, and higher realized pre-answer dependency costs than open-source datasets. SFT on FORT trajectories yields the strongest overall performance among the compared 30B-scale agents and retains an advantage at matched retrieval cost, with the same overall pattern across diagnostic solvers.

AI USE STATEMENT

Generative language models are used within the proposed research pipeline for task synthesis, adversarial solving, and trajectory-based analysis. An AI assistant also contributed to literature collection and manuscript revision. The authors manually reviewed all AI-assisted content and remain fully responsible for the final manuscript.

REFERENCES

Raviteja Anantha, Soheil Hor, Teodor Nicola Antoniu, and Layne C Price. Nanoflux: Adversarial dual-llm evaluation and distillation for multi-domain reasoning. *arXiv preprint arXiv:2509.23252*, 2025.

BIGBANG team. Bigbang: Pursuing open-ended intelligence through self-evolving synthesis of verifiable frontier tasks. Technical report, 2026. URL GitHub-endless-frontier/BigBang-v1. Preprint. Technical Report.

Guoxin Chen, Minpeng Liao, Peiying Yu, Dingmin Wang, Zile Qiao, Chao Yang, Xin Zhao, and Kai Fan. C-3PO: Compact plug-and-play proxy optimization to achieve human-like retrieval-augmented generation. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=hlpwAmQ4wr.

Guoxin Chen, Zile Qiao, Xuanzhong Chen, Donglei Yu, Haotian Xu, Wayne Xin Zhao, Ruihua Song, Wenbiao Yin, Huifeng Yin, Liwen Zhang, et al. Iterresearch: Rethinking long-horizon agents via markovian state reconstruction. *arXiv e-prints*, pp. arXiv–2511, 2025b.

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, et al. xbench: Tracking agents productivity scaling with profession-aligned real-world evaluations. *arXiv preprint arXiv:2506.13651*, 2025c.

Zheng Chu, Xiao Wang, Jack Hong, Huiming Fan, Yuqi Huang, Yue Yang, Guohai Xu, Chenxiao Zhao, Cheng Xiang, Shengchao Hu, et al. Redsearcher: A scalable and cost-efficient framework for long-horizon search agents. *arXiv preprint arXiv:2602.14234*, 2026.

Zikun Cui, Tianyi Huang, Chia-En Chiang, and Cuiqianhe Du. Toward verifiable misinformation detection: A multi-tool llm agent framework. In *Proceedings of the 2025 International Conference on Generative Artificial Intelligence for Business*, pp. 179–185, 2025.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

Yuwen Du, Rui Ye, Shuo Tang, Xinyu Zhu, Yijun Lu, Yuzhu Cai, and Siheng Chen. Openseeker: Democratizing frontier search agents by fully open-sourcing training data. *arXiv preprint arXiv:2603.15594*, 2026.

Xinyu Geng, Xuanhua He, Sixiang Chen, Yanjing Xiao, Fan Zhang, Shijue Huang, Haitao Mi, Zhenwen Liang, Tianqing Fang, and Yi R Fung. Deepsearch-world: Self-distillation for deep search agents in a verifiable environment. *arXiv preprint arXiv:2607.07820*, 2026.

Yuyang Hu, Hongjin Qian, Shuting Wang, Jiongnan Liu, Tong Zhao, Xiaoxi Li, Zheng Liu, and Zhicheng Dou. Agentfugue: Agent scaling for long-horizon tasks through collective reasoning, 2026a. URL https://arxiv.org/abs/2605.24486.

Yuyang Hu, Hongjin Qian, Shuting Wang, Jiongnan Liu, Ziliang Zhao, Jiejun Tan, Zheng Liu, and Zhicheng Dou. Sam: State-adaptive memory for long-horizon reasoning agent, 2026b. URL https://arxiv.org/abs/2605.24468.

Roie Kazoom, Ofir Cohen, Rami Puzis, Asaf Shabtai, and Ofer Hadar. Vault: Vigilant adversarial updates via llm-driven retrieval-augmented generation for nli. *arXiv preprint arXiv:2508.00965*, 2025.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*, 2025.

Qi Liu, Jiaxin Mao, Fengbin Zhu, and Tat-Seng Chua. Diagnosing search behavior and failure modes in long-horizon search agents. *arXiv preprint arXiv:2608.01913*, 2026.

Rui Lu, Zhenyu Hou, Zihan Wang, Hanchen Zhang, Xiao Liu, Yujiang Li, Shi Feng, Jie Tang, and Yuxiao Dong. Deepdive: Advancing deep search agents with knowledge graphs and multi-turn rl. *arXiv preprint arXiv:2509.10446*, 2025.

Lang Mei, Xiaohan Yu, Chong Chen, Liyan Liu, Xiangnan Chen, Jinchao Ma, Chao Feng, Li Huang, Siyu Mo, Sichen Kang, et al. Searchart: Training long-horizon search agent with scalable synthetic and verified task. *arXiv preprint arXiv:2607.24850*, 2026.

Thinh Pham, Nguyen Nguyen, Pratibha Zunjare, Weiyuan Chen, Yu-Min Tseng, and Tu Vu. Sealqa: Raising the bar for reasoning in search-augmented language models. *arXiv preprint arXiv:2506.01062*, 2025.

Zile Qiao, Guoxin Chen, Xuanzhong Chen, Donglei Yu, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, Rui Min, Minpeng Liao, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents, 2025. URL https://arxiv.org/abs/2509.13309.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026b. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026c. URL https://qwen.ai/blog?id=qwen3.8.

Ben Rank, Hardik Bhatnagar, Ameya Prabhu, Shira Eisenberg, Karina Nguyen, Matthias Bethge, and Maksym Andriushchenko. Posttrainbench: Can llm agents automate llm post-training? *arXiv preprint arXiv:2603.08640*, 2026.

Zhiyu Shen, Jiyuan Liu, Yunhe Pang, and Yanghui Rao. Hopweaver: Cross-document synthesis of high-quality and authentic multi-hop questions. *arXiv preprint arXiv:2505.15087*, 2025.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in LLMs via reinforcement learning, 2025a. URL https://arxiv.org/abs/2503.05592.

Huatong Song, Jinhao Jiang, Wenqing Tian, Zhipeng Chen, Yuhuan Wu, Jiahao Zhao, Yingqian Min, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. Smart-Searcher: Incentivizing the dynamic knowledge acquisition of LLMs via reinforcement learning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 13572–13586, Suzhou, China, November 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.731. URL https://aclanthology.org/2025.findings-emnlp.731/.

Shuang Sun, Huatong Song, Yuhao Wang, Ruiyang Ren, Jinhao Jiang, Junjie Zhang, Fei Bai, Jia Deng, Xin Zhao, Zheng Liu, Lei Fang, Zhongyuan Wang, and Ji-Rong Wen. Simpledeepsearcher: Deep information seeking via web-powered reasoning trajectory synthesis. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 13705–13720. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-emnlp.739/.

Qiaoyu Tang, Hao Xiang, Le Yu, Bowen Yu, Yaojie Lu, Xianpei Han, Le Sun, WenJuan Zhang, Pengbo Wang, Shixuan Liu, et al. Beyond turn limits: Training deep search agents with dynamic context window. *arXiv preprint arXiv:2510.08276*, 2025.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, et al. Webshaper: Agentically data synthesizing via information-seeking formalization. *arXiv preprint arXiv:2507.15061*, 2025.

Apodex Team. Apodex-1.0: A verification-centric agent team for discoverative intelligence. 2026.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026a.

MiroMind Team, Song Bai, Lidong Bing, Carson Chen, Guanzheng Chen, Yuntao Chen, Zhe Chen, Ziyi Chen, Jifeng Dai, Xuan Dong, et al. Mirothinker: Pushing the performance boundaries of open-source research agents via model, context, and interactive scaling. *arXiv preprint arXiv:2511.11793*, 2025.

MiroMind Team, S Bai, L Bing, L Lei, R Li, X Li, X Lin, E Min, L Su, B Wang, et al. Mirothinker-1.7 & h1: Towards heavy-duty research agents via verification. *arXiv preprint arXiv:2603.15726*, 2026b.

Nandan Thakur, Zijian Chen, Xueguang Ma, and Jimmy Lin. Orbit: Scalable and verifiable data generation for search agents on a tight budget. *arXiv preprint arXiv:2604.01195*, 2026.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Shuai Wang, Haodong Chen, Yu Yin, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. Search, inspect, fetch: Exploiting boolean retrieval for deep-research agents. *arXiv preprint arXiv:2608.02751*, 2026a.

Yunhan Wang, Jiaan Wang, Lianzhe Huang, Xianfeng Zeng, and Fandong Meng. Evobrowsecomp: Benchmarking search agents on evolving knowledge. *arXiv preprint arXiv:2606.13120*, 2026b.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

Ryan Wong, Jiawei Wang, Li Chen, Yan Gao, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, Wenhao Huang, Yang Wang, et al. Widesearch: Benchmarking agentic broad info-seeking. In *International Conference on Learning Representations*, volume 2026, pp. 10012–10086, 2026.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhenglin Wang, Zhengwei Tao, Ding-Chu Zhang, Zekun Xi, Robert Tang, et al. Webdancer: Towards autonomous information seeking agency. *Advances in Neural Information Processing Systems*, 38:120957–120985, 2026a.

Tongzhou Wu, Yuhao Wang, Xinyu Ma, Xiuqiang He, Shuaiqiang Wang, Dawei Yin, and Xiangyu Zhao. Deepresearch-9k: A challenging benchmark dataset of deep-research agent. *arXiv preprint arXiv:2603.01152*, 2026b.

Xiaojun Wu, Cehao Yang, Xueyuan Lin, Chengjin Xu, Xuhui Jiang, Yuanliang Sun, Hui Xiong, Jia Li, and Jian Guo. Think-on-graph 3.0: Efficient and adaptive llm reasoning on heterogeneous graphs via multi-agent dual-evolving context retrieval. *arXiv preprint arXiv:2509.21710*, 2025.

Ziyi Xia, Kun Luo, Hongjin Qian, and Zheng Liu. Open data synthesis for deep research. *arXiv preprint arXiv:2509.00375*, 2025.

Jian Xie, Tianhe Lin, Zilu Wang, Yuting Ning, Yuekun Yao, Tianci Xue, Zhehao Zhang, Zhongyang Li, Kai Zhang, Yufan Wu, et al. Quest: Training frontier deep research agents with fully synthetic tasks. *arXiv preprint arXiv:2605.24218*, 2026.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

H Yen, A Paranjape, M Xia, T Venkatesh, J Hessel, D Chen, and Y Zhang. Lost in the maze: Overcoming context limitations in long-horizon agentic search. *arXiv preprint arXiv:2510.18939*, 2025. URL https://arxiv.org/abs/2510.18939v1.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026a.

Ziyang Zeng, Heming Jing, Jindong Chen, Xiangli Li, Hongyu Liu, Yixuan He, Zhengyu Li, Yige Sun, Zheyong Xie, Yuqing Yang, et al. Optimizing generative ranking relevance via reinforcement learning in xiaohongshu search. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2551–2561, 2026b.

Gongrui Zhang, Jialiang Zhu, Ruiqi Yang, Kai Qiu, Miaosen Zhang, Zhirong Wu, Qi Dai, Bei Liu, Chong Luo, Zhengyuan Yang, et al. Infoagent: Advancing autonomous information-seeking agents. *arXiv preprint arXiv:2509.25189*, 2025.

Xichen Zhang, Ziyi He, Yinghao Zhu, Sitong Wu, Shaozuo Yu, Meng Chu, Wenhu Zhang, Haoru Tan, and Jiaya Jia. Searchgym: Bootstrapping real-world search agents via cost-effective and high-fidelity environment simulation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18633–18665, 2026a.

Zhen Zhang, Liangcai Su, Zhuo Chen, Xiang Lin, Haotian Xu, Simon Shaolei Du, Kaiyu Yang, Bo An, Lidong Bing, and Xinyu Wang. Argus: Evidence assembly for scalable deep research agents. *arXiv preprint arXiv:2605.16217*, 2026b.

Jiarui Zhao, Rongzhi Zhang, Lingchuan Liu, Hao Yang, Xunliang Cai, and Xi Su. Lohosearch: Benchmarking long-horizon search agents beyond the human difficulty ceiling. *arXiv preprint arXiv:2606.12837*, 2026.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, et al. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*, 2025.

A Difficulty Framework and Proofs

This appendix consolidates the definitions and proofs underlying Section 2. We define the evidence-only reference solver and valid identifying routes, prove the structural and component lower bounds, and establish the one-query collapse case. We then relate the reference cost to the cost of a concrete solver.

A.1 Task and identifying subsets

We use the task shorthand $q = (\mathcal{X}, \mathcal{C}_q, \Sigma)$ and answer sets $\mathrm{Ans}(\mathcal{P})$ defined in Equations (1)–(2). The full constraint set uniquely identifies the gold answer: $\mathrm{Ans}(\mathcal{C}_q) = \{y^\star\}$. A subset is *identifying* if it already determines this answer. Write

$$\mathcal{I}_q = \{\mathcal{P} \subseteq \mathcal{C}_q : \mathrm{Ans}(\mathcal{P}) = \{y^\star\}\} \tag{7}$$

for the family of identifying subsets. A solver need not verify every constraint in the question if it can acquire evidence for a subset in $\mathcal{I}_q$.

A.2 Evidence-only reference solver

Queries and verification. Let Θ_Σ denote the queries supported by the retrieval interface, and let $\mathrm{Retrieve}_\Sigma(\theta)$ denote the evidence returned by query θ. Evidence items may be pages, snippets, or records. For accumulated evidence E, write

$$V_E = \{(x, c) \in \mathcal{X} \times \mathcal{C}_q : E \text{ verifies } c(x) = 1\}. \tag{8}$$

Thus, V_E records the candidate–constraint pairs verified by the retrieved evidence.

Executable queries. Let h contain the question and preceding queries and observations. Write $\mathcal{K}(h)$ for the constants available from the question and previously retrieved evidence, and $\mathrm{Const}(\theta)$ for those needed to instantiate query θ. The executable queries are

$$\mathrm{Exec}(h) = \{\theta \in \Theta_\Sigma : \mathrm{Const}(\theta) \subseteq \mathcal{K}(h)\}. \tag{9}$$

A reference policy issues only queries in $\mathrm{Exec}(h)$. For example, a query using an advisor's name requires that name to be available from the question or earlier evidence.

No-guessing termination. A reference policy may return x only after its accumulated evidence verifies an identifying subset for x. Formally, there must exist $\mathcal{P} \subseteq \mathcal{C}_q$ such that

$$\mathrm{Ans}(\mathcal{P}) = \{x\}, \qquad \{x\} \times \mathcal{P} \subseteq V_E. \tag{10}$$

Since $y^\star$ satisfies every constraint in $\mathcal{C}_q$, it also satisfies every constraint in $\mathcal{P}$. The singleton condition therefore implies $x = y^\star$.

No problem-specific prior knowledge. The policy may use basic reading and composition over retrieved evidence, but not entity-level prior knowledge specific to the question. In particular, it cannot name the target from memory or use a memorized target profile to bypass evidence acquisition.

Let Π_{post} denote the policies satisfying these restrictions and having nonzero probability of successfully solving q. We use $|\tau|$ to denote the total number of retrieval calls in trajectory τ, counting each retrieval invocation separately. The pure-posterior cost is

$$D_{\mathrm{post}}(q) = \inf_{\pi \in \Pi_{\mathrm{post}}} \mathbb{E}_{\tau \sim \pi} \left[|\tau| \mid \tau \text{ succeeds} \right]. \tag{11}$$

It measures the minimum expected retrieval cost over successful trajectories of the no-prior, no-guessing reference solver. An empty feasible-set minimum or infimum is assigned $+\infty$.

A.3 Valid routes and the structural lower bound

For an evidence-acquisition route $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_m)$, its cost is the number of retrieval calls, $|\boldsymbol{\theta}| = m$. For $\mathcal{P} \in \mathcal{I}_q$, an evidence-acquisition route is an ordered query sequence $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_m)$. Let h_{j-1}

contain the question and observations preceding query θ_j. The route is valid for $(y^\star, \mathcal{P})$ if every query is executable when issued,

$$\theta_j \in \mathrm{Exec}(h_{j-1}), \qquad j = 1, \ldots, m, \tag{12}$$

and the accumulated evidence verifies the subset:

$$\{y^\star\} \times \mathcal{P} \subseteq V_E, \qquad E = \bigcup_{j=1}^{m} \mathrm{Retrieve}_\Sigma(\theta_j). \tag{13}$$

Define the minimum route cost as

$$Q_\Sigma(\mathcal{P}) = \min_{\boldsymbol{\theta} \text{ valid for } (y^\star, \mathcal{P})} |\boldsymbol{\theta}|. \tag{14}$$

Unlike a realized trajectory τ, this route is an ideal reference for a fixed target and constraint subset. A solver searching adaptively may issue additional exploratory or redundant queries. Minimizing over identifying subsets gives

$$Q^\star_\Sigma = \min_{\mathcal{P} \in \mathcal{I}_q} Q_\Sigma(\mathcal{P}). \tag{15}$$

The minimum concerns all identifying subsets, rather than only the intended route in the construction graph.

Proposition A.1 (Structural lower bound)**.** *Under the reference-solver restrictions above,*

$$D_{\mathrm{post}}(q) \geq Q^\star_\Sigma. \tag{16}$$

Proof. Consider a trajectory generated by a policy $\pi \in \Pi_{\mathrm{post}}$ that returns an answer. By the no-guessing restriction, it returns $y^\star$ only after retrieving evidence that verifies some $\mathcal{P} \in \mathcal{I}_q$. Every query is executable when issued. Its query sequence is therefore a valid route for $(y^\star, \mathcal{P})$, and

$$|\tau| \geq Q_\Sigma(\mathcal{P}) \geq \min_{\mathcal{P}' \in \mathcal{I}_q} Q_\Sigma(\mathcal{P}') = Q^\star_\Sigma. \tag{17}$$

Taking the expectation over successful reference-solver trajectories preserves this lower bound. Taking the infimum over Π_{post} then gives

$$D_{\mathrm{post}}(q) = \inf_{\pi \in \Pi_{\mathrm{post}}} \mathbb{E}_{\tau \sim \pi} \left[|\tau| \mid \tau \text{ succeeds} \right] \geq Q^\star_\Sigma.$$

□

A sufficient condition for equality. If $Q^\star_\Sigma < \infty$ and a policy $\pi \in \Pi_{\mathrm{post}}$ uses exactly $Q^\star_\Sigma$ retrieval calls on its successful trajectories almost surely, its conditional expected cost is $Q^\star_\Sigma$. Together with Proposition A.1, this gives

$$D_{\mathrm{post}}(q) = Q^\star_\Sigma. \tag{18}$$

A policy that identifies and follows a cheapest valid route without extra retrieval satisfies this condition. A small $Q^\star_\Sigma$ indicates that a cheap identifying route exists; a particular solver may still need additional search to discover it.

A.4 Selectivity, evidence dispersion, and dependency depth

Subset selectivity. The candidate breadth remaining after applying a subset of clues is

$$s(\mathcal{P}) = |\mathrm{Ans}(\mathcal{P})|. \tag{19}$$

A smaller value means greater selectivity. If a single clue leaves only $y^\star$, that clue is itself identifying and enters the minimization in Equation (15). A small, non-identifying candidate pool can also focus subsequent exploration. Selectivity does not directly lower-bound route length; the evidence needed to verify the relevant clues must be considered separately.

Evidence dispersion. For a fixed subset, let

$$M_{\mathrm{ev}}(\mathcal{P}) = \min \left\{ |\boldsymbol{\theta}| : \boldsymbol{\theta} \text{ retrieves evidence verifying } (y^\star, \mathcal{P}) \right\}. \tag{20}$$

Queries are issued through Σ, but this minimization ignores executability: it counts the retrievals needed to obtain the evidence, regardless of whether their required constants are available when they are issued. Restoring executability can only restrict the feasible routes. If one retrieval supplies evidence for several constraints, those constraints can be verified together.

Dependency depth. A later query depends on an earlier one when it requires a name, attribute, or intermediate fact that is absent from the question but exposed by earlier retrieved evidence. Let $\mathrm{depth}(\boldsymbol{\theta})$ be the length of the longest such dependency chain in a valid route. An empty route has depth zero. Define

$$\mathrm{dep}(\mathcal{P}) = \min_{\boldsymbol{\theta} \text{ valid for } (y^\star, \mathcal{P})} \mathrm{depth}(\boldsymbol{\theta}). \tag{21}$$

For example, searching for an advisor's other students may first require finding the advisor's name. If the question already supplies that name, the downstream query becomes executable from the start and the required dependency chain may become shorter.

A.5 Component lower bound and one-query collapse

Proposition A.2 (Component lower bound)**.** *For every identifying subset $\mathcal{P}$,*

$$Q_\Sigma(\mathcal{P}) \geq \max\{M_{\mathrm{ev}}(\mathcal{P}), \mathrm{dep}(\mathcal{P})\}. \tag{22}$$

Consequently,

$$D_{\mathrm{post}}(q) \geq Q^\star_\Sigma \geq \min_{\mathcal{P} \in \mathcal{I}_q} \max\{M_{\mathrm{ev}}(\mathcal{P}), \mathrm{dep}(\mathcal{P})\}. \tag{23}$$

Proof. If $Q_\Sigma(\mathcal{P}) = +\infty$, the inequality is immediate. Otherwise, let $\boldsymbol{\theta}^\star$ be a shortest valid route for $(y^\star, \mathcal{P})$, so that $|\boldsymbol{\theta}^\star| = Q_\Sigma(\mathcal{P})$. First, it retrieves enough evidence to verify every constraint in $\mathcal{P}$ for $y^\star$. It is therefore feasible for the minimization defining $M_{\mathrm{ev}}(\mathcal{P})$, which ignores executability. Thus,

$$Q_\Sigma(\mathcal{P}) = |\boldsymbol{\theta}^\star| \geq M_{\mathrm{ev}}(\mathcal{P}). \tag{24}$$

Second, a dependency chain is contained within the route, so the route cannot be shorter than its longest dependency chain. The minimum depth over all valid routes cannot exceed the depth of this particular route. Combining these observations gives

$$Q_\Sigma(\mathcal{P}) = |\boldsymbol{\theta}^\star| \geq \mathrm{depth}(\boldsymbol{\theta}^\star) \geq \mathrm{dep}(\mathcal{P}). \tag{25}$$

Both bounds hold for the same route, establishing Equation (22). Taking the minimum over identifying subsets and applying Proposition A.1 establishes Equation (23). □

Evidence dispersion and dependency depth constrain route cost in different ways. The former measures evidence acquisition without executability; the latter arises from the information needed to make later queries executable. Selectivity affects which subsets are identifying, rather than supplying another route-length lower bound.

Corollary A.3 (One-query collapse)**.** *Assume that evidence verification requires at least one retrieval. If some $\mathcal{P} \in \mathcal{I}_q$ is verified by a single initially executable query, then*

$$Q^\star_\Sigma = 1. \tag{26}$$

Proof. That query forms a valid route of length one, giving

$$Q^\star_\Sigma = \min_{\mathcal{P}' \in \mathcal{I}_q} Q_\Sigma(\mathcal{P}') \leq Q_\Sigma(\mathcal{P}) \leq 1.$$

Since at least one retrieval is required for verification, $Q^\star_\Sigma \geq 1$. The two inequalities establish equality. □

This case illustrates the construction–search mismatch: a question built from many facts can still have a one-query identifying route when an initially executable query returns evidence for an identifying subset.

A.6 Solver-specific cost and trajectory diagnostics

A concrete solver π_0 may use prior knowledge or propose an answer before retrieval provides support. When its success probability is positive, its expected successful-trajectory cost is

$$\Omega(q, \pi_0) = \mathbb{E}_{\tau \sim \pi_0}[|\tau| \mid \tau \text{ succeeds}]. \tag{27}$$

When both $D_{\text{post}}(q)$ and $\Omega(q, \pi_0)$ are finite, define the solver-side cost difference as

$$U_{\pi_0}(q) = D_{\text{post}}(q) - \Omega(q, \pi_0). \tag{28}$$

A positive value means that the solver uses fewer retrieval calls on successful trajectories than the reference solver. Because this cost is conditioned on success, it should be interpreted together with the solver's success rate.

Rearranging the definition gives

$$\Omega(q, \pi_0) = D_{\text{post}}(q) - U_{\pi_0}(q). \tag{29}$$

Under the sufficient condition for equality above, if $U_{\pi_0}(q) = 0$, then

$$\Omega(q, \pi_0) = D_{\text{post}}(q) = Q^{\star}_{\Sigma}. \tag{30}$$

The theoretical quantities describe identifying routes and the reference cost, but are not all directly computable at scale. Section 2.4 therefore uses observable trajectory signatures: solving cost, answer hit time, and the answer-before-evidence indicator. Together, these diagnostics examine retrieval effort, early answer exposure, and answer-before-evidence behavior. The mechanisms and representative cases are discussed in Appendix B.

B Shortcut Risks and Diagnostic Cases

This appendix connects the four risks in Section 2.3 to Appendix A and illustrates their observable effects. The first three concern evidence acquisition: which constraints suffice, how their support is retrieved, and which queries are executable. Prior-knowledge binding concerns information supplied by the solver itself.

The cases retain the reported questions, answer labels, trigger queries, and trigger rounds. Descriptions of retrieved content and model behavior are summaries, not verbatim tool or model outputs. Round 0 denotes behavior before the first retrieval call.

B.1 Evidence co-coverage

Evidence co-coverage occurs when one retrieved item, page, or snippet verifies several intended constraints at once. Although a question may contain multiple clues, their verification need not require separate retrievals. The same observation can expose the answer together with several supporting facts, compressing the intended evidence-acquisition steps.

The corresponding quantity is the relaxed verification cost $M_{\text{ev}}(\mathcal{P})$ in Equation (20). Co-covered support can make this quantity small because fewer retrievals are needed to verify the selected constraints. If an initially executable query verifies an entire identifying subset, the minimum route cost is at most one, and is exactly one when verification requires at least one retrieval (Corollary A.3).

Figure 4 illustrates this mechanism: a single result surfaces Grey Investment Group together with several answer-side facts.

Case 1: Evidence co-coverage

Question. A corporate entity in the Pacific region, which was formally established to consolidate various holdings, traces its commercial lineage to entrepreneurial activities in Apia that began in the early decades of the 1900s. These foundational ventures were launched by a prominent individual whose career was primarily built in the local hospitality industry. What is the formal name of the resulting holding company that was incorporated?

Answer label: Grey Investment Group. **Trigger:** Round 1.

Reported query.

```
Apia Samoa early 1900s holding company founded by hospitality
entrepreneur
```

Explanation. A single returned snippet names Grey Investment Group and describes it as a Samoan investment enterprise with tourism and hospitality activities and assets across the Pacific region. The result exposes the answer and these attributes before the model names the answer.

Figure 4: Diagnostic example of evidence co-coverage.

B.2 Single-clue selectivity

Single-clue selectivity occurs when one clue, or a small clue subset, identifies the answer or narrows the candidate pool to a few plausible answers. The solver can then focus on those candidates before acquiring evidence for the remaining constraints.

Candidate breadth $s(\mathcal{P})$ is defined in Equation (19). A singleton identifying clue enters the minimization over $\mathcal{I}_q$ in Equation (15); if its evidence is cheaply accessible, it offers a cheap route. A clue that merely narrows the candidates can instead facilitate candidate discovery while leaving disambiguation and verification necessary.

Figure 5 shows a query using only the television-broadcast clue to surface the answer year, 1951, before the remaining threads are checked.

Case 2: Single-clue selectivity

Question. A specific calendar year in the middle of the twentieth century can be defined by four converging threads. First, it is the year a foundational international convention on the legal protection of displaced persons was opened for signature, administered by a UN agency. Second, that same UN agency has a mandate that significantly overlaps with a much larger intergovernmental body—one that formally became a UN-related organization more than five decades after the first human-crafted object reached another celestial body. Third, in this identical year, on a day in the final week of June, a major U.S. television network publicly transmitted its first regular programming using a full-color system. Finally, in that same twelve-month period, in a nation through which the Paraná River flows, a documented submission grappling contest took place between two high-profile figures from the world of martial arts. Determine which single year satisfies all four of these conditions.

Answer label: 1951. **Trigger:** Round 1.

Reported query.

```
first regular full-color programming US television network
final week June
```

Explanation. The query uses only the television-broadcast thread. Its results surface events in June 1951, after which the model treats 1951 as the principal candidate while checking the other threads.

Figure 5: Diagnostic example of single-clue selectivity.

B.3 EXPOSED CONSTANTS

A name, title, exact value, or distinctive phrase on the question surface can supply a query argument that would otherwise need to be discovered. Under Equation (9), exposing that constant makes queries that require it executable as soon as their other arguments are available. This can remove discovery calls and shorten dependencies as defined in Equation (21). A long relational description may therefore contain a directly searchable phrase that bypasses much of its intended traversal.

Figure 6 illustrates how a distinctive phrase supplied in the question is reused in a query that retrieves the answer, Jack Munro.

Case 3: Exposed constants

Question. Identify an individual who held a regional leadership role in a major North American forestry labor union, an acronym that could also denote an international water association or a wrestling organization, during the latter half of the 1980s. This person was publicly associated with a city whose name derives from an Indigenous word for "grizzly bear," located within a tourism region in the southern interior of the westernmost Canadian province. When the provincial government announced new labor legislation in the mid-1980s, this individual publicly stated that the people of this province want to work, not protest. The same city hosts an airport that provides temperature observations used to model evaporation rates for a glacially formed lake in a north-south oriented valley. This individual shares the same full name as a professional pickleball player from Austin, Texas. Who is this union leader?

Answer label: Jack Munro. **Trigger:** Round 2.

Reported query.

```
IWA union leader Kelowna 1980s "want to work not protest"
```

Explanation. The model reuses the phrase "want to work, not protest" nearly verbatim from the question, and the retrieved results directly name Jack Munro.

Figure 6: Diagnostic example of an exposed-constant shortcut.

B.4 PRIOR-KNOWLEDGE BINDING

Prior-knowledge binding occurs when a concrete solver associates clues with the answer before retrieval provides support. Such knowledge is excluded from the reference class in Appendix A.2. It may reduce the solver's retrieval cost, as described by $U_{\pi_0}(q)$ in Equation (28), without shortening the evidence-only route. The observable signature is the answer-before-evidence indicator in Equation (6).

Figure 7 shows the model proposing Cleopatra before its first retrieval call and subsequently issuing a query that already contains her name.

Case 4: Prior-knowledge binding

Question. In a coastal town where a mechanical marvel from the age of telegraphy still stands, the town's historical associations trace back to a Roman conqueror. That conqueror later defied the republic by crossing a boundary river, setting in motion conflicts that culminated in a naval battle whose aftermath led to the demise of a legendary queen. Who was that queen?

Answer label: Cleopatra. **Trigger:** Round 0, before the first tool call.

Explanation. Before issuing a query, the model proposes Cleopatra as a possible answer and links the historical clues to Caesar, the civil war, Actium, and Cleopatra's death. A retrieved observation containing the answer appears later, following a query that already includes her name.

Figure 7: Diagnostic example of prior-knowledge binding.

C Details of FORT Task Synthesis

This appendix provides the graph-expansion algorithm, fact and fuzzing examples, and refinement cases accompanying Section 3.

C.1 Entity Selection and Graph Initialization

Root selection follows the topical coverage, long-tail preference, and evidence-availability criteria in Section 3.1. To initialize a seed graph, FORT pre-mines a cycle library from Wikidata. It builds a pair index from Wikidata edges, enumerates small cycles, removes duplicate cycles with the same node set, and filters out hub-based, redundant, or mass-collaboration structures. Retained cycles are stored in an inverted index $\mathcal{I}$, where $\mathcal{I}[u]$ contains all retained cycles involving entity u. For a root r, the initializer uses one cycle from $\mathcal{I}[r]$ as the seed graph $\mathcal{G}_0$ if the set is nonempty. Otherwise, it uses the single-node seed $(\{r\}, \varnothing)$.

C.2 Evidence Graph Expansion

Algorithm 1 expands the seed graph under depth limit and node-addition budget. Seed-node depths are graph distances from the root; newly added nodes receive their parent's depth plus one. The queue $\mathcal{Q}$ holds expandable nodes, $\mathcal{V}_{\text{done}}$ records processed nodes, and b counts newly added nodes. For each selected fact (v, ρ, w), an edge and child node are added when w is new and the budget permits.

Algorithm 1 Evidence graph construction

Require: initialized subgraph $\mathcal{G}_0 = (V_0, E_0)$ with root $r \in V_0$, depth limit D, node-addition budget B
Ensure: evidence graph $\mathcal{G}$

$\mathcal{G} \leftarrow \mathcal{G}_0$
$\delta_{\mathcal{G}}(u) \leftarrow \text{dist}_{\mathcal{G}}(r, u)$ for all $u \in V(\mathcal{G})$ ▷ initialize node depth
$\mathcal{Q} \leftarrow \{u \in V(\mathcal{G}) : \delta_{\mathcal{G}}(u) < D\}$
$\mathcal{V}_{\text{done}} \leftarrow \varnothing$
$b \leftarrow 0$ ▷ number of newly added nodes
while $\mathcal{Q} \neq \varnothing$ **and** $b < B$ **do**
  $v \leftarrow \arg\max_{u \in \mathcal{Q}} \delta_{\mathcal{G}}(u)$ ▷ expand deepest node first
  $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{v\}$
  if $v \in \mathcal{V}_{\text{done}}$ **or** $\delta_{\mathcal{G}}(v) \geq D$ **then**
    continue
  end if
  $\mathcal{V}_{\text{done}} \leftarrow \mathcal{V}_{\text{done}} \cup \{v\}$
  $\mathcal{A}_v \leftarrow \textsc{CollectAtomicFacts}(v)$ ▷ collect facts from diverse sources
  $\mathcal{D}_v \leftarrow \textsc{ConstructDerivedFacts}(\mathcal{A}_v)$ ▷ construct facts from multiple atomic facts
  $\mathcal{P}_v \leftarrow \mathcal{A}_v \cup \mathcal{D}_v$
  $\mathcal{F}_v \leftarrow \textsc{VerifyFacts}(v, \mathcal{P}_v)$ ▷ filter inconsistent or mismatched facts
  $\mathcal{S}_v \leftarrow \textsc{SelectFacts}(\mathcal{F}_v)$ ▷ select low-shortcut facts
  for $(v, \rho, w) \in \mathcal{S}_v$ **do** ▷ ρ: selected fact, w: referenced entity
    if $w \notin V(\mathcal{G})$ **and** $b < B$ **then**
      $\mathcal{G} \leftarrow \mathcal{G} \cup \{(v, \rho, w)\}$
      $\delta_{\mathcal{G}}(w) \leftarrow \delta_{\mathcal{G}}(v) + 1$
      $b \leftarrow b + 1$
      if $\delta_{\mathcal{G}}(w) < D$ **then**
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{w\}$
      end if
    end if
  end for
end while
return $\mathcal{G}$

C.3 Fact Construction, Verification, and Selection

Multi-source enrichment. The enricher collects atomic facts from Wikidata, open web pages, structured databases, Google Scholar, and Google Maps. It avoids drawing multiple selected facts from the same evidence item and encourages different web facts to come from different pages. Root-node facts are distributed across sources whenever possible. For efficiency, non-root nodes use sampled sources, since only one or two facts are typically selected per node (Zhang et al., 2026b).

Derived fact construction. Derived facts combine atomic facts or inspect source texts to form constraints that are less likely to appear verbatim in a single retrieved item. Table 5 presents the four constructors and their examples (Shen et al., 2025).

Table 5: Derived fact constructors and examples.

Constructor	Definition	Example
Coincidence bridging	Match the same attribute across independently retrieved records.	A held a concert at venue V; B also held a concert at V. $\Rightarrow$ A and B held concerts at the same venue.
Count aggregation	Combine an entity set with independently verified attributes to derive a counting constraint.	The cast of work W includes A, B, and C, each of whom has won award M. $\Rightarrow$ W features at least two winners of award M.
Numerical relation	Retrieve numerical values and apply arithmetic comparison.	A was born in year Y_1; A's advisor B was born in year Y_2. $\Rightarrow$ A's advisor is $Y_1 - Y_2$ years older than A.
Meta-fact extraction	Inspect, count, or summarize the original text of a rich-content entity.	From the lyrics of song A: $\Rightarrow$ Word B appears at least three times in A. From Chapter 3 of novel C: $\Rightarrow$ The chapter begins with a memory-oriented passage.

Fact verification. Candidate facts are checked for source consistency and entity consistency. Source consistency rejects a fact if it is contradicted within its evidence item, contradicted by other retrieved evidence, or incompatible with accepted facts in the graph. Entity consistency requires the evidence to concern the same real-world entity as the target node. Table 6 illustrates four common failures. These checks assess factual validity before fact selection (Cui et al., 2025).

Generic fact selection. Because exact candidate breadth $s(\{c\})$ is not available in open-domain search, we use an LLM-based judgment as a generation-time surrogate for single-clue selectivity. For each verified candidate fact, the selector considers the target entity, the fact, and its supporting evidence, and judges whether the fact is overly distinctive or strongly narrows the target in isolation. Facts judged broadly applicable while remaining useful for graph expansion are preferred. This heuristic is used only during generation; realized selectivity is evaluated separately through the trajectory-level analysis in Appendix D.4.1. Table 7 contrasts generic and representative facts for the same entities.

Table 6: Entity-consistency failures considered during fact verification.

Failure mode	Definition	Example
Similar names	The page describes a different entity with a similar or partially overlapping name.	**Target:** *A Chemical Plant*. **Rejected:** *A Organic Chemical Plant*, a different company.
Abbreviation ambiguity	The same abbreviation refers to another subject.	**Target:** a research institute abbreviated as *IIS*. **Rejected:** *IIS* referring to *Internet Information Services*.
Temporal or geographic drift	The surface name matches, but the time, location, or entity instance differs.	**Target:** the N-th Director of Bureau X. **Rejected:** the $(N-1)$-th Director, or a same-named bureau in another region.
Series–edition mismatch	The page describes a different granularity or edition.	**Target:** *The Voice of China Season 3*. **Rejected:** the series-level page or pages about other seasons.

Table 7: Generic and representative facts for the same entities.

Entity	Generic fact	Representative fact
Marie Curie	Worked at a university in France.	Won Nobel Prizes in two different scientific fields.
The Eiffel Tower	Is located in a European capital city.	A landmark iron tower in Paris.
The Matrix	Is an English-language science fiction film.	Depicts humans living inside a simulated reality created by machines.

C.4 Question Formulation and Fuzzing

The formulation agent selects an answer-bearing subgraph $\mathcal{G}^* \subseteq \mathcal{G}$, chooses the answer node $y^\star$, and removes redundant facts. It retains clues that are jointly identifying while remaining individually generic. Intermediate entity names are withheld and replaced with referring expressions such as "the artist", "the institution", or "the work".

For remaining names and literal values, exact-value fuzzing produces truthful but less directly searchable constraints while preserving the answer to the complete question (Zhang et al., 2025). Table 8 gives the five strategies and examples. The agent chooses a strategy according to the value's semantic type, distinctiveness, and role in the question.

Table 8: Exact-value fuzzing strategies and examples.

Strategy	Description	Examples
Category generalization	Replace an entity name with a higher-level category, type, or domain.	*International Monetary Fund* → an international financial institution. *Harvard University* → an Ivy League university in the United States.
Range relaxation	Replace an exact number with an interval, approximate range, or open-ended constraint.	*1863* → the second half of the nineteenth century. *16,000* → more than ten thousand.
Meta-attribute description	Describe a value's intrinsic form or surface structure.	*September 9* → a date whose month and day use the same number. *Hannah* → a person whose given name is a palindrome.
Arithmetic encoding	Express a number through arithmetic, digit-level, or number-theoretic constraints.	*42 years old* → an age that is a multiple of six. *17,921* → a five-digit prime number whose digits sum to 20.
Contrastive exclusion	Exclude alternatives while leaving the target value implicit.	*Canada* → a country in North America that does not border Mexico. *Wednesday* → a weekday that is not Monday.

C.5 Adversarial Refinement Examples

Adversarial refinement uses the solver's trajectory to identify clues that expose an answer too early or prevent the question from being resolved. Figure 8 illustrates shortcut repair by broadening an aviation reference and relaxing a numerical constraint. Figure 9 illustrates ambiguity repair by narrowing over-fuzzed clues. The before and after questions are shown in full; bold text marks the revised spans.

Refinement Case 1: Repairing Shortcuts

Before. A certain film was released in a certain year. The film has a runtime of a three-digit minute count whose **digits sum to 10**, features protagonists launching business schemes and a scam escalating to large monetary levels. The film's composer is a certain individual who plays a certain instrument typically belonging to the keyboard family.
In that same release year, a certain aviation incident occurred involving a target being **shot down by the Soviet Union**. The incident originated from a certain institution **having 4 runways**. Who is the director of the film?

Refinement. The Soviet Union reference and runway count identify an aviation incident and hence the film's release year. Together with the runtime digit sum, these clues sharply restrict the film candidates. The revision broadens the aviation reference and relaxes the runtime constraint while retaining the plot, composer, and instrument clues.

After. A certain film was released in a certain year. The film has a runtime of a three-digit minute count whose **digits sum to a multiple of 5**, features protagonists launching business schemes and a scam escalating to large monetary levels. The film's composer is a certain individual who plays a certain instrument typically belonging to the keyboard family.
In that same release year, a certain aviation incident occurred involving a target being **shot down by a certain country during the Cold War**. The incident originated from **a certain international airport**. Who is the director of the film?

Figure 8: Shortcut repair by broadening the aviation reference and relaxing the runtime constraint.

Refinement Case 2: Narrowing Over-fuzzed Clues

Before. An individual was born in a certain year and died 61 years after that birth year. The individual was a member of an institution headquartered in a building in **a certain European capital**, built in a year in the 1910s. This institution was a junior coalition partner to another institution founded in **a certain year**. The individual served in a ministerial position involving economic and communications matters in **a region** where another individual also served. The other individual worked as a lawyer until 50 years after that other institution's founding year. The individual had been threatened multiple times before death. Who is the individual described above?

Refinement. The descriptions "a certain European capital", "a certain year", and "a region" are broad enough to admit an unrelated figure. The revision narrows these geographic, temporal, and administrative descriptions to exclude the spurious answer while keeping the relevant names implicit.

After. An individual was born in a certain year and died 61 years after that birth year. The individual was a member of an institution headquartered in a building in **the capital of a federal republic in Central Europe**, built in a year in the 1910s. This institution was a junior coalition partner to another institution founded in **the mid-20th century**. The individual served in a ministerial position involving economic and communications matters in **a federated state** where another individual also served. The other individual worked as a lawyer until 50 years after that other institution's founding year. The individual had been threatened multiple times before death. Who is the individual described above?

Figure 9: Ambiguity repair by narrowing over-fuzzed geographic, temporal, and administrative descriptions.

D EXPERIMENT DETAILS

D.1 EXPERIMENTAL SETUP

Models and training. We train FORT-Searcher using SFT only on search trajectories generated from FORT tasks. We use Qwen3.6-35B-A3B (Qwen Team, 2026b) as the primary backbone and further evaluate Qwen3.5-9B and Qwen3.5-4B (Qwen Team, 2026a) to examine scaling effects. We use DeepSeek-V4-Pro-0813 as the teacher model. Detailed training parameters are summarized in Table 9.

Benchmarks. We evaluate search capabilities on BrowseComp (Wei et al., 2025) and xbench-DeepSearch (Chen et al., 2025c) for long-context search, BrowseComp-ZH (Zhou et al., 2025) for Chinese search, and WideSearch (Wong et al., 2026) for multi-answer retrieval and Markdown organization. For robustness, we report the average performance over repeated evaluations, with one run for BrowseComp, ten runs for xbench-DeepSearch, and five runs for BrowseComp-ZH and WideSearch. We use Gemini-3.5-Flash as the evaluator. WideSearch is evaluated using item-level F1. To avoid unfair comparisons, we blocked domains from Hugging Face in the testing environment.

Baselines and evaluation protocol. We compare FORT-Searcher with large-scale search agents and 30B-scale models. The large-scale agents include DeepSeek-V4-Pro-0813 and DeepSeek-V4-Flash-0731 (DeepSeek-AI, 2026), GLM-5.3 (Zeng et al., 2026a), Kimi-K3 (Team et al., 2026a), and Qwen3.8-Max (Qwen Team, 2026c). The 30B-scale baselines include REDSearcher (Chu et al., 2026), Apodex-1.0-mini (Team, 2026), BigBang-v1 (BIGBANG team, 2026), and Qwen3.6-35B-A3B (Qwen Team, 2026b). We additionally evaluate Qwen3.5-9B and Qwen3.5-4B (Qwen Team, 2026a) before and after fine-tuning. All evaluations follow the official protocols when available, and otherwise use the same evaluation harness and budget as FORT-Searcher.

Context management. We follow the context-management framework of MiroThinker (Team et al., 2026b). Each rollout is allocated a certain budget of interaction turns, with tool observations retained to enable evidence reuse. If the model fails to produce a final answer within the budget, we reset the context and restart from the original question. Each evaluation instance allows up to five context resets.

Table 9: Key SFT hyperparameters for FORT-Searcher.

Hyperparameter	**Value**
Fine-tuning method	Full-parameter SFT (language model)
Training epochs	4
Maximum sequence length	262,144 tokens
Sequence packing	Enabled
Global batch size	32
Micro-batch size	1
Peak learning rate	3×10^{-5}
Minimum learning rate	2×10^{-6}
Learning-rate schedule	Cosine decay
Warmup steps	10
Weight decay	0.01
Adam coefficients (β_1, β_2)	$(0.9, 0.95)$
Gradient clipping threshold	1.0
MoE auxiliary loss coefficient	10^{-3}
MoE expert capacity factor	2.0

Data generation. We construct evidence graphs with a maximum depth of $D = 4$ and a node-addition budget of $B = 20$. A retrieval-cost threshold of $\tau_{\text{th}} = 30$ retrieval calls is used to identify

Table 10: Key data-generation settings for FORT.

Setting	Value
Graph depth limit D	4
Node-addition budget B	20
Refinement threshold τ_{th}	30
Maximum refinement rounds	1
Final training-set size	40K

shortcut-prone drafts for refinement, but does not determine whether a draft is retained. Each draft is refined for at most one round. The final training set contains 40K successful trajectories. Table 10 summarizes these settings.

D.2 Results on Additional Backbones

To examine whether the effectiveness of FORT training data extends beyond Qwen3.6-35B-A3B, we additionally fine-tune Qwen3-30B-A3B-Thinking-2507 and Qwen3.5-35B-A3B on FORT trajectories using SFT. We evaluate these models on BrowseComp and compare them with the REDSearcher and Apodex-1.0-mini reference results, respectively.

Table 11 shows that the two additional FORT-Searcher variants achieve 77.2 and 79.8 on BrowseComp, exceeding the corresponding reference scores. These results support the applicability of FORT data across backbone versions.

Table 11: BrowseComp results across backbone versions.

Model	Backbone	BC (%)
REDSearcher	Qwen3-30B-A3B-Thinking-2507	57.4
FORT-Searcher	Qwen3-30B-A3B-Thinking-2507	77.2
Apodex-1.0-mini	Qwen3.5-35B-A3B	71.5
FORT-Searcher	Qwen3.5-35B-A3B	79.8
BigBang-v1	Qwen3.6-35B-A3B	76.5
FORT-Searcher	Qwen3.6-35B-A3B	83.3

D.3 Cross-Solver Robustness of Dataset Difficulty Analysis

To examine whether the observed difficulty pattern depends on the diagnostic solver, we repeat the dataset comparison using Qwen3.5-397B-A17B under the same retrieval budget. For each dataset, we evaluate the same 200 sampled questions, computing accuracy over all 200 questions and trajectory diagnostics over the successful subset of the same runs.

Although absolute values and some baseline orderings change, FORT again yields the highest solving cost and latest answer exposure. Its accuracy is comparable to REDSearcher (60.5% vs. 59.5%), while $\widehat{\Omega}$ and T_{hit} remain substantially higher. This indicates that the observed increase in pre-answer search is not simply explained by lower solver accuracy and persists across diagnostic solvers.

D.4 Trajectory-Level Proxies and Diagnostic Validation

D.4.1 Trajectory-Level Proxies for Difficulty Factors

To examine how the difficulty factors in Appendix A are reflected in realized search behavior, we conduct a trajectory-level diagnosis on successful question–trajectory pairs. The exact values of the theoretical quantities in Appendix A are not feasible to compute in open-domain web search. In particular, exact evaluation of $s(\mathcal{P})$ requires determining the corresponding answer set, while

Table 12: Dataset difficulty comparison under trajectory signatures using Qwen3.5-397B-A17B as the diagnostic solver. For each dataset, accuracy is computed over 200 randomly sampled questions, while trajectory diagnostics are computed over the successful subset of the same runs.

Source	**Acc. (%)**	$\widehat{\Omega}$	T_{hit}	$\widehat{p}_{\text{prior}}(\%)$
InfoSeek (Xia et al., 2025)	97.5	20.6	5.7	2.1
MiroVerse-Voyager (Team et al., 2025)	99.0	30.6	5.9	5.6
DeepDive (Lu et al., 2025)	94.0	47.7	15.5	7.4
DeepResearch-9K (Pham et al., 2025)	49.5	47.8	3.4	27.3
OpenSeeker (Du et al., 2026)	67.5	84.7	9.3	31.9
REDSearcher (Chu et al., 2026)	59.5	92.1	18.7	11.8
FORT	60.5	**141.0**	**46.9**	10.7

$M_{\text{ev}}(\mathcal{P})$ and $\text{dep}(\mathcal{P})$ require optimization over evidence-acquisition routes. The solver-side cost difference $U_{\pi_0}(q)$ additionally depends on the reference cost $D_{\text{post}}(q)$, which is not directly observed in these experiments. This difference does not isolate the effect of prior knowledge.

We therefore do not treat the following measurements as direct estimates of the theoretical quantities. Instead, we use them as trajectory-level proxies for the realized effects of the difficulty factors analyzed in our framework. Specifically, we randomly sample 200 successful question–trajectory pairs from open-source deep-search data and 200 from FORT. We use GPT-5.5 to inspect each complete question and trajectory, decompose the question into atomic clues, and align each clue with the corresponding search steps and retrieved evidence. Based on this annotation, we define four observable proxies.

For *subset selectivity*, we examine whether the search evidence associated with each individual clue already narrows the candidate space to a very small set. We mark a clue as low-width if the clue-specific search evidence yields only one or two plausible candidate answers. We report the percentage of such clues as R_{low}. A lower R_{low} indicates that individual clues are less likely to identify the answer on their own, making small identifying subsets less available in realized search.

For *evidence dispersion*, we assign each verified clue to one primary evidence source that supports it. If the same web page or evidence source verifies multiple clues, it is counted only once. We then compute the normalized evidence dispersion:

$$R_{\text{ev}} = \widehat{M}_{\text{ev}}/n_c \times 100\%, \tag{31}$$

where $\widehat{M}_{\text{ev}}$ is the number of distinct primary evidence sources and n_c is the number of clues in the question. A higher R_{ev} indicates more dispersed evidence, while a lower value suggests that multiple clues are co-covered by a small number of sources.

For *dependency depth*, we do not estimate the oracle quantity $\text{dep}(\mathcal{P})$. Instead, we measure the retrieval cost of the most costly realized dependency chain, denoted by $\widehat{C}_{\text{dep}}$. We first keep only clue-relevant search steps before the first answer hit. We then check whether each step uses an entity name, attribute, date, title, alias, or intermediate fact first exposed by earlier retrieved evidence. If a later search depends on information exposed by an earlier step, the two steps are considered part of the same realized dependency chain. Among all realized dependency chains, we select the one with the highest retrieval cost and count the actual retrieval steps needed to complete it. A higher $\widehat{C}_{\text{dep}}$ indicates stronger serial dependence and greater retrieval effort before the answer is exposed.

To examine whether delayed answer exposure reflects stronger pre-answer evidence dependencies, we further compute the Spearman correlation between T_{hit} and $\widehat{C}_{\text{dep}}$ over the annotated trajectories. We report the correlation separately for open-source data and FORT.

The positive correlations show that later answer exposure is associated with more costly realized pre-answer dependency chains. This supports T_{hit} as an observable diagnostic of answer-exposure delay, rather than a direct estimate of the minimum identifying-route cost $Q^{\star}_{\Sigma}$.

Table 13: Correlation between answer hit time and realized pre-answer dependency cost.

Data	$\rho(\boldsymbol{T}_{\mathbf{hit}}, \widehat{\boldsymbol{C}}_{\mathbf{dep}})$
Open-source	0.31
FORT	0.36

Table 14: Trajectory-level proxies for realized difficulty factors on 200 successful question–trajectory pairs from each data source. $\widehat{\Omega}$ denotes the empirical average solving cost. R_{low} denotes the percentage of clues whose clue-specific search evidence yields only one or two plausible candidate answers. R_{ev} denotes normalized evidence dispersion. $\widehat{C}_{\mathrm{dep}}$ denotes the retrieval cost along the most costly realized pre-answer dependency chain. $\widehat{p}_{\mathrm{prior}}$ denotes the fraction of trajectories where the model proposes the gold answer before retrieved evidence exposes it.

Data	$\widehat{\Omega}$ $\uparrow$	R_{low} (%) $\downarrow$	R_{ev} (%) $\uparrow$	$\widehat{C}_{\mathrm{dep}}$ $\uparrow$	$\widehat{p}_{\mathrm{prior}}$ (%) $\downarrow$
Open-source	51.5	58.2	76.3	2.7	17.0
FORT	**100.3**	**43.0**	**88.2**	**5.5**	**13.0**

For answer-before-evidence behavior, we use the indicator defined in Section 2.4. We report the fraction of trajectories where GPT-5.5 judges, based on semantic equivalence rather than string matching, that the model proposes the gold answer before retrieved evidence exposes it, denoted by $\widehat{p}_{\mathrm{prior}}$.

Table 14 shows that FORT moves all trajectory-level proxies in the desired direction: compared with open-source deep-search data, it induces higher retrieval effort, less over-selective individual clues, more dispersed supporting evidence, more costly pre-answer dependency chains, and a lower answer-before-evidence rate. These proxies provide an operational connection between the theoretical difficulty factors and realized search behavior. They do not estimate the oracle minimum route cost in Appendix A, but show that FORT exhibits lower single-clue selectivity, greater evidence dispersion, more costly realized pre-answer dependency chains, and less answer-before-evidence behavior under the evaluated search setting.

D.4.2 Human Validation of the Answer-Before-Evidence Indicator

To validate the answer-before-evidence indicator, we manually inspect 200 trajectories, including 100 with $I_{\mathrm{prior}} = 1$ and 100 with $I_{\mathrm{prior}} = 0$. Three qualified annotators with graduate-level research training independently inspect each question and its corresponding search trajectory. The trajectories are randomly shuffled, and the annotators are not informed of the indicator value.

For each trajectory, annotators determine whether the model proposes the gold answer before retrieval provides supporting evidence. Semantically equivalent aliases are treated as the same answer. A case is not confirmed when the answer has already been supported by retrieved evidence before the model proposes it, or when the automatic answer match is incorrect.

Table 15: Human validation of the answer-before-evidence indicator.

Indicator	**Trajectories**	**Human-confirmed (%)**
$I_{\mathrm{prior}} = 1$	100	89.0
$I_{\mathrm{prior}} = 0$	100	3.0

Human annotation confirms answer-before-evidence behavior in 89.0% of trajectories with $I_{\mathrm{prior}} = 1$, compared with 3.0% of those with $I_{\mathrm{prior}} = 0$. These results support $\widehat{p}_{\mathrm{prior}}$ as a behavioral proxy for answer-before-evidence shortcuts rather than a direct measure of the model's internal prior knowledge.

Table 16: Manual validity audit of FORT questions. Values are percentages.

Group	Version	Correct	Supported	Unique	Unambiguous
Accepted	Final	96.0	96.0	94.0	92.0
Shortcut-prone	Before	98.0	96.0	96.0	98.0
	After	94.0	94.0	92.0	92.0
Initially unsolved	Before	88.0	86.0	72.0	74.0
	After	94.0	92.0	90.0	88.0

D.5 Question Validity and Uniqueness Audit

To examine whether increased search difficulty comes at the cost of question validity, we conduct a manual audit covering three groups. We randomly sample 50 questions accepted without refinement, 50 successfully refined shortcut-prone cases, and 50 successfully refined initially unsolved cases. For each successfully refined case, we inspect both the original pre-refinement draft and the corresponding post-refinement question, enabling paired comparisons before and after refinement.

Three qualified annotators with graduate-level research training independently inspect each question together with its gold answer and supporting evidence. Questions are randomly shuffled, and the annotators are not informed of the refinement category or whether a question is the pre- or post-refinement version. Each annotator evaluates four properties: answer correctness, source support, answer uniqueness, and ambiguity.

Answer correctness indicates whether the provided gold answer is factually correct for the complete question. Source support requires the provided evidence to support the constraints needed to establish the gold answer. Answer uniqueness requires that no plausible alternative answer satisfies the complete set of constraints. A question is marked unambiguous when its wording admits no materially different interpretation that leads to an alternative answer.

As shown in Table 16, questions accepted without refinement show consistently high validity across all four metrics. For shortcut-prone cases, validity remains high after refinement, with only modest decreases in correctness, uniqueness, and clarity. For initially unsolved cases, refinement substantially improves uniqueness from 72.0% to 90.0% and unambiguous formulation from 74.0% to 88.0%, while preserving answer correctness and source support. Overall, the audit shows that the two refinement modes serve distinct purposes without substantially compromising task validity.

D.6 Additional Ablation Analyses

Downstream evaluation. We further examine whether the cumulative configurations in Figure 3b differ in downstream training value. We select the full configuration, the configuration after removing Cycle, Long-tail, and Derived, and the final configuration after all six controls have been removed. For each setting, we use corresponding DeepSeek-V4-Pro-0813 trajectories and fine-tune Qwen3.5-4B with the same training recipe and inference setting.

Table 17: Downstream performance of representative configurations from the cumulative shortcut-resistance ablation.

Configuration	BrowseComp
Full	61.9
− Cycle − Long-tail − Derived	58.1
− Cycle − Long-tail − Derived − Diversity − Generic − Fuzzing	54.9

As shown in Table 17, BrowseComp decreases from 61.9 for the full configuration to 58.1 after removing Cycle, Long-tail, and Derived, and further to 54.9 after all six controls are removed. Together

Table 18: Isolated ablations of representative shortcut-resistant components.

Configuration	Acc. (%)	$\widehat{\Omega}$	T_{hit}	$\widehat{p}_{\text{prior}}(\%)$
Full	64.2	92.3	16.1	12.7
− Long-tail	72.0	76.4	13.0	15.3
− Derived	67.3	78.6	14.2	13.7
− Fuzzing	78.8	72.7	9.3	14.8

with Figure 3b, this shows that progressively removing shortcut-resistant controls is accompanied by easier realized search behavior and lower downstream training value.

Isolated component ablations. To complement the cumulative analysis in Figure 3b, we independently remove long-tail selection, derived-fact construction, and fuzzing from the full FORT pipeline. Each configuration removes only the specified component while keeping the remaining construction and evaluation settings fixed. We evaluate 3000 questions per configuration, computing accuracy over all questions and trajectory diagnostics over successful runs.

As shown in Table 18, removing long-tail selection increases the answer-before-evidence rate, while removing derived facts substantially reduces retrieval cost and removing fuzzing causes the largest drop in answer-hit time. Across all three isolated ablations, accuracy increases and the trajectory diagnostics shift toward easier search, with component-specific changes consistent with the intended shortcut risks.