Dossier: Deep Research via Ledger-Driven Branching Search and Query Encoding Learning

Om Chabra
MIT
Cambridge, MA, USA
omchabra@mit.edu

Noah Ziems
University of Notre Dame
Notre Dame, Indiana, USA
nziems2@nd.edu

Meng Jiang
University of Notre Dame
Notre Dame, Indiana, USA
mjiang2@nd.edu

Omar Khattab
MIT
Cambridge, MA, USA
okhattab@mit.edu

Hari Balakrishnan
MIT
Cambridge, MA, USA
hari@csail.mit.edu

Abstract

Deep research requires synthesizing information across fragmented sources. Existing ReAct-like agents for such multi-hop retrieval typically rely on long linear search trajectories, where early retrieval failures compound over time. We introduce **Dossier**, a deep research agent that replaces linear paths with *locally parallel, branching search* managed by a persistent **Research Ledger**. The Ledger explicitly tracks claims, contradictions, and information gaps, continuously updating as new evidence is synthesized. To improve the quality of each retrieval step, we introduce **Evidence-Aligned Query Learning (EAQL)**, a training mechanism that fine-tunes query encoders to condition on the Research Ledger. This ensures that generated queries are contextually grounded in the agent's evolving state rather than isolated prompts. Our evaluation demonstrates that Dossier's branching architecture improves end-to-end accuracy on BrowseComp-Plus by 23 percentage points and HoVer by 29 percentage points across multiple LLM models.

CCS Concepts

• **Information systems** → **Information retrieval**; **Web searching and information discovery**; • **Computing methodologies** → *Natural language processing*; *Machine learning*.

ACM Reference Format:
Om Chabra, Noah Ziems, Meng Jiang, Omar Khattab, and Hari Balakrishnan. 2026. Dossier: Deep Research via Ledger-Driven Branching Search and Query Encoding Learning. In *ACM Conference on AI and Agentic Systems (CAIS '26), May 26–29, 2026, San Jose, CA, USA*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3786335.3813122

1 Introduction

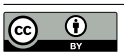
Deep research requires an LLM to iteratively retrieve and synthesize evidence across a document corpus to answer complex, multi-hop questions. For example, the BrowseComp benchmark [24] poses queries such as: "*What paper published at EMNLP between 2018-2023 did the first author do their undergrad at Dartmouth College and the fourth author do their undergrad at University of Pennsylvania?*" (Fig. 1a) Answering this question might require a system to find and enumerate numerous candidate papers and to verify biographies for multiple authors, as no single document encompasses evidence for all these constraints directly.

Current state-of-the-art approaches typically deploy a single agent operating in a linear Reasoning and Acting (ReAct; [27]) loop. In this method (Fig. 1b), the agent reasons over its current history (context), constructs a query, and appends the retrieved results to its context for the next iteration. While effective for short-horizon tasks, this strategy becomes brittle as retrieval hops, often set up to 100 iterations in recent work, increases [2]. Early retrieval or reasoning failures compound over time and steer the agent toward irrelevant paths.

We argue that a fundamental limitation in this paradigm is that the agent uses *unstructured context history* to track progress. Because it stores every retrieved document within a single context window without a representation of verified claims versus information gaps, its internal state eventually collapses under the weight of irrelevant tokens. To tackle this, we introduce **Dossier**, a deep research framework centered around tracking research state with a **Research Ledger**. This is a dynamic knowledge structure that partitions the agent's current belief state into *verified claims* and *unresolved information gaps*. By maintaining this, Dossier ensures that each retrieval iteration is strategically targeted toward specific unknowns and pruning distractors (Fig. 1c).

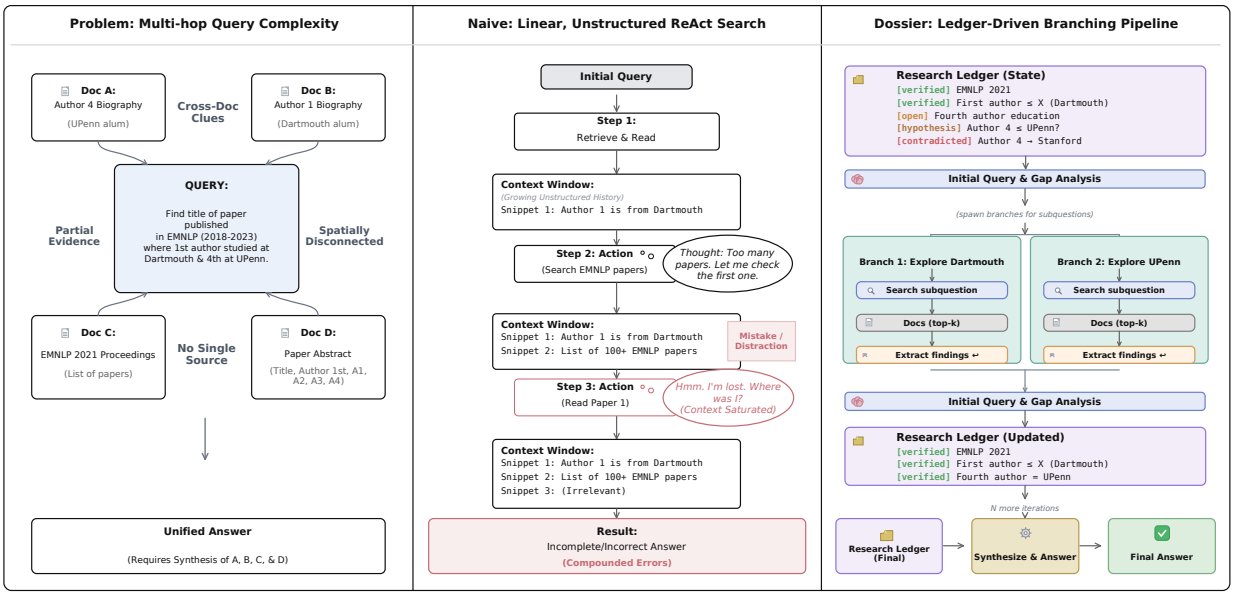
Dossier iterates through a cycle of *Gap Analysis, Search, and Information Extraction* phases, with the Research Ledger serving as the central state shared across all stages. In *Gap Analysis*, Dossier synthesizes fragmented findings and remaining information gaps from previous iterations into a unified belief state, which it then uses to generate a set of targeted textual subqueries. During the *Search* phase, a query encoder retrieves the top-k documents for each subquery independently. Then, the *Extraction* phase parses these documents in parallel to isolate atomic findings. Extraction is performed in isolation, deferring reconciliation across documents to the subsequent Gap Analysis stage.This loop continues until the Ledger reaches a terminal state, at which point Dossier uses the consolidated evidence to generate the final answer.

The efficacy of the Dossier loop depends on the retrieval model's ability to discern *incremental* relevance. Standard dense retrievers are context-blind; they encode queries in isolation, without awareness of the agent's evolving state. In deep research, however,



CAIS '26, San Jose, CA, USA

ACM ISBN 979-8-4007-2415-2/26/05
https://doi.org/10.1145/3786335.3813122



(A) Multi-hop reasoning requires connecting disparate evidence across documents.

(B) Linear approach relies on unstructured history, causing errors to compound and the agent to get "lost".

(C) Dossier maintains a structured ledger, enabling parallel branching and preventing error propagation.

Figure 1: Why ledger-driven branching scales for deep research. Dossier instead maintains a persistent Research Ledger of atomic, document-backed claims with explicit statuses (e.g., open/verified/contradicted). Gap analysis reads the ledger to identify remaining unknown factors, spawns parallel branches targeting specific gaps, and extracts findings back into the ledger for merge/deduplication/conflict resolution. Iterating this branch→merge loop keeps state compact and globally consistent, enabling reliable synthesis into the final answer.

document utility is conditional on the Research Ledger because a document is only relevant if it resolves an outstanding gap rather than merely recapitulating verified claims. To bridge this gap, we introduce the **Evidence-Conditioned Query Encoder**, which explicitly conditions retrieval on the current Ledger state. By making the encoder state-aware, we shift the retrieval objective from global semantic similarity to *informational incrementality*, ensuring each search step targets unresolved gaps rather than recapitulating known facts (§4).

Training this state-aware encoder requires an *Evidence-Aligned Query Learning (EAQL)* mechanism. The primary challenge lies in obtaining high-fidelity supervision signals over long horizons: in standard agent rollouts, early errors cause agent trajectories to drift, producing noisy and misaligned label pairs. To address this, we employ a technique during training that ensures accurate labels even in later hops (§3.3). This approach allows us to collect clean contrastive pairs for learning without allowing early retrieval errors to corrupt downstream supervision.

From these oracle-stabilized rollouts, we extract contrastive training tuples consisting of the generated subquery, the injected ground-truth document, and topically similar but non-gap-filling distractors. We then optimize the encoder using a Multiple Negatives Ranking (MNR) loss, teaching the model to prioritize documents that resolve specific Ledger gaps over those that exhibit surface-level similarity.

Overall, this paper makes the following contributions. First, **a state-aware research architecture:** Dossier replaces amorphous context histories with a persistent Research Ledger. By maintaining an explicit global state of verified claims, contradictions, and information gaps, it mitigates the error propagation and context saturation of ReAct-style agents. Second, **ledger-driven branching search:** We propose a three-step search cycle (*Gap Analysis, Search, and Extraction*) for parallel exploration. By dynamically querying the Research Ledger to resolve combinatorial dependencies, Dossier systematically prioritizes high-uncertainty frontiers, preventing redundant exploration and using the search budget efficiently. Third, an **Evidence-Aligned Query Learning (EAQL):** We develop a novel training objective that conditions the query encoder on the agent's internal belief state. Utilizing *Oracle-Guided Trajectory Generation* to stabilize long-horizon rollouts, EAQL trains the encoder to prioritize *informational incrementality* over surface-level similarity, ensuring each retrieval step yields novel evidence.

As a result of these design elements, Dossier consistently outperforms prior linear and hierarchical baselines on complex closed-corpus deep-research tasks. On BrowseComp-Plus, Dossier achieves a state-of-the-art **75.1%** success rate, improving over ReAct by +23 percentage points and over Smolagents Deep Research by +27 points, averaged across three backbone models. On the HoVer multi-hop fact verification benchmark, Dossier reaches **89.8%** accuracy, surpassing ReAct by +30 points and Smolagents Deep Research by +29 points. Additionally, although evaluated in closed-corpus settings, the architecture naturally extends to open-web retrieval.

2 Background & Related Work

Deep research. Deep research tasks represent a complex class of information retrieval and synthesis problems where a simple,

single-step search is insufficient. Formally, given an input research question q and a large corpus C, the objective is to generate an answer a that accurately answers q. These deep research agents require *iterative retrieval and synthesis*, where an agent must perform a sequence of actions including querying subsets of C, analyzing intermediate documents, and refining subsequent queries based on gathered information. A defining characteristic of this domain is the *multi-hop evidence requirement*: the final answer often relies on connecting disparate pieces of evidence $e_1, e_2, \ldots, e_n$ located across different documents within C, which may not share explicit lexical overlap with the original query q. Success in this regime is typically evaluated via end-to-end accuracy a and evidence recall, i.e. the percent of necessary documents successfully retrieved.

There has been growing interest around deep research in recent years with a number of benchmarks being released. These benchmarks can largely be broken into those that require the agent to interact with websites [3, 5, 24] and those that simply involve retrieving information from a provided corpus [2, 6].

Dense retrieval. Traditionally, dense retrievers have served as the primary search mechanism in deep research [9, 17, 26], and recently have made a resurgence in popularity for this task [2]. Dense retrievers often take in a natural language query, q, and a desired number of documents to retrieve, k. Under the hood, dense retrieval systems can largely be broken down into two components: a query encoder and a previously embedded corpus C. When the search tool is invoked, the query encoder takes the text query q and maps it into an internal vector representation $\mathbf{q}$. The corpus C consists of document passages that have already been mapped into this same vector space using a corresponding document encoder. When called, the search tool computes similarity between $\mathbf{q}$ and embeddings in C, performing a k-nearest neighbor lookup, returning the text of the top-k most similar documents. In Dossier, we focus on fine-tuning this query encoder (§3.3). Our work draws particular inspiration from Baleen [9], which introduced condensed retrieval for multi-hop reasoning, using Transformer encoders to extractively summarize pertinent facts from each retrieval step into a compact context for subsequent queries rather than carrying forward entire passages. Baleen also introduced a latent hop ordering training algorithm for the retriever.

ReAct agents. To address these complex deep research questions, ReAct-style [27] agents operate through an interleaved process of reasoning and acting, utilizing tools such as dense retrievers to bridge the knowledge gap. At each time step t, the agent generates a distinct *Thought* (th_t) which reasons about its current state based on the prior information in its context window. It will then, execute a specific tool *Action* (a_t)—such as issuing a specific search query—and receives an *Observation* (o_t) from the environment, typically in the form of retrieved document snippets. This process creates a linear trajectory $\tau = (th_1, a_1, o_1, \ldots, th_T, a_T, o_T)$. However, this architecture is inherently constrained by its linearity and memory management; the agent relies entirely on the LLM's limited context window to maintain the history of the investigation. Because these ReAct agents must store every retrieved document snippet and reasoning step within their current working context, the input prompt grows linearly with the research depth. As a result, long-horizon tasks quickly saturate the available token budget, forcing the model to either truncate critical earlier evidence or suffer from "lost-in-the-middle" phenomena where reasoning capabilities degrade over long contexts.

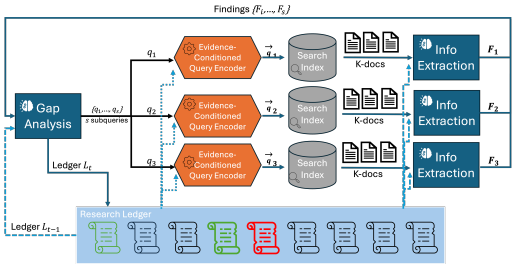


Figure 2: Dossier system overview. Starting from a global question Q, Gap Analysis generates a branching set of targeted sub-queries conditioned on the persistent Research Ledger L. Each branch performs retrieval using a ledger-conditioned query encoder to prioritize novel, gap-filling documents, then extracts findings to update L. Final Synthesis after a set budget will convert the conflict-resolved ledger into a final answer with citations.

Multi-agent systems. Multi-agent systems (MAS) attempt to overcome these single-agent limitations by decomposing the global research question q into sub-tasks distributed among distinct worker agents [16, 18, 22, 25]. Despite the advantages of parallelization, in existing MAS architectures the agents often lack a global view of the information landscape. Consequently, agents often converge on redundant evidence or narrow retrieval paths, leaving critical alternative perspectives within corpus C undiscovered.

Complementary approaches. Several recent systems are complementary to Dossier but target different parts of the deep-research pipeline. Search-o1 [13] uses a separate document-analysis module that returns findings to the main chain, but it does not maintain a persistent structure across hops. Search-R1 and WebThinker [14] focus on training with RL and DPO. RLMs [28] report results on a reduced version of BrowseComp-Plus, providing a strong alternative to standard LLM reasoning calls. Lastly, STORM [21] is closest in spirit because it builds structured artifacts, but it targets downstream synthesis through grounded outlines and long-form reports, whereas Dossier operates upstream by enforcing structured accumulation during evidence gathering.

3 System Overview

Dossier operates as a cyclic engine composed of three distinct functional components that form the core loop, shown in Figure 2 and discussed in detail below. First, **Gap Analysis** evaluates the current state of knowledge to identify missing information or conflicts, generating targeted sub-queries to resolve them. Second, **Evidence-Conditioned Retrieval** executes these searches using a learned encoder that conditions on both the query and the current ledger state to prioritize finding novel, high-value evidence. Third, **Information Extraction** distills relevant findings from raw documents and records them in isolation, deferring reconciliation to the Gap Analysis step. This loop repeats until a set number of iterations, at which point a final **Synthesis** step transforms the accumulated evidence into a final language answer.

3.1 The Research Ledger

The core of our architecture is the *Research Ledger* ($\mathcal{L}$), a structured, persistent data object that serves as the global belief state for the system. Unlike unstructured context windows that simply append raw text, $\mathcal{L}$ is maintained as a curated list of *provenance-tracked epistemic claims*. Formally, each entry $c \in \mathcal{L}$ is assigned a discrete status $s(c)$ reflecting the system's current confidence. We use the following states:

- `[verified]`: Claims explicitly confirmed by one retrieved document (with citation).
- `[corroborated]`: Claims supported by multiple independent sources, increasing confidence.
- `[unverified]`: Claims extracted from secondary sources or context that lack primary provenance.
- `[contradicted]`: Hypotheses that have been explicitly refuted by evidence.
- `[hypothesis]`: Analytical inferences or logical bridges proposed by the model to guide future search.

While these status categories were initially derived through automatic prompt optimization using a GEPA [1] run over our architecture, we find that these emergent categories transfer across deep research tasks without modification (§4), suggesting they provide a sufficiently general structure for multi-hop discovery. However, though this is not guaranteed, in practice, this prompt optimization process can be rerun for a new backbone or environment.

Importantly, the primary architectural grounding of Dossier is the Research Ledger itself: a consolidated, evolving state object that surfaces contradictions, enables branch communication, and grounds future retrieval. In our implementation, GEPA (or any prompt, whether human-tuned or automatic) is used only to optimize the specific policies operating within this framework (i.e., how findings are synthesized, entries are merged, and information is extracted). Thus, we observe that GEPA regularizes ledger behavior for a specific model; the core mechanism is ledger-driven search rather than simple prompt engineering.

3.2 Branching Search and Query Generation

To navigate the corpus, Dossier employs a cyclic process. At step t, the **Gap Analysis module** receives the current question q, the state of the ledger $\mathcal{L}_{t-1}$, and the isolated findings, $\mathcal{F}_1, ..., \mathcal{F}_s$, from all the s branches of the previous step. It first reconciles these findings into a unified ledger state $\mathcal{L}_t$ before jointly analyzing the original question and the current evidence in $\mathcal{L}_t$ to identify **outstanding gaps**—specific facts required to answer the question that are currently missing or labeled `[unverified]`.

The module then generates a set of s sub-queries $\{q'_1, \ldots, q'_s\}$, each spawning an independent *branch* b_i. Every branch b_i executes Retrieval and Extraction independently, producing a local set of candidate findings $\mathcal{F}_i$. At the start of the next cycle, Gap Analysis receives all branch outputs $\{\mathcal{F}_1, \ldots, \mathcal{F}_s\}$ and reconciles them into the unified ledger state $\mathcal{L}_{t+1}$. This decomposition serves two purposes. First, it breaks multi-faceted questions into distinct, atomic retrieval tasks (e.g., separating a query about "hardware specs" from "release dates"), ensuring the dense retriever focuses on one semantic target at a time. Second, it allows the system to generate queries specifically designed to verify or refute `[hypothesis]` tags currently in the ledger (e.g., "Check if 5TB drives existed in 2013").

Following retrieval, each branch b_i independently executes an *Evidence Extraction* step, ingesting the raw retrieved documents and distilling them into candidate findings $\mathcal{F}_i$, each tagged with a provisional status and source citation.

Then, after some set number of iterations, the Final Synthesis component uses the ledger $\mathcal{L}$ to output a final answer.

3.3 Evidence-Aligned Query Learning (EAQL)

To effectively retrieve evidence for deep research, the retrieval mechanism must not be confined to searching individual sub-queries in isolation of the broader deep research state. In a multi-hop trajectory, a generated query is weak without the grounding context of the original research goal and the findings accumulated so far. Therefore, rather than encoding only the search string q_{sub}, we employ an *Evidence-Conditioned Query Encoder*.

Formally, the input to the encoder is a concatenation of the current sub-query (at the start and at the end), the original global question Q, and the linearized state of the Ledger $\mathcal{L}$:

$$\mathbf{e}_q = \text{Encoder}(q_{sub} \oplus Q \oplus \mathcal{L} \oplus q_{sub})$$

However, utilizing this context presents a challenge: standard dense retrievers are not trained to process such multi-part inputs [4, 8, 11, 12]. Off-the-shelf models, typically optimized for simple self-contained queries, often fail to use the relevant details in the ledger. To correct this behavior, we fine-tune the encoder on this composite input structure using full agentic trajectories.

Oracle-Guided Trajectory Generation. To ensure the retriever learns from accurate workflows, we generate training data by actively executing the agent's full reasoning loop against the corpus. During this execution, the system constructs an idealized trajectory where an oracle ensures that every plausible step yields valid evidence, allowing us to capture high-quality query-document pairs even for complex, late-stage hops.

Formally, at each hop we record a training snapshot $(q_{sub}, D_{ret}, D_{pos}, D_{neg})$, where q_{sub} is the generated sub-query, D_{ret} is the retrieved candidate pool, D_{pos} is the ground-truth evidence, and D_{neg} is a set of hard negatives mined from high-ranking but incorrect documents in D_{ret}. For each query, the retriever produces a candidate pool (e.g., top-50), from which the top-k documents are passed to the agent. If $D_{pos} \notin$ top-k but $D_{pos} \in D_{ret}$, the oracle injects D_{pos} into the agent's top-k context. This ensures the agent follows a valid trajectory while exposing failures in the dense retriever, enabling the collection of contrastive training samples.

Contrastive Optimization. We train the query encoder using a Multiple Negatives Ranking (MNR) Loss. For a minibatch of size B, each sub-query q_i is paired with one positive document p_i and K hard negatives $h_{i,1}, \ldots, h_{i,K}$ sampled from distraction documents D_{neg}. We also treat positive documents from other query–document pairs in the batch as additional negatives. These hard negatives teach the encoder to distinguish documents that provide new evidence from those that are only topically similar.

Table 1: Performance on a 400-question sample of BrowseComp-Plus across three models. Dossier consistently outperforms linear (ReAct) and hierarchy-based (Smolagents) baselines, as well as a parallelized ReAct variant that mimics branching without structured ledger state. Gains are largest on weaker backbones, such as a +24.8-point Accuracy improvement over ReAct on Qwen3-8B, but remain substantial on stronger models, such as 75.1% versus 53.9% on GPT-5-mini. Dossier uniformly wins on Accuracy, while Parallel ReAct attains higher recall than Dossier on GPT-5-mini and GPT-oss-20B.

Backbone	Method	Accuracy	Evidence Recall	Gold Recall	All-or-Nothing Recall
GPT-5-mini	ReAct	53.9% ± 2.5%	66.9% ± 1.8%	69.0% ± 2.1%	58.1% ± 2.5%
	Smolagents	53.5% ± 2.5%	67.4% ± 1.8%	68.1% ± 2.1%	58.2% ± 2.5%
	Parallel ReAct (Dossier-Inspired)	65.2% ± 2.4%	**84.0%** ± 1.4%	**84.2%** ± 1.7%	**78.2%** ± 2.1%
	Dossier	**75.1%** ± 2.2%	76.0% ± 1.7%	79.2% ± 1.9%	74.4% ± 2.2%
GPT-oss-20B	ReAct	39.0% ± 2.4%	57.6% ± 1.9%	58.1% ± 2.3%	48.5% ± 2.5%
	Smolagents	33.4% ± 2.4%	41.5% ± 1.8%	44.6% ± 2.2%	33.2% ± 2.4%
	Parallel ReAct (Dossier-Inspired)	44.4% ± 2.5%	**72.8%** ± 1.7%	**73.1%** ± 2.0%	**65.1%** ± 2.4%
	Dossier	**62.5%** ± 2.4%	67.0% ± 1.9%	69.9% ± 2.2%	63.5% ± 2.4%
Qwen3-8B	ReAct	8.2% ± 1.4%	10.1% ± 1.0%	9.2% ± 1.2%	4.8% ± 1.1%
	Smolagents	1.5% ± 0.6%	16.6% ± 1.3%	15.0% ± 1.6%	9.7% ± 1.5%
	Parallel ReAct (Dossier-Inspired)	23.8% ± 2.1%	20.9% ± 1.4%	19.7% ± 1.7%	12.5% ± 1.7%
	Dossier	**33.0%** ± 2.4%	**46.4%** ± 1.8%	**51.4%** ± 2.2%	**39.5%** ± 2.4%

Table 2: Results on a 400-question sample of HoVer, a multi-hop fact verification benchmark. Across both LLM and embedding models, Dossier improves Accuracy by +27–33.1 points over ReAct and increases All-or-Nothing Recall by +16.3–60 points. *The marked backbone is evaluated on only a 100-question subset due to time/resource constraints.

Backbone	Method	Accuracy	Evidence Recall	All-or-Nothing Recall
GPT-oss-20B LLM + Qwen3-8B Embedding	ReAct	56.7% ± 2.5%	78.3% ± 1.7%	61.5% ± 2.4%
	Smolagents	70.4% ± 2.3%	73.5% ± 1.4%	42.3% ± 2.5%
	Dossier (Ours)	**89.8%** ± 1.5%	**91.7%** ± 0.9%	**77.8%** ± 2.1%
Qwen3-4B LLM + Qwen3-0.6B Embedding*	ReAct	60%	50.1%	15%
	Smolagents	49%	65.3%	29%
	Dossier (Ours)	**87%**	**88.4%**	**75%**

4 Experiments

We evaluate on two closed-corpus benchmarks to ensure controlled, reproducible comparisons and to enable query-encoder training. Dossier could naturally be used in an open-web setting where retrieved webpages would be treated as documents:

BrowseComp-Plus (BC+). BC+ is a controlled proxy for realistic web research derived from OpenAI's BrowseComp deep research task [2, 24]. Here questions are engineered to be hard for keyword search (rare conjunctions of constraints) but straightforward to verify once the right sources are located. Unlike live browsing, BC+ replaces the web with a fixed corpus (≈ 100k documents), enabling deterministic evaluation and reproducible rollouts. This static setting is essential for our training procedure. BC+ also includes mined hard negatives—topically related documents that do not resolve the missing constraint—which are critical for EAQL.

HoVer. HoVer [6] is a multi-hop verification benchmark built from Wikipedia abstracts, where each claim requires aggregating evidence from up to four disjoint articles. Unlike entity-answer QA, the output is a binary label (True or not True).

We report multiple retrieval-focused metrics and one end-to-end metric, with minor dataset-specific instantiations. First, **Evidence Recall:** fraction of annotated supporting evidence documents retrieved in the trajectory, macro-averaged over examples. Second, **Gold Recall (BC+ only)**: fraction of answer-containing document(s) which are retrieved (definition aligned to BC+ annotations). Third, **All-or-Nothing Recall:** percentage of questions where all required evidence documents are retrieved at least once (0 or 1 per question). And, lastly, **Accuracy:** answer correctness in BC+ and verification label correctness in HoVer.

These metrics are complementary: Evidence Recall measures coverage, All-or-Nothing captures completeness for multi-hop constraints, and Accuracy reflects whether the system ultimately synthesizes or verifies correctly. A question can be correct without retrieving all annotated evidence, since models may rely on partial evidence or parametric knowledge, but this reduces transparency and robustness, motivating emphasizing recall alongside accuracy.

4.1 Baselines

We compare Dossier against representative agentic baselines under a fixed compute budget. For all methods, we allocate a total of 100

Table 3: Ablation of learning mechanisms on a random sample of BC-Plus (GPT-oss-20B). A ledger-conditioned encoder without learning underperforms the default encoder (49% vs. 51% Accuracy), showing that naive conditioning is not sufficient. Evidence-Aligned Query Learning (EAQL) yields large improvements (Evidence Recall: 61.3%→64.4%; Gold Recall: 61.4% → 70.1%; All-or-Nothing Recall: 52% → 66%), nearly matching the Oracle upper bound (66% Accuracy). These highlight EAQL as a substantial contributor to overall performance.

Strategy	Accuracy	Evidence Recall	Gold Recall	All-or-Nothing Recall
Default Encoder	51%	61.3%	61.4%	52%
Evidence-Conditioned Query Encoder + no-learning	49%	53.6%	52.5%	48%
Evidence-Conditioned Query Encoder + EAQL Learning	64%	64.4%	70.1%	66%
Oracle	66%	80.4%	82.7%	76%

LLM calls per query for BC+ and 30 for HoVer, following prior work [2, 9]. This constraint ensures that improvements arise from architectural design rather than increased compute.

ReAct. ReAct [27] operates as a single linear reasoning-and-acting loop in which the model alternates between generating a thought, issuing a search query, and appending retrieved snippets into a growing context window. The full 100-call budget is used within this single trajectory, requiring the agent to manage long-horizon reasoning entirely through unstructured prompt history, making it susceptible to context saturation and early-error compounding.

For BC+, we use the official implementation and default prompts released with the benchmark [2]. Consistent with the benchmark, we provide the agent with trimmed document previews and a `get_document` tool for retrieving full document contents when needed. For HoVer, we use GEPA [1] to create the prompt and provide the model with full previews as the maximum document size is a few hundred tokens.

Smolagents Open Deep Research. This framework implements a manager–subagent architecture, where a controller decomposes tasks and delegates retrieval subtasks to worker agents through message-based coordination. We additionally equip the system with document summarization and history-compaction tools to mitigate context window growth, ensuring the baseline benefits from standard engineering best practices; however, coordination still occurs through unstructured conversational messages rather than a persistent, structured global state. The total interaction across manager and workers is constrained to the same 100-call budget [20].

Parallel ReAct (Dossier-inspired). To isolate the effect of parallelization alone, we run five independent ReAct agents in parallel and aggregate their outputs (e.g., union of retrieved evidence or majority voting over answers). To ensure fairness, the total 100-call budget is evenly divided across agents, giving each ReAct instance 20 calls. This baseline tests whether simply increasing search breadth via parallel trajectories is sufficient, or whether Dossier's explicit Research Ledger and coordinated gap analysis provide additional benefit beyond raw parallel exploration. We report the union of the retrieved documents for recall and a successful accuracy if any of the 5 are correct (benefiting the baseline).

4.2 Implementation Details

Dossier configuration. Dossier uses branching factor s = 5 and retrieves k = 10 documents per sub-query per hop.

Backbones and retrievers. We evaluate across multiple LLM backbones [15, 19, 23] (Table 1; Table 2) and pair them with dense retrievers as indicated in the tables. The query encoder is initialized from the respective encoders and fine-tuned using Evidence-Aligned Query Learning on 75 training questions with a batch size of 8, a learning rate of 5×10^{-6}, and 1 epoch. Each query is paired with 9 hard negatives sampled from the top-retrieved documents, and training uses a contrastive loss with temperature $\tau = 1$. We validate every 400 minibatches on a fixed set of 10 held-out questions and save the best checkpoint by evidence recall@10.

Prompt optimization (GEPA). We apply GEPA to tune the Dossier prompts. We also evaluate prompt transfer across tasks to test whether optimized prompting is dataset-specific or captures more general deep-research behavior. On both benchmarks, we use a training set of 20 questions and a validation set of 15 questions. For BC+, we optimize a reward signal that assigns a value of 1 to correct answers and otherwise uses Evidence Recall as the fallback objective from 0-1. Since HoVer is a binary classification task, we only use Evidence Recall. Our DSPy signatures are in Appendix A and represent the "baseline prompts" [10].

Evaluation uncertainty. In Table 1 and 2, we evaluate each method on $N = 400$ held-out questions. Each question is evaluated once with a complete agent rollout, producing a per-question score x_i. For Accuracy and All-or-Nothing Recall, x_i is binary; for Evidence and Gold Recall, x_i is the per-question recall value. We report the mean score together with the standard error of the mean, i.e., $100\bar{x} \pm 100\mathrm{SE}(\bar{x})$ in percentage points, where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$, $\mathrm{SE}(\bar{x}) = \frac{s}{\sqrt{N}}$, and $s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$. This estimates the sampling variability of the reported aggregate result.

4.3 Main Results

Table 1 summarizes performance on BC+ across backbones. Dossier achieves the best end-to-end accuracy across the reported backbones, improving substantially over linear ReAct and hierarchical baselines. We present 3 main takeaways from Table 1 and Table 2:

Deep research benefits from explicit global state beyond raw retrieval. Across backbones, systems with similar Evidence Recall can exhibit dramatically different Accuracy. For example, on BC+ with GPT-oss-20B, Parallel ReAct and Dossier achieve close All-or-Nothing Recall (65.1% vs. 63.5%), yet differ by 18.1 points in Accuracy (44.4% vs. 62.5%).

We believe this indicates that deep research performance depends not only on surfacing relevant documents, but on extracting and synthesizing the correct information from them. Dossier's ability to extract information from documents independently before combining multiple different lines of inquiry allows for structured reconciliation across long trajectories.

External state can compensate for limited backbone capacity. The advantage of Dossier becomes more pronounced as model capacity decreases. On BC+ with Qwen3-8B, Dossier improves Accuracy from 23.8% (Parallel ReAct) and 8.2% (ReAct) to 33.0%, while also substantially increasing Evidence Recall (20.9% → 46.4%). Similarly, improvements can be seen in the main HoVer results in Table 2, where Dossier has a particularly pronounced performance gap compared to the other backbones when using the smaller Qwen3-4B + Qwen3-0.6B Embedding. This widening gap suggests that explicit state tracking reduces reliance on long-context internal reasoning, effectively offloading constraint management from the LLM into a structured external memory.

Parallel search alone does not yield coordinated reasoning. Increasing search breadth improves recall (e.g., 84.0% Evidence Recall for Parallel ReAct on GPT-5-mini), but does not reliably convert into higher Accuracy, indicating independent trajectories lack a shared representation for reconciling partial findings or resolving contradictions. Dossier's unified Research Ledger synchronizes branches into a single global viewpoint, allowing parallel exploration to translate into coherent multi-constraint reasoning.

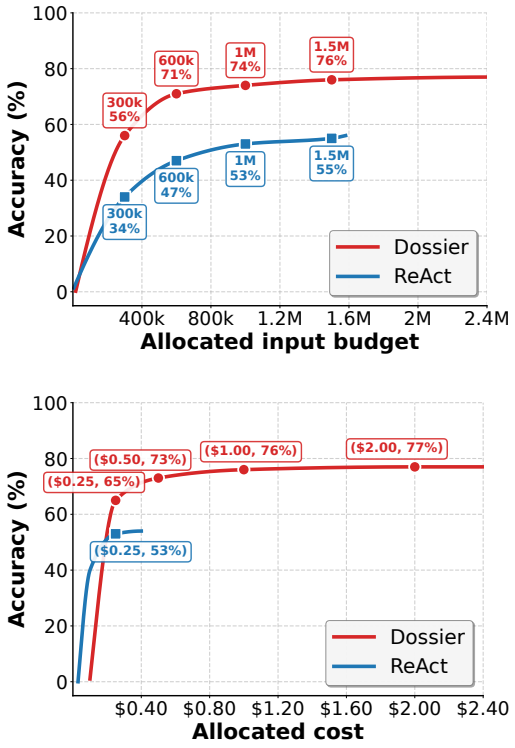


Figure 3: Accuracy as a function of cumulative input tokens (up) and cost (below) on 100 questions of BC+. Dossier reaches higher accuracy than ReAct at comparable token/cost budgets and achieves strong performance after only a small number of iterations.

4.3.1 Efficiency and Trajectory Cost. Fig. 3 shows cost by running answer generation after each iteration and measuring cumulative tokens consumed up to that iteration. At the same token budget, Dossier substantially outperforms ReAct and achieves strong performance with few iterations. Appendix C compares cost when matching LLM calls.

4.4 Ablation Studies

For the following subsections, we report results on a 100-question subset of the earlier 400-question subset for both BC+ and HoVer.

Table 4: Overall system ablation from ReAct to Dossier on BC+ using GPT-oss-20B. Performance improves as the system moves from a linear trajectory to the full Dossier execution pattern: persistent ledger state, always-on ledger updates, coordinated branching, and EAQL.

Method	Accuracy	Evidence Recall	Gold Recall	All-or-Nothing Recall
ReAct – benchmark prompts	33.0%	52.7%	55.8%	49.0%
ReAct – optimized prompts	32.0%	43.0%	45.2%	34.0%
Dossier – 1 branch	39.0%	43.5%	46.5%	38.0%
Dossier – 5 branches	51.0%	61.3%	61.4%	52.0%
Dossier – EAQL, 5 branches	64.0%	64.4%	70.1%	66.0%

4.4.1 Overall System Ablation. **Each component is incrementally beneficial where a single addition can provide immediate gains.** We conduct an incremental ablation starting from a standard ReAct baseline on BC+ in Table 4. We observe that optimizing prompts for the linear ReAct loop provides no significant benefit, with accuracy stagnating at 32-33%. Transitioning to a single-branch Dossier configuration improves accuracy to 39% despite nearly identical evidence recall to the optimized ReAct baseline (43.5% vs. 43.0%). This suggests that the Research Ledger improves the agent's ability to synthesize and extract information from retrieved documents even without the benefit of parallel search. The introduction of 5-way branching provides the most substantial performance boost, increasing evidence recall from 43.5% to 61.3% and boosting accuracy to 51%. Finally, Evidence-Aligned Query Learning (EAQL) provides the full performance of 64% accuracy.

4.4.2 Impact of Learning Mechanisms. Tables 3 and 8 isolate the effect of (i) conditioning retrieval on the ledger and (ii) training the query encoder with Evidence-Aligned Query Learning. We observe the following takeaways:

Conditioning without learning can hurt. The "Evidence-Conditioned Query Encoder (no-learning)" underperforms the default encoder (Table 3). This supports the core motivation: off-the-shelf embedding models are not trained to interpret long, structured ledger context. Naively concatenating the ledger can cause the encoder to be distracted by the ledger, which degrades retrieval.

Evidence-Aligned Query Learning restores and improves retrieval utility. Once the encoder is trained on oracle-guided rollouts, retrieval becomes state-aware in the intended sense: it ranks gap-filling documents above merely topically similar documents already represented in the ledger. This is reflected most in All-or-Nothing Recall and Accuracy gains (for example, +15%, Table 8), the metrics most sensitive to missing a single critical hop.

4.4.3 Structured Ledger Behavior vs. Unstructured Note Passing. As we mentioned in §3.1, we do not impose an absolute structure on the system or the research ledger. So here we evaluate the impact of GEPA-tuned prompts (Table 5) and the structure's ability to transfer across different tasks (Table 6). We see the following:

GEPA enforces structured ledger behavior rather than improving retrieval. When retrieval is held fixed on BC+, GEPA substantially improves Accuracy (54.0% → 64.0%) and All-or-Nothing Recall (61.0% → 66.0%) while leaving Evidence Recall unchanged (64.4% → 64.4%). This suggests that the additional structure imposed

Table 5: Impact of GEPA Prompt Optimization on BC+ using GPT-oss-20B. We observe that for Dossier, GEPA improves Accuracy and All-or-Nothing Recall without affecting Evidence Recall. This indicates that better prompting and structure enhances ledger management and information extraction rather than retrieval ranking alone. GEPA only partially explains Dossier's gains and provides only a modest improvement for Parallel ReAct.

Prompt Setting	Method	Accuracy	Evidence Recall	Gold Recall	All-or-Nothing Recall
No GEPA	ReAct	33.0%	52.7%	55.8%	49.0%
	Parallel ReAct	38.0%	63.8%	65.1%	55.0%
	Dossier	**54.0%**	**64.4%**	**71.1%**	**61.0%**
GEPA	ReAct	32.0%	43.0%	45.2%	34.0%
	Parallel ReAct	40.0%	**66.6%**	**70.8%**	57.0%
	Dossier	**64.0%**	64.4%	70.1%	**66.0%**

Table 6: Impact of GEPA under Oracle Retrieval (GPT-oss-20B). Both GEPA variants achieve nearly identical performance, indicating that once sufficient prompts are available, gains from modifying the prompts governing the ledger are marginal. This suggests the ledger design generalizes across deep research tasks.

Strategy	Accuracy	Evidence Recall	Gold Recall	All-or-Nothing Recall
HoVer Oracle Results				
GEPA on BC+ (Dossier)	91.0%	96.6%	96.0%	91.0%
GEPA on HoVer (Dossier)	89.0%	96.2%	95.7%	89.0%

by the prompt strengthens how retrieved evidence is incorporated into the Research Ledger.

Without GEPA, the default ledger will degrade into a series of loosely structured notes passed between components; while technically persistent, such unstructured memory fails to reliably enforce constraint reconciliation. The improvements therefore demonstrate that Dossier's effectiveness depends on disciplined, schema-consistent ledger updates rather than mere memory accumulation.

The ledger abstraction generalizes beyond benchmark-specific prompting. Under oracle retrieval (Table 6), GEPA tuned on BC+ and GEPA tuned on HoVer achieve nearly identical performance (e.g., 96.6% vs. 96.2% Evidence Recall and 91.0% vs. 89.0% Accuracy on HoVer). The small remaining gaps are likely attributable to differences in GEPA optimization runs (e.g., prompt initialization and search stochasticity) rather than structural mismatch. The near-alignment across datasets suggests that improvements stem from enforcing general deep-research principles—explicit gap tracking, contradiction marking, and consistent ledger updates—rather than dataset-specific heuristics. This suggests that the Research Ledger is a fundamental architectural abstraction, with prompt optimization regularizing its behavior rather than redefine it per benchmark.

Table 7: Impact of Branching Factor on BC-Plus. Increasing width to a factor of 5 improves recall but incurs higher cost.

Branch	Accuracy	Evidence Recall	Gold Recall	All-or-Nothing Recall
1	52.0%	54.9%	61.8%	52.0%
3	60.0%	63.4%	68.8%	62.0%
5	**64.0%**	**64.4%**	**70.1%**	**66.0%**
7	62.0%	63.7%	68.1%	62.0%

4.4.4 Impact of Branching Factor. In Table 7, we vary s, the maximum number of subqueries allowed in an iteration. Importantly, we do not adjust the number of global iterations, so a branching factor of 1 has roughly 1/5 the amount of LLM calls as a branching factor of 5. We observe that increasing the branching factor to a moderate width (5) improves both recall and accuracy by allowing the system to pursue independent evidence gaps in parallel. This minimizes the risk of a single poor query derailing the entire trajectory. However, expanding beyond a moderate width causes performance gains to saturate or decline where branches converge on similar evidence.

5 Trajectory Analysis

This section provides trajectory-level analysis for BC+ comparing the agentic trajectories using GPT-5-mini traces.

5.1 Baseline: Context Saturation and Giving Up

> **Abridged Query:** *Actor (as of Dec. 31, 2023): taxicab driver; TV debut in 2004; girlfriend played volleyball; appeared in a film with a Disney Channel star; switched schools at least 10 times.*
> **Ground Truth:** Cory Monteith

We observe that both ReAct and Smolagents process evidence linearly. When they retrieve documents verifying some constraints but failing others, they struggle to generate targeted subqueries for the missing pieces. As their context window fills with failed attempts, they prematurely abandon the hypothesis and once partial matches appear without closure, the trajectory devolves into broad, repetitive searches. For example, ReAct starts with an over-constrained query which attempting answering the whole question at once:

> **ReAct search query:** actor was a taxicab driver TV debut 2004 girlfriend used to play volleyball switched schools 10 times 'taxicab driver' 'switched schools' 'volleyball' girlfriend 'TV debut 2004' actor

When this "mega-query" fails to return a single golden document (non-existent), ReAct correctly breaks the query down into slightly smaller chunks (Step 3: `"switched schools" "10 times" actor interview`). However, even when it retrieves correct documents, it fails to link these back to a clear hypothesis, continually trying random partial searches. After 26 steps and reading 120 documents, it gives up entirely, asking the user for help.

Smolagents shows a complementary failure mode: extreme breadth without convergence. The manager continues to delegate aggressively (nearly 600 documents in 23 steps), but the coordination fails to converge, also yielding a "no-solution" judgment.

In contrast, Dossier surfaces Cory Monteith early but refuses to finalize until all constraints are verified or explicitly marked as unresolved. This behavior is driven by the *Research Ledger*:

> **Ledger (Iteration 1):**
> `[verified]` The profile titled "Monteith's Life: In His Own Words" states he worked as a roofer and a taxi driver (docid: 77808)
> `[contradicted]` The requirement that the actor's TV debut occurred in 2004 is contradicted; the context places his first TV credit in 2006
> `[outstanding gap]` There is no evidence in the context that any actor switched schools at least 10 times
> `[hypothesis]` The complete set of personal details is likely contained in a single feature interview or long-form profile

`[outstanding gap]` No context document confirms that Monteith's girlfriend (Lea Michele) used to play volleyball

Because those unresolved constraints are codified as `[outstanding gap]` entries, they shape the next hop's subqueries, Dossier very quickly transitions from broad discovery to highly targeted searches. For instance, in the next iteration, subquery (0) closes the last outstanding gap in the ledger above:

Dossier subqueries (Iteration 2):
`(0)` "Lea Michele" volleyball roster OR "played volleyball" biography
`(1)` Cory Monteith "television debut" 2004 OR earliest TV appearance
`(2)` "Cory Monteith" "switched schools" interview

By the 10th iteration, Dossier resolves the gaps and gracefully outputs the correct answer, recovering from an initial state of uncertainty that completely doomed the baseline agents.

5.2 Recall ≠ Accuracy

An important observation is that Dossier enforces quality evidence synthesis instead of only improving document retrieval. In the following trace, both ReAct and Smolagents achieve 100% evidence recall, yet confidently return the wrong entity. The failure is therefore not due to missing information, but to how the retrieved evidence is analyzed and integrated.

Abridged Query: *2010s international soccer match: winning team by >2 goals; 40k+ attendance; pre-match injuries; a substitute scored after <10 goals on loan in the prior season. Which loan club?*
Ground Truth: FC Arsenal Tula

Both ReAct and Smolagents retrieve the decisive documents but still answer incorrectly. This failure is not due to missing evidence, but to the absence of a structured *falsification* step in the trajectory.

In fact, both systems correctly identified that the match is Russia vs. Saudi Arabia (2018 World Cup). However, two substitutes scored for Russia: Denis Cheryshev and Artem Dzyuba. The baselines "lock-on" to Cheryshev who had a loan at a different team (Villarreal) a few years prior. However, despite future documents verifying Artem Dzyuba (who spent the prior season on loan at FC Arsenal Tula), the unstructured trajectories lack a mechanism to refute the plausible partial match.

Instead, Dossier is specifically designed to work with this imperfect hypothesis but does not allow synthesis to proceed unless the ledger can satisfy all constraints coherently:

`[verified]` The Russia 5–0 Saudi Arabia World Cup opener on 14 June 2018 had an attendance of 78,011 and a five-goal winning margin
`[contradicted]` The requirement that the match included more than 30 fouls is contradicted by the Russia v Saudi Arabia report, which records only ten fouls
`[outstanding gap]` None of the provided documents contain pre-match injury lists confirming a defender who made more than 25 league appearances

Dossier avoids early commitment because of the `[contradicted]` foul count (it had retrieved incomplete stats). Over 10 hops, it finds the correct comprehensive match report, resolves all constraints, and correctly synthesizes Dzyuba's loan.

5.3 The Limits of Parallel Breadth

Query Prompt: *As of Dec. 2023, identify the historical structure: erected by a renowned Portuguese navigator; named in his honor; first arrival in a non-Christian region; located ∼1 mile from a fort museum.*
Ground Truth: Magellan's Cross

Parallel ReAct is intended to isolate the effect of exploration breadth by running five independent trajectories and unioning their results. In this setting, each run is executed with high temperature, producing materially different reasoning traces and query sequences. This example shows that even substantial stochastic diversity does not correct semantic drift in unstructured agents.

Despite following distinct trajectories, all five runs failed to retrieve Magellan's Cross and instead converged on the same semantically similar but factually incorrect distractor: Vasco da Gama Pillar (Malindi, Kenya). The single-run ReAct baseline made the identical error.

The failure is therefore not due to insufficient exploration or lack of diversity, but to missing coordination and constraint enforcement. Once a trajectory retrieves Vasco da Gama Pillar—a Portuguese explorer's marker near a fort—it satisfies superficial aspects of the query and becomes self-reinforcing. Because each branch reasons in isolation, parallelism simply replicates the same confirmation-biased attractor under different surface-level reasoning paths.

Increasing parallel breadth improves recall coverage but does not address state management or global constraint reconciliation. In contrast, Dossier synchronizes evidence through a structured ledger and anchors on Magellan's Cross at Hop-1 with near-perfect evidence recall, demonstrating that the correct documents were fully accessible; what the baselines lacked was structured cross-branch synchronization and explicit falsification.

6 Conclusion

We introduced Dossier, a ledger-driven deep research architecture that replaces linear, unstructured trajectories with structured branching search over a persistent Research Ledger, combined with Evidence-Aligned Query Learning to train state-aware retrieval. On BrowseComp-Plus, Dossier improved over ReAct by +23 percentage points and over Smolagents Deep Research by +27 points. Likewise, on HoVer, Dossier improved over ReAct by +30 points and Smolagents Deep Research by +29 points.

Acknowledgments

This work was partially funded by the MIT Generative AI Impact Consortium (MGAIC), the Laude Institute, Quanta Computer Inc. under the AIR Project, and NSF grants IIS-2142827 and IIS-2234058.

References

[1] Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. 2025. Gepa: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457* (2025).

[2] Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, et al. 2025. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600* (2025).

[3] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems* 36 (2023), 28091–28114.

[4] Jorge Gabín, Javier Parapar, and Craig Macdonald. 2026. Beyond questions: Leveraging ColBERT for keyphrase search. *Information Processing & Management* 63, 2 (2026), 104480. doi:10.1016/j.ipm.2025.104480

[5] Boyu Gou, Zanming Huang, Yuting Ning, Yu Gu, Michael Lin, Weijian Qi, Andrei Kopanev, Botao Yu, Bernal Jiménez Gutiérrez, Yiheng Shu, et al. 2025. Mind2web 2: Evaluating agentic search with agent-as-a-judge. *arXiv preprint arXiv:2506.21506* (2025).

[6] Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 3441–3460.

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

[8] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://aclanthology.org/2020.emnlp-main.550/

[9] Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. *Advances in Neural Information Processing Systems* 34 (2021), 27670–27682.

[10] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714* (2023).

[11] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. https://arxiv.org/abs/2004.12832

[12] Millicent Li, Tongfei Chen, Benjamin Van Durme, and Patrick Xia. 2025. Multi-Field Adaptive Retrieval. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=3PDklqqqfN

[13] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. 5420–5438.

[14] Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. 2025. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776* (2025).

[15] OpenAI. 2025. gpt-oss-120b & gpt-oss-20b Model Card. arXiv:2508.10925 [cs.CL] https://arxiv.org/abs/2508.10925

[16] Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. 2024. Adapt: As-needed decomposition and planning with language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*. 4226–4252.

[17] Peng Qi, Haejun Lee, Tg Sido, and Christopher D Manning. 2021. Answering open-domain questions of varying reasoning steps from text. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 3599–3614.

[18] Zile Qiao, Guoxin Chen, Xuanzhong Chen, Donglei Yu, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, et al. 2025. Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents. *arXiv preprint arXiv:2509.13309* (2025).

[19] Qwen Team. 2025. Qwen3 Technical Report. arXiv:2505.09388 [cs.CL] https://arxiv.org/abs/2505.09388

[20] Aymeric Roucher, Albert Villanova del Moral, Thomas Wolf, Leandro von Werra, and Erik Kaunismäki. 2025. 'smolagents': a smol library to build great agentic systems. https://github.com/huggingface/smolagents.

[21] Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. 2024. Assisting in writing wikipedia-like articles from scratch with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 6252–6278.

[22] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems* 36 (2023), 8634–8652.

[23] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, Alex Makelov, Alex Neitz, Alex Wei, Alexandra Barr, Alexandre Kirchmeyer, Alexey Ivanov, Alexi Christakis, Alistair Gillespie, Allison Tam, Ally Bennett, Alvin Wan, Alyssa Huang, Amy McDonald Sandjideh, Amy Yang, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrei Gheorghe, Andres Garcia Garcia, Andrew Braunstein, Andrew Liu, Andrew Schmidt, Andrey Mereskin, Andrey Mishchenko, Andy Applebaum, Andy Rogerson, Ann Rajan, Annie Wei, Anoop Kotha, Anubha Srivastava, Anushree Agrawal, Arun Vijayvergiya, Ashley Tyra, Ashvin Nair, Avi Nayak, Ben Eggers, Bessie Ji, Beth Hoover, Bill Chen, Blair Chen, Boaz Barak, Borys Minaiev, Botao Hao, Bowen Baker, Brad Lightcap, Brandon McKinzie, Brandon Wang, Brendan Quinn, Brian Fioca, Brian Hsu, Brian Yang, Brian Yu, Brian Zhang, Brittany Brenner, Callie Riggins Zetino, Cameron Raymond, Camillo Lugaresi, Carolina Paz, Cary Hudson, Cedric Whitney, Chak Li, Charles Chen, Charlotte Cole, Chelsea Voss, Chen Ding, Chen Shen, Chengdu Huang, Chris Colby, Chris Hallacy, Chris Koch, Chris Lu, Christina Kaplan, Christina Kim, CJ Minott-Henriques, Cliff Frey, Cody Yu, Coley Czarnecki, Colin Reid, Colin Wei, Cory Decareaux, Cristina Scheau, Cyril Zhang, Cyrus Forbes, Da Tang, Dakota Goldberg, Dan Roberts, Dana Palmie, Daniel Kappler, Daniel Levine, Daniel Wright, Dave Leo, David Lin, David Robinson, Declan Grabb, Derek Chen, Derek Lim, Derek Salama, Dibya Bhattacharjee, Dimitris Tsipras, Dinghua Li, Dingli Yu, DJ Strouse, Drew Williams, Dylan Hunn, Ed Bayes, Edwin Arbus, Ekin Akyurek, Elaine Ya Le, Elana Widmann, Eli Yani, Elizabeth Proehl, Enis Sert, Enoch Cheung, Eri Schwartz, Eric Han, Eric Jiang, Eric Mitchell, Eric Sigler, Eric Wallace, Erik Ritter, Erin Kavanaugh, Evan Mays, Evgenii Nikishin, Fangyuan Li, Felipe Petroski Such, Filipe de Avila Belbute Peres, Filippo Raso, Florent Bekerman, Foivos Tsimpourlas, Fotis Chantzis, Francis Song, Francis Zhang, Gaby Raila, Garrett McGrath, Gary Briggs, Gary Yang, Giambattista Parascandolo, Gildas Chabot, Grace Kim, Grace Zhao, Gregory Valiant, Guillaume Leclerc, Hadi Salman, Hanson Wang, Hao Sheng, Haoming Jiang, Haoyu Wang, Haozhun Jin, Harshit Sikchi, Heather Schmidt, Henry Aspegren, Honglin Chen, Huida Qiu, Hunter Lightman, Ian Covert, Ian Kivlichan, Ian Silber, Ian Sohl, Ibrahim Hammoud, Ignasi Clavera, Ikai Lan, Ilge Akkaya, Ilya Kostrikov, Irina Kofman, Isak Etinger, Ishaan Singal, Jackie Hehir, Jacob Huh, Jacqueline Pan, Jake Wilczynski, Jakub Pachocki, James Lee, James Quinn, Jamie Kiros, Janvi Kalra, Jasmyn Samaroo, Jason Wang, Jason Wolfe, Jay Chen, Jay Wang, Jean Harb, Jeffrey Han, Jeffrey Wang, Jennifer Zhao, Jeremy Chen, Jerene Yang, Jerry Tworek, Jesse Chand, Jessica Landon, Jessica Liang, Ji Lin, Jiancheng Liu, Jianfeng Wang, Jie Tang, Jihan Yin, Joanne Jang, Joel Morris, Joey Flynn, Johannes Ferstad, Johannes Heidecke, John Fishbein, John Hallman, Jonah Grant, Jonathan Chien, Jonathan Gordon, Jongsoo Park, Jordan Liss, Jos Kraaijeveld, Joseph Guay, Joseph Mo, Josh Lawson, Josh McGrath, Joshua Vendrow, Joy Jiao, Julian Lee, Julie Steele, Julie Wang, Junhua Mao, Kai Chen, Kai Hayashi, Kai Xiao, Kamyar Salahi, Kan Wu, Karan Sekhri, Karan Sharma, Karan Singhal, Karen Li, Kenny Nguyen, Keren Gu-Lemberg, Kevin King, Kevin Liu, Kevin Stone, Kevin Yu, Kristen Ying, Kristian Georgiev, Kristie Lim, Kushal Tirumala, Kyle Miller, Lama Ahmad, Larry Lv, Laura Clare, Laurance Fauconnet, Lauren Itow, Lauren Yang, Laurentia Romaniuk, Leah Anise, Lee Byron, Leher Pathak, Leon Maksin, Leyan Lo, Leyton Ho, Li Jing, Liang Wu, Liang Xiong, Lien Mamitsuka, Lin Yang, Lindsay McCallum, Lindsey Held, Liz Bourgeois, Logan Engstrom, Lorenz Kuhn, Louis Feuvrier, Lu Zhang, Lucas Switzer, Lukas Kondraciuk, Lukasz Kaiser, Manas Joglekar, Mandeep Singh, Mandip Shah, Manuka Stratta, Marcus Williams, Mark Chen, Mark Sun, Marselus Cayton, Martin Li, Marvin Zhang, Marwan Aljubeh, Matt Nichols, Matthew Haines, Max Schwarzer, Mayank Gupta, Meghan Shah, Melody Huang, Meng Dong, Mengqing Wang, Mia Glaese, Micah Carroll, Michael Lampe, Michael Malek, Michael Sharman, Michael Zhang, Michele Wang, Michelle Pokrass, Mihai Florian, Mikhail Pavlov, Miles Wang, Ming Chen, Mingxuan Wang, Minnia Feng, Mo Bavarian, Molly Lin, Moose Abdool, Mostafa Rohaninejad, Nacho Soto, Natalie Staudacher, Natan LaFontaine, Nathan Marwell, Nelson Liu, Nick Preston, Nick Turley, Nicklas Ansman, Nicole Blades, Nikil Pancha, Nikita Mikhaylin, Niko Felix, Nikunj Handa, Nishant Rai, Nitish Keskar, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Oona Gleeson, Pamela Mishkin, Patryk Lesiewicz, Paul Baltescu, Pavel Belov, Peter Zhokhov, Philip Pronin, Phillip Guo, Phoebe Thacker, Qi Liu, Qiming Yuan, Qinghua Liu, Rachel Dias, Rachel Puckett, Rahul Arora, Ravi Teja Mullapudi, Raz Gaon, Reah Miyara, Rennie Song, Rishabh Aggarwal, RJ Marsan, Robel Yemiru, Robert Xiong, Rohan Kshirsagar, Rohan Nuttall, Roman Tsiupa, Ronen Eldan, Rose Wang, Roshan James, Roy Ziv, Rui Shu, Ruslan Nigmatullin, Saachi Jain, Saam Talaie, Sam Altman, Sam Arnesen, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Sarah Yoo, Savannah Heon, Scott Ethersmith, Sean Grove, Sean Taylor, Sebastien Bubeck, Sever Banesiu, Shaokyi Amdo, Shengjia Zhao, Sherwin Wu, Shibani Santurkar, Shiyu Zhao, Shraman Ray Chaudhuri, Shreyas Krishnaswamy, Shuaiqi, Xia, Shuyang Cheng, Shyamal Anadkat, Simón Posada Fishman, Simon Tobin, Siyuan Fu, Somay Jain, Song Mei, Sonya

Egoian, Spencer Kim, Spug Golden, SQ Mah, Steph Lin, Stephen Imm, Steve Sharpe, Steve Yadlowsky, Sulman Choudhry, Sungwon Eum, Suvansh Sanjeev, Tabarak Khan, Tal Stramer, Tao Wang, Tao Xin, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Degry, Thomas Shadwell, Tianfu Fu, Tianshi Gao, Timur Garipov, Tina Sriskandarajah, Toki Sherbakov, Tomer Kaftan, Tomo Hiratsuka, Tongzhou Wang, Tony Song, Tony Zhao, Troy Peterson, Val Kharitonov, Victoria Chernova, Vineet Kosaraju, Vishal Kuo, Vitchyr Pong, Vivek Verma, Vlad Petrov, Wanning Jiang, Weixing Zhang, Wenda Zhou, Wenlei Xie, Wenting Zhan, Wes McCabe, Will DePue, Will Ellsworth, Wulfie Bain, Wyatt Thompson, Xiangning Chen, Xiangyu Qi, Xin Xiang, Xinwei Shi, Yann Dubois, Yaodong Yu, Yara Khakbaz, Yifan Wu, Yilei Qian, Yin Tat Lee, Yinbo Chen, Yizhen Zhang, Yizhong Xiong, Yonglong Tian, Young Cha, Yu Bai, Yu Yang, Yuan Yuan, Yuanzhi Li, Yufeng Zhang, Yuguang Yang, Yujia Jin, Yun Jiang, Yunyun Wang, Yushi Wang, Yutian Liu, Zach Stubenvoll, Zehao Dou, Zheng Wu, and Zhigang Wang. 2025. OpenAI GPT-5 System Card. arXiv:2601.03267 [cs.CL] https://arxiv.org/abs/2601.03267

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

[25] Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, et al. 2025. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648* (2025).

[26] Wenhan Xiong, Xiang Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, et al. 2021. Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval. In *International Conference on Learning Representations*.

[27] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

[28] Alex L Zhang, Tim Kraska, and Omar Khattab. 2025. Recursive language models. *arXiv preprint arXiv:2512.24601* (2025).

A DSPy Signatures

Listing 1: DSPy Signatures Used in Dossier

```
class GenerateSubquerySignature(dspy.Signature):
    question: str = dspy.InputField(
        desc="The original question being asked"
    )
    notes: List[str] = dspy.InputField(
        desc="A cummulation of knoweldge so far which may be useful for generating the next subquery"
    )
    past_subqueries: List[str] = dspy.InputField(
        desc="A list of previous subqueries asked so far"
    )
    updated_notes: List[str] = dspy.OutputField(
        desc="An updated notes which is summarized for your plan"
    )
    subqueries: List[str] = dspy.OutputField(
        desc="A list of subqueries to ask in future steps. The subqueries should be as far from each
            other as possible. Should be exactly 5"
    )


class AppendNotesSignature(dspy.Signature):
    question: str = dspy.InputField(
        desc="The original question being asked"
    )
    notes: List[str] = dspy.InputField(
        desc="A cummulation of knoweldge so far which may be useful for generating the next subquery"
    )
    subquery: str = dspy.InputField(
        desc="The subquery that was just asked to the search engine"
    )
    context: List[str] = dspy.InputField(
        desc="The list of documents returned from the search engine for the given subquery"
    )
    new_notes: List[str] = dspy.OutputField(
        desc="The new notes to append to the existing notes"
    )
    titles: List[str] = dspy.OutputField(
        desc="The titles of the documents retrieved from the search engine"
    )


class FinalAnswerSignature(dspy.Signature):
    question: str = dspy.InputField(
        desc="The original question being asked"
    )
    titles: List[str] = dspy.InputField(
        desc="The titles of all documents retrieved during the hops"
    )
    notes: List[str] = dspy.InputField(
        desc="A cummulation of knoweldge gathered during all hops"
    )
    answer: str = dspy.OutputField(
        desc="The final answer to the original question"
    )
```

Table 8: Ablation of Learning Mechanisms on HoVer across backbones.

Backbone	Strategy	Accuracy	Evidence Recall	All-or-Nothing Recall
GPT-oss-20B + Qwen3-8B Embed	Default Query Encoding	91.0%	92.1%	81.0%
	Evidence-Conditioned Query Encoder + EAQL on BC+	**93.0%**	91.0%	80.0%
	Evidence-Conditioned Query Encoder + EAQL on HoVer	**93.0%**	92.7%	84.0%
	Oracle	91.0%	**96.6%**	**91.0%**
Qwen3-4B + Qwen3-0.6B Embed	Default Query Encoding	87.0%	80.9%	60.0%
	Evidence-Conditioned Query Encoder + EAQL on HoVer	87.0%	88.4%	**75.0%**
	Oracle	**88.0%**	**89.3%**	73.0%

Table 9: Efficiency comparison on BrowseComp-Plus using GPT-5-mini. We report median input/output tokens (thousands) and number of search and LLM calls per query. Dossier exhibits higher token usage because it retrieves and injects substantially more evidence into context, reflecting improved coverage rather than redundant search.

Method	Input (kTok)	Output (kTok)	Search Calls	LLM Calls
ReAct	314	4	10	11
Smolagents	675	74	168	42
Parallel ReAct (Dossier-Inspired)	1543	18	50	55
Dossier	2243	207	55	68

B HoVer Training Ablation

Table 8 reports an ablation of learning mechanisms on HoVer across backbones, comparing default query encoding, Evidence-Conditioned encoding with EAQL, and an oracle upper bound. The results show that EAQL consistently improves All-or-Nothing Recall (and matches or slightly improves Accuracy) over the default encoder, while the oracle establishes the ceiling for retrieval completeness.

C Efficiency Comparison

Table 9 reports the median input/output tokens and search/LLM call counts on BrowseComp-Plus using GPT-5-mini under a fixed 100 LLM call budget. Dossier uses more tokens than the linear baselines because it retrieves and processes more evidence, rather than terminating early or compressing the trajectory into an unstructured context history. This higher usage reflects the system's ability to sustain longer-horizon evidence gathering: Dossier performs 55 search calls and 68 LLM calls per query, compared to 10 and 11 for ReAct. Parallel ReAct also incurs a high input-token cost, but lacks a shared state for reconciling evidence across branches, so its additional computation does not provide the same coordinated synthesis benefits. Overall, the table shows that Dossier is more reliable for long-horizon reasoning.

D Generative AI

We used assistance from generative AI tools in preparing portions of this manuscript. All outputs were reviewed, verified, and edited by the authors.