PatchScout: Thematic Web Data Collection via Information Foraging

Michael West **Eduard Dragut**
Department of Computer and Information Science
Temple University
Philadelphia, PA
{michael.west0004, edragut}@temple.edu

Abstract

Researchers often need web corpora that are not answers to a single query, but reusable collections of documents spanning many decentralized sources. We refer to this problem as *thematic web data collection*: assembling semantically relevant documents that share a common theme but are scattered across heterogeneous and structurally disconnected regions of the web. Existing approaches, including traversal-based methods and LLM-driven web agents, are limited by local exploration or short-horizon stopping, preventing comprehensive collection. We formulate this task as a long-horizon information foraging problem over query-induced semantic patches, where the central challenge is deciding when to continue local exploitation and when to switch to new regions. We propose PatchScout, a multi-agent framework that instantiates a class of patch-switching policies to balance local exploitation and adaptive patch switching under a fixed budget. Experiments on four live-web topics show that PatchScout substantially improves yield and domain coverage over traversal-based and search-augmented reasoning agent baselines. In an applied social science setting with an incomplete prior collection, PatchScout discovers 102 previously unidentified entities, illustrating its ability to expand existing thematic datasets.

1 Introduction

Many scientific and societal investigations require constructing reusable thematic corpora from decentralized web sources, where relevant documents satisfy shared semantic criteria yet remain structurally dispersed across heterogeneous domains [Lazer et al., 2009]. For example, a public health researcher studying pandemic-related communication may need to collect documents from hundreds of municipal health departments, while an education researcher identifying Autism Support Programs (ASPs) must gather information scattered across hundreds of university websites [Kusters et al., 2023, West et al., 2025]. The same problem arises when assembling evaluation data for a model capability, collecting scientific sources that report a specific phenomenon, or identifying entities that satisfy a given criteria. In each case, the objective is not answering a single query, but assembling a collection of semantically relevant documents drawn from diverse, decentralized sources.

We refer to this task as *thematic web data collection*. We define a *theme* as a user-provided specification of the target collection, which functions as an operational relevance criterion that defines for each encountered document whether it belongs in the collection. For example, "identify colleges that host an Autism Support Program" is a theme: the target is not a broad subject area, but a concrete set of institutions satisfying a specified condition. The theme is not submitted directly as a search-engine query, instead it is used to guide the entire collection process. Unlike traditional retrieval, where success is measured by the relevance of a ranked list, thematic collection requires simultaneously optimizing *depth* (maximizing thematically relevant yield within discovered sources) and *breadth*

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

(maximizing coverage across distinct sources) under a limited interaction budget. This dual objective creates a fundamentally different search problem: relevant information is sparse, decentralized, and distributed across semantically related but structurally disconnected regions of the web.

One may attempt to address this problem using existing paradigms for web exploration. Traversal-based approaches, such as focused crawling [Chakrabarti et al., 1999, Diligenti et al., 2000, Aggarwal et al., 2001], prioritize link expansion from a set of seed pages based on topical relevance signals. However, these methods rely on the assumption that relevant content is reachable through graph connectivity from initial seeds, an assumption that breaks in decentralized settings where semantically related sources are structurally disconnected. Alternatively, query-driven approaches based on search-augmented reasoning generate adaptive queries to retrieve relevant information [Jin et al., 2025, Qian and Liu, 2025, Zheng et al., 2025]. These systems build on recent work that equips language models with search, browsing, and web-interaction tools to complete question-answering tasks [Nakano et al., 2021, Deng et al., 2023]. They often revisit classical retrieval challenges such as selecting promising entry points, iteratively updating relevance estimates, and allocating limited interaction [Dragut et al., 2022]. However, they are fundamentally designed for *task-bounded* objectives: their stopping criterion is answer sufficiency, not collection completeness. Once sufficient evidence is gathered to answer a user request or generate a coherent report, exploration terminates. As a result, they do not support the sustained exploration required for constructing comprehensive thematic collections. These observations highlight a fundamental gap: effective thematic web data collection requires jointly reasoning over local exploitation and global re-entry into new regions of the web under a shared budget constraint.

We argue that thematic web data collection is not a page-ranking or task-completion problem; it is a *budget-constrained, long-horizon information foraging problem*. Information Foraging Theory (IFT) models information seeking as movement among information patches guided by information scent and expected marginal gain [Pirolli and Card, 1999]. Under this perspective, effective thematic collection poses three interdependent challenges: (1) *patch discovery* – identifying promising new semantic regions; (2) *patch exploitation* – efficiently extracting high-yield documents within a discovered region; and (3) *patch switching* – determining when diminishing local returns justify abandoning the current patch in favor of exploring elsewhere. Therefore, the central problem is not relevance estimation or traversal optimization, but strategic patch-switching under budget constraints.

Motivated by this perspective, we propose PatchScout, a multi-agent framework that formulates thematic web data collection as a sequential decision problem over semantic patches and instantiates patch-switching policies for long-horizon exploration. PatchScout enters patches through adaptive search engine querying, exploits discovered patches through local traversal, and explicitly models when to abandon a patch and switch to a new one. This framing transforms thematic collection from traversal-centric crawling or short-horizon search into a long-horizon foraging problem over disjoint semantic patches. We implement this strategy through a multi-agent framework composed of specialized agents for patch discovery, relevance assessment, patch exploitation, and patch switching.

Our contributions are:

- We introduce thematic web collection as a distinct problem: given a user-defined theme, the objective is to discover and assemble a distributed set of relevant web resources over a long horizon, rather than answer a single query or remain within one topical neighborhood.
- We formulate it as a budget-constrained information-foraging problem, treating the decision to continue exploiting the current semantic patch or re-seed into a new region of the web as the central control problem.
- We propose PatchScout, whose Scout–Forager–Critic architecture couples adaptive query generation, within-patch exploitation, and between patch exploration through a family of patch-switching policies.
- On four live-web topics, PatchScout improves document yield, domain coverage and collection diversity over classical crawlers and deep-research agents.

2 Related Work

Building collections of semantically related web documents has been studied in several related contexts, particularly in traversal-based focused crawling and domain-specific web data acquisition [Olston and Najork, 2010, Pham et al., 2018, Chen et al., 2024, 2025b, West and Dragut, 2026]. Early

work explores hybrid strategies that combine graph exploration with query-based discovery. Qin et al. [2004] consider the construction of scientific digital libraries and identify structural limitations of traversal-only approaches, incorporating periodic query-based discovery via search engines. However, their approach relies on a static set of user-provided queries, limiting adaptability and preventing sustained exploration across dynamically discovered regions of the web. More recently, Pham et al. [2019] introduce DISCO, which combines query-based discovery with local graph exploration and models action selection as a multi-armed bandit based on observed marginal yield [Auer et al., 2002]. Their results demonstrate that interleaving these actions substantially improves over traversal-only baselines.

While these approaches recognize the need to combine query-based discovery and local exploration, they treat these operations as competing alternatives with isolated rewards. Thus, the decision problem is framed as a greedy selection between actions based on historical yield. This formulation fails to capture their complementary structure: query-based discovery exposes new regions of the information space, while local graph exploration efficiently exploits those regions for high-yield extraction. The relevant policy is therefore not selecting between actions, but optimizing their coordinated composition over time. More fundamentally, prior work does not explicitly formulate the problem as a long-horizon, budget-constrained collection task over decentralized semantic targets.

Information Foraging Theory (IFT) provides a useful abstraction by modeling information-seeking as navigation over an information landscape composed of *information patches*, regions containing relevant information, and *information scent*, signals that estimate the value of nearby content [Chi et al., 2001, Pirolli and Fu, 2003]. In this setting, a query induces a semantic patch by exposing a set of entry points, while local graph exploration exploits the neighborhood reachable from those points. Under this view, query actions perform patch discovery, traversal actions perform patch exploitation, and the central policy decision is when to switch between patches.

Modern search-augmented reasoning systems address related information-seeking problems. Trivedi et al. [2023] interleave retrieval with chain-of-thought steps to answer multi-hop questions. Jiang et al. [2023] make this process adaptive by introducing forward-looking retrieval triggers based on generation uncertainty that issue queries when model confidence falls below a threshold. Asai et al. [2023] uses supervised fine-tuning to train a model to emit retrieval tokens, allowing the model to learn *when* to search, while Jin et al. [2025] achieves this through reinforcement learning. More recent works have extended this reasoning-search paradigm by incorporating live web search, rather than retrieval from static corpora [Li et al., 2025, Zheng et al., 2025, Chen et al., 2025a]. Qian and Liu [2025] interpret such systems through the lens of IFT, modeling intermediate reasoning steps and subqueries as evolving expressions of information scent that guide retrieval toward increasingly useful patches. However, these approaches optimize patch switching implicitly as part of a reasoning trajectory toward a single answer. In contrast, our setting requires treating patch switching as an explicit policy decision and optimizing over long-horizon collection objectives across disjoint regions of the web graph.

Taken together, prior work addresses components of this problem, either through traversal-based exploration, query-based discovery, or short-horizon reasoning, but does not provide a unified formulation of thematic web data collection as a long-horizon, budget-constrained decision process. We address this gap by formalizing thematic web data collection as an information foraging problem over query-induced semantic patches.

3 Problem Definition

A thematic collection task involves multi-step semantic search. Consider identifying Autism Support Programs (ASPs) across universities [West et al., 2026, Nachman et al., 2022]. An agent must first discover which university domains are likely to contain relevant programs. However, domain discovery alone is insufficient. Many universities do not offer dedicated ASPs, while others embed autism-specific services within broader student support ecosystems that also encompass motor, intellectual, visual, hearing, or mental health accommodations. As a result, target relevance is both decentralized and highly heterogeneous: autism-related support may appear as a standalone program, a subprogram within a larger support structure, or may be entirely absent despite superficially relevant institutional signals. Consequently, the agent must determine whether relevant support exists, distinguish autism-specific offerings from broader but non-target programs, and efficiently

locate the small number of semantically relevant pages containing critical details such as support type, eligibility, pricing, and accommodations within larger, noisy site structures. Thus, thematic web data collection requires solving both inter-domain discovery (finding promising domains) and intra-domain exploitation (extracting sparse target pages within semantically complex domains), making it a hierarchical search problem characterized by domain uncertainty, semantic nesting, and sparse target information.

We model thematic web data collection as an iterative search process in which a system alternates between patch exploitation and patch switching under a limited interaction budget. Given a thematic objective, the system issues queries to identify promising entry points, locally explores the web regions induced by those entry points, evaluates discovered pages for thematic relevance, and accumulates relevant documents into a growing collection.

In this framework, a semantic patch is induced by a search query and consists of the retrieved seed pages together with their locally reachable neighborhood. Exploiting a patch corresponds to local traversal and relevance-guided extraction within that region, while patch switching corresponds to issuing a new query to enter a different semantic region of the web. Because query-based reseeding provides relatively low-cost access to new regions, the central policy problem is determining when the expected marginal value of continued local exploitation falls below the expected value of switching.

We formalize this decision through four components: (i) semantic patches, (ii) realized information gain, (iii) information scent as an estimate of future within-patch utility, and (iv) the reseed policy governing adaptive patch switching.

3.1 Semantic Patches

We model the collection process as alternating between *patch discovery* and *patch exploitation*. While patches can be defined in multiple ways, e.g., via semantic clustering or embedding-based neighborhoods, we adopt a query-induced, traversal-based definition that aligns with how agents access the web through search queries and explore it via link traversal. Patch discovery begins when the system issues a search query Q, and the search engine returns the top-k results $S(Q) = \mathrm{SEARCH}_k(Q) = \{s_1, \ldots, s_k\}$. Let $N_d(s)$ denote the set of pages reachable from search result $s \in S(Q)$ by link traversal within depth d. The patch is then:

$$\mathcal{P}_{k,d}(Q) = \bigcup_{s \in S(Q)} N_d(s).$$

3.2 Patch Explosion Under Transitive Closure

A central challenge in patch-based exploration is that local neighborhoods expand rapidly under transitive closure. Consider a patch induced by a query Q and its top-k results. Let b denote the average branching factor (i.e., the number of outgoing links per page). The size of the reachable neighborhood within depth d grows approximately as $|\mathcal{P}_{k,d}(Q)| \leq k \cdot \frac{b^{d+1}-1}{b-1} = O(kb^d)$.

Even for modest values (e.g., $k = 10$, $b = 8$, $d = 3$), this yields approximately $5{,}850$ pages. Although overlap between neighborhoods may reduce this number in practice, shallow transitive expansion can still produce thousands of candidate pages within a single patch. We refer to this phenomenon as **patch explosion**: a combinatorial growth effect in which local exploration quickly becomes computationally and cognitively impractical. Hence, exhaustive within-patch exploration is not only inefficient, but suboptimal under a limited interaction budget. This constraint fundamentally shapes the exploration strategy. To avoid over-investing in a single dense region, an effective agent must balance local exploitation with timely patch switching. The core challenge is thus not simply discovering relevant patches, but managing the tradeoff between exploiting the current patch and re-entering the search space to discover new ones.

3.3 Patch Utility and Re-Seeding Policy

We quantify the utility of exploration through both realized and expected information gain, and use these signals to define a policy for patch switching. A *Critic* assigns each visited page x a binary relevance label $r(x) \in \{0, 1\}$, where $r(x) = 1$ indicates a relevant page. Let $\mathcal{R}_t$ denote the set of relevant pages discovered up to time t, such that $\mathcal{R}_t = \{x : r(x) = 1\}$. The cumulative information

gain is: $G_t = |\mathcal{R}_t|$. Let $G_m(\tau)$ denote the number of relevant pages discovered during the first τ steps in patch m. We define the *patch productivity* as the average gain per step: $\rho_m(\tau) = \frac{G_m(\tau)}{\tau}$.

While $\rho_m(\tau)$ captures the realized return from the current patch, effective decision-making also requires estimating the remaining value of continued exploration. To this end, we model *information scent* as a predictive signal over unexplored links. At time t, let F_t denote the frontier of candidate links within the current patch. For each link $u \in F_t$, the Forager assigns an information-scent score representing the probability that following u yields a relevant page:

$$\sigma_t(u) = \Pr\big(r(x_u) = 1 \mid u, c(u), \mathcal{H}_{<t}\big),$$

where x_u is the page reached by following link u, $c(u)$ is the textual context in which the link appears, and $\mathcal{H}_{<t}$ is the collection history prior to time t.

Let $u_{t,1}, \ldots, u_{t,h}$ denote the top-h frontier links ranked by $\sigma_t(\cdot)$. We define the *patch-level information scent* as the expected future gain rate under h additional exploration steps:

$$\gamma_t(h) = \frac{1}{h} \sum_{j=1}^{h} \sigma_t(u_{t,j}). \tag{1}$$

Together, $\rho_m(\tau)$ and $\gamma_t(h)$ provide complementary signals: $\rho_m(\tau)$ reflects the realized value of past exploration, while $\gamma_t(h)$ estimates the expected return from continued exploitation. At each step t, the agent observes $s_t = (\rho_m(\tau), \gamma_t(h))$ and selects $a_t \in \{\text{stay}, \text{switch}\}$ according to a threshold-based policy that switches when trigger$(s_t) <$ baseline$(\mathcal{H}_{<t})$, where the trigger is given by either $\rho_m(\tau)$ or $\gamma_t(h)$. We consider three baselines. A fixed threshold η provides a static reference. A global baseline is given by the mean productivity of previously completed patches:

$$\bar{\rho}_{<m} = \frac{1}{m-1} \sum_{i=1}^{m-1} \rho_i(T), \tag{2}$$

where $\rho_i(T)$ is the final productivity of patch i. To allow the baseline to adapt to changes in patch quality over the course of collection, we also define an exponential moving average (EMA):

$$\theta_{m+1} = (1-\beta)\theta_m + \beta \rho_m(T), \tag{3}$$

where $\beta \in (0, 1]$ controls the update rate and θ_1 is an initial estimate. The re-seeding policy is then defined as: *switch patch if trigger < baseline*. This policy can be viewed as a threshold-based stopping rule [Shiryaev, 2008], where the agent switches when the estimated marginal utility of the current patch falls below an adaptive reference level derived from prior experience. This general form induces a family of threshold-based policies obtained by combining different trigger signals and baseline estimators:

Table 1: Re-seeding policies derived from different trigger-baseline combinations. Scent conditions use forward-looking estimates, while realized conditions use observed within-patch performance.

Name	**Condition**	**Description**
Threshold	$\rho_m(\tau) < \eta$	Realized productivity below a fixed topic-dependent threshold.
Realized-Mean	$\rho_m(\tau) < \bar{\rho}_{<m}$	Realized productivity below the mean productivity of prior patches.
Realized-EMA	$\rho_m(\tau) < \theta_m$	Realized productivity below the EMA of prior patches.
Scent-Mean	$\gamma_t(h) < \bar{\rho}_{<m}$	Information scent below the mean productivity of prior patches.
Scent-EMA	$\gamma_t(h) < \theta_m$	Information scent below the EMA of prior patches.

These variants allow us to evaluate the relative impact of realized versus predictive signals, as well as static versus adaptive baselines, on long-horizon collection performance. This formulation makes patch switching an explicit decision driven by diminishing returns. The trigger signals capture both realized and expected utility, while the baselines provide adaptive reference points grounded in past exploration. Together, they implement the tradeoff between continued exploitation of the current patch and re-entry into new regions of the web under a limited budget.

We view patch-based exploration as a sequential decision problem under uncertainty [Cassandra et al., 1994, Puterman, 2014]. At each step, the agent must decide whether to continue exploiting

Algorithm 1 Thematic Web Data Collection Loop

Require: collection theme t, collection budget B

initialize visited set $V \leftarrow \emptyset$, relevant set $R \leftarrow \emptyset$, patch history $H \leftarrow \emptyset$
while B not exhausted **do**
    $q \leftarrow \textsc{Scout}(t, H)$
    $S \leftarrow \textsc{Search}_k(q)$
    initialize patch $\mathcal{P}$ from S
    while $\mathcal{P}$ not exhausted and budget remains **do**
        $u \leftarrow \textsc{Forager}(\mathcal{P}, H)$
        visit page u and add u to V
        $r \leftarrow \textsc{Critic}(u, t, H)$
        if $r = 1$ **then**
            add u to R
        update patch state, frontier, and patch productivity
        if $\textsc{ReSeed}(\mathcal{P}, H)$ **then**
            break
    record completed patch in H

the current patch or switch to a new one based on partial observations of realized gain and estimates of future utility. This can be interpreted as an instance of optimal stopping or a two-action decision problem, where the agent terminates local exploration when its expected marginal return falls below that of alternative patches.

4 Methodology

We present PatchScout, a multi-agent framework for *thematic web data collection*. Guided by the information foraging formulation in Section 3, the system organizes collection as an iterative process over *semantic patches*. At a high level, PatchScout alternates between *patch discovery*, where new queries identify promising regions of the web, and *patch exploitation*, where the system traverses links within a patch to collect relevant pages. A dynamic re-seeding policy determines when to abandon low-yield patches and initiate new ones. The objective is to maximize cumulative information gain under a fixed interaction budget.

4.1 System Overview

PatchScout has three cooperating agents: a *Scout*, a *Forager*, and a *Critic*. These agents implement complementary roles in the information-foraging loop and interact through structured state.

- **Scout:** Performs *patch discovery* by generating queries that open new semantic patches and adapts query generation based on past patch outcomes.
- **Forager:** Performs *within-patch exploitation* by traversing links in the current patch, ranking candidate outlinks using information scent, and triggering re-seeding when the current patch becomes unproductive.
- **Critic:** Converts visited pages into realized information gain via relevance classification and provides feedback that informs both query generation and link ranking.

Algorithm 1 summarizes the collection process. Starting from a natural-language theme description, the Scout generates a search engine query which induces a semantic patch. The Forager explores this patch via local link traversal, while the Critic evaluates visited pages for relevance. During exploration, the system tracks expected future gain (Eq. 1) and past productivity (Eq. 2 or Eq. 3). When the re-seeding policy is triggered, control returns to the Scout and the process repeats, with the objective of maximizing cumulative information gain under a fixed interaction budget.

4.2 Agent Implementation

Scout: The Scout takes as input the user-defined theme and a history of previously explored patches, including queries and their realized productivity. It generates a new query that is both distinct from prior queries and likely to lead to relevant content. The output is issued directly to the search engine

to initiate a new patch.
Forager: The Forager manages within-patch exploration through frontier ranking and re-seeding decisions. It maintains a frontier of candidate links annotated with contextual signals such as anchor text, surrounding text, and page metadata. Using recent exploration history and Critic feedback, it assigns a score to each link, determining its priority in the frontier. At each step, the Forager either continues exploration by selecting a new link or triggers re-seeding according to the specified policy.
Critic: The Critic performs binary relevance classification on visited pages. Given the page content (e.g., title and body) and the collection theme, it outputs a relevance label and optionally a short rationale. These outputs are incorporated into the system history and guide both query generation and link ranking.

4.3 Crawling safeguards

PatchScout is a read-only research prototype that retrieves publicly accessible web pages. It performs no write actions: it does not authenticate, submit forms, post content, execute transactions, bypass access controls or paywalls, or interact with resources requiring credentials. Its only external actions are HTTP GET requests and search queries issued through a commercial search API. To support polite crawling and limit load on external services, PatchScout reads and respects `robots.txt` whenever it is available, enforces a global concurrency cap of eight requests, and does not revisit a URL within a run. Requests that return access-control or rate-limit responses, such as `HTTP 403` or `429`, are not retried. These controls reduce repeated access, respect site-level crawling restrictions, and avoid continuously requesting blocked or unavailable resources. PatchScout also treats retrieved content as untrusted data rather than executable instructions. Page content cannot alter the system's permissions, trigger write actions, or introduce new tools or credentials. Thus, malicious or prompt-injection content may at most affect a relevance judgment, but it cannot cause external actions beyond the bounded, read-only retrieval process.

5 Experimental Setup

We evaluate PatchScout on *thematic web data collection* tasks on the live web. Our goal is to assess whether dynamic re-seeding improves long-horizon collection performance compared to traversal-centric crawlers and task-bounded agents. We compare a focused crawler, a modern search-augmented reasoning agent, and variants of our framework that differ in their re-seeding policies.

5.1 Tasks and Evaluation Topics

We evaluate across four topics selected to vary in *density* (number of relevant domains) and *generality* (breadth of the relevance definition). A key challenge in thematic collection is the absence of ground truth for coverage, since the size and location of the target set are typically unknown. For evaluation, we thus select topics with authoritative entity lists, allowing us to estimate coverage. Coverage is measured at the *entity level* rather than by exact domain matching. A target entity may expose relevant content across multiple domains or subdomains (e.g., `pepsico.com` vs. `pepsicojobs.com`), so discovered pages are mapped to entities prior to computing coverage [Dong et al., 2019]. Details of this verification process are provided in Appendix B.

- **Four Year Universities (US):** Dense ($N = 1{,}928$) and broad; any page within a university domain is relevant [National Center for Education Statistics, 2025].
- **Fortune 500 Employment:** Moderately dense ($N = 500$) but narrow; relevance is restricted to employment-related pages [Fortune, 2025].
- **Accredited Law Schools (US):** Sparse ($N = 197$) with broad relevance (any page within the domain) [American Bar Association, 2025].
- **Autism Support Programs (ASPs):** Extremely sparse ($N = 133$) and niche; authoritative list is estimated using datasets from Nachman et al. [2022] and Viezel et al. [2020].

5.2 Baselines and Variants

Traversal-centric baseline. We use ACHE [Barbosa and Freire, 2007] to represent traversal-based crawling. ACHE relies on a fixed seed set and expands solely through link traversal without query-based re-entry. We generate seeds using the Scout (query generation) and Critic (filtering). ACHE is

Table 2: Harvest rate (HR%) and gold-standard domain coverage (Cov%) at a 10,000-page budget. Deep-research systems are evaluated at completion by coverage over final and intermediate URLs; HR is omitted for them because they have no comparable page-budget denominator. Best completed result per topic per metric in bold.

	Universities		**Fortune 500**		**Law Schools**		**ASPs**	
System	HR	Cov	HR	Cov	HR	Cov	HR	Cov
ACHE	18.3	3.6	4.0	2.6	6.0	6.6	0.0	0.0
DeepResearcher	–	0.3	–	0.8	–	1.0	–	17.3
Claude Deep Research	–	1.1	–	15.0	–	91.9	–	17.3
Gemini Deep Research	–	1.8	–	3.0	–	**93.4**	–	9.8
No-Reseed	18.7	1.7	0.8	3.0	0.1	1.0	0.1	1.5
Fixed-Reseed	62.3	16.1	15.3	18.4	26.8	72.6	4.1	30.8
Threshold	52.6	14.8	22.5	23.8	**38.1**	85.3	**6.1**	**39.1**
Realized-Mean	73.1	2.4	**30.1**	17.0	29.5	72.6	4.9	31.6
Realized-EMA	**86.2**	4.4	23.2	19.8	25.3	70.6	3.7	26.3
Scent-Mean	77.7	**20.3**	17.5	20.6	33.2	79.2	4.8	33.1
Scent-EMA	68.3	15.5	20.1	**24.6**	37.2	86.8	5.6	35.3

run using the public implementation with default settings.

Task-bounded agent. We include DeepResearcher [Zheng et al., 2025], an open-source search-augmented reasoning agent trained for open-domain question answering, which we run locally and can fully inspect. We also evaluate two commercial deep-research systems, Claude Deep Research which is accessible only through the web interface and Gemini Deep Research accessed through API. These closed systems do not expose intermediate reasoning, so we compare them based on their final reports. All three systems receive the same theme specification and prompts which attempt to induce long exploration (Appendix D), have no page budget and terminate when they produce a report.

PatchScout variants. To isolate the impact of re-seeding, we evaluate multiple variants of our framework, which we interpret as instances of a broader policy class over patch-switching decisions: *No-Reseed*, which disables patch switching and performs only initial query-based entry followed by traversal; *Fixed-Reseed*, which triggers patch switching after a fixed number of pages per patch; and the five re-seeding policies of Table 1 (Threshold, Realized-Mean, Realized-EMA, Scent-Mean, Scent-EMA), each evaluated as a separate system in Table 2 under the same name. Additionally, we conduct component-level ablations, reported in Appendix E, to isolate the effects of policy-guided re-seeding, feedback-driven Scout query generation, and Forager link ranking.

5.3 Implementation Details

All agents (Scout, Forager, Critic) are implemented using DeepSeek-V3.1 via API. Queries are issued to Google using SearchAPI. Each method is provided the same theme description and prompt configuration (see Appendix G). Our implementation is available at `https://github.com/JustusWest/PatchScout`. All systems operate under a fixed budget of 10,000 page visits per run. The budget controls page fetches rather than model compute; token usage for all systems is reported in Section 6 and Appendix C. For consistency, all collected pages from baseline methods are evaluated post-hoc using our Critic Agent. All experiments follow the crawling safeguards specified in Section 4.3.

Our framework introduces a small number of policy-specific hyperparameters. For *Fixed-Reseed*, we use a patch budget of 50 pages. *Threshold* uses a topic-specific value η, selected based on stable performance observed in preliminary runs. The *Scent* and *EMA* policies introduce h (frontier size) and β (smoothing rate). We perform a small sensitivity analysis on representative dense (Universities) and sparse (ASP) topics (see Appendix A). We find that $h = 10$ and $\beta = 0.3$ yield stable performance across topics, and use these values in all experiments.

6 Results

We evaluate all systems on two main metrics: *harvest rate* (the fraction of visited pages classified as relevant) and *domain coverage* (the fraction of ground-truth domains discovered). Table 2 reports endpoint performance after a fixed 10,000-page budget. Figure 1 shows coverage growth over time.

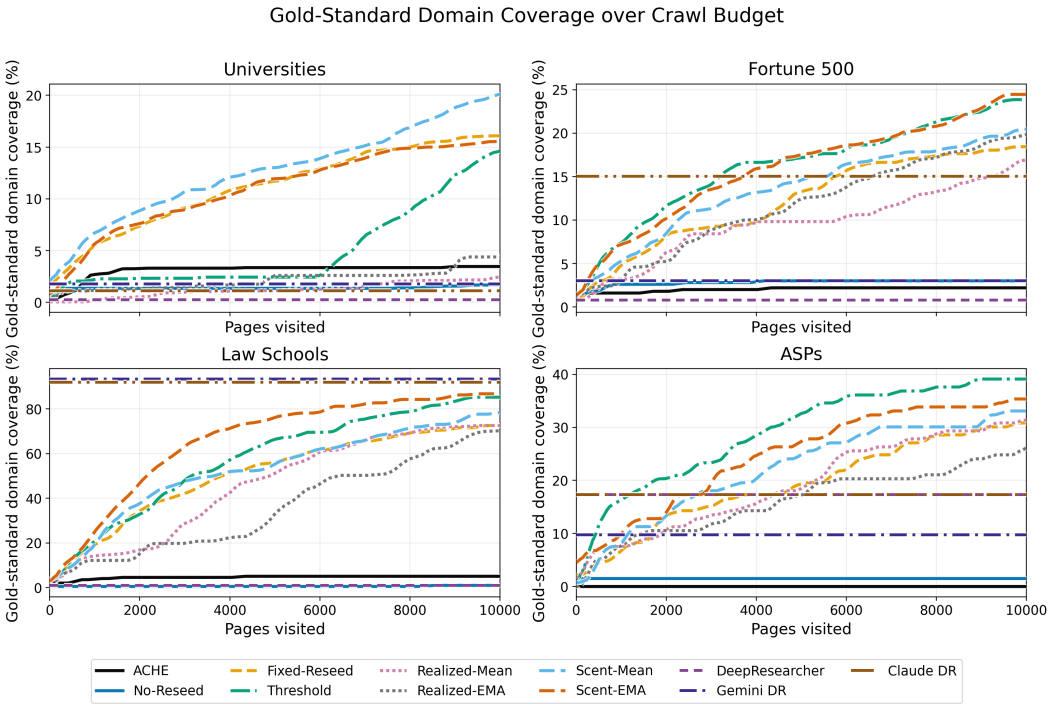


Figure 1: Gold-standard domain coverage (%) over pages visited. Deep-research baselines reported as completion endpoints. Traversal-only methods plateau early, while re-seeding sustains discovery, particularly on sparse topics.

Baseline Performance Traversal-only methods (*ACHE* and *No-Reseed*) plateau rapidly across all topics, especially for sparse settings where weak connectivity limits discovery beyond initial seeds. As shown in Figure 1, both methods saturate within the first few hundred pages, while re-seeding variants sustain coverage growth throughout collection. The task-bounded agent baselines do not dominate, but they outperform our best variant on one topic. Law Schools has a small target set ($N = 197$) enumerated by a single authoritative accreditation directory: an agent that locates that page can recover nearly the full set without sustained exploration (Gemini 93.4%, Claude 91.9%, best PatchScout 86.8%). When PatchScout reaches the same directory, its roughly 200 outlinks compete in a shared frontier, rather than recognizing the list-structure and extracting the record directly [Chen et al., 2022]. Where no such source exists, the ordering reverses: PatchScout reaches $11.3\times$ the coverage of the best deep-research system on Universities (20.3% vs. 1.8%), whose target set is too large for any single directory, $2.3\times$ on ASPs (39.1% vs. 17.3%), for which no complete list exists, and $1.6\times$ on Fortune 500 (24.6% vs. 15.0%). This exposes a boundary condition: a theme that is already enumerated reduces to a directory lookup.

Breadth and Depth Harvest rate captures local yield, while coverage measures breadth across the target population. These objectives often diverge, reflecting a mismatch between page-level relevance and coverage across distinct domains. For example, *Realized-Mean* achieves the highest harvest rate on Fortune 500 (30.1%), but lags in coverage relative to *Scent-Mean* and *Scent-EMA*. This occurs because productivity-based triggers over-exploit dense patches. On Universities, *Realized-Mean* spends 2,457 pages in a single patch (`uchicago.edu`), yielding 2,399 relevant pages but only 28 unique domains, high local yield but poor coverage.

This tradeoff is dependent on topic structure. Dense, broad topics (Universities) favor depth, while sparse or niche topics (ASPs) require aggressive exploration. Fortune 500 Employment lies between these extremes, requiring both domain discovery and within-domain navigation. These results confirm that optimizing for relevance alone can produce narrow, unrepresentative collections.

Scent-based policies generally improve coverage by estimating remaining patch value. *Scent-Mean* achieves the highest coverage on Universities, while *Scent-EMA* performs best on Fortune 500 and

Law Schools. However, no policy dominates across all topics. On ASPs, patches are often entirely unproductive, causing adaptive baselines to quickly degrade to near zero values which allows local traversal in unproductive patches. In this case fixed thresholding performs best, suggesting that aggressive switching may outperform fine-grained estimation in highly sparse settings.

For Universities, Fortune 500, and Law Schools, gold-standard lists are treated as closed sets. For ASPs, the list is incomplete and somewhat outdated; we report coverage against the known set and manually verify additional discoveries. PatchScout identifies 102 new ASP domains absent from the prior list, demonstrating its ability to expand existing datasets.

Overall, the best re-seeding policy depends on the target theme and the desired balance between local yield and global breadth. This reinforces the central claim: thematic web-data collection is not solved by relevance estimation alone, but by explicitly modeling patch discovery, patch exploitation, and patch switching under a limited budget.

Cost Analysis The page budget does not control model compute, so we also compare systems by token consumption (Appendix C). Truncating each PatchScout run at the token budget DeepResearcher consumed on the same topic, the best PatchScout variant matches or exceeds DeepResearcher on every topic (Table 8). PatchScout is thus more efficient at matched compute and continues converting budget into coverage after task-bounded agents stop, however, these returns decline as the accessible target set is exhausted.

Ablations and Robustness Ablation studies show that each component of the PatchScout framework contributes to the systems success (Table 9). Random patch switching reduces Law Schools coverage from 86.8% to 25.4% and ASP coverage from 35.3% to 9.0%; generating all queries upfront reduces Fortune 500 coverage from 24.6% to 6.6%; and random link selection reduces Universities harvest rate from 86.2% to 53.5%. Results do not depend on a large Critic: binary agreement with our production Critic is 86.1%, 87.4%, and 86.9% for 9B, 27B, and 397B reference models (Table 10).

7 Conclusion

We introduced PatchScout and formulated thematic web data collection as a long-horizon information foraging problem over query-induced semantic patches. This framing highlights the need to balance local exploitation with adaptive patch switching under a limited budget. Our experiments show that patch-switching policies fundamentally change collection dynamics: traversal-only methods plateau early, while re-seeding continues discovering new domains. We also find that yield and coverage are distinct objectives, and that scent-based policies improve coverage, though no single strategy dominates across topics. In the ASP setting, PatchScout discovered 102 previously unidentified entities, demonstrating its ability to expand incomplete scientific datasets rather than merely recover known targets. Overall, thematic web data collection is not just a retrieval or reasoning problem, but a decision problem over exploration. The patch-based foraging perspective provides a foundation for building systems that construct diverse, reusable web corpora.

8 Limitations

Our study reflects several practical constraints of web-scale collection. Experiments are conducted on the live web, where evolving content and rankings limit exact reproducibility. PatchScout relies on an LLM-based Critic, introducing variability in relevance assessment, and incurs higher per-page cost than traditional classifiers (see Appendix C for cost analysis). Finally, re-seeding effectiveness depends on the collection objective and topic structure. Policies based on patch productivity capture local yield but may diverge from coverage, which is difficult to estimate without knowing the target set size. Designing strategies that better balance yield and coverage remains an important direction for future work.

Acknowledgments and Disclosure of Funding

This work was supported by the U.S. National Science Foundation awards III-2107213, ITE-2333789, and 2026513. The authors declare no competing interests.

References

Charu C Aggarwal, Fatima Al-Garawi, and Philip S Yu. Intelligent crawling on the world wide web with arbitrary predicates. In *Proceedings of the 10th international conference on World Wide Web*, pages 96–105, 2001.

American Bar Association. ABA-Council-Approved Law Schools, 2025. URL `https://www.americanbar.org/groups/legal_education/accreditation/approved-law-schools/alphabetical/`. Accessed: 2025-07-21.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2023.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Luciano Barbosa and Juliana Freire. An adaptive crawler for locating hidden-web entry points. In *Proceedings of the 16th international conference on World Wide Web*, pages 441–450, 2007.

Anthony R Cassandra, Leslie Pack Kaelbling, and Michael L Littman. Acting optimally in partially observable stochastic domains. In *Aaai*, volume 94, pages 1023–1028, 1994.

Soumen Chakrabarti, Martin Van den Berg, and Byron Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Computer networks*, 31(11-16):1623–1640, 1999.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Zhou Yijie, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. Research: Learning to reason with search for LLMs via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL `https://openreview.net/forum?id=OuGAwwAT8G`.

Zhijia Chen, Weiyi Meng, and Eduard Dragut. Web record extraction with invariants. *Proceedings of the VLDB Endowment*, 16(4), 2022.

Zhijia Chen, Lihong He, Arjun Mukherjee, and Eduard Dragut. Comquest: Large scale user comment crawling and integration. In *Companion of the 2024 International Conference on Management of Data*, pages 432–435, 2024.

Zhijia Chen, Weiyi Meng, and Eduard Dragut. Comcrawler: General crawling solution for article comments. In *Proceedings of the International Conference on Extending Database Technology*, 2025b.

Ed H Chi, Peter Pirolli, Kim Chen, and James Pitkow. Using information scent to model user information needs and actions and the web. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 490–497, 2001.

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

Laura Dietz, Oleg Zendel, Peter Bailey, Charles LA Clarke, Ellese Cotterill, Jeff Dalton, Faegheh Hasibi, Mark Sanderson, and Nick Craswell. Principles and guidelines for the use of llm judges. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, pages 218–229, 2025.

Michelangelo Diligenti, Frans Coetzee, Steve Lawrence, C Lee Giles, Marco Gori, et al. Focused crawling using context graphs. In *VLDB*, pages 527–534, 2000.

Yongquan Dong, Eduard C. Dragut, and Weiyi Meng. Normalization of duplicate records from multiple sources. *IEEE Transactions on Knowledge and Data Engineering*, 31(4):769–782, 2019.

Eduard C Dragut, Weiyi Meng, and Clement Yu. *Deep web query interface understanding and integration*. Springer Nature, 2022.

Fortune. Fortune 500, 2025. URL https://fortune.com/ranking/fortune500/. Accessed: 2025-07-01.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 7969–7992, 2023.

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

Isabelle S Kusters, Amanda M Gutierrez, Julianna M Dean, Mark Sommer, and Anna Klyueva. Spanish-language communication of covid-19 information across us local health department websites. *Journal of Racial and Ethnic Health Disparities*, 10(5):2482–2489, 2023.

David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-László Barabási, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. Computational social science. *Science*, 323(5915):721–723, 2009.

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

Brett Ranon Nachman, Catherine Tobin McDermott, and Bradley E Cox. Brief report: Autism-specific college support programs: Differences across geography and institutional type. *Journal of Autism and Developmental Disorders*, 52(2):863–870, 2022.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

National Center for Education Statistics. Integrated Postsecondary Education Data System (IPEDS), 2025. URL https://nces.ed.gov/ipeds/. Accessed: 2025-07-12.

Christopher Olston and Marc Najork. Web crawling. *Foundations and Trends in Information Retrieval*, 4(3):175–246, 2010.

Kien Pham, Aécio Santos, and Juliana Freire. Learning to discover domain-specific web content. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 432–440, 2018.

Kien Pham, Aécio Santos, and Juliana Freire. Bootstrapping domain-specific content discovery on the web. In *The World Wide Web Conference*, pages 1476–1486, 2019.

Peter Pirolli and Stuart Card. Information foraging. *Psychological review*, 106(4):643, 1999.

Peter Pirolli and Wai-Tat Fu. Snif-act: A model of information foraging on the world wide web. In *International Conference on User Modeling*, pages 45–54. Springer, 2003.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Hongjin Qian and Zheng Liu. Scent of knowledge: Optimizing search-enhanced reasoning with information foraging. *arXiv preprint arXiv:2505.09316*, 2025.

Jialun Qin, Yilu Zhou, and Michael Chau. Building domain-specific web collections for scientific digital libraries: a meta-search enhanced focused crawling method. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 135–141, 2004.

Albert N Shiryaev. *Optimal stopping rules*. Springer, 2008.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 10014–10037, 2023.

Karane Vieira, Luciano Barbosa, Altigran Soares Da Silva, Juliana Freire, and Edleno Moura. Finding seeds to bootstrap focused crawlers. *World Wide Web*, 19:449–474, 2016.

Kathleen D Viezel, Elizabeth Williams, and Wesley H Dotson. College-based support programs for students with autism. *Focus on Autism and Other Developmental Disabilities*, 35(4):234–245, 2020.

Michael West and Eduard Dragut. A demo of interactive thematic data collection on the live web. *Proceedings of the VLDB Endowment*, 19(12):4554–4557, 2026.

Michael West, Matthew Tincani, Donald Hantula, Sungsoo Ray Hong, Slobodan Vucetic, and Eduard Dragut. Applying data science practices to identify characteristics of postsecondary autism support programs from their websites. *Journal of Autism and Developmental Disorders*, pages 1–47, 2025.

Michael West, Katie Kostin, Matt Tincani, Donald Hantula, and Eduard Dragut. Evaluating autism support programs in higher education through the perspectives of students, parents, counselors and staff. In *Frontiers in Education*, volume 11, page 1921007. Frontiers Media SA, 2026.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 414–431, 2025.

A Hyperparameters

Table 3 lists all hyperparameters that may be set for different variants of PatchScout. Table 4 shows the results from the hyperparameter sensitivity analysis for h and β.

Table 3: List of hyperparameters for **PatchScout**.

Symbol	Role	Default Value
k	Number of search results retained per query.	10
b	Maximum branching factor (number of links to enque in frontier from a single page).	10
η	Fixed productivity threshold for re-seeding.	Topic dependent.
ρ_0	Initial patch productivity estimate.	Topic dependent.
h	Number of frontier links used to estimate information scent.	10
β	Update rate for patch productivity.	0.3

Table 4: Topic-separated totals for relevant pages and relevant domains across scent-dynamic hyperparameter settings. Balanced columns are computed across both topics: for each topic, values are normalized by that topic's best value across settings (relevant/max relevant, domains/max domains), then averaged equally across topics.

		University		**ASP**		**Balanced**	
h	β	Relevant	Domain	Relevant	Domain	Relevant	Domain
5	0.1	743	190	72	36	0.884	0.764
5	0.3	813	172	73	**50**	0.930	0.866
5	0.5	868	159	49	30	0.807	0.638
10	0.1	777	204	**78**	43	0.941	0.864
10	0.3	**880**	**235**	74	42	**0.974**	**0.920**
10	0.5	729	187	60	38	0.799	0.778
25	0.1	838	183	30	19	0.668	0.579
25	0.3	618	153	70	42	0.800	0.746
25	0.5	849	218	32	19	0.688	0.654

B Entity-Domain Verification

Our gold standard lists for each topic contain a single unique root-domain, each representing an entity that belongs to the target theme. For example, Fortune 500 contains `amazon.com`. For initial coverage estimation, we perform exact root-domain matching between gold standard lists and the set of relevant pages from each collection ($N = 77,532$). In this case, both `amazon.com/jobs` and `jobs.amazon.com` would match the gold standard list. However, we notice that many entities contained in the list maintain alternate domains, such as `amazon.jobs` or `amazonjobs.com`, so we check relevant pages which are not matched to a gold standard domain ($N = 20,427$). We strip the root domains to only the stem (i.e. `amazon`) and compare to every domain in the gold standard list for exact matches and prefix/suffix dependent matches (i.e. `amazonjobs.com` would become `amazonjobs` which matches `amazon`). The domain is marked as a match if it matches exactly one gold standard domain. We utilize an LLM to process the remaining unmatched pages ($N = 5,931$). First, we search the text of the page content and title for the gold standard stems, and add them to the candidate match list if they are found. If this finds fewer than 3 candidates, we fill the remaining slots based on lexical similarity between the domain stem and gold standard list stems. The LLM is then given the domain, example page URLs, corresponding content from those URLS, and the candidate list. It is prompted to choose `match, no-match or uncertain`, the matching candidate domain if match is chosen, some brief rationale or evidence, and optionally flag the instance for manual review. Instances that are flagged by the LLM for the manual review flag ($N = 14$) and those in which the LLM proposes a matching domain which was not in the candidate list ($N = 48$) are manually reviewed by researchers. The remaining relevant pages which do not match to a given gold standard entity ($N = 2,259$) generally fall into one of three categories: third party directory such as the ABA list of accredited law schools, news or other article content about the theme generally, but not one entity specifically, or similar entity not on the gold standard list such as companies which are part of the Fortune Global 500 or Universities located outside the US.

C Computational Cost

Table 5: Per component cost metrics. Token usage is aggregated across all 28 collection runs included in Table 2. Latency per call is in seconds.

Component	Total Tokens	Tokens per Page	Total Calls	Latency per Call
Forager	198,415,172	707.71	49,015	6.44
Critic	44,390,930	158.33	38,834	4.68
Scout	3,798,127	13.54	1,666	4.02
Fetch	0	0.0000	344,588	0.75
Search	0	0.0000	7,571	2.01

Table 6: Token usage per approach, aggregated across all 4 topics included in Table 2.

Approach	Forager Tokens	Critic Tokens	Scout Tokens	Total Tokens
DeepResearcher	-	-	-	2,258,943
No-Reseed	39,005,430	5,442,936	2,472	44,450,838
Fixed-Reseed	23,891,382	6,762,778	563,082	31,217,242
Threshold	21,800,783	6,173,678	822,458	28,796,919
Realized-Mean	29,978,153	6,195,291	341,875	36,515,319
Realized-EMA	30,342,823	6,054,647	259,603	36,657,073
Scent-Mean	25,317,440	6,779,714	925,704	33,022,858
Scent-EMA	28,079,161	6,981,886	882,933	35,943,980

We report token usage and latency by system component to characterize the computational cost of PatchScout.

Forager This component accounts for the largest share of LLM usage, consuming 198.4M tokens, or approximately 708 tokens per visited page, with an average latency of 6.44 seconds per call. This reflects the fact that the Forager is invoked throughout local exploration to rank frontier links, and

Table 7: Full-budget token efficiency per topic. Tok is total LLM tokens; Pg/1K is relevant pages discovered per 1,000 tokens; Cov/1M is the *number* of reference-set entities discovered per million tokens (not a percentage). Claude Deep Research exposes no usage data. Best value per column in bold.

	Universities			**Fortune 500**			**Law Schools**			**ASPs**		
System	Tok	Pg/1K	Cov/1M	Tok	Pg/1K	Cov/1M	Tok	Pg/1K	Cov/1M	Tok	Pg/1K	Cov/1M
DeepResearcher	477K	0.16	10.5	340K	0.01	11.7	680K	0.00	2.9	761K	**0.08**	**30.2**
Claude Deep Research	–	–	–	–	–	–	–	–	–	–	–	–
Gemini Deep Research	707K	0.16	46.7	1.1M	0.08	14.1	1.8M	0.14	**102.7**	1.2M	0.07	10.7
No-Reseed	9.9M	0.19	3.1	11.6M	0.01	1.3	11.6M	0.00	0.2	11.4M	0.00	0.2
Fixed-Reseed	6.7M	0.93	44.4	7.6M	0.20	12.0	9.1M	0.30	15.7	7.8M	0.05	5.3
Threshold	3.4M	**1.53**	**79.3**	7.9M	0.29	**15.2**	9.7M	**0.39**	17.4	7.8M	**0.08**	6.6
Realized-Mean	10.4M	0.71	4.3	8.9M	**0.34**	9.6	9.6M	0.31	14.9	7.6M	0.06	5.5
Realized-EMA	10.9M	0.79	7.4	7.9M	0.29	12.6	10.2M	0.25	13.7	7.7M	0.05	4.6
Scent-Mean	7.3M	1.07	51.4	7.4M	0.24	13.9	10.4M	0.32	15.0	8.0M	0.06	5.5
Scent-EMA	7.6M	0.90	37.6	8.9M	0.23	13.8	11.0M	0.34	15.6	8.4M	0.07	5.6

Table 8: Token-matched comparison: pages visited (Pg) and gold-standard domain coverage (Cov%) for each PatchScout system if stopped at the same total LLM token budget the DeepResearcher baseline used for that topic. Claude and Gemini Deep Research rows show final, unmatched results for reference: Gemini reports only total usage and Claude reports none, so neither can be truncated. ACHE has no LLM token cost and is omitted. Best completed Pg/Cov per topic in bold (baseline reference rows excluded from bolding).

	Universities		**Fortune 500**		**Law Schools**		**ASPs**	
System	Pg	Cov	Pg	Cov	Pg	Cov	Pg	Cov
DeepResearcher	78	0.3	56	0.8	28	1.0	64	17.3
Claude Deep Research	-	1.1	-	15.0	-	91.9	-	17.3
Gemini Deep Research	113	1.8	82	3.0	247	93.4	87	9.8
No-Reseed	716	1.4	893	2.6	1,556	0.5	1,063	1.5
Fixed-Reseed	1,072	5.7	1,405	6.0	1,995	34.0	1,568	8.3
Threshold	1,914	2.3	1,364	**9.0**	1,871	31.0	1,579	**18.8**
Realized-Mean	687	0.1	1,205	3.2	1,891	16.2	1,606	9.0
Realized-EMA	645	1.1	1,375	4.6	1,780	12.2	1,597	10.5
Scent-Mean	982	**6.7**	1,477	6.6	1,756	36.0	1,529	11.3
Scent-EMA	951	5.6	1,223	7.6	1,664	**40.1**	1,491	12.8

thus processes many more links than the system ultimately visits. Each URL is passed to the Forager with context which includes the anchor text, surrounding text, source page URL, source page title and source page content snippet. The Forager was implemented with a batch based prompt, with a maximum batch size of 16. Across all 28 runs, the Forager scored $659,022$ URLs in $49,015$ API calls (batches) for an average batch size of 13.4. On average, the *Scent* based re-seeding policies scored approximately 24% fewer URLs than the *Realized* variants, indicating that Forager scores are a good early indicator of patch degradation.

Critic This component consumed 44.4M tokens, or approximately 158 tokens per page, with an average latency of 4.68 seconds per call. This cost is incurred because every visited page must be classified for relevance in order to compute realized information gain and update patch productivity. We implement a heuristic content extractor based on HTML structure which prioritizes meaningful page content such as title, body, paragraph and article tags, and avoids noise such as navigation elements and metadata. For each page, the Critic is given the full title and the first 600 characters of remaining extracted content. The Critic was implemented with a batch based prompt, with a maximum batch size of 12. Across all it scored $280,000$ visited pages in $38,834$ calls for an average batch size of 7.2.

Scout Finally, the Scout is invoked only when entering a new patch, resulting in substantially lower total usage: 3.8M tokens, or approximately 13.5 tokens per visited page, with an average

latency of 4.02 seconds per call. The Scout prompt includes recent query performance history and requests a batch of 5 search engine queries. The relatively small Scout cost indicates that adaptive re-seeding itself is not the dominant source of computation; rather, the main cost comes from making fine-grained semantic decisions during sustained local exploration, which is required of any web interacting system.

Fetching and search engine calls introduce non-LLM latency, averaging 0.75 and 2.01 seconds per call, respectively, but do not contribute to token usage. The discrepancy between $280,000$ total visited pages and $344,588$ fetch calls is indicative of the challenges of operating on the live web; roughly 19% of fetches fail, with the majority due to 403 error (46.2%) and 404 error (33.9%), none of which are retried (Section 4.3).

All experiments were run from a single Linux workstation with an AMD Ryzen 5965WX 24-Core processor and 512 GB RAM running Ubuntu 20.04.5 and Python 3.12.13. LLM calls were made to the DeepSeek-V3.1 API and search queries were issued through SearchAPI over a standard internet connection. Reported latencies are client-observed wall-clock times and include network overhead, provider-side queuing, API response time, and local processing, rather than isolated model inference time. We limited concurrent DeepSeek API requests to at most 4 in flight at a time, as preliminary experiments showed that larger concurrency significantly increased per call latency. In total, the 28 runs reported in Table 2 completed in 83.69 hours for an average of 2.98 hours per run.

Tables 6–7 compare PatchScout's compute with the deep-research baselines. The gap is more than an order of magnitude over a full run and is concentrated in the Forager, which is invoked throughout local exploration rather than only at patch boundaries. At matched token budgets (Table 8), PatchScout visits 8–70$\times$ more pages than DeepResearcher. Over the full budget its tokens-per-entity is higher on Law Schools and ASPs, reflecting continued exploration after task-bounded agents stop. The Critic sensitivity study (Appendix F) suggests that a much smaller Critic model would reduce cost with little change in labels.

D Deep-Research Baselines

We evaluate DeepResearcher-7B in an inference-only setting using the model's native tool-use protocol, without any additional training or modification to the checkpoint. We load the public release model weights from Hugging Face, which is released under the Apache 2.0 License, run locally on a single NVIDIA A6000 and iteratively parsed model generations as either final answers or structured tool calls. Web search and browsing were handled through the repository's existing Flask-based handler: search queries were sent to the Serper API, while requested webpages were fetched with the repository's simple text browser and summarized by a separate OpenAI-compatible reader model. The original work uses `qwen-plus` for this reader model, but we choose to use DeepSeek-V3.1 to remain consistent with our other experiments. The architecture requires a number of hyperparameters, which we left on default settings: `search-top-k=10`, `max-new-tokens=12,000`, `temperature=0`, `top-p=0.95`, `max-turns=12`. All model outputs, tool calls, tool observations, timing events and token counts were logged for later analysis. We tested a variety of prompting strategies, included below, to try to induce the model to perform longer horizon discovery. Under all prompting strategies the model chose to terminate before utilizing the maximum tool use turns, and performed only shallow exploration. We utilized the highest coverage result from among the prompting strategies for each topic for the main results reported in Table 2.

We evaluated Claude Deep Research and Gemini Deep Research API preview, runs on on all four topics. From each final report we extracted all cited URLs, and for Gemini also the intermediate URLs it visited. These URLs were judged by the same Critic and mapped to entities with the same procedure as all other systems (Appendix B). Claude Deep Research exposes no token or tool-call statistics; Gemini exposes total token consumption only, so neither can be compared at a matched token budget.

D.1 DeepResearcher Prompts

DeepResearcher Collection Oriented Prompt

You are not answering a question. You are performing thematic web data collection.
Your goal is to discover as many distinct relevant entities and source domains as possible for the following theme:

`<theme-description>`

Do not stop after finding a few examples. Continue searching, browsing, and expanding until you have exhausted the available tool-use budget or can no longer find new relevant entities.
For each search/browse step:

1. Identify new candidate entities or domains.
2. Follow promising links to verify whether they are relevant.
3. Avoid repeatedly visiting the same domain unless it is needed to verify a new entity.
4. Prefer breadth across distinct entities over depth within a single entity.
5. Keep track of entities already discovered and actively search for new ones.

Your final answer should be a structured collection report containing:

- A list of discovered relevant entities.
- The source URL or domain for each entity.
- A brief relevance justification.

DeepResearcher Anti-Answer-Sufficiency Prompt

The task below is not a question-answering task. Do not stop when you have enough information to write a summary.
Your task is to discover a broad collection of relevant entities and source domains for:

`<theme-description>`

You should treat each discovered relevant page as evidence for one entity, but the goal is to maximize the number of distinct relevant entities. A good response is not a polished answer; a good response is a large, diverse, verified collection.
Continue searching and browsing even after you can describe the topic. Early stopping is a failure unless additional exploration is unlikely to discover new entities.
During exploration:

- Maintain a running list of discovered entities.
- Maintain a list of used queries.
- Generate new queries that differ meaningfully from earlier ones.
- Follow directory pages, institutional pages, and links that may reveal additional entities.
- Switch away from sources that only produce duplicate entities.

Your final answer should be a structured collection report containing:

- A list of discovered relevant entities.
- The source URL or domain for each entity.
- A brief relevance justification.

DeepResearcher Exploration Policy Prompt

You are a thematic web data collection agent.
Your task is to build a broad collection of relevant entities and source URLs for the theme below. This is not a question-answering task: do not stop once you can summarize the topic. Continue exploring to maximize the number of distinct relevant entities discovered.
Theme:

<THEME>

Collection objective:

- Maximize distinct relevant entities.
- Prefer breadth across domains over depth within one domain.
- Verify relevance using source pages whenever possible.
- Avoid duplicate entities and duplicate domains unless needed for verification.
- Use search results, directory pages, and discovered links as entry points into new regions of the web.

Exploration policy:

1. Generate a search query likely to discover relevant entities.
2. Browse promising results and record distinct relevant entities.
3. Follow links that may reveal additional entities.
4. When the current source or query stops producing new entities, switch to a meaningfully different query.
5. Use prior discoveries and failures to choose later queries.
6. Continue until the tool budget is exhausted or several diverse searches produce no new entities.

Your final answer should be a structured collection report containing:

- A list of discovered relevant entities.
- The source URL or domain for each entity.
- A brief relevance justification.

E Ablation Study

Table 9: Ablation results. Each topic reports harvest rate (HR%) and gold-standard domain coverage (Cov%), matching the main results table. Best value within each topic/metric is bolded.

	Universities		**Fortune 500**		**Law Schools**		**ASPs**	
System	HR	Cov	HR	Cov	HR	Cov	HR	Cov
Random-Reseed	72.1	4.6	7.6	2.2	27.8	25.4	4.9	9.0
Random-Forager	53.5	1.2	10.6	2.0	21.5	11.7	4.6	12.0
Upfront-Scout	64.3	8.3	16.9	6.6	28.6	28.9	3.7	10.5
Realized-EMA	**86.2**	4.4	**23.2**	19.8	25.3	70.6	3.7	26.3
Scent-EMA	68.3	**15.5**	20.1	**24.6**	**37.2**	**86.8**	**5.6**	**35.3**

We evaluate three ablations to isolate the main decision components of PatchScout. *Random-Reseed* replaces the policy-guided re-seeding decision with random patch switching, while preserving repeated entry into new search-induced patches. Analysis of our re-seeding policy variants revealed that across topics and policies reseeding is triggered on roughly 10% of decision points, so we enforce a 10% chance to reseed randomly. *Upfront-Scout* generates all queries before collection begins, similar to prior work on seed finding [Vieira et al., 2016], removing feedback from observed patch outcomes and Critic judgments during later patch discovery. These two ablations utilize the *Scent-EMA* re-seeding policy, as main experiments find it to have the best consistent performance. *Random-Forager* replaces information-scent based frontier ranking with random link selection, testing the value of guided within-patch exploitation. Given that this ablation does not utilize Forager scores and thus has no *scent* signal, we utilize *Realized-EMA* as the reseeding policy. These ablations respectively test whether performance depends on patch switching policy, adaptive query generation and information-scent based link selection.

As shown in Table 9, all three ablations reduce performance relative to the full PatchScout system. The performance under *Random-Reseed* shows that coverage gains are not explained solely by the

number of re-seeding events; rather, the timing of patch switches matters. The weaker performance of *Upfront-Scout* suggests that collection history is informative for the Scout, allowing it to formulate better queries over the course of collection. Finally, the degradation of harvest rate under *Random-Forager* indicates that the Forager agent link priority scores are meaningful predictions of URL relevance, thus guiding local exploitation to high-yield pages and serving as informative information-scent. Overall, the ablations show that PatchScout's performance depends on the interaction between adaptive discovery, guided exploitation, and policy-based re-seeding.

Table 10: Pairwise agreement between critic models on the binary relevance decision. For each pair, agreement (Agr%), precision (Prec%), and recall (Rec%) treat the *row* model as the prediction and the *column* model as the reference. All four models judged the same 10,000 pages, with the same binary Yes/No prompt; only the model differs.

	Reference model											
	DeepSeek			**Qwen 9B**			**Qwen 27B**			**Qwen 397B**		
Prediction	Agr	Prec	Rec	Agr	Prec	Rec	Agr	Prec	Rec	Agr	Prec	Rec
DeepSeek	–	–	–	86.1	81.1	86.9	87.4	79.9	90.8	86.9	77.6	91.9
Qwen 9B	86.1	86.9	81.1	–	–	–	88.7	83.6	88.6	88.0	80.9	89.4
Qwen 27B	87.4	90.8	79.9	88.7	88.6	83.6	–	–	–	92.0	87.8	91.5
Qwen 397B	86.9	91.9	77.6	88.0	89.4	80.9	92.0	91.5	87.8	–	–	–

Table 11: Critic 4-way relabel of sampled pages, reclassified in their original critic batches (same pages, same order, same batch-mates as the first pass): full breakdown of where originally *relevant* and originally *not-relevant* pages land under the four graded labels (not_relevant / topically_related_no_evidence / partial_evidence / direct_evidence). Rows sum to 100% (rounding). Combined pools all four topics.

Topic	**Orig. label**	**Not rel.**	**Topical, no ev.**	**Partial ev.**	**Direct ev.**
Universities	Relevant	5.8	37.9	9.3	47.1
Universities	Not relevant	72.9	19.5	2.0	5.6
Fortune 500	Relevant	5.8	9.8	36.3	48.1
Fortune 500	Not relevant	85.4	10.3	3.8	0.4
Law Schools	Relevant	2.8	22.4	41.8	32.9
Law Schools	Not relevant	66.6	28.0	5.2	0.2
ASPs	Relevant	1.4	12.4	58.0	28.2
ASPs	Not relevant	77.8	17.9	4.3	0.0
Combined	Relevant	**4.2**	**21.6**	**33.8**	**40.4**
Combined	Not relevant	**75.9**	**18.8**	**3.9**	**1.4**

F Critic Sensitivity

Graded relevance. We relabelled 9,983 pages—4,457 originally judged relevant and 5,526 originally judged not relevant—on a four-point evidence scale (not relevant; topically related without theme evidence; partial evidence; direct evidence). Pages were sampled randomly across all four topics, targeting balance between the two original labels; because sampling operated on whole Critic batches, balance could not be maintained on ASPs, where relevant pages are scarce. Pages were reclassified in their original batches with the same ordering and batch-mates, so batching effects do not confound the comparison. Table 11 shows that the binary boundary is meaningful but loose at the margin: 74.2% of pages labelled relevant carry at least partial theme evidence and 40.4% carry direct evidence, while 25.8% are either not relevant (4.2%) or topically related without theme evidence (21.6%). The false-negative rate is low: 1.4% of pages labelled not relevant carry direct evidence. Universities is the outlier, with 43.7% of relevant pages lacking direct evidence; this is partly a rubric artefact, since that topic's criterion is any page within a university domain, so topically related pages without specific evidence still satisfy the task definition.

Critic model. We re-judged 10,000 pages with three additional models using the identical binary prompt (Table 10). Agreement with the production Critic is flat in model scale (86.1%, 87.4%, and 86.9% for 9B, 27B, and 397B), so binary judgment over a title and short snippet appears to saturate well below the scale we use [Dietz et al., 2025]. Within-family agreement (88.0–92.0%) exceeds cross-family agreement (86.1–87.4%), so the cross-family figures are the conservative estimate of reproducibility. Disagreement is systematic: treating our Critic as the prediction, its precision against successively larger references falls (81.1, 79.9, 77.6) while recall rises (86.9, 90.8, 91.9), so it labels relevant a superset of what larger models accept. Both analyses indicate that harvest rate is optimistic at the margin; since baseline pages are scored by the same Critic, this affects absolute rates rather than the comparison between systems.

G Agent Prompts

PatchScout uses three LLM-backed agents. The templates below show the prompts used in the experiments, with run-specific values represented by angle-bracketed placeholders.

G.1 Scout Agent

Scout System Prompt

You are the Scout Agent in a patch-based thematic web-data collection system. Your job is to propose high-precision search queries that discover new relevant patches. Return only valid JSON. Do not include markdown, prose, or code fences.

Scout Discovery Prompt

Task: Generate exactly `<n_queries>` distinct search queries for patch discovery.

Theme description: `<theme_description>`

`<topic-specific rules>`

Requirements:

1. Return exactly `<n_queries>` executable search queries.
2. Prefer precision over recall.
3. Prioritize authoritative, information-dense sources.
4. Do not produce queries that differ only by term order, pluralization, or synonym swaps.
5. At most two queries in this batch may start with the same operator prefix (e.g., `site:*.edu`, `inurl:...`).
6. Use exclusion terms selectively; do not repeat long negative suffix bundles across the whole batch.
7. Prefer HTML-first discovery pages.
8. Avoid filetype filters by default; do not use `filetype:pdf`.
9. Do not output explanations outside JSON.

Return JSON with exactly this schema:

```
{
  "queries": ["q1", "q2"],
  "reason": "short rationale"
}
```

Scout Re-seed Prompt

Task: Generate exactly `<n_queries>` new re-seed queries for discovering fresh patches.

Theme description: `<theme_description>`

`<topic-specific rules>`

Here are the most recent queries you've suggested and their outcomes: <recent_query_outcomes>

Requirements:

1. Return exactly <n_queries> new queries.
2. Queries must be meaningfully different from recently used ones.
3. Do not produce queries that differ only by term order, pluralization, or synonym swaps.
4. At most two queries in this batch may start with the same operator prefix (e.g., site:*.edu, inurl:...).
5. Use exclusion terms selectively; avoid repeating long negative suffix bundles across the batch.
6. If recent high-yield queries are topically narrow, expand laterally to adjacent entities/subdomains instead of refining the same pattern.
7. Bias toward unexplored domains/entities/subtopics suggested by recent outcomes.
8. Reduce patterns associated with low-yield patches.
9. Prefer precise, verifiable source discovery with HTML-first pages.
10. Avoid filetype filters by default; do not use filetype:pdf unless absolutely necessary.
11. Do not output explanations outside JSON.

Return JSON with exactly this schema:

```
{
  "queries": ["q1", "q2"],
  "reason": "short rationale tied to recent outcomes"
}
```

Scout No-Fresh-Results Retry Instruction

Retry condition: The previous discovery queries returned no fresh pages after filtering. This is retry attempt <retry_attempt>. Generate substantially more diverse query structures and entity angles than before. Do not reuse operator skeletons from the previous batch. Avoid near-duplicates of prior phrasing and operator patterns. If recent high-yield queries are topically narrow, expand laterally to adjacent entities/subdomains instead of refining the same pattern. Prioritize novelty while staying on-topic.

Scout Recent Query Outcomes Section

The adaptive Scout re-seed prompt includes recent query outcomes from the run state. The section contains up to <query_history_limit> recent queries, with a default limit of 12. Each row is serialized as:

```
- query: "<query>" | relevant pages: <count> | gain-cost ratio: <ratio>
```

If a recent query has no associated patch record, the row is serialized as:

```
- query: "<query_text>" | relevant pages: n/a | gain-cost ratio: n/a
```

If there is no query history, the section is:

```
- <no prior queries>
```

G.2 Forager Agent

Forager Link-Scoring System Prompt

You are the Forager Agent in a thematic web data collection system. You score candidate links for traversal priority. Precision is more important than recall. Return only valid JSON.

Forager Link-Scoring Prompt

Task: Score candidate links for thematic relevance and expected utility for local traversal.

Theme description: `<theme_description>`

`<topic-specific rules>`

`<recent scored-link outcomes for calibration>`

Candidate links JSON:

```
{
  "theme_description": "<theme_description>",
  "candidate_links": [
    {
      "url": "...",
      "anchor_text": "...",
      "surrounding_text": "...",
      "parent_url": "...",
      "source_url": "...",
      "source_title": "...",
      "source_snippet": "..."
    }
  ]
}
```

Scoring guidelines:

- Score each URL in $[0, 1]$.
- Use URL, anchor text, surrounding text, source title, and source snippet jointly.
- Prefer links likely to contain substantive thematic content.
- Penalize generic navigation/account/help/legal/cart/login pages, weak-context links, or likely low-information targets.
- If uncertain, score lower.

Use exactly these four score bins:

1. 0.85 to 1.00: Highly likely relevant and high utility for patch exploitation.
2. 0.60 to 0.84: Likely relevant, moderate-to-high utility.
3. 0.30 to 0.59: Weak/indirect relevance, low utility.
4. 0.00 to 0.29: Likely irrelevant or unproductive.

Output constraints:

1. Score every input URL exactly once.
2. Do not invent or omit URLs.
3. Do not include explanations or reasons.
4. Return JSON only.

Return JSON with exactly this schema:

```
{
  "scored_links": [
    {"url": "...", "score": 0.0}
  ]
}
```

Forager Feedback Calibration Section

Recent scored-link outcomes for calibration: These are examples of what was previously sent to the Forager scorer and what happened after Critic classification.

```
{
  "positive_examples": [
    {
      "url": "...",
      "anchor_text": "...",
      "surrounding_text": "...",
      "source_title": "...",
      "source_snippet": "...",
      "assigned_score": 0.0,
      "critic_relevance": true,
      "critic_reason": "..."
    }
  ],
  "negative_examples": [...]
}
```

G.3 Critic Agent

Critic System Prompt

You are the Critic Agent for binary relevance classification in thematic web-data collection. Classify each page as relevant or not relevant to the theme. Return only valid JSON.

Critic Batch Prompt

Task: Theme classification for multiple pages. Return JSON only.

Theme description: `<theme_description>`

`<topic-specific rules>`

Decision rule:

- "Yes" if directly relevant to the theme or a strong bridge page.
- "No" if weak relevance, generic/navigation/legal-only, or low-information.

Input JSON:

```
{
  "theme_description": "<theme_description>",
  "pages": [
    {"id": 0, "title": "...", "snippet": "..."}
  ]
}
```

Output constraints:

1. Return exactly one result for each input id.
2. Do not invent or omit ids.
3. Confidence in $[0, 1]$.
4. Reason ≤ 40 words.

Return JSON schema:

```
{
  "results": [
    {"id": 0, "pred": "Yes" or "No", "reason": "...", "confidence": 0.0}
  ]
}
```

G.4 Theme-Specific Prompt Elements

The following **theme descriptions** were used for each theme, across all configurations:

- University: "Four year colleges or universities in the United States"
- Fortune 500: "Fortune 500 company job postings, career pages, and employment opportunity pages that specifically describe jobs or hiring opportunities at Fortune 500 companies"
- Law Schools: "Accredited law schools in the United States (ABA-approved)"
- ASP: "Autism support programs at colleges and universities in the United States that provide additional support beyond standard disability services specifically for autistic students"

The following addenda are inserted conditionally based on the configured theme. Each theme contributes separate instructions to the Scout, Forager, and Critic prompts.

Topic Addendum: U.S. Four-Year Colleges

Scout guidance. Prioritize official U.S. four-year institution pages; use negatives for community-college, two-year, law, medical, and graduate-only drift; avoid alumni, athletics, giving, bookstore, events, login, portal, and help surfaces.

Forager guidance. Score official institution home, about, admissions, undergraduate, and academics pages highest; score auxiliary institution pages lower; penalize two-year, professional-only, alumni, athletics, giving, events, login, and help pages.

Critic rule. Classify pages hosted by qualifying U.S. four-year college or university domains as relevant, including auxiliary pages; reject two-year, standalone professional/graduate-only, and unrelated third-party sites.

Topic Addendum: U.S. Accredited Law Schools

Scout guidance. Prioritize official U.S. law school pages and authoritative accreditation/listing sources; favor ABA-approved schools, J.D. admissions, and school-of-law domains; avoid directories, forums, generic legal news, bar prep, law firms, and legal aid.

Forager guidance. Score official U.S. law school, ABA/accreditation, J.D. admissions, curriculum, tuition, faculty, and accreditation pages highest; score clear university routes to law schools moderately; penalize unrelated legal content and boilerplate.

Critic rule. Classify official pages for U.S. accredited/ABA-approved law schools or their university law-school sections as relevant; reject law firms, bar prep, legal aid, legal news/blog pages, and non-U.S. or non-accredited institutions.

Topic Addendum: College Autism Support Programs

Scout guidance. Prioritize official college/university autism, ASD, and neurodiversity support program pages; require support beyond ordinary accommodations; avoid K–12, clinics, therapy providers, research-only centers, awareness pages, and parent-resource directories.

Forager guidance. Score official autism, ASD, neurodiversity support program pages and program admissions/services/coaching/transition/mentoring/eligibility/FAQ pages highest; penalize generic accommodations, K–12, clinic, research-only, awareness, parent-resource, login/help, and directory-only pages.

Critic rule. Classify pages describing autism-specific, ASD-specific, or neurodiversity support programs for college students as relevant, including pages listing such programs; reject generic disability services unless autism-specific structured programming is clear.

Topic Addendum: Fortune 500 Job Postings

Scout guidance. Prioritize official company career sites, job-detail pages, and ATS surfaces with clear employer and opportunity signals; avoid news, investor pages, aggregators, salary/review sites, recruiting agencies, and generic career advice.

Forager guidance. Score official job-detail, career-search, application, ATS, internship, early-career, team, location, and hiring-program pages highest; score career landings moderately; penalize culture-only pages, aggregators, recruiters, news, investor, press, advice, login/help, and product pages.

Critic rule. Classify official company career pages or ATS-hosted pages describing employment opportunities at Fortune 500 companies as relevant; reject aggregators, salary/review pages, recruiters, advice, investor/news/press pages, and pages without employment/application content.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction accurately reflect the results reported in Section 6 based on the experiments detailed in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations introduced by the system design and experimental setting are discussed in Section 8.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: We provide theoretical framing of the problem in Section 3 with a full set of assumptions, but do not claim to make any theoretical result or proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The framework and experiments are described in sufficient detail for reproducibility of the experiments in Sections 4 and 5 and Appendix G. As mentioned in Section 8, experiments were conducted on the live web, so exact reproducibility of the reported results is limited.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is available on GitHub.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 describes the experimental setting in sufficient detail to understand the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Each system configuration is evaluated in a single run due to the high computational cost of collection (approximately 3 hours and 8M LLM tokens per run). Running multiple trials per configuration was not feasible within the available budget. We report endpoint metrics and coverage curves across four distinct topics to provide breadth of evidence.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C gives a full account of the resources utilized to execute experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research involved in this work conforms to the Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We note that the reported work has the potential to benefit social science researchers. We acknowledge that comprehensive thematic web data collection is dual-use and carries inherent risk for surveillance and malicious scraping. We mitigate this risk by implementing the system with a read-only design, and do not release any model or corpus.

Additionally, we implement standard practices of 'crawler etiquette'; our system reads and respects robots.txt where available, enforces a global concurrency cap of eight requests, does not revisit a URL within a run, and does not retry requests returning access-control or rate-limit responses such as HTTP 403 or 429. This was implemented throughout all reported experiments.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release any dataset or models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The only existing assest used, DeepResearcher, is cited in the main text and the Apache 2.0 License it is released under is explicity mentioned in Appendix D.

    Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The only asset associated with the paper is the code, released on GitHub. It is well documented.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper describes the use of LLMs in detail in Sections 4 and 5. Full prompts are given in Appendix G.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.