Network-Level Prompt and Trait Leakage in Local Research Agents

Abstract

We show that Web and Research Agents (WRAs)—language model-based systems that investigate complex topics on the Internet—are vulnerable to inference attacks by passive network adversaries such as ISPs. These agents could be deployed internally by organizations and individuals for privacy, legal, or cost reduction purposes. Unlike sporadic human browsing, we show that a typical WRA visits $70-140$ domains with distinguishable timing correlations, enabling unique fingerprinting attacks.

We propose prompt and user trait leakage attack that only uses the IP address metadata of a WRA's session. We, first, obtain new datasets of WRA traces based on user search queries and queries generated by synthetic personas. We introduce OBELS, a behavioral metric for comprehensively assessing similarity between original and inferred prompts, showing that our attack successfully recovers functional and domain knowledge of user prompts. Extending to a multi-session setting, we recover XX of 32 latent traits with high accuracy.**[[Amir: are traits clear here?]]** Our attack remains effective under partial observability and noisy conditions. Finally, we discuss mitigation strategies that limit visited domains or obfuscate traces showing negligible utility impact but reduced attack effectiveness.

1 Introduction

Web and Research Agents (WRAs) are reshaping how users interact with information. Unlike traditional assistants that answer isolated queries, these Large Language Model (LLM)-powered systems autonomously plan, browse, and synthesize knowledge across the web [34]. Major AI providers are integrating such agents directly into their flagship assistants or browsers, including OpenAI [56], Google Gemini [27], Grok [82], Mistral [45], and Perplexity [59]. Open-source counterparts such as LangChain's Local Deep Researcher [39], GPT Researcher [25], and Agent Laboratory [65] are also widely adopted in academic and developer communities. Their growing deployment in assistants, productivity platforms, and enterprise environments, particularly within privacy-sensitive sectors such as healthcare, law, and finance, raises urgent concerns about their security and privacy implications. While on-device execution through locally-deployable models and TLS-encrypted access could suggest privacy guarantees, this perception is misleading. Yet WRAs leave unavoidable, structured metadata traces that can reveal sensitive user intent, creating an underexplored leakage vector.

As autonomous agent systems, WRAs inherit vulnerabilities from both LLMs and traditional web browsing, such as system prompt extraction [35], model memorization [16], metadata exposure [68], and behavioral fingerprinting [12]. Yet they also expose a distinct and underexplored vulnerability: their behavioral traces. They reflect the agent's internal reasoning process and can reveal sensitive or stigmatized user intent [54,72]. Even if prompts and page contents remain fully encrypted, externally visible metadata, such as domain names, access order, payload size, and timing, can silently expose user intent [54, 72]. Because WRAs autonomously execute multi-step research plans [87], they produce cascades of semantically related domain visits in dense temporal bursts [59]. These structured, high-throughput access patterns [21, 46] are unlike human browsing behavior and make WRAs particularly amenable to profiling and inference attacks [21,46] while introducing new risks through autonomous, high-throughput interaction with live websites.

We adopt a realistic threat model in which a passive adversary, such as an ISP, enterprise firewall, or local network operator, can observe only domain-level traffic metadata, without access to prompts, page contents, or model internals. Despite this extremely limited visibility, we show that adversaries can reliably recover what users asked and who they are.

Motivating Examples. Consider a user seeking reproductive health information: their research agent may visit clinics, support forums, and medical sites. Even without access to page contents, these traces alone can expose the agent's internal reasoning process and deeply stigmatized intent. As

Code available at https://anonymous.4open.science/r/wra-BD06

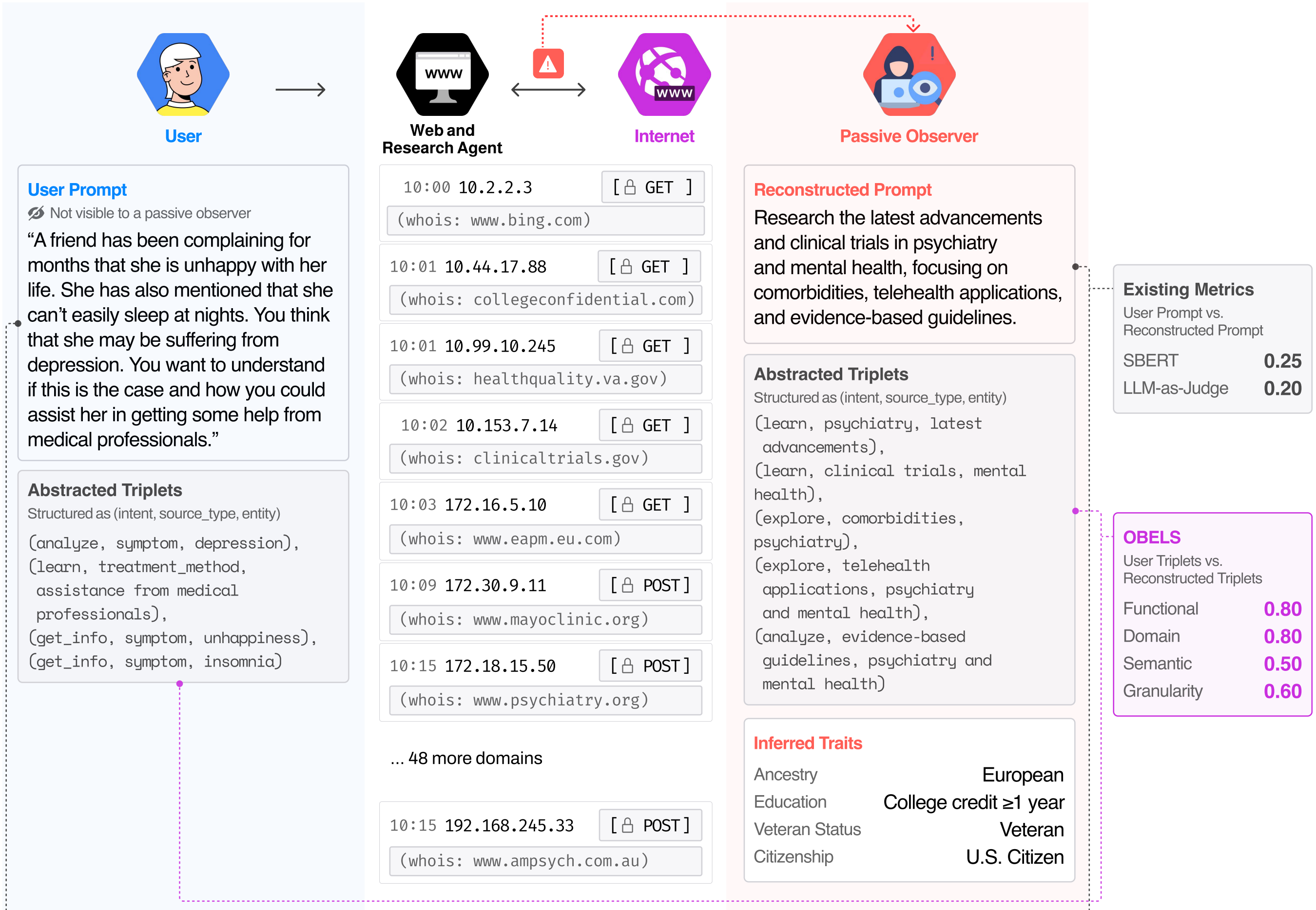

Figure 1: **Overview of the attack.**

WRAs become integrated into daily decision-making, their behavioral traces present a new leakage channel. We show that WRA visits 70-140 distinct domains per each query in a short period of time making an easy target to isolate. In essence, privacy failures emerge not from what agents say, but from how they act.

Our Work. We present privacy leakage attacks that exploit metadata traces of web and research agents deployed locally. The attacks include **(1) Prompt Reconstruction**, which recovers a user's original prompt from observed domain sequences, and **(2) Trait Inference**, which profiles latent attributes (e.g., gender, ideology) from the agent's multi-session browsing patterns. The adversary trains an inference model using task-specific strategies such as In-Context Learning (ICL) and fine-tuning to map browsing traces to inference targets. Our results show that the attacks remain effective even when 40% of the traces are masked or fully obfuscated, revealing that encrypted content alone cannot guarantee privacy. We observe that WRAs visit dozens of marginal or redundant domains that do not improve the final report but still expand the adversary's visibility. To mitigate these risks, we propose and evaluate practical defenses that either **hide traces**, by injecting LLM-generated decoy prompts and virtual personas to confuse inference, or **block traces**, by redirecting tasks to multipurpose sources or LLM knowledge, avoiding uniquely identifying domains.

Our contributions are as follows:

- **New attack surface:** We introduce metadata-based privacy attacks from behavioral traces produced by WRAs and propose a general two-stage pipeline for prompt reconstruction and trait inference.
- **Benchmark dataset and OBELS metric:** We construct and will release a dataset linking domain traces to prompts and persona traits, and introduce OBELS, a new ontology-aware metric for evaluating privacy leakage through behavioral traces.
- **Ablation and countermeasures:** We conduct extensive experiments across agents, LLMs, and masking conditions, and outline potential mitigation directions.

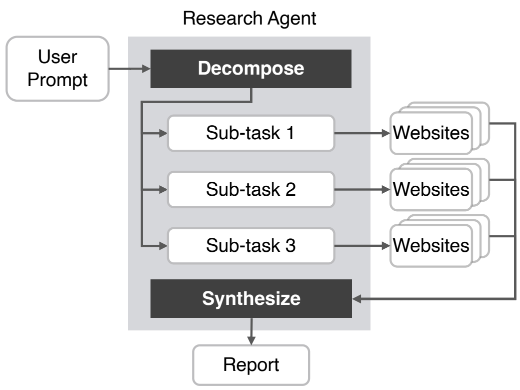

Figure 2: **Workflow of Web and Research Agents.**

2 Background and Related Work

2.1 AI, Web, and Research Agents

Standalone LLMs to AI Agents. LLMs excel at language understanding but are limited as static knowledge systems [6,74]. They cannot access real-time information, call external APIs, or execute multi-step plans, but produce responses based only on their pretrained knowledge. Early solutions such as RAG [28,42] and tool-augmented LLMs [20,60,63,67] improved grounding and tool use, but remain static and reactive, invoking tools only when explicitly instructed. AI agents address this by embedding LLMs into a perception–action loop [69,76,83], where the model plans, executes, and adapts across multiple steps, enabling autonomous reasoning and dynamic interaction.

Web Agents. Generic agents can reason about which queries to run, but often lack robust capabilities for controlling browsers, parsing dynamic content, handling JavaScript-heavy pages, or managing session state across multiple sites. Web agents integrate LLM planning with browser control: issuing queries, navigating between sources, extracting content, and adapting to dynamic pages [21]. Frameworks like AutoGen [81] and Browser-Use [47] exemplify these capabilities, though evaluations often emphasize narrow navigation tasks.

Research Web Agents (WRAs). WRAs operationalize deep evidence synthesis through a *search–retrieve–extract–synthesize* loop (Figure 2). Given a user query, the agent generates sub-queries and issues them via APIs such as Tavily [4] to minimize token overhead and retrieve relevant documents efficiently. Retrieved content is parsed into evidence, distilled into notes, and evaluated to decide whether further exploration is needed. This process repeats until sufficient coverage is achieved, at which point the agent composes a comprehensive, citation-rich output [21,34,59,64,87]. Systems exemplifying this paradigm include GPT Researcher [25], LangChain's Local Deep Researcher [39], OpenAI Deep Research [56], Grok DeepSearch [82], and Gemini Deep Research [27].

Because WRAs generate dense browsing traces over dozens of domains, they amplify metadata-leakage risks. This risk was underscored when Grok and OpenAI misconfigurations exposed private prompts to public indexing [44,71]. Locally deployed agents mitigate such threats. Running on personal machines or servers, they preserve confidentiality while offering lower latency, offline use, and lower cost. Examples include GPT-Researcher [25] with self-hosted backends such as Ollama [55] and AMD's GAIA [7].

2.2 Privacy Attacks

Threats in LLMs. LLMs introduce unique privacy risks due to their training data and the ways users interact with them. One major class of attacks focuses on hidden-input inference, recovering sensitive inputs such as training data, system prompts, or user queries [15, 16, 35, 43, 58, 79, 86]. Other work has focused on recovering hidden system prompts used internally by providers [35,86], where adversaries probe model input and output patterns to elicit protected instructions. Unlike these approaches, our setting assumes *no direct access to the model* and relies instead on indirect behavioral signals.

Beyond direct extraction, studies show that semantic leakage occurs even when the attacker only observes generated outputs. Pasquini et al. [58] and Liu et al. [43] found that reconstructed prompts often preserve enough semantic content for intent inference, while Carlini et al. [15] and Weiss et al. [79] demonstrated that response timing and token-length patterns can reveal coarse-grained information about prompt topics. These attacks, however, assume access to model outputs or fine-grained response timings. By contrast, we show that **passively observable network-level metadata alone**, without accessing queries or responses, can be sufficient to infer hidden prompts.

LLMs also introduce privacy risks through trait inference and user profiling. Their outputs carry author-identifying signals [10, 32, 49, 66], allowing inference of attributes such as age, gender, and personality. Other risks include speculative decoding revealing fragments of private inputs [78] and long-context memory retaining user data across sessions for later retrieval [75]. Unlike these works, we assume a **strictly weaker threat model**: the adversary does **not** observe model inputs, outputs, or internal states, relying solely on external traffic traces (e.g., domain sequences, timing, payload sizes).

Threats in Web Privacy. Metadata leakage has long been recognized as a central privacy risk on the web. Classical work on anonymous communications (e.g., MIX Nets [19] and Tor [23]) demonstrated that encrypting content is insufficient when traffic patterns remain exposed. Website-fingerprinting studies [13,29,37,80] showed that adversaries can infer visited pages based on encrypted flow patterns, while large-scale deanonymization attacks like the Netflix attack [50] revealed that sparse behavioral traces are often enough to re-identify individuals. Subsequent work confirmed that traffic features, such as packet sizes, burst patterns, and inter-arrival timing, can fingerprint websites even under protections like Tor [30, 57]. Parallel work in web tracking exposed persistent profiling vectors, including cookie-based tracking [41], cookieless fingerprinting [5, 51], and evercookies [8].

Recent work has shifted to domain-level metadata as a source of behavioral leakage. Oliveira et al. [54] showed that as few as four domains can uniquely re-identify most users in real-world datasets. Crichton et al. [22] found that domain sequence fingerprints remain stable across years, even after cookie resets. Cross-context linkage is also possible: Naini et al. [48] and Su et al. [72] demonstrated that anonymized browsing logs can be matched across platforms, breaking assumed protections. Unlike fingerprinting, we attempt to reconstruct prompts or user traits that WRAs leak.

While protocol-level defenses such as HTTPS, DNS-over-HTTPS (DoH), and DNS-over-TLS (DoT) aim to hide domain names too, they often fail to conceal key identifiers. Hoang et al. [31] showed that TLS handshakes frequently leak domain information, while Sunahara et al. [73] pointed out widespread misconfigurations in encrypted DNS deployments, enabling persistent metadata leakage. Similarly, Wang et al. [77] demonstrated that in-app telemetry (e.g., navigation events, click logs, and tracking APIs) leaks fine-grained behavioral patterns to platforms. Unlike these works, we assume a weaker adversary: a passive network observer without privileged telemetry access, relying only on network-level domain sequences and packet metadata.

Agentic Privacy and Security Threats. AI agents amplify privacy risks by combining LLM reasoning with active browsing and external tool integration [24]. Kim et al. [38] showed that compromised agents can scrape PII from visited pages and reuse it in phishing, while InjecAgent [85] demonstrated that crafted payloads in webpages can trigger secret retrieval or unauthorized service access. Defenses such as AirGapAgent [11] and CHeaT [9] aim to counter these threats through context isolation, behavior cloaking, and honey-token traps.

However, these prior studies assume an adversary with visibility into the agent's queries, outputs, or injected prompts. In contrast, our work identifies a **new and underexplored attack surface**: even without access to agent inputs, outputs, or model internals, **domain sequences and packet-level metadata alone** can be exploited to reconstruct hidden prompts and infer sensitive user traits. This highlights a distinct dimension of privacy leakage introduced by web and research agents, where autonomous decision-making produces metadata-rich browsing patterns vulnerable to adversarial inference. To our knowledge, this is the first study to identify and exploit metadata-driven leakage in WRAs.

3 Overview: Prompt and Trait Leakage Attack

We study a *passive network adversary* that monitors encrypted traffic between the AI agent (acting on behalf of the user) and the internet. Without access to page contents or search queries, the adversary can exploit metadata leakage to recover sensitive information about the user's inputs and attributes.

Problem Definition. WRAs execute user prompts by issuing queries, following links, and retrieving information, leaving behind browsing traces that encode sensitive signals. Even if the underlying page contents or prompts are hidden, the sequence and diversity of domains can reveal the intent behind a query, while recurring patterns across sessions expose persistent user traits. An adversary has clear incentives to exploit these inferences: prompt recovery compromises user intent privacy, while trait inference enables long-term profiling that can be monetized, surveilled, or used for manipulation.

Prompt Reconstruction. With proxy dataset $D_{\mathrm{PR}} = \{(WRA(p_j), p_j)\}$ consisting of pairs indexed by j, the adversary constructs an inference model M_{PR} (via ICL or fine-tuning). For a new prompt p with trace $t = WRA(p)$ (optionally filtered to $\tilde{t}$), the adversary recovers the prompt $\hat{p}$, aiming to maximize functional similarity:

$$\hat{p} = M_{\mathrm{PR}}(\tilde{t}), \quad \max \ \mathrm{Sim}\big(M_{\mathrm{PR}}(\tilde{t}), p\big).$$

Trait Inference. Let τ denote a user's latent trait vector and $\{p_i\}_{i=1}^{N}$ the prompts across N sessions, yielding $t^{1:N} = WRA(\{p_i\}_{i=1}^{N})$. From proxy dataset $D_{\mathrm{TI}} = \{(WRA(\{p_i^{(j)}\}_{i=1}^{N}), \tau_j)\}$, the adversary instantiates an inference model M_{TI} (ICL). On a new multi-session trace $t^{1:N}$ and infers trait $\hat{\tau}$, maximizing agreement:

$$\hat{\tau} = M_{\mathrm{TI}}(\tilde{t}^{1:N}), \quad \max \ \mathrm{Score}\big(M_{\mathrm{TI}}(\tilde{t}^{1:N}), \tau\big).$$

where Sim and Score denote task-appropriate semantic similarity and trait-agreement metrics, respectively.

3.1 Threat Model

Adversary Goals. The adversary's objective is to infer sensitive information about the user from the browsing trace. We focus on two representative leakage goals that capture both short- and long-term risks. The first is **prompt reconstruction**, where the adversary seeks to infer queries functionally equivalent to the user's original input (e.g., detecting that a user asked about "signs of depression and seeking professional mental health support"). The second is **trait inference**, where the adversary profiles latent attributes such as gender, religion, and political ideology by correlating patterns across multiple browsing sessions. Together, these goals illustrate how metadata can expose both immediate user intent and persistent personal characteristics.

Adversary Capabilities. The adversary can observe domain names accessed by the agent, obtained through plaintext DNS or TLS handshake metadata such as Server Name Indication (SNI). In addition, the adversary can track the sequence of domains accessed, capturing the order of domain visits during the agent's browsing sessions, as well as packet-level metadata including timing and payload lengths. By contrast, the adversary does *not* see full URLs, such as query strings (e.g., `/search?q=...`), page contents including articles or forms, the plaintext of the user's original prompts, or the agent's intermediate queries. Visibility may also be reduced by Content Delivery Networks (CDNs), reverse proxies, or load balancers that aggregate or mask requests. To capture these cases, we also consider partial-visibility settings where only a subset of domains (e.g., 80%, 60%) is observable. These capabilities align with realistic adversaries such as local ISPs, corporate firewalls, VPN providers, or other on-path observers that collect encrypted metadata without decryption or manipulation.

Adversary Assumptions. We assume the user interacts with a local web or research agent that autonomously performs multi-hop web browsing in response to a natural language prompt. The adversary is aware of the agent's general browsing capabilities, such as issuing web searches, clicking links, scrolling pages, or scraping structured data from web pages. We further assume that the adversary observes the entire session between the agent and the internet rather than isolated flows, that DNS and TLS handshake metadata remain visible (i.e., no DoH, DoT, or ECH), and that users do not employ VPNs, Tor, or other anonymizing proxies that would conceal domain sequences completely. Even under full-session monitoring, however, proxy infrastructures such as CDNs may still limit visibility, exposing only a subset of traces.

3.2 Attack Scenarios

In this section, we illustrate when a passive adversary would actually deploy our prompt and trait leakage attack. Locally deployed WRAs are especially relevant in settings where users avoid reliance on remote APIs, reduce recurring costs, maintain control over proprietary workflows, or process highly confidential information that cannot be shared externally. In such cases, browsing traffic is generated directly on the client machine, exposing metadata to on-path observers.

Personal WRAs. An individual running a local WRA for personal tasks, such as health research, financial planning, or private study, may assume that confidentiality is preserved by avoiding cloud services. Yet even with local deployment, ISPs, enterprise firewalls, or other intermediaries can still observe traces and reconstruct prompts or infer private traits, enabling targeted advertising, discrimination, or phishing.

Academic WRAs. In university, students and faculty use local WRAs for daily tasks, coursework, or research, leaking information about health, politics, or unpublished work. External observers such as ISPs or competing research entities could exploit traces to reconstruct prompts or profile academic interests.

Industrial WRAs. At the company level, employees use local WRAs for competitor analysis, strategic planning (e.g., product roadmaps, M&A interests, financial planning), or handling confidential information (e.g., legal case files or patient records). These traces can be mined for business intelligence by competitors, ISPs, cloud providers, or even internal IT monitors. In startup companies, small teams could rely on local WRAs for cost savings or planning, but this may lead to leaks about funding strategies, partnership negotiations, or technical stack decisions. ISPs or data brokers could harvest and resell the information or insights to competitors.

State WRAs. At the government or state level, state-operated firewalls or regulators monitoring domestic traffic could reconstruct politically sensitive prompts or identify activists and vulnerable communities for closer surveillance.

Overall, local deployment makes browsing activity directly observable in ways that remote, server-side agents do not. For adversaries, such visibility creates actionable opportunities to reconstruct immediate user prompts or build long-term trait profiles, depending on the target and setting.

4 Attack Methodology

We present the end-to-end pipeline for our privacy attacks against AI agents. Both tasks—prompt reconstruction and trait inference—follow a shared two-stage process: (1) **offline model construction** on proxy datasets D_{PR} or D_{TI}, yielding M_{PR} or M_{TI}; and (2) **online attack execution** on real user traces t or $t^{1:N}$ to produce $\hat{p}$ or $\hat{\tau}$. This separation reflects adversary constraints: real user prompts are private and unavailable for training, so the adversary must first build an inference model on proxy data before deploying it. The pipeline remains structurally consistent across tasks, though the input sources and inference targets differ.

Figure 3 illustrates our two-stage attack pipeline, separating the adversary's model construction from the attack execution. A two-stage design is necessary because the adversary cannot directly obtain a large set of real user prompts paired with their browsing traces. Instead, the adversary first trains an inference model on proxy data that mimics real-world interactions, and then applies this trained model to actual users.

4.1 Proxy Prompt Generation

To train the inference model in Stage 1, the adversary generates *proxy prompts* that realistically simulate user behavior and produce informative traces t. Effective proxy prompts must balance realism, browsing induction, and inferential utility. (a) Prompts resemble natural-language queries real users might issue (health, education, planning), forming $(t, p) \in D_{\mathrm{PR}}$ or $(t^{1:N}, \tau) \in D_{\mathrm{TI}}$. (b) They must reliably trigger multi-page browsing, rather than being satisfied by a single source

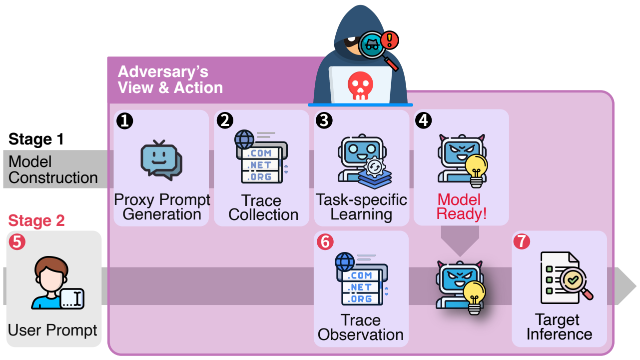

Figure 3: **Two-stage attack pipeline.**

(e.g., Wikipedia) or by the agent's prior knowledge; to enforce this, we prepend instructions requiring at least five distinct page visits. (c) Prompts must be sufficiently detailed to produce distinguishable traces; underspecified inputs are rewritten into instruction-complete forms to avoid trivial queries that yield little signal. (d) Prompts and resulting traces remain semantically linked, so the trace encodes sufficient signal for recovering p or inferring τ.

Prompt reconstruction. We draw proxy prompts from public query datasets (FEDWEB13, SESSION14, DD16) to build $D_{\text{PR}} = \{(WRA(p_j), p_j)\}_j$. Some prompts are underspecified to reliably induce meaningful browsing, and agents differ in how they handle such inputs. GPT-Researcher [25] internally refines vague prompts, but ChatGPT Deep Research UI asks clarifying questions, while Deep Research API, AutoGen, and Browser-Use execute inputs as-is. To align behaviors, Deep Research API prompts are rewritten into detailed variants (denoted -DR) to match the interactive Deep Research interface.

Trait inference. From persona profiles, we construct $D_{\text{TI}} = \{(WRA(\{p_i^{(j)}\}_{i=1}^N), \tau_j)\}_j$, embedding selected traits (e.g., age, citizenship, insurance coverage) either explicitly or implicitly into prompts. This mirrors real user behavior, where personal context guides information seeking (e.g., health questions tied to age or history). A persistent adversary linking sessions over time (e.g., by IP) would observe recurring patterns tied to these traits (τ_j). Executing such prompts through agents produces labeled traces that reveal stable user attributes, enabling training of trait inference models (M_{TI}). Dataset-specific sampling and generation details are described in Section 6.1.

4.2 Domain Trace Collection

To understand how different prompts yield distinct browsing behavior, we collect domain-level metadata from agents via Playwright logs or built-in system logs, yielding a trace $t = \langle (d_k, u_k, s_k) \rangle_{k=1}^L$ for each prompt p, where (d_k, u_k, s_k) capture the ordered sequence of accessed domains, timestamps (at millisecond-level precision), and payload size in bytes, corresponding to the life cycle of a single user prompt. This captures whether the agent revisits the same domains or explores diverse sources, the features that adversaries could exploit. Although TLS application data does not expose full URLs or domain names directly, domain attribution remains feasible through DNS queries, SNI fields in TLS handshakes, or correlated DNS/HTTPS timing, all of which are observable to ISPs or other on-path entities. This matches realistic adversarial capabilities today, though we note that CDN consolidation or future deployment of DoH/DoT/ECH could reduce visibility.

Noise Filtering. A passive adversary observes all outbound requests, including primary domains, embedded resources, ads, and trackers. Because much of this traffic is incidental rather than agent-driven, an effective adversary would attempt to filter noise to isolate the domains most indicative of user intent. We apply two simple heuristics for the noise filtering: removing well-known advertising and analytics services (e.g., `doubleclick.net`, `onetrust.com`, `taboola.com`), and discarding packets smaller than 7KB, which in pilot runs rarely corresponded to substantive browsing activity. These steps represent the simplest plausible filtering approach; more advanced adversaries could employ refined techniques (e.g., traffic classification or resource-type analysis) to separate incidental requests from meaningful domains. We leave such stronger filtering strategies to future work.

4.3 Inference Model Construction and Use

The adversary builds an inference model from proxy datasets, then applies it to new traces simulating real-user activity.

For **prompt reconstruction**, the task is to recover a natural-language prompt that captures the functional intent of the user's original input. The adversary constructs an inference model (M_{PR}) from proxy (trace, prompt) pairs in the proxy dataset (D_{PR}), using either ICL, where a small set of examples is included in the inference prompt without parameter updates, or fine-tuning, where the same proxy pairs are used to train a direct mapping from traces to prompts. Given a new user prompt p producing trace $t = WRA(p)$ (optionally filtered to $\tilde{t}$ per Section 4.2), the resulting model M_{PR} is then applied to new traces to reconstruct real user prompts $\hat{p} = M_{\text{PR}}(\tilde{t})$.

For **trait inference**, the goal is to infer latent user attributes ($\hat{\tau}$) from browsing traces ($\tilde{t}^{1:N}$). Here, the proxy dataset (D_{TI}) is generated, consists of proxy (trace, traits) pairs for each annotated persona profile, and the adversary adopts ICL to construct M_{TI}, since it outperformed fine-tuning in the prompt reconstruction task. For a new user with sessions $\{p_i\}_{i=1}^N$ and trace $t^{1:N} = WRA(\{p_i\}_{i=1}^N)$, the adversary output $\hat{\tau} = M_{\text{TI}}(\tilde{t}^{1:N})$. This enables an adversary to recover persistent attributes such as demographics, ideology, or lifestyle from observed browsing activity.

5 OBELS: Ontology-aware Behavioral Leakage Score

From the adversary's perspective, a reconstructed prompt poses a significant privacy risk if it induces the same browsing behavior as the user's original input. Even when the textual form differs, the sensitive intent is revealed through replicated domain visits and task flows. Existing metrics focus on textual resemblance and therefore miss such cases. We introduce **OBELS (Ontology-Aware Behavioral Leakage Scores)**, a multi-metric scoring scheme that decomposes leakage into dimensions of preserved functionality, domain targets, and semantic entities, providing a structured alignment with attacker goals and real-world agent behavior.

5.1 Limitations of Existing Metrics

In AI agent contexts, leakage risk stems from *functional equivalence*: a reconstruction is risky if it causes the agent to visit similar sites, access comparable services, or pursue equivalent goals—regardless of its phrasing. Existing metrics fall short because they evaluate what prompts *say*, not what they *do*.

SBERT score, computed as cosine similarity between sentence embeddings [62], captures high-level semantic alignment but suffers from paraphrastic inflation. It misleadingly assigns high scores to verbose, generic, or loosely related prompts [70, 84], overlooking task-specific behavior or intent.

LLM-as-Judge methods leverage LLMs (e.g., GPT-4) to rate similarity through natural language instructions. While more tolerant to paraphrase and flexible, they are prompt-sensitive and often lack interpretability [40, 61], producing scalar scores without indicating which semantic elements were preserved or lost.

These shortcomings are evident in practice. For example, rephrasing "Compare Italy and Spain's digital nomad visas" as "How do work visas differ across southern Europe?" yields low cosine similarity, yet the agent navigates to similar government portals and retrieves overlapping content. From a privacy standpoint, this is a successful reconstruction—yet traditional metrics fail to capture it.

5.2 Semantic Triplet Scoring with OBELS

To address these gaps, we propose **OBELS**, a set of four complementary scores designed to capture behavioral alignment. Rather than collapsing leakage into a single scalar, OBELS reports distinct dimensions so that failures in one area (e.g., missing entity granularity) are not obscured by high alignment elsewhere. This decomposition provides interpretable diagnostics and better reflects adversarial risk. It does so by abstracting prompts into structured triplets:

```
(intent, source_type, entity)
```

Here, `intent` denotes the high-level goal (e.g., learn, explore, analyze, compare, summarize, plan, decide, book, watch, read, evaluate); `source_type` identifies the class of information or service sought (e.g., travel, symptom, policy_area, event, visa_process, treatment_method, academic_field, cuisine, etc); and `entity` specifies the core topic (e.g., Rome, Italy, depression, face transplant, immigration, PhD, Business, Swahili dish, etc). (Figure 10).

We evaluate each reconstruction across four dimensions:

- **Functional Equivalence (E_{func}):** Do the two prompts express the same high-level intent or task (e.g., finding a product, booking a service)?
- **Domain-Type Equivalence (E_{domain}):** Do they engage with the same category of services or information sources (e.g., travel agencies, health databases)?
- **Semantic Equivalence ($E_{semantic}$):** Are entities semantically aligned (e.g., "side effects of Prozac" vs. "adverse reactions to fluoxetine")?
- **Entity Granularity Tolerance (T_{entity}):** Are differences in specificity (e.g., "Italy" vs. "Rome and Venice") minor enough to preserve meaning?

Each dimension is scored from 0.0 (unrelated) to 1.0 (fully equivalent) using a standardized evaluation template applied to GPT-4, which compares triplet sets holistically and assigns fine-grained similarity scores (see Figure 11).

Unlike conventional metrics, OBELS captures reconstructions that preserve function and downstream behavior, even when their exact wordings diverge. By decomposing leakage into interpretable components, it offers a more transparent measure of privacy risk in prompt reconstruction.

6 Experimental Setup

We evaluate our metadata-based prompt and trait leakage attacks on held-out test prompts. The objective is to assess how well an adversary can reconstruct user prompts that induce the same downstream behavior as the originals, or infer sensitive latent traits from domain traces, thereby capturing both immediate intent leakage and longer-term profiling risks.

6.1 Datasets

We target two leakage scenarios: prompt reconstruction and trait inference. Table 1 summarizes all datasets; full details are in Appendix A.1.

Prompt Reconstruction (TREC). We use three TREC datasets: FedWeb 2013 (50 topics) [53], Session 2014 (60) [2], and Dynamic Domain 2016 (53) [3], referred to as FEDWEB13, SESSION14, and DD16. For FEDWEB13 and DD16, we concatenate the `<description>` and `<narrative>` fields to match SESSION14's instruction-style and prompt specificity. To align with OpenAI's Deep Research API [56], we also generate **-DR variants** by rewriting

Table 1: **Traces of Web and Research Agents collected from different base datasets.**

Base Dataset	Target	# Total	# Test	Agent	Avg. # Dom	Avg. # URLs	Prompts × Sessions
FEDWEB13 [53]	Prompt	50 prompts	20	GPT-Researcher	77.5	143.0	1 prompt × 1 session
SESSION14 [2]	Prompt	60 prompts	20	GPT-Researcher	69.9	145.9	1 prompt × 1 session
DD16 [3]	Prompt	53 prompts	20	GPT-Researcher	64.8	155.8	1 prompt × 1 session
PERSONA [18]	Trait	50 personas (of 997)	3	AutoGen	144.1	173.7	3–5 prompts × 7 sessions = 21–35 queries

each prompt with GPT-4.1, while scoring is always against the original prompts. For each dataset, 20 prompts are reserved for testing, with the remainder used for training and ICL examples.

Trait Inference (PERSONA). We sample 50 profiles from the PERSONA_subset [18] (997 total), each annotated with 32 traits. For each, we select 5 traits and use GPT-4o to generate 3–5 prompts across 7 sessions, simulating a week of browsing. Three personas are held out for ICL, and after filtering to ages 18–70, 35 remain for testing. The prompt template generating queries per user (persona) is provided in Appendix E.

6.2 WRAs and Backbone LLMs

We evaluate domain-trace leakage in settings where a passive adversary can observe network-level metadata (IPs, domains, timing, payload size). To capture the traces, we use three open-source WRAs and one proprietary API agent (details in Appendix A.2):

GPT Researcher [25]. An open-source autonomous research agent for deep multi-hop investigation. We run it in `deep` mode, which produces structured research traces. GPT Researcher strategically uses multiple LLM backbones, and we evaluate both GPT-family and open-source backbones for the prompt reconstruction task.

Browser-Use [47]. An open-source web agent that automates browsing through a local browser. To ensure stable traces, we configure it to start from Bing and require at least five distinct page visits before summarization. Browser-Use is used for prompt reconstruction with GPT-4o as the backbone.

AutoGen [81]. An open-source web agent system we used for both prompt reconstruction (GPT-4o) and trait inference (Gemini 2.0 Flash), instrumented to log domains and metadata. Unlike the other agents, it produces the real packet-level traces, noisier with ads and trackers.

OpenAI Deep Research API. A proprietary research agent that serves as a commercial baseline. To fully exercise its capabilities, we provide the **-DR** prompt variant (e.g., SESSION14-DR), designed to approximate the behavior of accessing Deep Research through the ChatGPT interface. While browsing is remote and not locally visible, it serves as a benchmark for high-capability proprietary systems.

Each open-source agent is paired with both open and proprietary LLM backbones to reflect realistic local deployments. OpenAI Deep Research API serves as a proprietary baseline. Larger local LLMs (e.g., LLaMA 3.1 70B, DeepSeek-R1) are excluded because they require substantial resources and often bypass web search by relying on internal knowledge.

6.3 Evaluation Metrics

Metrics for Prompt Reconstruction. We evaluate prompt reconstruction using three metrics: (1) **SBERT**, computed as cosine similarity over mean-pooled Sentence-BERT (SBERT) embeddings [62]; (2) **LLM-as-Judge** (template in Figure 12), where an LLM rates prompt similarity based on natural language instructions; and (3) **OBELS**, our proposed behavior-aware scoring scheme to capture functional equivalence across multiple dimensions (i.e., whether the reconstructed prompt would induce similar agent behavior).

Metrics for Trait Inference. For trait inference, we adopt a type-aware scoring strategy tailored to each trait's structure:

- *Numeric traits* are evaluated using a normalized absolute difference: smaller relative errors yield higher scores, linearly decreasing toward zero.
- *Ordinal traits* are mapped to integer levels and scored by normalized distance on a 5-point scale. Closer ratings yield higher similarity scores.
- *Categorical traits* use exact match for single-token values; multi-word categories are scored using the SBERT score to account for semantic equivalence.
- *Free-text traits* are assessed using the SBERT score, which computes semantic overlap via contextual token embeddings.

This trait-type–aware evaluation avoids penalizing structured attributes with embedding-based metrics and ensures robust semantic assessment for open-ended descriptions. In contrast to prompt reconstruction, trait inference does not require alignment in downstream behavior; semantic similarity alone suffices to assess privacy leakage.

7 Attack Results

7.1 Prompt Reconstruction

Attack success is measured not by textual similarity alone, but by semantic and downstream behavior similarity, whether reconstruct prompts preserve the functional intent behind the original (see Section 6.3). The results quantify how much information about a user's input is revealed through observable browsing activity, and how reconstruction performance varies across learning strategies, datasets, agents, and agents' LLM backbones. GPT-4o is used as the inference LLM unless stated otherwise. Appendix A.3 details the ICL setup, and additional results are reported in Appendix B.

Table 2: 5-shot ICL vs. supervised fine-tuning across multiple inference LLMs. ICL outperforms fine-tuning on all metrics, making it the stronger approach for prompt reconstruction.

Models (Method)	SBERT	LLM-Judge	OBELS			
			E_{func}	E_{dom}	E_{sem}	T_{ent}
GPT-4o (ICL)	**0.492**	**0.415**	**0.770**	**0.735**	0.520	0.640
GPT-4o (FT)	0.469	0.340	0.675	0.690	0.485	0.615
GPT-4.1 nano (ICL)	0.430	0.330	0.735	0.700	0.500	0.665
GPT-4.1 nano (FT)	0.369	0.210	0.610	0.605	0.440	0.510
Llama 3.1-8B (ICL)	0.466	0.310	0.695	0.675	0.475	0.630
Llama 3.1-8B (FT)	0.416	0.230	0.705	0.710	0.470	0.525
Qwen 2.5-32B (ICL)	0.489	0.400	0.770	0.72	**0.525**	**0.670**
Qwen 2.5-32B (FT)	0.404	0.225	0.595	0.640	0.410	0.530
Gemma 3-4B (ICL)	0.429	0.280	0.675	0.670	0.475	0.635
Gemma 3-4B (FT)	0.335	0.175	0.510	0.485	0.355	0.420

ICL vs. Fine-Tuning Across Models. To motivate our choice of ICL over fine-tuning as the primary task-specific approach, we compare their performance across closed-source models (GPT-4o and GPT-4.1 nano) and open-source models (Llama 3.1 8B, Gemma 3 4B, and Qwen 2.5 32B) (Table 2). We create our final training set by combining the training samples from the prompt reconstruction datasets discussed in Section 6.1. We test ICL by utilizing the setup in Appendix A.3. For closed-source models, fine-tuning is performed through OpenAI's fine-tuning API [1] for 3 epochs with batch size 1. For open-source models, we apply LoRA [33] with rank 16 and $\alpha = 12$. Results show that ICL generally outperforms fine-tuning across nearly all metrics, though models like Llama occasionally favor fine-tuning for functional and domain equivalence. We attribute this to limited sample diversity, which likely causes overfitting in fine-tuned models.

Impact of Example Count. Table 3 compares performance across different numbers of in-context examples. 5-shot ICL provides a balanced trade-off between functional equivalence, domain alignment, and semantic similarity, while avoiding the performance drop observed at smaller or greater counts.

Impact of Agents. As described in Sections 6.1 and 6.2, we collect traces from GPT-Researcher, AutoGen, and Browser-Use (20 prompts, 5-shot ICL) and from Deep Research (5 prompts, 3-shot) using SESSION14 prompts. Table 4 shows that GPT-Researcher traces leak the most, consistent with its broader domain coverage and richer source diversity. Deep Research exhibits comparable leakage despite fewer prompts and shots, while AutoGen and Browser-Use generate leaner traces that result in weaker reconstruction. These differences highlight how agent design and exploration strategy directly shape the degree of leakage.

Table 3: Prompt reconstruction performance for different numbers of ICL examples. 5-shot offers the best overall balance across evaluation metrics.

No. of Examples	SBERT	LLM-Judge	OBELS			
			E_{func}	E_{dom}	E_{sem}	T_{ent}
0-shot	0.395	0.295	0.665	0.630	0.465	0.615
5-shot	**0.492**	**0.415**	**0.770**	**0.735**	0.520	0.640
8-shot	0.490	0.400	0.760	0.730	**0.540**	0.610
12-shot	0.504	0.405	0.750	0.710	0.530	0.640
15-shot	0.487	0.385	0.770	0.685	0.525	**0.645**

Table 4: Prompt reconstruction on SESSION14. All agents are evaluated on 20 prompts with 5-shot ICL, except Deep Research* (5 prompts, 3-shot). GPT-Researcher traces leak the most, while Deep Research shows comparable leakage despite fewer shots.

Agent	SBERT	LLM-Judge	OBELS			
			E_{func}	E_{dom}	E_{sem}	T_{ent}
AutoGen	0.356	0.225	0.565	0.625	0.400	0.495
Browser-Use	0.368	0.240	0.540	0.590	0.390	0.485
GPT-Researcher	0.492	**0.415**	**0.770**	**0.735**	**0.520**	**0.640**
Deep Research*	**0.574**	0.360	0.680	0.720	0.480	0.640

Table 5: Reconstruction performance of GPT-Researcher traces across datasets, comparing GPT-4 and local LLM backbones. GPT-4 traces consistently yield higher scores, while local backbones leak less but remain informative.

Model	Dataset	SBERT	LLM-Judge	OBELS			
				E_{func}	E_{dom}	E_{sem}	T_{ent}
GPT4	SESSION	0.492	0.415	0.770	0.735	0.520	0.640
Local	SESSION	0.512	0.355	0.735	0.745	0.520	0.630
GPT4	FEDWEB	0.505	0.305	0.755	0.735	0.540	0.695
Local	FEDWEB	0.493	0.310	0.750	0.745	0.515	0.645
GPT4	DD	0.453	0.190	0.585	0.615	0.430	0.545
Local	DD	0.445	0.180	0.550	0.635	0.430	0.515

Impact of Agent Backbone: Proprietary vs. Open-Source. Using GPT-Researcher, we evaluated traces across datasets with either GPT-4 or local open-source backbones (DeepSeek-V3, Mistral family). As shown in Table 5, GPT-4 traces yield higher reconstruction scores, reflecting stronger reasoning capacity and broader exploration. Local models leak less overall, but still provide exploitable cues. For the remainder of our ICL experiments, we use GPT-Researcher with a GPT-4 backbone on SESSION14, as this setup offers both representativeness and clearer leakage signals for analysis.

Table 6: Prompt reconstruction with different ICL inference models. Proprietary models (Claude-opus-4-1, Gemini-2.5-pro) yield the highest scores, while GPT-5, GPT-5-mini, and GPT-4o remain competitive.

Models	SBERT	LLM-Judge	OBELS			
			E_{func}	E_{dom}	E_{sem}	T_{ent}
Claude-opus-4-1	**0.544**	**0.445**	0.765	**0.785**	**0.570**	**0.670**
Gemini-2.5-pro	0.506	0.420	0.710	0.705	0.505	0.645
GPT-5	0.470	0.240	0.710	0.715	0.535	0.670
GPT-5-mini	0.479	0.315	0.710	0.685	0.500	0.635
GPT-4o	0.492	0.415	**0.770**	0.735	0.520	0.640
GPT-4.1 nano	0.430	0.330	0.735	0.700	0.500	0.665
Llama 3.1 8B	0.466	0.310	0.695	0.675	0.475	0.630
Qwen 2.5-32B	0.489	0.400	0.770	0.72	0.525	0.670
Gemma 3-4B	0.429	0.280	0.675	0.670	0.475	0.635

Impact of Inference LLM. Table 6 compares inference engines for ICL-based prompt reconstruction, showing that model choice strongly affects reconstruction quality. Claude-opus-4-1 and Gemini-2.5-pro achieve the highest scores, while GPT-5, GPT-5-mini, and GPT-4o remain competitive. Larger models leak more information but incur higher cost and latency, especially in the Gemini, Claude, and GPT-5 families. Balancing accuracy, cost, and latency, we identify GPT-4o as the most practical option, offering strong reconstruction performance with lower overhead than top-scoring but slower systems.

7.2 Trait Inference

We assess how much and accurately an adversary can recover latent attributes of the user from a week-long browsing trace (see Section 6.3). The results quantify trait exposure using *inference accuracy* (similarity score), where higher values mean more reliable adversarial inference. Gemini 2.5 pro is used as the inference LLM.

Figure 4 ranks the fifteen most exposed traits, showing that exposure varies by attribute. Health insurance (0.98), veteran status (0.90), employment status (0.88), and household language (0.86) are inferred with near-perfect accuracy, with employment status nearly as reliable as binary traits despite being multi-class. A second tier (marital status, sex, race, religion, household type) scores 0.73–0.77, while political views, citizenship, education, family presence/age, place of birth, and income range from 0.66–0.70, still well above random. Behavioral and psychographic traits do not appear in the top 15, consistent with their lower exposure in Figure 6. Overall, the results suggest that demographic and occupational traits are most vulnerable to leakage, while lifestyle and personality attributes remain less exposed but at nontrivial levels.

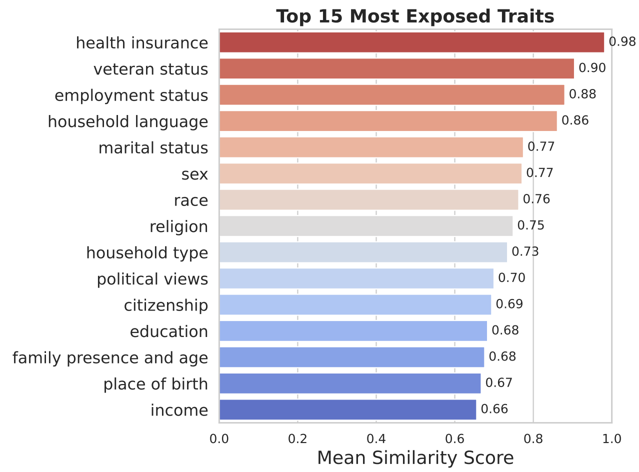

Figure 4: Top 15 traits with the highest exposure risk, measured by mean similarity scores across all personas and sessions. Higher scores indicate stronger inference.

Figure 5 compares similarity scores for selected and unselected traits. Selected traits typically exceed 0.7, with some personas (012, 024, 037, 045) surpassing 0.8, indicating consistent high-risk leakage [14]. Unselected traits generally fall below but often remain above 0.5, showing moderate exposure even when traits are not explicitly embedded. In a few cases (007, 025, 041), unselected traits approach or exceed selected ones, confirming leakage extends beyond explicitly mentioned attributes.

Figure 6 compares model confidence and accuracy across categories. As shown in the left plot, occupational and demographic traits are assigned very high confidence (0.8–1.0), while the other two peak in the mid-confidence range (0.6–0.8). The accuracy distributions (right plot) further distinguish the categories: occupational traits average 0.70 but with a lower median (0.53), psychographic traits are inferred with moderate accuracy (mean 0.56, median 0.46) but show extreme variability (spread 0–1). Behavioral traits are uniformly moderate to weak (mean 0.46, median 0.45) with a narrowest spread of 0.25 to 0.61, and demographic traits show the highest median (0.90), but occasional errors lower the mean (0.44). Trait categories differ not only in accuracy but also in distribution: demographic traits are most reliably inferred, occupational traits are less consistent but often highly accurate, psychographic traits are highly variable, and behavioral traits are the least exposed.

Different Number of Sessions. As part of our ablation study, Figure 7 shows the distribution of traits per persona with similarity scores above 0.7 under three vs. seven sessions. With three sessions (yellow), most personas reveal 6–10 traits with a long tail of higher counts. With seven sessions (purple), the

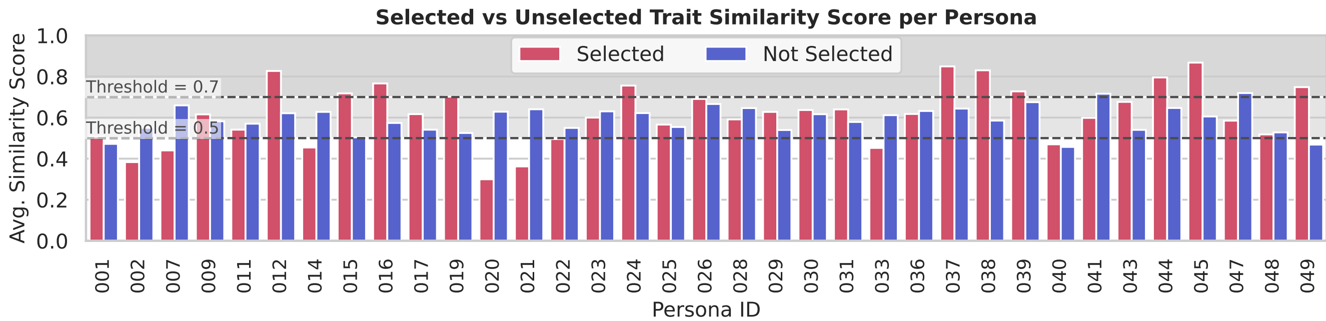

Figure 5: Average similarity scores per persona for selected (red) and unselected (blue) traits. Dashed lines at 0.5 and 0.7 indicate moderate and high-risk leakage thresholds. Many personas have selected traits exceeding 0.7, while unselected traits frequently remain above 0.5, reflecting moderate but non-negligible exposure.

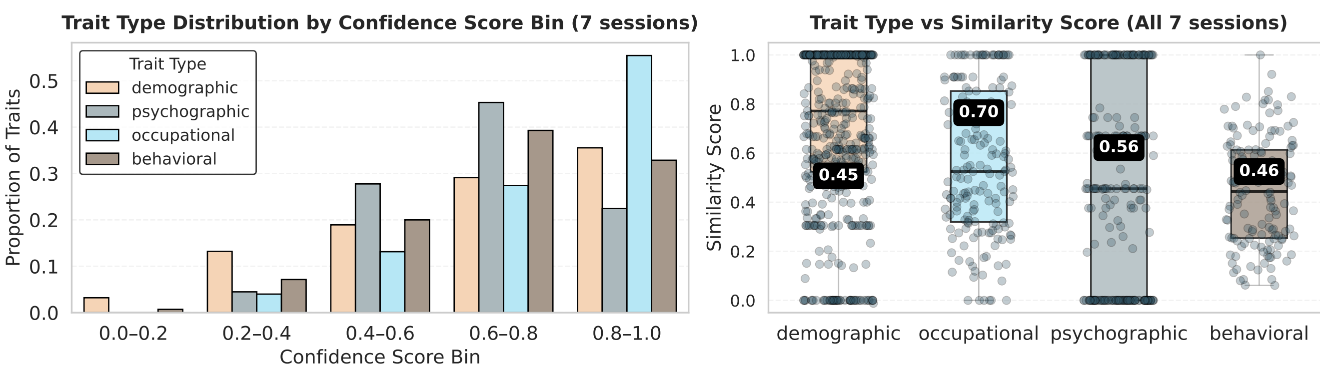

Figure 6: **Left:** Proportion of inferred traits in each confidence bin. Occupational and demographic traits are most often assigned high confidence (0.8–1.0), while the other two peak in middle ranges (0.6–0.8). **Right:** Similarity scores by trait category. Boxes show the interquartile range, black lines, squared numbers, and dots represent medians, means, and all data points, respectively.

distribution shifts rightward and concentrates around 12–14 traits, with some reaching 19. This demonstrates that additional sessions, and thus more domain traces, enable inference of more traits at high accuracy.

Table 7 confirms consistent gains across categories, with psychographic (+16.7%) and demographic (+15.4%) improving most, followed by behavioral (+12.2%) and occupational (+7.4%) traits. Overall, longer observation windows strengthen inference, increasing both the number of high-similarity traits and average accuracy.

Table 7: Average similarity scores grouped by trait category. Δ shows the improvement of full sessions over three sessions.

Trait Category	3 Sessions	7 Sessions	Δ (%)
Demographic	0.39	0.45	+15.4%
Occupational	0.68	0.73	+7.4%
Psychographic	0.48	0.56	+16.7%
Behavioral	0.41	0.46	+12.2%

Overall, our results show that metadata leakage enables both immediate and long-term risks. For prompt reconstruction, prompts can be reliably recovered regardless of whether traces are produced by proprietary or local LLM backbones, or by agents with varying exploration strategies. For trait inference, exposure levels differ across categories, but even traits not explicitly embedded in prompts can still be inferred at moderate to high accuracy, as shown by the recovery of unselected traits. Together, these findings demonstrate that domain traces alone reveal sensitive user intent and identity signals, underscoring metadata leakage as a significant privacy threat.

8 Defense and Realistic Deployment

Motivation. If a task requires visiting a uniquely identifying domain, the defense should hide that interaction via VPNs or randomized background queries. By contrast, if the information can be obtained without distinctive traces, blocking is preferable; for example, by querying an LLM directly or gathering information from Wikipedia. We thus identify two conceptual strategies for mitigating metadata-based inference attacks: (1) **hiding traces**, where sensitive activity must occur but is camouflaged with plausible noise; and (2) **blocking traces**, where sensitive activity is avoided by relying on

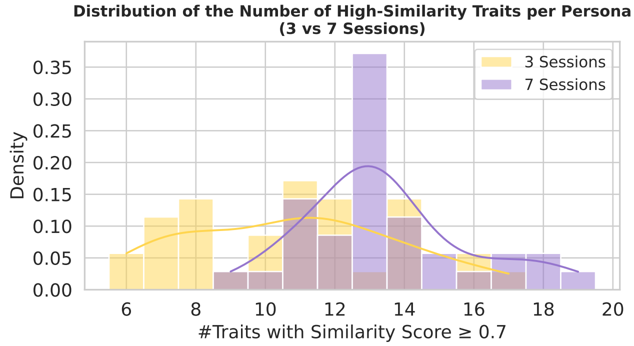

Figure 7: Distribution of the number of traits per persona with similarity scores above 0.7 under 3- and 7-session settings. The 7-session distribution (purple) shifts rightward and is more concentrated, showing that additional sessions allow inference of more traits at high similarity.

alternative, less revealing sources.

8.1 Hiding via Decoy Prompts

Inspired by TrackMeNot [52], for each real prompt the system generates and executes additional decoy prompts that remain within the same topical space but differ in framing, intent, or implied traits. Because OBELS evaluates whether reconstructions preserve intent, domain type, and entity, breaking this alignment is critical; otherwise, an adversary can still recover the user's functional behavior despite wording changes. For instance, a prompt about "recognizing depression and seeking medical help for a friend" can be paired with a decoy like "compare the effectiveness of yoga versus meditation for improving sleep quality", which shifts both domain and entity, lowering OBELS score and obscuring the true goal. This ambiguity in observed traffic undermines an adversary's ability to infer the user's true objective, attributes, or target entity. We explore a lightweight approach using LLM-based decoy generation and a trait-aware variant that leverages virtual personas to introduce long-term behavioral divergence.

Mechanism. The defense operates in three stages. (1) **Decoy prompt generation**: an LLM produces multiple decoys that remain within the same topical space but differ in context, intent, or entity granularity, thereby obscuring both high-level intent and fine-grained attributes such as geographic focus or institutional target (e.g., a prompt on "Swahili food" may be paired with decoys on "Italian cuisine" or "modern American dishes"). (2) **Trait-conflicting guidance** (optional): decoy generation can be steered by a virtual persona, with the system maintaining a rolling estimate of the user's traits via lightweight keyword heuristics and selecting a persona that diverges on key dimensions (e.g., ideology or religion). These divergent decoys introduce consistent but misleading patterns, weakening long-term profiling. (3) **Concurrent execution**: real and decoy prompts are issued in parallel; while only the real output is returned to the user, all queries leave observable traces, blending plausible yet misleading activity into the traffic and complicating adversarial inference of the user's true intent or identity.

Table 8: **Effect of decoy prompts and domain shuffling on prompt reconstruction attacks.**

Defense	SBERT	LLM-Judge	OBELS			
			E_{func}	E_{dom}	E_{sem}	T_{ent}
Without Defense	0.492	0.415	0.770	0.735	0.520	0.640
1 Decoy	0.390	0.190	0.610	0.615	0.420	0.545
3 Decoys	0.376	0.210	0.645	0.615	0.420	0.555
5 Decoys	0.368	0.175	0.565	0.590	0.395	0.500
1 Decoys + shuffle	**0.455**	**0.285**	**0.725**	**0.715**	**0.510**	**0.610**
3 Decoys + shuffle	0.415	0.235	0.685	0.645	0.455	**0.610**
5 Decoys + shuffle	0.403	0.220	0.650	0.695	0.465	0.570

Impact of Defense on Prompt Reconstruction. Table 8 summarizes the reconstruction scores under defense. The defense inserts a varying number of decoy prompts, each contributing its own set of visited domains. Original domains are randomly interleaved within these sets while preserving their internal order, as shown in rows 2–4 of the table. To model a stronger defender, we also consider a condition where all domains, both original and decoy, are fully shuffled (rows 5–7).

The results show a consistent reduction in inference accuracy once decoys are added. With only one decoy prompt, SBERT drops by 0.10 and LLM-Judge by 0.23 relative to the baseline. More decoys generally strengthen the defense, with the largest effect observed at five decoys. When domains are shuffled, attack performance recovers somewhat for semantic and domain-level metrics, but still remains below baseline.

Table 9: **Median inference accuracy across trait categories.**

Trait Category	No Defense	w. Defense	Δ (%)
Demographic	0.4440	0.4080	−8.1%
Occupational	0.8975	0.6790	−24.3%
Psychographic	0.5250	0.5030	−4.2%
Behavioral	0.4550	0.4550	0.0%

Impact of Defense on Trait Inference. We evaluate the defense by re-running the trait inference attack with decoy prompts injected. In this setting, each real prompt triggers a concurrent decoy prompt from a conflicting virtual persona, selected based on the defender-side trait estimate. Table 9 reports median similarity scores between inferred and ground-truth traits across four categories. The defense yields the largest median reduction for occupational traits (−24.3%), followed by demographic traits (−8.1%). Psychographic traits drop slightly (−4.2%), while behavioral traits remain unchanged. Additional results are provided in Appendix C.

8.2 Blocking via Alternative Sources

A complementary defense is to prevent sensitive traces from arising in the first place. When a user's objective can be met without contacting uniquely identifying domains, the agent can instead draw from large multipurpose repositories (e.g., Wikipedia, StackExchange, Reddit) or from the LLM's internal knowledge. By retrieving information from resources that serve diverse intents or by leveraging the model's internal knowledge, the resulting traffic no longer maps cleanly to a specific user goal. This "blocking" approach avoids generating distinctive traces altogether, reducing the adversary's ability to associate domain visits with private attributes.

Table 10: Prompt reconstruction under different trace visibility. URL-level traces leak the most, while domain-only or partial traces reduce leakage. Adding timing metadata shows no consistent benefit.

Trace Visibility	SBERT	LLM-Judge	OBELS			
			E_{func}	E_{dom}	E_{sem}	T_{ent}
Domains	0.492	0.415	0.770	0.735	0.520	0.640
Domains + Timing	0.486	0.455	0.725	0.695	0.525	0.640
URLs (100%)	**0.624**	**0.600**	0.810	**0.805**	0.620	**0.695**
Partial URLs (80%)	0.618	0.590	0.795	0.795	**0.635**	0.680
Partial URLs (60%)	0.622	0.590	**0.820**	0.755	0.585	0.690

Table 11: Leakage vs. utility under varying domain visibility. Reducing visibility lowers prompt reconstruction accuracy but leaves the quality of generated reports nearly unchanged.

Visibility	Utility	SBERT	LLM-Judge	OBELS			
				E_{func}	E_{dom}	E_{sem}	T_{ent}
100%	8.55	0.492	0.415	0.770	0.735	0.520	0.640
80%	8.72	0.491	0.415	0.750	0.725	0.515	0.645
60%	8.54	0.489	0.355	0.765	0.730	0.520	0.650
40%	8.50	0.453	0.350	0.765	0.700	0.515	0.650
20%	8.60	0.438	0.300	0.725	0.695	0.475	0.615
10%	8.54	0.375	0.225	0.610	0.555	0.395	0.485
5%	8.32	0.319	0.185	0.530	0.455	0.335	0.395
2%	7.65	0.238	0.095	0.345	0.410	0.275	0.315

Table 10 and Table 11 together show that visibility of browsing traces strongly shapes leakage but has little effect on utility. Utility is how much the WRA's output helps a human reader reach their research goals. We measure it with an LLM using the prompt template in Figure 17. URL-level traces leak substantially more than domain-only traces, making prompt inference easier and more accurate. Reducing visibility (e.g., observing only 80% or 60% of domains) lowers leakage but still exposes useful information. Adding timing metadata to domains yields no consistent gains, suggesting that domain identity is more informative than temporal order. In contrast, the utility of the generated reports remains relatively stable across visibility levels down to 10%. However, beyond this threshold, utility begins to drop, revealing diminishing returns from additional domain coverage. This suggests that *many extra domains contribute little to report quality while disproportionately increasing exposure to adversaries*. Thus, WRA behaviors that aggressively visit many domains are especially problematic: they create substantial leakage risks while providing only marginal gains in utility.

9 Discussion and Conclusion

Privacy and Security Implications. Our results show that metadata leakage is an inherent risk in WRA ecosystems. Agent design choices—including domain exploration breadth, backbone model capacity, and orchestration logic—directly affect the amount of information exposed. High-capacity backbones produce more comprehensive outputs but leave richer traces. Substituting locally deployable models reduces but does not eliminate leakage: domain-level traffic is structurally unavoidable. This shifts the attack surface from *content* to *behavior*. Whereas prompt injection and output filtering can be mitigated through isolation, network-level traces cannot be trivially suppressed. At scale, such leakage enables profiling by advertisers, ISPs, or state actors, raising urgent questions about how to balance agent utility with user privacy. By showing that leakage occurs from metadata rather than content, our study exposes a new attack surface largely overlooked in prior LLM and agent security work.

Defense Effectiveness and Limitations. Our study has several limitations. We rely on proxy datasets and synthetic personas with limited session counts; longer-term aggregation or multimodal attacks may amplify risks. Defense evaluation is confined to decoys and blocking, and cost/latency tradeoffs are only partially measured. Nonetheless, our results show that decoy prompts and trait-conflicting personas reduce both prompt reconstruction and trait inference accuracy, while blocking strategies help when tasks can rely on broad, non-unique sources (e.g., Wikipedia) without sacrificing utility. Yet defenses remain partial: adversaries still recover useful signals, and decoys introduce overhead in latency, bandwidth, and behavioral realism. As with prior obfuscation systems, mitigation is meaningful but incomplete; full network-level protections (e.g., VPNs or anonymity systems) remain the only robust safeguard.

Future Directions and Open Challenges. Future work should explore hybrid defenses that combine obfuscation, blocking, and timing perturbations; study long-term aggregation attacks across extended user histories; and integrate formal privacy guarantees such as differential privacy or traffic-analysis resistance directly into agent design. Multimodal agents introduce additional risks, for example, screenshot-based profiling that bypasses text-based defenses, which merit dedicated mitigation. Overall, progress requires viewing AI agents and the web as a coupled ecosystem rather than orthogonal components.

Conclusion. Metadata leakage is a first-class privacy risk for WRA ecosystems. Even with defenses, residual exposure persists because traces are structurally generated and behaviorally rich. Our study demonstrates both the feasibility of reconstruction attacks and the incompleteness of current defenses, underscoring the need for systemic, privacy-aware agent design. Until such frameworks are developed, strong network-level protections remain the only reliable safeguard.

A Ethical Considerations

Our research investigates privacy risks in Web and Research Agents (WRAs) by studying metadata-based prompt and trait leakage attacks. The goal is to better understand how browsing traces may expose sensitive information and to encourage the development of more secure and privacy-preserving agents. By identifying concrete risks and suggesting defenses, we aim to make a constructive contribution to AI safety.

The main stakeholders are end-users of AI agents, who risk exposure of personal traits or prompts; developers and providers, who may need to adapt their systems to mitigate these risks; and the broader public, which benefits from safer deployments of agent technologies. Our experiments used only public datasets (FEDWEB13, SESSION14, DD16) and synthetic personas, ensuring that no real user data or personally identifiable information was involved. Also, no live systems were disrupted.

We recognize that the methods presented could, in principle, be misused by malicious actors. To balance reproducibility with responsibility, we release the full code to support transparency and community evaluation, but note that it operates only on public datasets and synthetic personas. This minimizes direct risk of harm while enabling other researchers to validate our findings and develop stronger defenses.

We believe that the benefits of publishing this research, raising awareness of risks, guiding the design of defenses, and informing regulatory discussions, significantly outweigh the potential harms. By conducting the research within strict ethical and legal boundaries, and by emphasizing mitigations, we ensure that the work contributes constructively to the responsible advancement of AI systems.

B Open Science

Implementations, evaluation code, prompts, and the synthetic datasets derived from our study will be made publicly available. Code available at: https://anonymous.4open.science/r/wra-BD06/.

References

[1] Fine-tuning now available for GPT-4o — openai.com. https://openai.com/index/gpt-4o-fine-tuning/. [Accessed 21-08-2025].

[2] TREC 2014 session track data. Dataset files (topics, session→topic mapping, relevance judgments), 2014. Available from the TREC data repository, NIST.

[3] TREC 2016 dynamic domain track data. Dataset files, overview materials, 2016. Available from the TREC data repository, NIST.

[4] Tavily: The web access layer for ai agents. https://www.tavily.com/, 2025. Accessed: 2025-08-22.

[5] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *CCS*, 2014.

[6] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv:2303.08774*, 2023.

[7] AMD. GAIA: Generative AI is awesome (open-source project). https://github.com/amd/gaia, 2025. Accessed: 2025-08-23.

[8] Mika D Ayenson, Dietrich James Wambach, Ashkan Soltani, Nathan Good, and Chris Jay Hoofnagle. Flash cookies and privacy ii: Now with HTML5 and etag respawning. *Available at SSRN 1898390*, 2011.

[9] Daniel Ayzenshteyn, Roy Weiss, and Yisroel Mirsky. Cloak, honey, trap: Proactive defenses against LLM agents. In *USENIX Security*, 2025.

[10] Danny Azucar, Davide Marengo, and Michele Settanni. Predicting the big 5 personality traits from digital footprints on social media: A meta-analysis. *Personality and individual differences*, 124:150–159, 2018.

[11] Eugene Bagdasarian, Ren Yi, Sahra Ghalebikesabi, Peter Kairouz, Marco Gruteser, Sewoong Oh, Borja Balle, and Daniel Ramage. AirGapAgent: Protecting privacy-conscious conversational agents. In *CCS*, 2024.

[12] Sarah Bird, Ilana Segall, and Martin Lopatka. Replication: whywe still can't browse in peace: on the uniqueness and reidentifiability of web browsing histories. In *SOUPS*, 2020.

[13] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a distance: Website fingerprinting attacks and defenses. In *CCS*, 2012.

[14] Tristan JB Cann, Ben Dennes, Travis Coan, Saffron O'Neill, and Hywel TP Williams. Using semantic similarity to measure the echo of strategic communications. *EPJ Data Science*, 14(1):20, 2025.

[15] Nicholas Carlini and Milad Nasr. Remote timing attacks on efficient language model inference. *arXiv:2410.17175*, 2024.

[16] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security*, 2021.

[17] Ben Carterette, Evangelos Kanoulas, Mark M Hall, and Paul D Clough. Overview of the TREC 2014 session track. In *TREC*, 2014.

[18] Louis Castricato, Nathan Lile, Rafael Rafailov, Jan-Philipp Fränken, and Chelsea Finn. PERSONA: A reproducible testbed for pluralistic alignment. In *COLING*, 2025.

[19] David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[20] Sijia Chen, Yibo Wang, Yi-Feng Wu, Qingguo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Lijun Zhang. Advancing tool-augmented large language models: Integrating insights from errors in inference trees. *NeurIPS*, 2024.

[21] Jeffrey Yang Fan Chiang, Seungjae Lee, Jia-Bin Huang, Furong Huang, and Yizheng Chen. Why are web AI agents more vulnerable than standalone llms? a security analysis. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

[22] Kyle Crichton, Lorrie Faith Cranor, and Nicolas Christin. Rethinking fingerprinting: An assessment of behavior-based methods at scale and implications for web tracking. *Proceedings on Privacy Enhancing Technologies*, 2025.

[23] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *SSYM*, 2004.

[24] Santiago (Sal) Díaz, Christoph Kern, and Kara Olive. Google's approach for secure ai agents. Technical report, 2025.

[25] Assaf Elovic. gpt-researcher. Code repository: https://github.com/assafelovic/gpt-researcher.

[26] Xiang Gao and Kamalika Das. Customizing language model responses with contrastive in-context learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2024.

[27] Google. Gemini Deep Research: Your personal AI research assistant. https://gemini.google/overview/deep-research/, 2025. Accessed: 2025-07-16.

[28] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *ICML*, 2020.

[29] Jamie Hayes and George Danezis. k-fingerprinting: A robust scalable website fingerprinting technique. In *USENIX Security*, 2016.

[30] Andrew Hintz. Fingerprinting websites using traffic analysis. In *International workshop on privacy enhancing technologies*, pages 171–178. Springer, 2002.

[31] Nguyen Phong Hoang, Arian Akhavan Niaki, Nikita Borisov, Phillipa Gill, and Michalis Polychronakis. Assessing the privacy benefits of domain name encryption. In *AsiaCCS*, 2020.

[32] Dirk Hovy. Demographic factors improve classification performance. In *ACL-IJCNLP*, 2015.

[33] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. *ICLR*, 2022.

[34] Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, et al. Deep research agents: A systematic examination and roadmap. *arXiv:2506.18096*, 2025.

[35] Bo Hui, Haolin Yuan, Neil Gong, Philippe Burlina, and Yinzhi Cao. Pleak: Prompt leaking attacks against large language model applications. In *CCS*, 2024.

[36] Glory Jain and Kevin Alwell. Introduction to deep research in the openai api. https://cookbook.openai.com/examples/deep_research_api/introduction_to_deep_research_api, 2025. OpenAI Cookbook, accessed: 2025-06-25.

[37] Marc Juarez, Mohsen Imani, Mike Perry, Claudia Diaz, and Matthew Wright. Toward an efficient website fingerprinting defense. In *European Symposium on Research in Computer Security*, pages 27–46. Springer, 2016.

[38] Hanna Kim, Minkyoo Song, Seung Ho Na, Seungwon Shin, and Kimin Lee. When llms go online: The

emerging threat of web-enabled llms. *arXiv preprint arXiv:2410.14569*, 2024.

[39] LangChain AI. Local Deep Researcher: A fully local web research assistant. https://github.com/langchain-ai/local-deep-researcher, 2025. Accessed: 2025-07-16.

[40] Yoonjoo Lee, Kihoon Son, Tae Soo Kim, Jisu Kim, John Joon Young Chung, Eytan Adar, and Juho Kim. One vs. many: Comprehending accurate information from multiple erroneous and inconsistent ai generations. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 2518–2531, 2024.

[41] Ada Lerner, Anna Kornfeld Simpson, Tadayoshi Kohno, and Franziska Roesner. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In *25th USENIX Security Symposium (USENIX Security 16)*, 2016.

[42] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[43] Yupei Liu, Yuqi Jia, Jinyuan Jia, and Neil Zhenqiang Gong. Evaluating large language model based personal information extraction and countermeasures. In *USENIX Security Symposium*, 2025.

[44] Liv McMahon. Hundreds of thousands of grok chats exposed in google results. *BBC News*, August 21 2025.

[45] Mistral AI. Le chat dives deep. Mistral AI blog, Jul 2025. Announcing "Le Chat," a powerful deep reasoning model by Mistral AI.

[46] Mykyta Mudryi, Markiyan Chaklosh, and Grzegorz Wójcik. The hidden dangers of browsing ai agents. *Available at SSRN 5290351*.

[47] Magnus Müller and Gregor Žunič. Browser use: Enable ai to control your browser, 2024.

[48] Farid M Naini, Jayakrishnan Unnikrishnan, Patrick Thiran, and Martin Vetterli. Where you are is who you are: User identification by matching statistics. *IEEE Transactions on Information Forensics and Security*, 11(2):358–372, 2015.

[49] Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. On the feasibility of internet-scale author identification. In *2012 IEEE Symposium on Security and Privacy*, pages 300–314. IEEE, 2012.

[50] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *S*&P, 2008.

[51] Nick Nikiforakis, Alexandros Kapravelos, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *2013 IEEE Symposium on Security and Privacy*, pages 541–555. IEEE, 2013.

[52] Helen Nissenbaum and Howe Daniel. Trackmenot: Resisting surveillance in web search. *Lessons from the Identity Trail*, 2009.

[53] NIST. TREC 2013 federated web track: 50 topics (fedweb13). XML file 'fedweb13_50topics.xml', 2013. Available from the TREC data repository, National Institute of Standards and Technology (NIST).

[54] Marcos Oliveira, Junran Yang, Daniel Griffiths, Denis Bonnay, and Juhi Kulshrestha. Browsing behavior exposes identities on the web. *arXiv preprint arXiv:2312.15489*, 2023.

[55] Ollama Team. Ollama: Run large language models locally. https://ollama.com, 2025. Accessed: 2025-08-23.

[56] OpenAI. Introducing Deep Research. https://openai.com/index/introducing-deep-research/, February 2025. Accessed: 2025-07-16.

[57] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, pages 103–114, 2011.

[58] Dario Pasquini, Evgenios M Kornaropoulos, and Giuseppe Ateniese. LLMmap: Fingerprinting for large language models. In *USENIX Security*, 2025.

[59] Perplexity AI. Introducing *Deep Research*. Perplexity AI blog, Feb 2025. "Deep Research performs dozens of searches, reads hundreds of sources, and delivers comprehensive reports".

[60] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *The Twelfth International Conference on Learning Representations*.

[61] Amirhossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. Benchmarking prompt sensitivity in large language models. In *ECIR*, 2025.

[62] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. ACL, 2019.

[63] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *NeurIPS*, 2023.

[64] Samuel Schmidgall and Michael Moor. AgentRxiv: Towards collaborative autonomous research. *arXiv:2503.18102*, 2025.

[65] Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using LLM agents as research assistants. *arXiv:2501.04227*, 2025.

[66] H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791, 2013.

[67] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. HuggingGPT: Solving AI tasks with ChatGPT and its friends in Hugging Face. *NeurIPS*, 2023.

[68] Sandra Siby, Marc Juarez, Claudia Diaz, Narseo Vallina-Rodriguez, and Carmela Troncoso. Encrypted dns–> privacy? a traffic analysis perspective. In *Proceedings 2020 Network and Distributed System Security Symposium*. Internet Society, 2020.

[69] Significant-Gravitas. Autogpt: Build, deploy, and run ai agents. https://github.com/Significant-Gravitas/AutoGPT. GitHub repository, accessed August 2025.

[70] Harald Steck, Chaitanya Ekanadham, and Nathan Kallus. Is cosine-similarity of embeddings really about similarity? In *Companion Proceedings of the ACM Web Conference 2024*, 2024.

[71] Chris Stokel-Walker. Exclusive: Google is indexing chatgpt conversations, potentially exposing sensitive user data. *Fast Company*, July 30 2025.

[72] Jessica Su, Ansh Shukla, Sharad Goel, and Arvind Narayanan. De-anonymizing web browsing data with social networks. In *Proceedings of the 26th international conference on world wide web*, pages 1261–1269, 2017.

[73] Satoru Sunahara, Yong Jin, Katsuyoshi Iida, and Yoshiaki Takai. A framework for institutional privacy considered full dns over https architecture. *IEEE Access*, 2025.

[74] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv:2403.05530*, 2024.

[75] Bo Wang, Weiyi He, Pengfei He, Shenglai Zeng, Zhen Xiang, Yue Xing, and Jiliang Tang. Unveiling privacy risks in llm agent memory. 2025.

[76] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models.

[77] Mona Wang, Pellaeon Lin, Jeffrey Knockel, Will Greenberg, Jonathan Mayer, and Prateek Mittal. What WeChat knows: Pervasive first-party tracking in a billion-user super-app ecosystem. *PETS*, 2025.

[78] Jiankun Wei, Abdulrahman Abdulrazzag, Tianchen Zhang, Adel Muursepp, and Gururaj Saileshwar. Privacy risks of speculative decoding in large language models. *arXiv:2411.01076*, 2024.

[79] Roy Weiss, Daniel Ayzenshteyn, and Yisroel Mirsky. What was your prompt? a remote keylogging attack on AI assistants. In *USENIX Security*, 2024.

[80] Charles V Wright, Scott E Coull, and Fabian Monrose. Traffic morphing: An efficient defense against statistical traffic analysis. In *NDSS*, volume 9, 2009.

[81] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations.

[82] xAI. Grok 3 beta — the age of reasoning agents. xAI blog, Feb 2025. Announcing Grok 3, a reasoning-optimized version of the Grok chatbot.

[83] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[84] Kisung You. Semantics at an angle: When cosine similarity works until it doesn't. *arXiv:2504.16318*, 2025.

[85] Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *ACL*, 2024.

[86] Yiming Zhang, Nicholas Carlini, and Daphne Ippolito. Effective prompt extraction from language models. In *COLM*, 2024.

[87] 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. *arXiv:2504.03160*, 2025.

A Experimental Setup Details

A.1 Dataset Details

Prompt Reconstruction. For all TREC datasets used in the prompt reconstruction task, we create **-DR** variants by inputting each original prompt, along with the `suggested_rewriting_prompt` from [36], to GPT-4.1. These rewritten prompts are used for the Deep Research API runs to match its prompt handling, and all evaluations are scored against the original, unmodified prompts.

TREC FedWeb 2013 Topics (FEDWEB13) contains 50 topics originally designed to evaluate federated web search. We concatenate the `<description>` and `<narrative>` fields to form a single prompt, and apply GPT-4.1 rewriting to richer instruction-complete variants (FEDWEB13-DR).

TREC Session 2014 Topics (SESSION14) contains 60 realistic user topic descriptions spanning domains such as travel, health, and education. We use the original `description` fields as user prompts and their rewritten counterparts (SESSION14-DR).

TREC Dynamic Domain 2016 Truth Data (DD16) provides 53 topics focused on interactive information retrieval in evolving domains such as medicine and cybersecurity. Following the same procedure as FEDWEB13, we concatenate the `<description>` and `<narrative>` fields and rewrite them with GPT-4.1 (DD16-DR).

For each dataset, 20 prompts are reserved for evaluation, with the remainder used for training and ICL examples.

Trait Inference Personas. We use **SynthLabsAI/PERSONA_subset** to simulate benign users who use a web agent over time. The dataset includes 997 synthetic user profiles annotated with 32 traits, grouped into four categories:

- *Demographic*: age, sex, race, ancestry, religion, place of birth, citizenship, income, education, marital status, household type, household language, veteran status, disability, family presence and age, and health insurance
- *Occupational*: employment status, industry, occupation, class of worker, and detailed job description
- *Psychographic*: ideology, political views, and Big Five personality traits
- *Behavioral*: lifestyle, personal time, mannerisms, and defining quirks

For trait inference, we adopt a type-aware scoring strategy tailored to each trait's structure:

- *Numeric traits* (e.g., age, income) are evaluated using a normalized absolute difference: smaller relative errors yield higher scores, linearly decreasing toward zero. Age differences are scaled by 30, and income by 200K.
- *Ordinal traits* (e.g., big five scores) are mapped to integer levels and scored by normalized distance on a 5-point scale (e.g., `low` to `high`). Closer ratings yield higher similarity scores.
- *Categorical traits* (e.g., gender, marital status) use exact match for single-token values; multi-word categories are scored using the SBERT score to account for semantic equivalence.
- *Free-text traits* (e.g., job description, personal time) are assessed using the SBERT score, which computes semantic overlap via contextual token embeddings.

In our experiments, we sample 50 personas. For each, including those reserved for ICL examples, we randomly select 5 traits and use GPT-4o (temperature = 0.7) to generate 21–35 trait-revealing prompts that explicitly or implicitly embed selected traits. The produced traces serve as the input for trait inference.

A.2 Agent and LLM Backbone Choice

When browsing is executed locally, the network-level outbound traffic—accessed domains, timing, and payload length—is visible to the adversary; when browsing is server-side (e.g., OpenAI's Operator), only connections to the provider's infrastructure are exposed. To study the leakage risks in the local setting, we collect traces using three open-source WRAs, GPT Researcher [25], AutoGen [81], and Browser-Use [47], and one proprietary system, the OpenAI Deep Research API.

GPT Researcher [25]. GPT Researcher is a research agent designed for deep multi-hop investigation. Unlike lightweight browsing agents, it decomposes prompts into sub-questions, iteratively gathers evidence, and synthesizes reports. We run it in `report_type="deep"` mode, which enables multi-turn exploration over multiple sources. The agent logs visited URLs with timestamps, from which we extract domains. While it does not expose packet-level traces, these logs still capture the trajectory of its structured reasoning.

https://trec.nist.gov/data/federated/2013/fedweb13_50topics.xml
https://trec.nist.gov/data/session2014.html
https://trec.nist.gov/data/domain2016.html

A distinctive feature of GPT Researcher is its strategic use of different LLMs for different sub-tasks: `FAST_LLM` (e.g., GPT-4o-mini, DeepSeek-V3-0324) for lightweight operations such as summarization, `SMART_LLM` (e.g., GPT-4.1, Mistral-7B-Instruct v0.3) for reasoning and report generation, and `STRATEGIC_LLM` LLMs (e.g., o4-mini, Mistral-Nemo-12B) for higher-level planning such as research planning. Its default configuration assigns OpenAI models, but we also evaluate alternative combinations using local models to assess privacy leakage when deployed outside the proprietary ecosystem.

Browser-Use [47]. Browser-Use is an open-source web agent that automates browsing through a local browser, in contrast to GPT Researcher. By default, it queries Google via `search_web`, but this frequently triggers reCAPTCHA and stalls sessions. To ensure stability, we configure it to start from Bing and automatically fall back to Bing when blocked.

We further modify its system prompt to require at least five distinct page visits before summarization, preventing it from shortcutting tasks with prior knowledge or superficial searches. After each run, we extract domains and timestamps from the browser's recorded history to build ordered traces. This setup makes Browser-Use suitable for evaluating how web agents' lightweight browsing patterns leak information.

AutoGen [81]. AutoGen is another open-source web agent system; we configure it as a two-agent system: an `AssistantAgent` that interprets prompts and a `MultimodalWebSurfer` that executes browsing. It is faster and more controllable than Browser-Use, which makes it useful for both tasks. By default, AutoGen tends to summarize results directly without clicking through, so we enforce the same rule, visiting at least five pages.

To collect metadata, we extend the web surfer into a custom `LoggingWebSurfer` that hooks Playwright APIs to record domains, timestamps, and IPs for all network events. Unlike GPT Researcher or Browser-Use, AutoGen logs not only primary domains but also auxiliary requests (ads, trackers, analytics), producing noisier but realistic traces that an actual network observer would see. We run AutoGen with GPT-4o for prompt reconstruction and Gemini 2.0 Flash for trait inference.

OpenAI Deep Research API. As a proprietary reference point, we evaluate the OpenAI Deep Research API. Unlike the ChatGPT Deep Research interface, which engages in interactive follow-up questions, the API is stateless and relies entirely on instruction-rich inputs. To align its behavior with the UI, we provide **-DR** prompt variants (e.g., SESSION14-DR), rewritten to simulate the iterative refinement normally triggered by user interaction. Because its browsing occurs remotely, only provider connections are visible locally; we nonetheless analyze these traces to benchmark inference difficulty against a high-capability proprietary system.

Rationale for Agent and Model Selection. We evaluate both research (GPT Researcher) and web agents (Browser-Use, AutoGen) to capture differences in browsing style and leakage. Each open-source agent is paired with both proprietary (GPT-4o, Gemini 2.0 Flash) and open-source (DeepSeek-V3, Mistral family) LLMs, reflecting realistic local deployments where users might expect stronger privacy protections. The OpenAI Deep Research API serves as a commercial baseline. Also, we deliberately exclude larger open-source models such as LLaMA 3.1 70B or DeepSeek-R1, as they require substantial computational resources and often bypass web search by relying on internal knowledge, making them less suitable for our local, browsing-driven evaluation. Together, these choices ensure our study captures attacker-visible metadata across diverse agents and for realistic local deployments, while also showing results against strong proprietary baselines.

A.3 Inference Configuration

We evaluate our inference attacks using LLMs as an inference engine for few-shot ICL. A summary of ICL configurations is provided in Table 12, and the full ICL prompt templates are in Appendix F and Appendix G.

Prompt Reconstruction. To recover the user's original prompt from a trace, we use a 5-shot ICL configuration. Each ICL prompt consists of five (trace, prompt) pairs, where the trace includes an ordered domain sequence and associated timing information, followed by an unseen trace for prediction. The format is fully natural language, with traces presented as newline-separated domain lists (and timing metadata when present). Examples are drawn from distinct topics to prevent overfitting and to test generalization across different intent types. This format encourages the model to infer high-level task semantics from the structure, content, and temporal progression of visited domains.

Trait Inference. For trait inference, we adopt a 3-shot ICL format. Each example provides a domain trace and corresponding persona traits. The reduced shot count reflects the greater regularity of trait-disclosing behavior over time and helps minimize prompt length for complex structured outputs. The model is asked to generate a structured trait list in the format `- Trait: Value`, and is explicitly instructed to infer as many traits as possible, even under partial evidence.

B Additional Prompt Reconstruction Results

Effect of ICL Contrastive Configurations. Gao et al. [26] introduced contrastive ICL, showing that adding negative examples to the in-context setup can improve performance, even when negatives must be synthesized by the LLM. Since our datasets lack native negatives, we follow their procedure to generate them automatically.

In our baseline, each 5-shot prompt includes five (trace, prompt) pairs followed by a new trace for prediction (Figure 14 and **??**). Contrastive variants augment these with one or three negatives, either in a simple setup or in our proposed **Quality-Filtered (QF) contrastive ICL**, where negatives

Table 12: Few-shot ICL configuration for each attack task.

Task	ICL Shots	Output Format	Prompt Details
Prompt Reconstruction	5	Natural language query	Domain trace → prompt
Trait Inference	3	Structured list (`-Trait: Value`)	Domain trace → trait list

are regenerated until their SBERT similarity to the original prompt falls below a threshold, ensuring sufficient dissimilarity.

As shown in Table 13, adding negatives did not improve overall results. While domain-type alignment (E_{dom}) occasionally improved, most other metrics decreased. Even with larger numbers of negatives and QF filtering, performance remained below the 5-shot baseline. We attribute this to the difficulty of the prompt inference task: the model may already operate near its limits, so additional negatives increase confusion rather than sharpening topical discrimination.

Table 13: Prompt reconstruction performance for the 5-shot ICL baseline compared to contrastive ICL variants. The results show that the 5-shot baseline achieves better overall performance than the contrastive variants.

Type of ICL	SBERT	LLM-Judge	OBELS			
			E_{func}	E_{dom}	E_{sem}	T_{ent}
5-shot ICL	0.492	**0.415**	**0.770**	0.735	0.520	0.640
Simple (1 neg)	**0.495**	0.405	0.740	0.745	**0.525**	**0.645**
Simple (3 neg)	0.490	0.355	0.755	**0.750**	0.510	0.625
QF (1 neg)	0.493	0.385	0.765	0.680	0.520	0.590
QF (3 neg)	0.492	0.355	0.750	0.740	0.505	0.640

Effect of Example Selection. We compare random versus embedding-based selection of ICL examples. In the random strategy, examples are sampled from the training set for each test instance. For embedding-based selection, all training and test traces are encoded with the `all-MiniLM-L6-v2` sentence transformer, and the training examples with the highest cosine similarity to the test trace are chosen. As shown in Table 14, embedding-based selection does not improve prompt reconstruction and in some metrics performs slightly worse, suggesting that random sampling, by providing more diverse examples, is the more effective strategy.

Effect of Example Ordering. We test whether the ordering of embedding-selected examples influences ICL performance, comparing random order, ascending similarity, and descending similarity. As shown in Table 15, ordering has little effect: scores remain close across all metrics. This suggests that once relevant examples are chosen, their sequence contributes minimally to reconstruction quality.

Table 14: Prompt reconstruction with random versus embedding-based ICL example selection. Embedding-based selection fails to improve results, while random sampling yields stronger overall performance by providing greater diversity.

Ex Selection Strategy	SBERT	LLM-Judge	OBELS			
			E_{func}	E_{dom}	E_{sem}	T_{ent}
Random	**0.492**	**0.415**	**0.770**	0.735	**0.520**	**0.640**
Embedding-based	0.481	0.375	0.740	**0.755**	0.515	0.605

Table 15: Prompt reconstruction with different orderings of embedding-selected examples. Ordering shows only minor variations, indicating limited impact on performance.

Ex Ordering	SBERT	LLM-Judge	OBELS			
			E_{func}	E_{dom}	E_{sem}	T_{ent}
Random	0.481	0.375	0.740	**0.755**	0.515	0.605
Ascending	**0.493**	**0.415**	0.725	0.735	**0.525**	**0.645**
Descending	0.481	0.380	**0.755**	0.745	0.495	0.620

C Additional Virtual Persona Defense Results

Figure 8 and Figure 9 provide additional experimental results of the proposed virtual persona defense beyond the median similarity score reductions reported in the main text.

D OBELS Prompt Templates

Figure 10 shows the prompt used in the OBELS framework to convert natural language queries into structured semantic triplets of the form (intent, source_type, entity). These triplets capture the user's goal, the type of resource sought, and the core entity involved. Prompts are drawn from the TREC Session Track 2014 dataset [17] and span a variety of user intents, including planning, learning, comparing, and exploring. For brevity, some prompts are abbreviated with ellipses ("...") to fit within the template box, though the actual prompts used were complete.

Figure 11 then shows the complementary OBELS scoring prompt, which evaluates behavioral similarity between reconstructed and original prompts. The model is instructed to align triplets and score their correspondence along four dimensions:

Available via Hugging Face: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2. Fine-tuned on 1B sentence pairs with a contrastive objective, this model maps each trace into a 384-dimensional dense vector space.

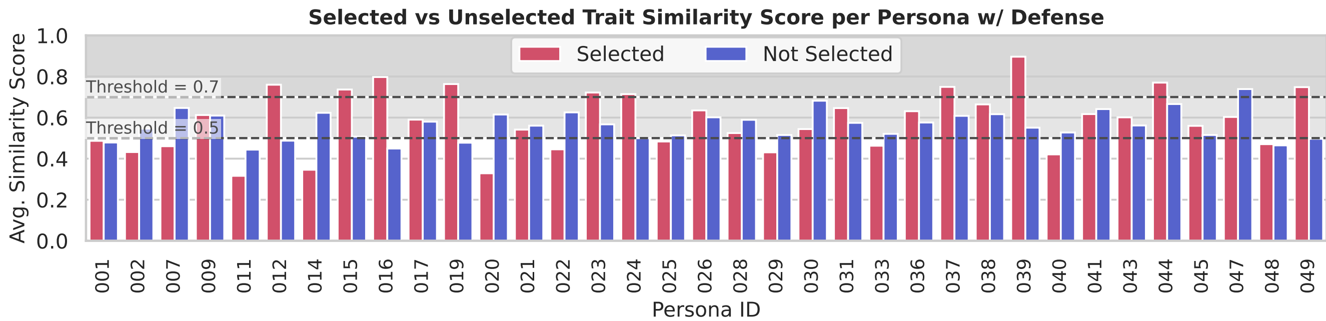

Figure 8: Average inference accuracy for selected vs. unselected traits per persona, showing generally lower accuracy than Figure 5 with the proposed virtual persona defense applied.

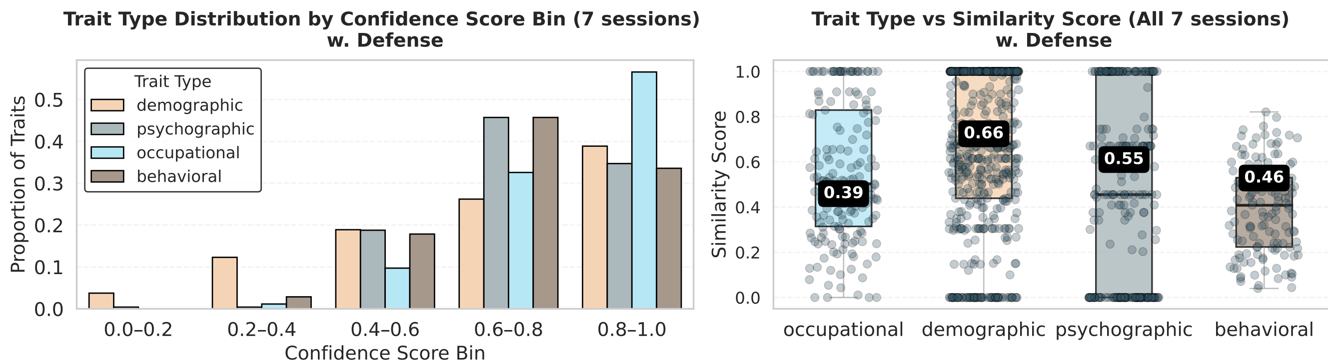

Figure 9: Trait-level breakdown of inference performance with the virtual persona defense. Confidence distributions remain similar to Figure 6, but average similarity scores are lowered across all categories.

functional equivalence, domain type, semantic alignment, and entity granularity. This structured format ensures consistent evaluation across diverse prompt reconstructions.

E Prompt Template for Generating Persona-Based Queries

Figure 13 shows the full prompt used to elicit naturalistic web search queries from a simulated user. The goal is to mimic realistic multi-session browsing behavior based on a given persona, with selected traits subtly reflected in the generated queries.

F 5-Shot ICL Prompt Templates (Prompt Reconstruction)

Figure 14 presents the 5-shot ICL prompt used for reconstructing user intent from observed domain traces. Each example contains a list of domains followed by the original prompt that likely triggered them. The model is prompted to generate a plausible user prompt consistent with the browsing behavior.

G 3-Shot ICL Prompt Templates (Trait Inference)

?? illustrates the 3-shot ICL prompt used for inferring personal attributes from browsing data. Each example pairs a list of visited domains with a set of traits across 32 predefined dimensions. The model is instructed to extract as many traits as possible based solely on behavioral cues.

The examples used in the few-shot template are based on `persona_000`, `persona_003`, and `persona_004`. For brevity, we omit some traits from `persona_003` in **??**. In the actual prompt used during inference, the full list of traits for `persona_003` is included.

OBELS Prompt Template for Triplet Abstraction

```
You are an AI system that abstracts natural language prompts into semantic triplets of the
form: (intent, source_type, entity)

Where:
- `intent` captures the user's primary semantic goal. Instead of using a generic label like
"get_info", choose more specific intents when appropriate. Use one of:
  - "learn": to understand or gain knowledge about a topic
  - "explore": to investigate options or alternatives
  - "analyze": to understand causes, effects, or implications
  - "compare": to contrast two or more entities or options
  - "summarize": to find concise descriptions or overviews
  - "plan": to organize steps toward a future action
  - "decide": to weigh alternatives with the goal of making a choice
  - "book", "watch", "read", "evaluate", etc. as appropriate
  - Retain "get_info" only for truly generic factual lookups

- `source_type` is the type of domain, service, or information requested (e.g., travel,
symptom, policy_area, cooking_method, event, visa_process, treatment_method, academic_field,
cuisine, etc).

- `entity` is the specific concept, item, or group of interest (e.g., Italy, depression, cold
turkey, face transplant, immigration, Dulles airport, cold turkey, PhD in Business, Swahili
dish, etc).

- If the prompt includes modifiers such as price, audience, date, location, or purpose,
extract separate `target` triplets:
  ("target", source_type, entity)

Your output should be a list of triplets capturing each atomic semantic intent.

**Prompt:**
{prompt}
**Triplets:**
```

Figure 10: Prompt template for OBELS triplet abstraction. The prompt guides the model to convert natural language queries into structured (intent, source_type, entity) triplets.

OBELS Prompt Template for Scoring (Continued)

```
You are an expert evaluator assessing the behavioral similarity between two user prompts.
Each prompt has been abstracted into a **set of semantic triplets** of the form:

(intent, source_type, entity)

Your job is to compare the two sets holistically across four dimensions of behavioral
similarity.

---

## Evaluation Guidelines

### Step 1: Set-Level Alignment
Compare the full **triplet sets** for Prompt A and Prompt B. Use semantic similarity (not
strict string match) to align triplets. Triplets may align even if only **one or two fields**
are semantically similar (e.g., both target energy sources, or both analyze environmental
impacts). You may align multiple triplets as long as they reflect overlapping behavior or
goal.

**Be generous in identifying partial matches**--this alignment is used to assess user intent,
not exact wording.

Align greedily based on **overall behavioral similarity**, and include all meaningful pairs
even if imperfect.

### Step 2: Score the Four Dimensions
Score from 0.0 to 1.0 using the following definitions:

1. **Functional Equivalence**:
Do the prompts express the same high-level user intent across their triplets?

2. **Domain Type Equivalence**:
Do the prompts rely on similar types of services or sources of information?

3. **Semantic Equivalence**:
Do the `entity` fields refer to semantically similar or related concepts?

4. **Entity Granularity Tolerance**:
Do the `entity` fields differ in specificity but still refer to compatible ideas (e.g., Honda
vs. Honda Civic)?

## Scoring Scale:
- 1.0 = completely equivalent
- 0.8 = very similar
- 0.5 = somewhat related
- 0.2 = weakly related
- 0.0 = unrelated or contradictory

---
```

OBELS Prompt Template for Scoring

```
**Prompt A Triplets:**
{triplets_a}

**Prompt B Triplets:**
{triplets_b}

---

## Please Return:

1. A list of aligned triplet pairs used in your comparison.
2. A JSON object with four fields:
   - `functional_equivalence`
   - `domain_type_equivalence`
   - `semantic_equivalence`
   - `entity_granularity_tolerance`
3. A short 1-2 sentence rationale for each score.

Format your response like:

```json
{
  "aligned_triplets": [
    [["search", "flight", "Europe"], ["search", "flight", "international flights to Europe"]],
    [["target", "price_range", "cheapest"], ["target", "price_range", "low cost"]]
  ],
  "scores": {
    "functional_equivalence": 1.0,
    "domain_type_equivalence": 0.8,
    "semantic_equivalence": 0.6,
    "entity_granularity_tolerance": 0.7
  },
  "rationale": {
    "functional_equivalence": "...",
    "domain_type_equivalence": "...",
    "semantic_equivalence": "...",
    "entity_granularity_tolerance": "..."
  }
}
```
```

Figure 11: Prompt template for OBELS scoring. Given two sets of triplets derived from user prompts, the model evaluates their behavioral similarity across four dimensions: functional equivalence, domain type equivalence, semantic equivalence, and entity granularity tolerance.

Prompt Template for LLM as a Judges

```
You are an evaluator comparing two user prompts for task-oriented web agents. The goal is to
assess whether the **Reconstructed Prompt** successfully captures the **core intent** of the
**Original Prompt**, even if wording, style, or structure differ. You should focus on **overall
task equivalence** -- whether a web agent executing the reconstructed prompt would accomplish
essentially the same task the user intended in the original.

Give a similarity score from **0.0 to 1.0**, where:
- 1.0 → Clearly the same task, just worded differently.
- 0.9-0.8 → Minor differences that wouldn't affect the outcome.
- 0.7-0.5 → Generally the same topic or direction, but with some important drift.
- 0.4-0.1 → Only loosely connected; task likely wouldn't work the same way.
- 0.0 → Completely different or unrelated task.

Err on the side of **leniency** -- small phrasing or structure changes are acceptable if the
intended action is preserved. Output **only the similarity score** (a number between 0.0 and
1.0). Do not include explanations.

Now evaluate the following prompts:

Original Prompt: {original}
Reconstructed Prompt: {reconstructed}
```

Figure 12: Prompt template for LLM as a Judge to measure reconstruction accuracy.

Generate Queries for Persona

```
You are simulating a user with the following profile. Your task is to write realistic web
search queries across 7 sessions that this person might ask an AI browsing assistant.

The goal is to ensure that the following 5 selected traits are naturally embedded in the
queries. The queries may explicitly mention or implicitly suggest these traits.

----
Persona Summary:
{persona_text}

Selected Traits:
{
  "trait_1": "...",
  "trait_2": "...",
  "trait_3": "...",
  "trait_4": "...",
  "trait_5": "..."
}
----

Instructions:

1. Generate 3-5 realistic, informative web queries for each of 7 sessions.
2. Across all sessions, ensure that each of the selected traits is revealed through at least
one query.
3. Avoid listing traits directly -- instead, embed them naturally in the context of the
queries.
4. Your output should start with "User prompt:" and format the queries session-wise as:

User prompt:
Session 1:
- ...
- ...

...

Session 7:
- ...
```

Figure 13: Prompt template used for generating trait-revealing web queries across sessions.

5-shot ICL Prompt Template (Prompt Reconstruction) (Continued)

```
Objective:
You are given a list of domains visited by a web agent during a browsing session.
Your task is to reconstruct the most plausible **original user prompt** that would have
caused the agent to visit these domains.
This prompt should reflect the likely information need or task of the user, as inferred
from the domains.
Here are some examples:

Example 1:
Visited Domains:
- dev.radiate.sanuker.com
- www.heephong.org
- webchat.stellabot.com
- crossmark.crossref.org
- www.ncbi.nlm.nih.gov
- dap.digitalgov.gov
- www.unicef.org

Reconstructed Prompt:
"What are typical developmental milestones for infants age 0-12 months in East Asian countries
(for example, China, South Korea, Japan, Thailand, etc)"

Example 2:
Visited Domains:
- www.americanweatherstar.com
- searchlocalroofers.com
- blog.architizer.com
- tru.am
- architizer.com
- architizer-prod.imgix.net

Reconstructed Prompt:
"What is silicone roof coating? What is it best for? What are other solutions regarding roof
coating? What is the best material to use if the purpose is cooling down the roof so that the
house temperature remains low?"

Example 3:
Visited Domains:
- en.wikipedia.org
- www.ft.com
- consent-notice.ft.com
- consent-manager.ft.com
- ft.coral.coralproject.net
- public.flourish.studio
- i.guim.co.uk
- sourcepoint.theguardian.com
```

5-shot ICL Prompt Template (Prompt Reconstruction)

```
Reconstructed Prompt:
"Portugal, Spain, Ireland, Greece, and Cyprus are some of the countries that have experienced
severe economic downturns during the ongoing Eurozone crisis. What actions are political
leaders in Europe taking to resolve their respective economic troubles? How are these actions
meant to help?"

Example 4:
Visited Domains:
- africageographic.com
- ukuri.travel
- edge.aditude.io
- syndicatedsearch.goog
- raven-edge.aditude.io
- onesignal.com
- linguafonica.com

Reconstructed Prompt:
"What are some traditional Swahili dishes? What ingredients do they use to cook them? Are
swahili people using any particular herb in their dishes? Could you find these ingredients
in your country? Are there any recipes you can find online?"

Example 5:
Visited Domains:
- www.aireserv.com
- www.datadoghq-browser-agent.com
- tags.srv.stackadapt.com
- www.sylvane.com
- vcc-na7.8x8.com
- shopper.shop.pe

Reconstructed Prompt:
"You would like to buy a dehumidifier. You want to know what makes a dehumidifier good value
for money. Find web pages that will help you do so."

Now, for the following set of visited domains, reconstruct the most plausible original user
prompt.

Important: Only output the reconstructed prompt itself. Do not output the visited domains,
reasoning, or any other text.

Visited domains:
{list_of_domains}

Reconstructed Prompt:
```

Figure 14: 5-shot ICL template used for reconstructing user prompts from domain traces. The template provides five examples of (domain list, prompt) pairs, guiding the model to generate a naturalistic prompt that aligns with the observed browsing behavior.

5-shot Contrastive ICL Prompt Template (Prompt Reconstruction) (Continued)

```
Objective:
You are given a list of domains visited by a web agent during a browsing session.
Your task is to suggest a good, preferred prompt (query) for the information need that best
matches the set of visited web domains provided.
You will see examples with visited domains, a preferred (specific) query, multiple less
preferred (vague or off-topic) queries, and reasoning.
The preferred query should be specific, actionable, and match the domains content.
The less preferred queries should be vague, off-topic, or missing critical details.
Here are some examples:

Example 1:
Visited Domains:
- dev.radiate.sanuker.com
- www.heephong.org
- webchat.stellabot.com
- crossmark.crossref.org
- www.ncbi.nlm.nih.gov
- dap.digitalgov.gov
- www.unicef.org

Preferred Query:
"What are typical developmental milestones for infants age 0-12 months in East Asian countries
(for example, China, South Korea, Japan, Thailand, etc)"

Less Preferred Query:
"What are some things babies do?"

Reasoning:
The preferred query is specific, actionable, and matches domains content. The less preferred
queries are vague, off-topic, or missing critical details.

Example 2:
Visited Domains:
- www.americanweatherstar.com
- searchlocalroofers.com
- blog.architizer.com
- tru.am
- architizer.com
- architizer-prod.imgix.net

Preferred Query:
"What is silicone roof coating? What is it best for? What are other solutions regarding roof
coating? What is the best material to use if the purpose is cooling down the roof so that the
house temperature remains low?"

Less Preferred Query:
"What are some ways to improve a building's appearance?"

Reasoning:
The preferred query is specific, actionable, and matches domains content. The less preferred
queries are vague, off-topic, or missing critical details.
```

5-shot Contrastive ICL Prompt Template (Prompt Reconstruction) (Continued)

```
Example 3:
Visited Domains:
- en.wikipedia.org
- www.ft.com
- consent-notice.ft.com
- consent-manager.ft.com
- ft.coral.coralproject.net
- public.flourish.studio
- i.guim.co.uk
- sourcepoint.theguardian.com

Preferred Query:
"Portugal, Spain, Ireland, Greece, and Cyprus are some of the countries that have experienced
severe economic downturns during the ongoing Eurozone crisis. What actions are political
leaders in Europe taking to resolve their respective economic troubles? How are these actions
meant to help?"

Less Preferred Query:
"What are some European countries?"

Reasoning:
The preferred query is specific, actionable, and matches domains content. The less preferred
queries are vague, off-topic, or missing critical details.

Example 4:
Visited Domains:
- africageographic.com
- ukuri.travel
- edge.aditude.io
- syndicatedsearch.goog
- raven-edge.aditude.io
- onesignal.com
- linguafonica.com

Preferred Query:
"What are some traditional Swahili dishes? What ingredients do they use to cook them? Are
swahili people using any particular herb in their dishes? Could you find these ingredients in
your country? Are there any recipes you can find online?"

Less Preferred Query:
"What are some foods people eat in Africa?"

Reasoning:
The preferred query is specific, actionable, and matches domains content. The less preferred
queries are vague, off-topic, or missing critical details.
```

5-shot Contrastive ICL Prompt Template (Prompt Reconstruction) (continued)

```
Example 5:
Visited Domains:
- www.aireserv.com
- www.datadoghq-browser-agent.com
- tags.srv.stackadapt.com
- www.sylvane.com
- vcc-na7.8x8.com
- shopper.shop.pe

Preferred Query:
"You would like to buy a dehumidifier. You want to know what makes a dehumidifier good value
for money. Find web pages that will help you do so."

Less Preferred Query:
"What are some popular household appliances?"

Reasoning:
The preferred query is specific, actionable, and matches domains content. The less preferred
queries are vague, off-topic, or missing critical details.

Now, for the following set of visited domains, suggest a good, preferred prompt (query) that
best matches the information need implied by these domains.

Important: Only output the preferred query itself. Do not output the less preferred query,
reasoning, or any other text.

Visited domains:
{list of domains}

Preferred Query:
```

Figure 15: 5-shot contrastive ICL template for reconstructing user prompts from domain traces, with preferred vs. less-preferred queries to guide the model.

3-shot ICL Prompt Template (Trait Inference) (Continued)

```
Objective:
You are given a list of domains visited by a person during web browsing sessions.
Your task is to infer as many traits as possible about the person based on their browsing
behavior.

You are simulating an adversary attempting to reconstruct the user's profile.
Your goal is to **maximize coverage**: infer traits across as many of the following 32
categories as possible, even if some evidence is weak, partial, or indirect.

The 32 possible traits include:
age, sex, race, ancestry, religion, place of birth, citizenship, income, education,
employment status, industry category, occupation category, class of worker, veteran status,
marital status, household type, household language, family presence and age, health
insurance, disability, ideology, political views, lifestyle, personal time, mannerisms,
defining quirks, detailed job description, and the Big Five traits (Agreeableness,
Conscientiousness, Extraversion, Neuroticism, Openness).

Output as many traits as possible using this format:
**One trait per line**, each prefixed with a dash ('-').
Do **not** include explanations or confidence levels.
Strictly use the format:

- Trait: Value

---

Example 1:

Visited Domains:
"catholicclimatecovenant.org", "www.crs.org", "www.california.com", "www.oceanviewfarms.net",
"www.selfsufficiencyinthecity.com", "www.catholicshare.com", "catholicshare.substack.com",
"catholicweekly.com.au", "www.sydneycatholic.org", "www.epa.gov", "www.youtube.com",
"www.conferencelists.org", "www.backyardgardenlover.com", "www.thespruce.com",
"www.programsforelderly.com", "universaldesignstyle.com", "www.pbs.org", "bookauthority.org",
"earthwatch.org", "seedsavers.org", "exchange.seedsavers.org", "www.coursera.org",
"www.edx.org", "sonic-ui.highereducation.com", "www.usccb.org", "realitypathing.com",
"www.jointhecollective.com"

Inferred Traits:
- Age: 89
- Ancestry: American
- Big Five Scores: Openness: Extremely Low, Conscientiousness: High, Extraversion: Average,
Agreeableness: Extremely High, Neuroticism: Average
- Citizenship: Born in the United States
- Class Of Worker: Retired
- Defining Quirks: Has a vast collection of rare plants
- Detailed Job Description: Conducted research on environmental issues and advocated for
environmental protection
- Disability: With a disability
```

3-shot ICL Prompt Template (Trait Inference) (continued)

```
- Education: Doctorate degree
- Employment Status: Not in labor force
- Family Presence And Age: No related children
- Health Insurance: With health insurance coverage
- Household Language: English only
- Household Type: Married couple household, no children of the householder less than 18
- Ideology: Progressive
- Income: 519400.0
- Industry Category: Academia
- Lifestyle: Quiet and intellectual
- Mannerisms: Often lost in thought, tends to speak in academic jargon
- Marital Status: Married
- Occupation Category: Environmental Scientist
- Personal Time: Spends free time in her home garden or reading
- Place Of Birth: California/CA
- Political Views: Democrat
- Race: White alone
- Religion: Catholic
- Sex: Female
- Veteran Status: Non-Veteran

---

Example 2:

Visited Domains:
"vehiclefreak.com", "www.autocheatsheet.com", "www.rydeshopper.com", "learnlaughspeak.com",
"www.indeed.com", "thegreatestbooks.org", "www.fluentu.com", "bookclubs.com",
"mylanguageexchange.com", "www.irs.gov", "www.keepertax.com", "www.uscis.gov", "www.dhs.gov",
"www.healthcare.gov", "www.forbes.com", "www.meetup.com", "www.sandiego.gov",
"www.sddemocrats.org", "www.paloma-aguirre.com", "www.limo.org", "www.nerdwallet.com",
"www.xero.com", "www.umasstransportationcenter.org", "www.taxibutler.com",
"www.flextaxis.com", "www.limoanywhere.com"

Inferred Traits:
- Age: 34
- Ancestry: Bulgarian
- Big Five Scores: Openness: Average, Conscientiousness: Average, Extraversion: Extremely
High, Agreeableness: Low, Neuroticism: Average
- Citizenship: U.S. citizen by naturalization
- Class Of Worker: Self-employed in incorporated business
- Defining Quirks: Deep love for literature and reading
- Detailed Job Description: Drives customers to destinations and maintains vehicle
- Disability: None
- Education: Some college credit, no degree
- Employment Status: Civilian employed, at work
- Family Presence And Age: No family in the U.S.
- Health Insurance: With health insurance coverage
- Household Language: Other Indo-European languages
- Household Type: Male householder, living alone
- Ideology: Liberal
```

3-shot ICL Prompt Template (Trait Inference) (Continued)

```
- Income: 60000.0
- Industry Category: Taxi and Limousine Service
- Lifestyle: Active and independent
- Mannerisms: Highly expressive, talks often while driving
- Marital Status: Never married
- Occupation Category: Shuttle Drivers and Chauffeurs
- Personal Time: Spends time reading or exploring the city
- Place Of Birth: Bulgaria
- Political Views: Democrat
- Race: White alone
- Religion: Religiously unaffiliated
- Sex: Male
- Veteran Status: Non-Veteran

Example 3:

Visited Domains:
"www.forbes.com", "www.constructionbusinessowner.com", "www.brainzmagazine.com",
"kbscpa.com", "www.engineering.com", "www.truxnow.com", "buildertrend.com",
"www.procore.com", "www.indeed.com", "modernmrsdarcy.com", "www.kiplinger.com",
"www.greatseniorliving.com", "seniorsreports.com", "www.constructionplacements.com",
"www.gop.com", "rnc.org", "www.nbcnews.com", "www.valuepenguin.com", "www.lendingtree.com",
"www.humana.com", "www.scripps.org", "www.webmd.com", "www.ncoa.org", "www.verywellmind.com",
"www.nimh.nih.gov", "www.businessresearchinsights.com", "xpro.mit.edu", "www.coursera.org",
"www.pmi.org", "www.constructiondive.com"

Inferred Traits:
- Age: 71
- Ancestry: Irish
- Big Five Scores: Openness: Average, Conscientiousness: Average, Extraversion: High,
Agreeableness: Average, Neuroticism: Average
- Citizenship: Born in the United States
- Class Of Worker: Employee of a private for-profit company or individual
- Defining Quirks: Keen interest in vintage fashion
...
- Place Of Birth: California/CA
- Political Views: Republican
- Race: White alone
- Religion: Protestant
- Sex: Male
- Veteran Status: Non-Veteran

---

Visited_domains:
{visited_domains}

Inferred Traits:
```

Figure 16: 3-shot ICL template for trait inference from domain traces. Each example consists of a (domain traces, inferred traits) pair. The model is guided to infer as many traits as possible based solely on the user's browsing behavior.

Prompt Template for Utility Measurement

```
You are an impartial evaluator tasked with assessing the **utility** of a research
report generated by an autonomous web research agent.

## Definition of Utility
Utility means how helpful the report is for a human reader to achieve their research goals.
Key aspects of utility include:
- **Coverage**: How well does the report cover relevant aspects of the research question?
- **Depth**: How detailed and well-explained are the findings?
- **Accuracy & Evidence**: Are claims supported with sources, citations, or reasoning?
- **Clarity**: Is the report clearly structured and easy to understand?
- **Actionability**: Does the report provide insights or conclusions that a user could act
upon?

## Instructions
1. Read the research question and the report.
2. Evaluate the report's utility using the dimensions above.
3. Assign a **utility score** between 1 and 10.
- `1` = very low utility, incomplete, or misleading.
- `10` = extremely useful, comprehensive, reliable.
4. Provide your output strictly in **valid JSON**.

## Input
Research Question: {query}
Report: {report}

## Output Format
"research_question": "...",
"coverage_score": 0,
"depth_score": 0,
"accuracy_score": 0,
"clarity_score": 0,
"actionability_score": 0,
"overall_utility_score": 0,
"justification": "..."

Note: make sure to use "[]" instead of "()" to have valid JSON. Your output should be a valid
JSON object.
```

Figure 17: Prompt template for measuring the utility of the output report of the web agent. Utility means how helpful the report is for a human reader to achieve their research goals.