Network-Level Prompt and Trait Leakage in Local Research Agents

Anonymous Authors[1]

Abstract

We show that Web and Research Agents (WRAs) are vulnerable to inference attacks by passive network observers. Deployment of WRAs *locally* for privacy, legal, or financial purposes exposes them to DNS resolvers, malicious ISPs, VPNs, web proxies, and corporate or government firewalls. However, unlike sporadic and scarce web browsing by humans, WRAs visit $70-140$ domains per request with a distinct timing pattern, creating unique privacy risks. Specifically, we demonstrate a novel prompt and trait leakage attack that only leverages WRAs' network-level metadata. We start by building a new dataset of WRA traces based on real and synthetic user search queries. We define a behavioral metric (called OBELS) to comprehensively assess similarity between original and inferred prompts, showing that our attack recovers over 73% of the functional and domain knowledge of user prompts and up to 19 of 32 traits in a multi-session setting. Finally, we discuss mitigation strategies that constrain domain diversity or obfuscate traces, showing negligible utility impact while reducing attack effectiveness by an average of 29%.

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 (Huang et al., 2025). 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. While on-device execution through locally-deployable models and TLS-encrypted access could suggest privacy guarantees, this perception is misleading. WRAs leave unavoidable, structured metadata traces that can reveal sensitive user intent, creating an underexplored leakage vector.

As autonomous agents, WRAs inherit vulnerabilities from both LLMs and traditional web browsing (Hui et al., 2024; Carlini et al., 2021; Siby et al., 2020; Bird et al., 2020). Yet they also expose a distinct, underexplored vulnerability: behavioral traces left by an LLM acting on a user's query. Prior work has shown that user intent and goals can be inferred from search queries and browsing activity (Rose & Levinson, 2004; Jones et al., 2007). Even when content is encrypted, metadata such as domain names, access order, payload size, and timing can silently reveal what a user is trying to accomplish (Oliveira et al., 2023; Su et al., 2017).

Crucially, these behavioral traces are operationally distinguishable. WRAs generate dense cascades of semantically related domain visits within short time windows (Perplexity AI, 2025). Unlike cognitively paced, irregular, and opportunistic human browsing (Kellar et al., 2007), WRAs follow deterministic retrieval-and-synthesis loops with low variance in per-step timing. As a result, a single WRA session often traverses 40–100 distinct domains, whereas human sessions typically rely on few sites (Wang et al., 2025). These structured, high-throughput access patterns make WRA traffic reliably separable from human activity (Chiang et al., 2025; Mudryi et al., 2025).

We adopt a realistic threat model in which a passive adversary (e.g., DNS resolver, ISP, enterprise firewall, or local network operator) observes only domain-level traffic metadata, without access to prompts, page contents, or model internals. Despite limited visibility, we show that attackers can reliably recover what users asked and who they are.

Motivating Examples. Consider a user seeking reproductive health information: their WRA may visit clinics, support forums, and medical sites. Even with encryption, these traces alone can expose the agent's internal reasoning process and intent. In essence, privacy failures emerge not from what agents say, but from how they act.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review at the Second Workshop on Agents in the Wild: Safety, Security, and Beyond (AIWILD) at ICML 2026. Do not distribute.

Code, prompts available at https://anonymous.4open.science/r/wra-6E76

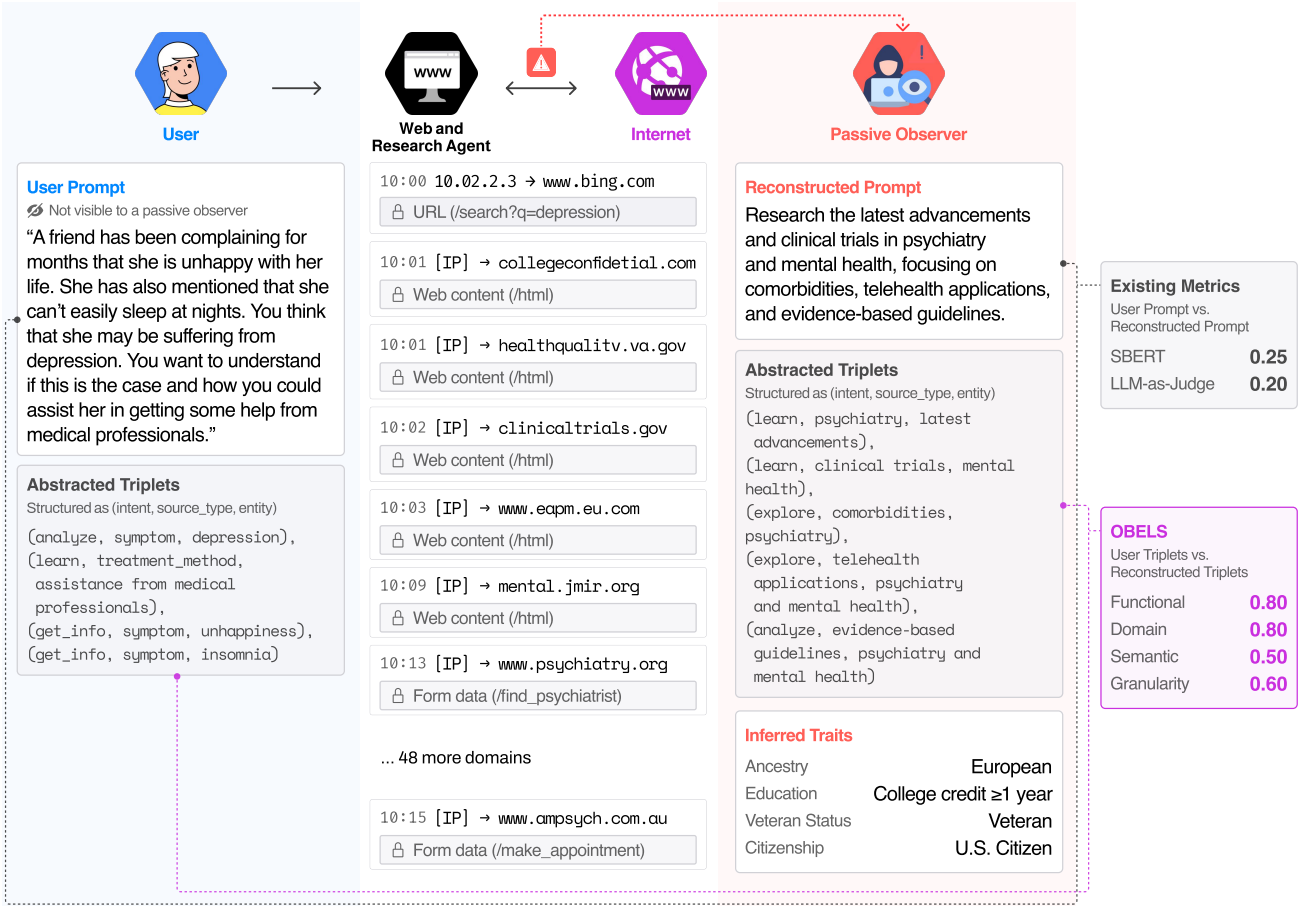

Figure 1. **Overview of the attack.**

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 recovery and trait inference.
- **Benchmark dataset and OBELS metric:** We construct and release a dataset linking domain traces to prompts and persona traits, and introduce OBELS, an ontology-aware metric for evaluating privacy leakage through behavioral traces.
- **Ablation and mitigation:** We conduct extensive experiments across agents, LLMs, and visibility, and evaluate mitigation that constrains domain diversity or obfuscates traces while negligibly affecting utility.

2. Background and Related Works

2.1. Web and Research Agents

WRAs operationalize deep evidence synthesis through a *search–retrieve–extract–synthesize* loop across many sources (Figure 2). Given a user query, the agent generates sub-queries and issues them via APIs such as Tavily (Tavily, 2025) 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 (Huang et al., 2025; Schmidgall & Moor, 2025; Zheng et al., 2025; Perplexity AI, 2025; Chiang et al., 2025). Systems exemplifying this paradigm include GPT Researcher (Elovic), LangChain's Local Deep Researcher (LangChain AI, 2025), OpenAI Deep Research (OpenAI, 2025), Grok DeepSearch (xAI, 2025), and Gemini Deep Research (Google, 2025).

Because WRAs generate dense browsing traces over dozens of domains, they amplify metadata-leakage risks. This risk has been underscored when Grok and OpenAI misconfigurations exposed private prompts to public indexing (McMahon, 2025; Stokel-Walker, 2025). Locally deployed agents can be one of the mitigation strategies against such threats. Running on personal machines or servers, they preserve confidentiality while offering lower latency, offline use, and lower cost. Examples include GPT-Researcher (Elovic) with self-hosted backends such as Ollama (Ollama Team, 2025) and AMD's GAIA (AMD, 2025) to operate fully on-device,

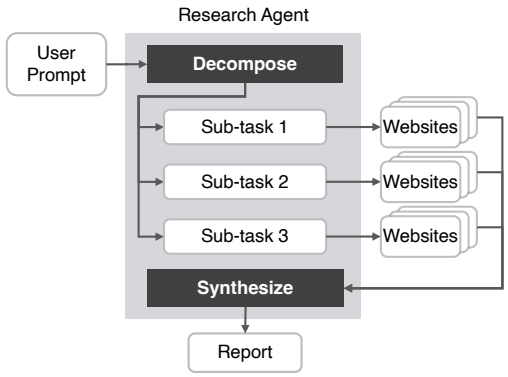

Figure 2. The workflow of Web and Research Agents.

giving users more control over both their data and their agent workflows.

2.2. Privacy Attacks

Agentic Privacy and Security. AI agents amplify privacy risks by combining LLM reasoning with active browsing and external tool integration (Díaz et al., 2025). Kim et al. (Kim et al., 2025) showed that compromised agents can scrape PII from visited pages and reuse it in phishing, while InjecAgent (Zhan et al., 2024) demonstrated that crafted payloads in webpages can trigger secret retrieval or unauthorized service access. Defenses such as AirGapAgent (Bagdasarian et al., 2024) and CHeaT (Ayzenshteyn et al., 2025) 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. Threat Model

Adversary Goals. The adversary's objective is to infer sensitive user information from agents' browsing traces. The first is **prompt recovery**, 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 illustrate how metadata can expose both immediate user intent and persistent personal characteristics.

Adversary Capabilities. The adversary observes domain-level metadata generated by the agent's web activity, including domain names (via plaintext DNS or TLS handshake such as SNI), domain visit order, and coarse packet-level features such as timing and payload sizes. They do *not* observe full URLs (e.g., query strings), page contents, user prompts, or the agent's intermediate queries. These capabilities correspond to realistic on-path observers such as local ISPs, corporate firewalls, DNS resolvers, VPN providers, or shared network operators that collect encrypted metadata without decryption or manipulation.

Adversary Assumptions. We assume the user interacts with a local WRA that autonomously performs web browsing in response to a natural language prompt. The adversary is aware of the agent's web browsing capabilities, such as issuing web searches, clicking links, scrolling pages, or scraping structured data. The adversary monitors the session between the agent and the Internet but does not alter traffic. We treat full-session metadata visibility as an **upper bound**: weaker adversaries such as DNS resolvers or enterprise firewalls may see only subsets of domains, yet our experiments show that WRA traffic remains highly distinguishable due to its structured, high-volume access patterns. We further assume 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.

3.1. Attack Scenarios

Local WRAs are often deployed to avoid remote APIs, reduce cost, preserve control over proprietary workflows, or process confidential information that cannot be shared externally.

Personal WRAs. Individuals may use local WRAs for daily tasks, health research, or legal questions, assuming that avoiding cloud services protects confidentiality. 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 other forms of profiling.

Academic WRAs. Students and researchers may use local WRAs for coursework, literature review, grant preparation, or early-stage research. External observers like ISPs or competing research entities could exploit traces to reconstruct prompts or profile academic interests, leaking unpublished projects, political interests, academic direction, or strategically valuable research topics.

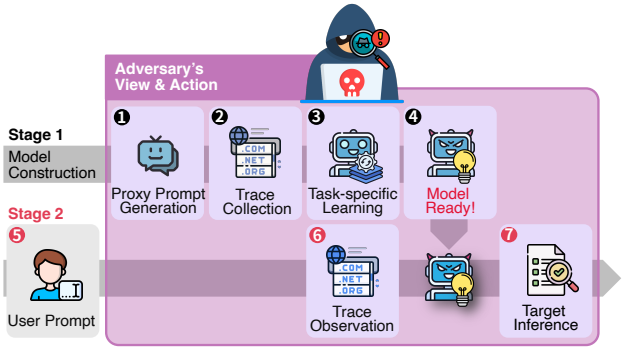

Figure 3. **Two-stage attack pipeline.**

Industrial WRAs. Companies may deploy local WRAs for competitor analysis, product planning, legal review, finance, or customer and patient-related workflows. Such traces, exploited by adversaries, may expose product roadmaps, M&A interests, legal strategies, financial planning, or other business-sensitive information to ISPs, cloud providers, data brokers, competitors, or internal monitors.

State and civic WRAs. Government agencies, journalists, activists, or civil society groups may use WRAs to investigate politically sensitive issues. In monitored networks, domain traces could reveal sensitive inquiries, organizational priorities, or vulnerable communities of interest.

These scenarios show why agent privacy in the wild cannot be reduced to hiding prompts or page contents. Even when content remains encrypted and the model runs locally, the agent's observable behavior can provide actionable signals for immediate prompt recovery and long-term trait profiling.

4. Attack Methodology

The end-to-end pipeline for our privacy attacks follow two-stage process: (1) **offline model construction** on proxy datasets and (2) **online attack execution** on real user traces. 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 mimic real-world interactions, and then applies it to actual users.

4.1. Proxy Prompt Generation

To train the model in Stage 1, the adversary generates *proxy prompts* that realistically simulate user behavior and produce informative traces.Effective proxy prompts must balance realism, browsing induction, and inferential utility. (a) Prompts resemble natural-language queries real users might issue (health, education, planning). (b) They must reliably trigger multi-page browsing, rather than being satisfied by a single source (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 inference.

Prompt recovery. We draw proxy prompts from public query datasets (FEDWEB13 (Demeester et al., 2013), SESSION14 (NIST, 2014), DD16 (NIST, 2016)). Some prompts are underspecified to reliably induce meaningful browsing, and agents differ in how they handle such inputs.

Trait inference. From persona profiles, we synthesize a user query, embedding selected traits 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). Executing such prompts on agents produces labeled traces that reveal stable user attributes, enabling training of trait inference models.

4.2. Domain Trace Collection

To characterize how different prompts induce distinct browsing behavior, we collect domain-level metadata from agents via Playwright or built-in system logs. The ordered trace captures whether an agent revisits the same domains or explores the set of sources during a single prompt execution. 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.

4.3. Inference Model Construction and Use

The adversary builds a model from proxy datasets, then applies it to new traces simulating real-user activity. For **prompt recovery**, the task is to reconstruct a natural-language prompt that captures the functional intent of the user's original input. The adversary constructs an inference model from proxy (trace, prompt) pairs in the proxy dataset using either ICL or fine-tuning (Table 2). Given a user prompt, the resulting inference model is then applied to new traces to reconstruct real user prompts. For **trait inference**, the goal is to infer latent user attributes from browsing traces. Here, a proxy dataset is generated, consist-

ing of proxy (trace, traits) pairs for each annotated persona profile, and the adversary adopts ICL to construct an inference model. Against another user, the adversary infers user traits from the agent's multi-session browsing activity.

5. OBELS: Measuring Behavioral Leakage

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*: prompt recovery is risky if it leads the agent to similar sites, services, or goals, despite differences in wording. Existing metrics fall short because they assess what prompts *say*, not what they *do*.

SBERT score, computed as cosine similarity between sentence embeddings (Reimers & Gurevych, 2019), captures high-level semantic alignment but suffers from paraphrastic inflation. It misleadingly assigns high scores to verbose, generic, or loosely related prompts (Steck et al., 2024; You, 2025), overlooking task-specific behavior or intent.

LLM-as-Judge methods leverage LLMs to rate similarity through natural language instructions. While more tolerant to paraphrase and flexible, they are prompt-sensitive and often lack interpretability (Razavi et al., 2025; Lee et al., 2024), 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 decomposes alignment into distinct dimensions so that failures in one area (e.g., entity granularity) are not masked by strong similarity elsewhere. This decomposition yields interpretable diagnostics and aligns more directly with attacker goals and real-world agent behavior. 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, event, visa_process, treatment_method, academic_field, cuisine, etc); and `entity` specifies the core topic (e.g., Rome, Italy, depression, immigration, PhD, Swahili dish, etc). Removing any triplet component collapses distinct browsing behaviors, whereas variations in tone or sentiment do not meaningfully alter domain traversal.

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.

Different applications may weight these components differently. For instance, entity granularity and semantic precision dominate in high-risk healthcare scenarios; domain-type alignment matters most in regulatory tasks; and intent alignment is sufficient for lower-risk consumer scenarios such as travel planning. Collapsing these dimensions into a single score would obscure such task-sensitive variation and misrepresent leakage risk.

6. Experimental Setup

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

6.1. Datasets

Prompt Recovery (TREC). We use three TREC datasets: FedWeb 2013 (FEDWEB13) (NIST, 2013), Session 2014 (SESSION14) (NIST, 2014), and Dynamic Domain 2016 (DD16) (NIST, 2016). For FEDWEB13 and DD16, we concatenate the `<description>` and `<narrative>` fields to match SESSION14's style and prompt specificity.

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	Prompt	50 prompts	20	GPT-Researcher	77.5	143.0	1 prompt × 1 session
SESSION14	Prompt	60 prompts	20	GPT-Researcher	69.9	145.9	1 prompt × 1 session
DD16	Prompt	53 prompts	20	GPT-Researcher	64.8	155.8	1 prompt × 1 session
PERSONA	Trait	50 personas (of 997)	3	AutoGen	144.1	173.7	3–5 prompts × 7 sessions = 21–35 queries

To align with OpenAI's Deep Research API (OpenAI, 2025), we also generate **-DR variants** by rewriting prompts with GPT-4.1, while scoring is always on the original prompts. For each dataset, 20 prompts are reserved for testing, with the remainder used for training and ICL examples.

TREC query text may appear in LLM pre-training corpora, but the domain-level traces we evaluate on are newly collected and do not exist in any dataset. The attack exploits fresh behavioral traces, not memorized prompt content.

Trait Inference (PERSONA). We sample 50 profiles from PERSONA_subset (Castricato et al., 2025) (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.

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 B.2):

GPT Researcher (Elovic). 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 recovery task.

Browser-Use (Müller & Žunič, 2025). 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 the prompt recovery task with GPT-4o as the backbone.

AutoGen (Wu et al., 2024). An open-source web agent system we used for both prompt recovery (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 Recovery. We evaluate prompt recovery performance using three metrics: (1) **SBERT**, computed as cosine similarity over mean-pooled Sentence-BERT embeddings (Reimers & Gurevych, 2019); (2) **LLM-as-Judge** , where an LLM rates prompt similarity based on natural language instructions; and (3) **OBELS**, our proposed behavior-aware scoring scheme to capture functional equivalence (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 SBERT to account for semantic equivalence.
- *Free-text traits* are assessed using SBERT, computing 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 recovery, trait inference does not require alignment in downstream behavior; semantic similarity alone suffices to assess privacy leakage.

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

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
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

Table 3. **Prompt recovery performance on SESSION14 across WFAs.** 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

7. Results

7.1. Prompt Recovery from Agent Traces

WRA vs. Human Attribution Analysis. To assess whether WRA traffic is distinguishable from human browsing, we clustered agent traces together with human sessions from SESSION14 using only timing, burstiness, and domain-transition features. WRAs consistently remain highly separable with 0.86–1.0 cluster purity across topics and 97.9% global classification accuracy. WRAs thus form a distinct, easily identifiable behavioral mode, exposing a richer and more attributable metadata footprint than humans.

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 models (GPT-4o, GPT-4.1 nano, Qwen 2.5 32B) (Table 2). We create our final training set by combining the training samples from the prompt recovery datasets discussed in Section 6.1. For GPTs, fine-tuning is performed through OpenAI's fine-tuning API for 3 epochs with batch size 1. For Qwen, we apply LoRA (Hu et al., 2022) 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 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 3 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 learner traces that result in weaker leakage. These differences highlight how agent design and exploration strategy directly shape the degree of leakage.

Table 4. **Prompt recovery with different ICL inference models.** Proprietary models yield the highest scores.

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-4o	0.492	0.415	**0.770**	0.735	0.520	0.640
Qwen 2.5-32B	0.489	0.400	0.770	0.72	0.525	0.670

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

7.2. Trait Inference Across Sessions

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 compares model confidence and accuracy across categories. As shown in the left plot, occupational and demographic traits are assigned very high confidence, while the other two peak in the mid-confidence range. The accuracy distributions (right plot) further distinguish the categories: occupational traits average 0.70, 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

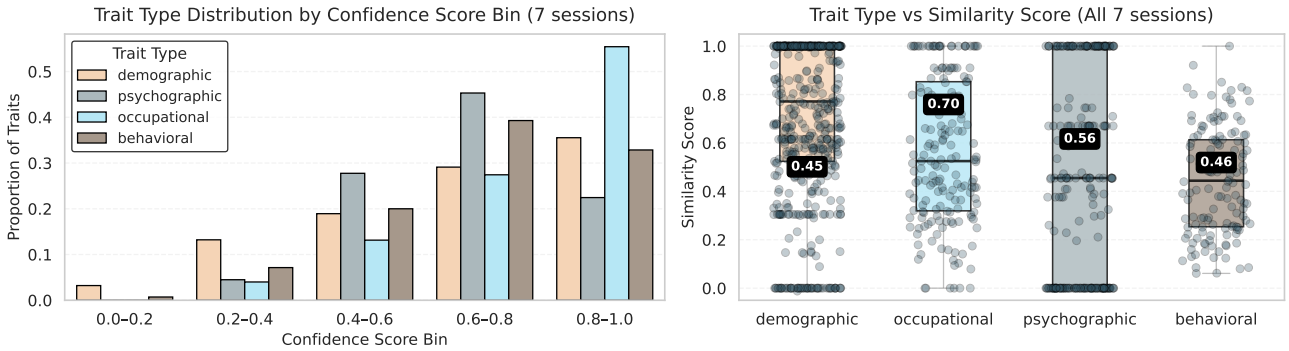

Figure 4. **Left:** Proportion of inferred traits in each confidence bin. Occupational and demographic traits are most often assigned high confidence (0.8–1.0), while the other two peak in middle ranges (0.6–0.8). **Right:** Similarity scores by trait category. The boxes, black line, numbers, and dots represent the interquartile range, median, mean, and all data points, respectively.

traits are the least exposed.

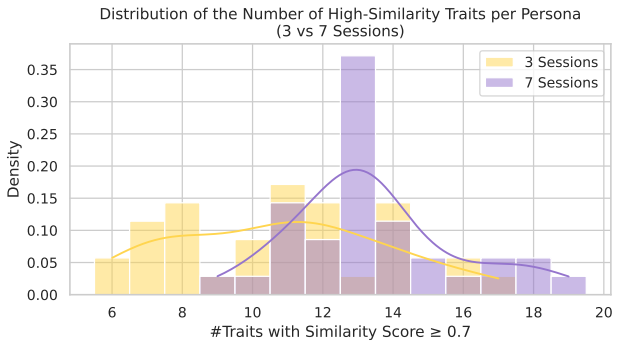

Figure 5. Distribution of traits per persona with similarity ≥ 0.7 under 3- and 7-session settings. The 7-session case shifts rightward, indicating more traits inferred at high similarity.

Different Number of Sessions. As part of our ablation study, Figure 5 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 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.

8. Defense and Realistic Deployment

8.1. Hiding via Decoy Prompts

Inspired by TrackMeNot (Nissenbaum & Daniel, 2009), for each real prompt the system generates and executes additional decoy prompts. 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

Table 5. **Effect of decoy prompts and domain shuffling on prompt leakage 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**

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 to introduce long-term behavioral divergence.

Mechanism. The defense operates in two stages. (1) **Decoy prompt generation**: an LLM produces multiple decoys 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) **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 5 shows 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.

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 in-

Table 6. **Prompt recovery under different trace visibility.** URL-level traces leak the most; 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 7. **Leakage vs. utility under varying visibility.** Reducing visibility lowers prompt recovery accuracy but leaves the report quality 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
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
2%	7.65	0.238	0.095	0.345	0.410	0.275	0.315

ternal 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.

From the adversary's perspective, this strategy effectively reduces visibility by collapsing domain diversity, yielding traces similar to those observed by weaker observers that see limited or non-discriminative domain information. This "blocking" approach therefore both mitigates leakage and illustrates how reduced observability weakens inference, reducing the adversary's ability to associate domain visits with private attributes.

Table 6 and Table 7 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. 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 for locally deployed web and research agents. We find that design choices—including domain exploration breadth, backbone model capacity, and orchestration logic—directly affect the amount of information exposed. Substituting locally deployable models does not eliminate leakage: domain-level traffic is structurally unavoidable. This shifts the attack vector from *content* to *behavior*. Whereas prompt injection and data exfiltration could be mitigated through isolation (Tsai & Bagdasarian, 2025), network-level traces cannot be trivially suppressed. At scale, such leakage enables profiling by advertisers, ISPs, or state actors, raising unique questions about how to balance agent utility with user privacy.

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 trade-offs are only partially measured. Nonetheless, our results show that decoy prompts reduce prompt recovery 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 for agents (Pham et al., 2024)) 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, as passed multi-modal content could reveal even more information (Schuster et al., 2017). Overall, progress requires viewing AI agents and the web as a coupled ecosystem rather than orthogonal components.

Conclusion. We show prompt and trait leakage from network traces of local WRAs. Even with defenses, residual exposure persists because traces are structurally generated and behaviorally rich. Our study demonstrates both the feasibility of inference 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.

Impact Statement

Our research examines how WRAs can inadvertently leak sensitive information through metadata such as domain access patterns, timing, and interaction traces. While our goal is to strengthen privacy protections for agent-based systems, we recognize the potential for harmful use if such inference techniques were deployed maliciously.

Stakeholders. We identify four stakeholder groups affected by this research: (1) End-users of WRAs, who may have personal traits, goals, or intentions inferred without their awareness. This risk is not uniform; individuals in politically sensitive contexts, marginalized communities, or high-surveillance environments may face disproportionately higher harm. (2) Developers and framework maintainers, who may need to modify or harden system architectures, logging defaults, and data retention practices in response to identified risks. (3) Organizations deploying agents (e.g., educational platforms, customer service systems), which may unknowingly violate privacy expectations or regulatory obligations if leakage is not addressed. (4) The broader public, which has a collective interest in ensuring that emerging agent ecosystems evolve in privacy-preserving and autonomy-respecting directions.

Mitigating Potential Harm. To reduce risks to end-users, we designed our study so that no real user data were collected or analyzed; all experiments use public benchmark datasets (FEDWEB13, SESSION14, DD16) and synthetic personas, ensuring that no identifiable individuals can be profiled or re-identified. To support developers and framework maintainers, we present concrete mitigation strategies, such as trace obfuscation and domain suppression, that can be incorporated into WRA architectures to reduce metadata leakage. For organizations that deploy agents in educational, corporate, or customer-service environments, we do not release any tools for live traffic monitoring, and the accompanying code is limited to offline, dataset-based replay experiments, preventing misuse in operational settings. Finally, by identifying these risks early and providing actionable defenses, the work serves broader public interests by promoting privacy-preserving WRA design, rather than end-user surveillance or behavioral profiling as agents become widely deployed.

Researcher Well-being and Responsibilities. Our research did not involve direct interaction with online communities, covert participation, or exposure to disturbing or sensitive content. However, we continuously discussed the dual-use concerns throughout the project and made design choices to minimize the potential for misuse, for example, by limiting our evaluation to public datasets and not designing tools that can be directly applied to live user traffic.

Societal Impact and Justification. As WRAs become integrated into productivity tools, classrooms, workplaces, and personal browsing workflows, metadata-based inference risks could scale rapidly and invisibly. By identifying and characterizing this attack vector under controlled conditions, we enable developers, regulators, and practitioners to recognize and address these vulnerabilities early, before they are exploited in real-world deployments. By focusing on public data, transparent methodology, and concrete mitigation guidance, we ensure that the research advances privacy-preserving agent design while minimizing the likelihood of harmful use.

References

AMD. GAIA: Generative AI is awesome (open-source project). https://github.com/amd/gaia, 2025.

Ayzenshteyn, D., Weiss, R., and Mirsky, Y. Cloak, honey, trap: Proactive defenses against LLM agents. In *USENIX Security*, 2025.

Bagdasarian, E., Yi, R., Ghalebikesabi, S., Kairouz, P., Gruteser, M., Oh, S., Balle, B., and Ramage, D. AirGapAgent: Protecting privacy-conscious conversational agents. In *CCS*, 2024.

Bird, S., Segall, I., and Lopatka, M. Replication: whywe still can't browse in peace: on the uniqueness and reidentifiability of web browsing histories. In *SOUPS*, 2020.

Cann, T. J., Dennes, B., Coan, T., O'Neill, S., and Williams, H. T. Using semantic similarity to measure the echo of strategic communications. *EPJ Data Science*, 14(1):20, 2025.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In *USENIX Security*, 2021.

Carterette, B., Kanoulas, E., Hall, M. M., and Clough, P. D. Overview of the TREC 2014 session track. In *TREC*, 2014.

Castricato, L., Lile, N., Rafailov, R., Fränken, J.-P., and Finn, C. PERSONA: A reproducible testbed for pluralistic alignment. In *COLING*, 2025.

Chiang, J. Y. F., Lee, S., Huang, J.-B., Huang, F., and Chen, Y. Why are web AI agents more vulnerable than standalone llms? a security analysis. In *ICLR Workshop*, 2025.

Demeester, T., Trieschnigg, D., Nguyen, D., and Hiemstra, D. Overview of the TREC 2013 federated web search track. In *TREC*, 2013.

Díaz, S. S., Kern, C., and Olive, K. Google's approach for secure AI agents. Technical report, Google, 2025.

Elovic, A. GPT researcher. URL https://gptr.dev. Code repository: https://github.com/assafelovic/gpt-researcher.

Gao, X. and Das, K. Customizing language model responses with contrastive in-context learning. In *AAAI*, 2024.

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

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. LoRA: Low-rank adaptation of large language models. *ICLR*, 2022.

Huang, Y., Chen, Y., Zhang, H., Li, K., Fang, M., Yang, L., Li, X., Shang, L., Xu, S., Hao, J., et al. Deep research agents: A systematic examination and roadmap. *arXiv:2506.18096*, 2025.

Hui, B., Yuan, H., Gong, N., Burlina, P., and Cao, Y. PLeak: Prompt leaking attacks against large language model applications. In *CCS*, 2024.

Jain, G. and Alwell, K. Introduction to deep research in the openai api. https://cookbook.openai.com/examples/deep_research_api/introduction_to_deep_research_api, 2025.

Jones, R., Kumar, R., Pang, B., and Tomkins, A. "i know what you did last summer" query logs and user privacy. In *CIKM*, 2007.

Kellar, M., Watters, C., and Shepherd, M. A field study characterizing web-based information-seeking tasks. *Journal of the American Society for information science and technology*, 58(7):999–1018, 2007.

Kim, H., Song, M., Na, S. H., Shin, S., and Lee, K. When LLMs go online: The emerging threat of web-enabled LLMs. In *USENIX Security*, 2025.

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

Lee, Y., Son, K., Kim, T. S., Kim, J., Chung, J. J. Y., Adar, E., and Kim, J. One vs. many: Comprehending accurate information from multiple erroneous and inconsistent ai generations. In *FAccT*, 2024.

McMahon, L. Hundreds of thousands of grok chats exposed in google results. *BBC News*, August 21 2025. URL https://www.bbc.com/news/articles/cdrkmk00jy0o.

Mudryi, M., Chaklosh, M., and Wójcik, G. The hidden dangers of browsing AI agents. *arXiv:2505.13076*, 2025.

Müller, M. and Žunič, G. Browser use: Enable ai to control your browser. https://github.com/browser-use/browser-use, 2025.

Nissenbaum, H. and Daniel, H. Trackmenot: Resisting surveillance in web search. *Lessons from the Identity Trail*, 2009.

NIST. TREC 2013 federated web track: 50 topics (fedweb13). XML file 'fedweb13_50topics.xml', 2013. URL https://trec.nist.gov/data/federated/2013/fedweb13_50topics.xml. Available from the TREC data repository, NIST.

NIST. TREC 2014 session track data. Dataset files (topics, session→topic mapping, relevance judgments), 2014. URL https://trec.nist.gov/data/session2014.html. Available from the TREC data repository, NIST.

NIST. TREC 2016 dynamic domain track data. Dataset files, overview materials, 2016. URL https://trec.nist.gov/data/domain2016.html. Available from the TREC data repository, NIST.

Oliveira, M., Yang, J., Griffiths, D., Bonnay, D., and Kulshrestha, J. Browsing behavior exposes identities on the web. *arXiv:2312.15489*, 2023.

Ollama Team. Ollama: Run large language models locally. https://ollama.com, 2025.

OpenAI. Introducing Deep Research. https://openai.com/index/introducing-deep-research/, 2025.

Perplexity AI. Introducing *Deep Research*. Perplexity AI blog, Feb 2025. URL https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research. "Deep Research performs dozens of searches, reads hundreds of sources, and delivers comprehensive reports".

Pham, D., Sheffey, J., Pham, C. M., and Houmansadr, A. ProxyGPT: Enabling anonymous queries in AI chatbots with (un)trustworthy browser proxies. *arXiv:2407.08792*, 2024.

Razavi, A., Soltangheis, M., Arabzadeh, N., Salamat, S., Zihayat, M., and Bagheri, E. Benchmarking prompt sensitivity in large language models. In *ECIR*, 2025.

Reimers, N. and Gurevych, I. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *EMNLP-IJCNLP*, 2019.

Rose, D. E. and Levinson, D. Understanding user goals in web search. In *WWW*, 2004.

Schmidgall, S. and Moor, M. AgentRxiv: Towards collaborative autonomous research. *arXiv:2503.18102*, 2025.

Schuster, R., Shmatikov, V., and Tromer, E. Beauty and the burst: Remote identification of encrypted video streams. In *USENIX Security*, 2017.

Siby, S., Juarez, M., Diaz, C., Vallina-Rodriguez, N., and Troncoso, C. Encrypted DNS–¿ privacy? a traffic analysis perspective. In *NDSS*, 2020.

Steck, H., Ekanadham, C., and Kallus, N. Is cosine-similarity of embeddings really about similarity? In *WWW*, 2024.

Stokel-Walker, C. Exclusive: Google is indexing chatgpt conversations, potentially exposing sensitive user data. *Fast Company*, July 30 2025. URL https://www.fastcompany.com/91376687/google-indexing-chatgpt-conversations.

Su, J., Shukla, A., Goel, S., and Narayanan, A. De-anonymizing web browsing data with social networks. In *WWW*, 2017.

Tavily. Tavily: The web access layer for ai agents. https://www.tavily.com/, 2025.

Tsai, L. and Bagdasarian, E. Contextual agent security: A policy for every purpose. In *HotOS*, 2025.

Wang, Z. Z., Shao, Y., Shaikh, O., Fried, D., Neubig, G., and Yang, D. How do ai agents do human work? comparing ai and human workflows across diverse occupations. *arXiv preprint arXiv:2510.22780*, 2025.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *COLM*, 2024.

xAI. Grok 3 beta — the age of reasoning agents. xAI blog, Feb 2025. URL https://x.ai/news/grok-3. Announcing Grok 3, a reasoning-optimized version of the Grok chatbot.

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

Zhan, Q., Liang, Z., Ying, Z., and Kang, D. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *ACL*, 2024.

Zheng, Y., Fu, D., Hu, X., Cai, X., Ye, L., Lu, P., and Liu, P. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv:2504.03160*, 2025.

A. 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 Recovery. With proxy dataset $D_{\text{PR}} = \{(WRA(p_j), p_j)\}$ consisting of pairs indexed by j, the adversary constructs an inference model M_{PR}. 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:

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

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 $D_{\text{TI}} = \{(WRA(\{p_i^{(j)}\}_{i=1}^{N}),\, \tau_j)\}$, the adversary instantiates an inference model M_{TI}. On a new multi-session trace $t^{1:N}$ and infers trait $\hat{\tau}$, maximizing agreement:

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

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

B. Experimental Setup Details

B.1. Dataset Details

Prompt Recovery. For all TREC datasets used in the prompt recovery task, we create **-DR** variants by inputting each original prompt with `suggested_rewriting_prompt`(Jain & Alwell, 2025), to GPT-4.1. The 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 contains 50 topics originally designed to evaluate federated web search. We concatenate the `<description>` and `<narrative>` fields to form a single prompt, and rewritten with GPT-4.1 (FEDWEB13-DR).

TREC Session 2014 Topics 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 provides 53 topics focused on interactive information retrieval in evolving domains such as medicine and cybersecurity. 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 in four categories (demographic, occupational, psychographic, and behavioral). We adopt a type-aware scoring for each trait for evaluation:

- *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. Closer ratings yield higher similarity scores.
- *Categorical traits* (e.g., gender, race) use exact match for single-token values; multi-word categories are scored using 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 ($T = 0.7$) to generate 21–35 trait-revealing prompts that explicitly or implicitly embed the traits. The produced traces serve as the input for trait inference.

Table 8. Few-shot ICL configuration for each attack task.

Task	ICL Shots	Output Format	Prompt Details
Prompt Recovery	5	Natural language query	Trace → prompt
Trait Inference	3	Structured list	Trace → trait list

B.2. Agent and LLM Backbone Choice

When browsing is executed locally, encrypted outbound traffic remains visible to an on-path adversary; when browsing is server-side (e.g., OpenAI's Operator), only connections to the provider's infrastructure are exposed. We collect traces using both open-source and proprietary WRAs.

GPT Researcher (Elovic). GPT Researcher is a research agent designed for multi-hop investigation that decomposes prompts into sub-questions, gathers evidence, and synthesizes reports. We run it in `report_type=``deep''` mode, enabling multi-turn exploration across sources. GPT Researcher employs different LLMs for sub-tasks: `FAST_LLM` for lightweight operations (e.g., GPT-4o-mini, DeepSeek-V3-0324), `SMART_LLM` for reasoning and report generation (e.g., GPT-4.1, Mistral-7B-Instruct v0.3), and `STRATEGIC_LLM` for higher-level planning (e.g., o4-mini, Mistral-Nemo-12B). While the default configuration uses GPT-4, we also evaluate local LLM combinations to assess leakage outside proprietary ecosystems.

Browser-Use (Müller & Žunič, 2025). Browser-Use is an open-source agent that automates browsing through a local browser. By default, it issues Google queries via `search_web`, which frequently triggers reCAPTCHA; we therefore configure it to use Bing with automatic fallback. 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 construct ordered traces. This setup makes Browser-Use suitable for evaluating how web agents' lightweight browsing patterns leak information.

AutoGen (Wu et al., 2024). We configure AutoGen as a two-agent system consisting of an `AssistantAgent` and a `MultimodalWebSurfer`. Compared to Browser-Use, AutoGen is faster and more controllable, making 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 recovery and Gemini 2.0 Flash for trait inference.

OpenAI Deep Research API. As a proprietary baseline, we evaluate the OpenAI Deep Research API. Unlike the interactive ChatGPT Deep Research interface, 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 that would have been normally triggered by user interaction. Because browsing occurs remotely, only provider-level 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-oriented (GPT Researcher) and web-oriented (Browser-Use, AutoGen) agents to capture variation in browsing behavior 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 may expect stronger privacy guarantees. The Deep Research API serves as a commercial baseline. We deliberately exclude larger open-source models (e.g., LLaMA 3.1 70B, DeepSeek-R1), as they require substantial computational resources and often bypass web search by relying on internal knowledge, making them less suitable for our browsing-driven evaluation. Together, these choices ensure coverage of attacker-visible metadata across diverse agents and realistic local deployment settings.

B.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 8.

Prompt Recovery. 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 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.

C. Additional Prompt Recovery Results

Table 9. **Prompt recovery performance depending on 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**

Effect of Example Count. Table 9 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.

Table 10. **Prompt recovery on trace by GPT-Researcher with different backbones across datasets.** Local backbones leak less than GPT-4 traces 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

Effect 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 10, GPT-4 traces yield higher recovery 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 11. **Prompt recovery: 5-shot vs. contrastive ICL.**

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 ICL Configurations. Contrastive ICL(Gao & Das, 2024) showed that adding negative examples to the in-context setup can improve performance, even when negatives must be synthesized by the LLM. In our baseline, each 5-shot prompt

includes five (trace, prompt) pairs followed by a new trace for prediction. 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 are regenerated until their SBERT similarity to the original prompt falls below a threshold, ensuring sufficient dissimilarity.

As shown in Table 11, 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 12. **Prompt recovery: example selection strategies.**

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

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`[1] sentence transformer, and the training examples with the highest cosine similarity to the test trace are chosen. As shown in Table 12, embedding-based selection does not improve prompt recovery accuracy and, in some metrics, performs slightly worse, suggesting that random sampling, by providing more diverse examples, is the more effective strategy.

Table 13. **Prompt recovery: different example orderings**

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

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 13, ordering has little effect: scores remain close across all metrics. This suggests that once relevant examples are chosen, their sequence contributes minimally to recovery quality.

D. Additional Trait Recovery Results

Figure 6 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. It suggest that demographic and occupational traits are most vulnerable to leakage, while lifestyle and personality attributes remain less exposed but at nontrivial levels.

Figure 7 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 (Cann et al., 2025). 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.

Table 14 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.

[1] Fine-tuned on 1B sentence pairs with a contrastive objective, this model maps each trace into a 384-dimensional vector space.

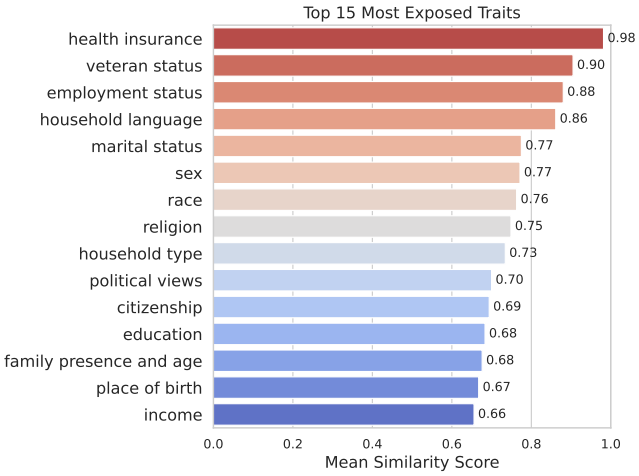

Figure 6. **Top 15 traits with highest exposure risk.** Inference risk is measured by mean similarity scores across all personas and sessions. Higher scores indicate stronger inference.

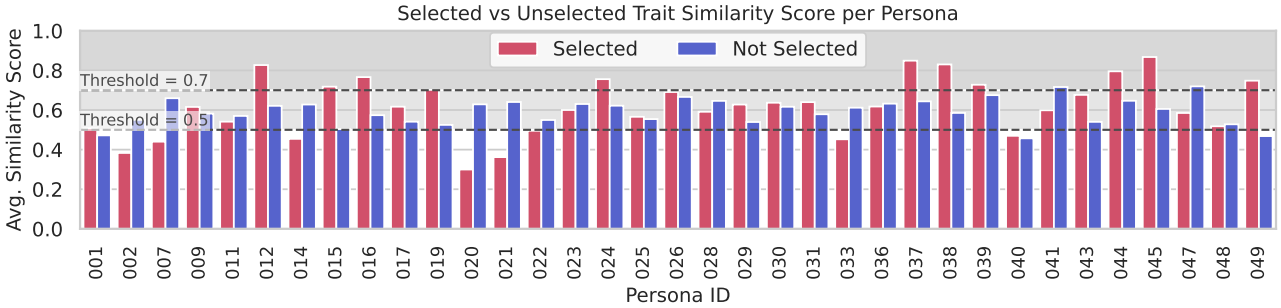

Figure 7. **Average similarity scores per persona for selected (red) and unselected (blue) traits.** Dashed lines at 0.5 and 0.7 indicate moderate and high-risk leakage thresholds. Many personas have selected traits exceeding 0.7, while unselected traits frequently remain above 0.5, reflecting moderate but non-negligible exposure.

E. Additional Virtual Persona Defense Results

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 15 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.

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.

F. 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 (Carterette et al., 2014) 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: functional equivalence, domain type, semantic alignment, and entity granularity. This structured format ensures consistent

Table 14. **Average similarity scores across trait categories.** 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%

Table 15. **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%

evaluation across diverse prompt reconstruction.

G. 5-Shot ICL Prompt Templates (Prompt Recovery)

Figure 13 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.

H. Prompt Template for Generating Persona-Based Queries

Figure 15 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.

I. 3-Shot ICL Prompt Templates (Trait Inference)

Figure 16 illustrates the 3-shot ICL prompt used for inferring user traits 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 Figure 16. In the actual prompt used during inference, the full list of traits for `persona_003` is included.

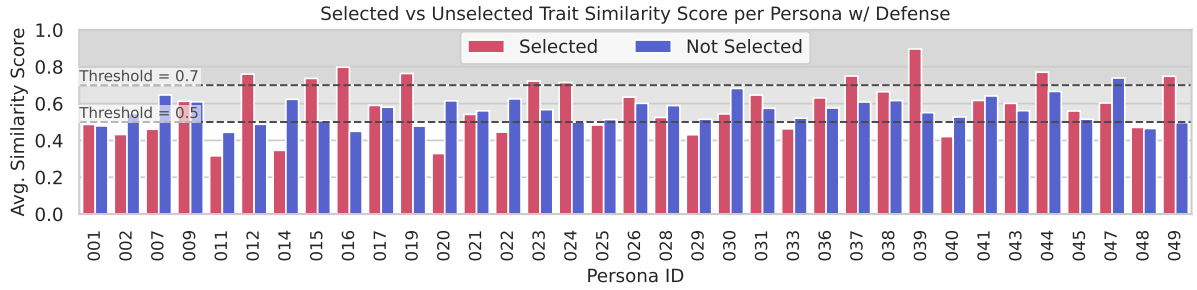

Figure 8. Average inference accuracy for selected vs. unselected traits per persona, showing generally lower accuracy than Figure 7 with the proposed virtual persona defense applied.

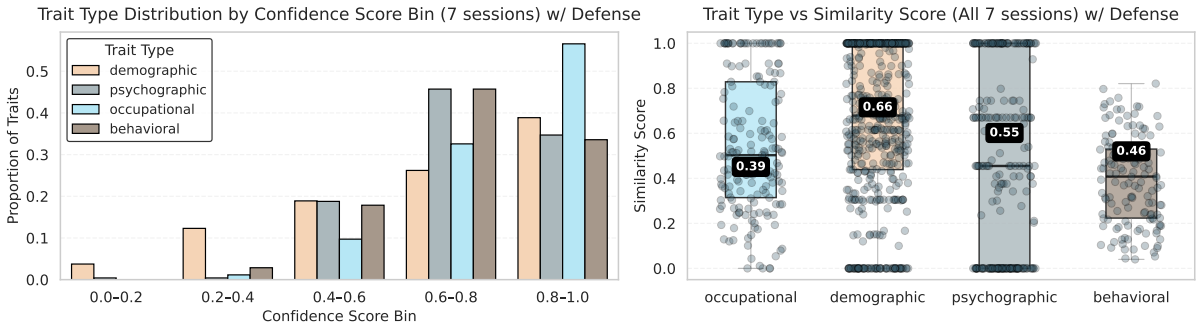

Figure 9. Trait-level breakdown of inference performance with the virtual persona defense. Confidence distributions remain similar to Figure 4, but average similarity scores are lowered across all categories.

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, 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 recovery performance.

5-shot ICL Prompt Template (Prompt Recovery) (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 Recovery)

```
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 13. 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 Recovery) (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 Recovery) (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 Recovery) (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 14. 5-shot contrastive ICL template for reconstructing user prompts from domain traces, with preferred vs. less-preferred queries to guide the model.

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 15. Prompt template used for generating trait-revealing web queries across sessions.

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.