Redefining Website Fingerprinting Attacks with Multi-Agent LLMs

Chuxu Song
Rutgers University
cs1346@scarletmail.rutgers.edu

Dheekshith Dev Manohar Mekala
Rutgers University
dm1653@scarletmail.rutgers.edu

Hao Wang
Rutgers University
hw488@cs.rutgers.edu

Richard P. Martin
Rutgers University
rmartin@rutgers.edu

Abstract

Website Fingerprinting (WFP) attacks exploit side-channel leakage from encrypted traffic to infer visited websites. While deep learning has driven major accuracy gains in controlled settings, we show that such progress does not transfer to modern, highly interactive websites. Traffic generated by scripted browser automation—the de facto standard for WFP datasets—fails to capture the behavioral diversity of real users, leading to severe overestimation of attack effectiveness. We introduce a novel data generation pipeline that uses a multi-agent system powered by large language models (LLMs) to simulate persona-driven browsing behavior. These agents coordinate high-level decision making with low-level browser interaction, producing semantically grounded traffic traces at 3–5× lower cost than human collection. Using traffic collected from 20 modern websites browsed by 30 users, we evaluate nine state-of-the-art WFP models across human, scripted, and LLM-generated datasets. All models achieve under 10% accuracy when trained only on scripted traffic and tested on human traffic. In contrast, training with LLM-generated traces boosts accuracy into the 80% range, demonstrating strong transfer to real browsing patterns. We further conduct an open-world evaluation with 40 unseen websites and find that augmenting human data with LLM-generated traces reduces accuracy degradation by 50% compared to training on human data alone. Our results indicate that the core bottleneck in modern WFP is data realism rather than model design, and that scalable, behavior-rich synthetic traffic is essential for accurately assessing privacy risks under contemporary web conditions.

Keywords

website fingerprinting, LLM, Computer agent, Privacy

1 Introduction

Website Fingerprinting attacks [3, 8, 11, 18–20, 23, 29, 33, 36] exploit side-channel information from encrypted traffic to infer which website a user is visiting. Even though content and destination IPs are hidden, traffic patterns such as packet sizes, directions (incoming vs. outgoing), burst lengths, and inter-arrival times can serve as discriminative features for identifying web activity. Early WFP techniques used traditional machine learning classifiers on hand-engineered features [5, 24], but recent work has demonstrated that deep learning models—such as CNNs and LSTMs—can achieve significantly higher accuracy under controlled conditions [2, 16, 28]. These models typically operate on network traces corresponding to full webpage loads and achieve classification accuracy higher than 90% in closed-world settings.

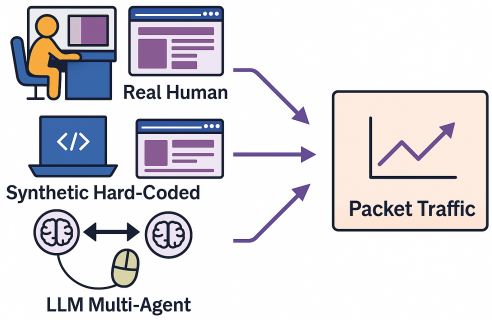

Figure 1: We collect WFP traffic dataset generated by real human users, traditional scripted robots, and our LLM-based multi-agent system.

However, most of these techniques implicitly assume a static, page-oriented web browsing model. In this model, users navigate between clearly delineated web pages, and each page load corresponds to a discrete unit of classification. Although researchers have introduced enhancements such as webpage segmentation [21], early-stage prediction [7, 10, 31], and online learning [8], these methods fail to consider that many websites have evolved well past the model of a collection of static pages. Today's websites are dynamic, interactive, and highly personalized. Single Page Applications (SPAs) and streaming platforms dominate user traffic, where content is updated dynamically without changing URLs. On platforms like YouTube, Reddit, or Amazon, users may stay on a single webpage for extended periods, interacting with embedded media players, infinite-scroll feeds, or personalized content panels. These interactions create rich and varied traffic that breaks the clean boundaries required for traditional WFP analysis.

We revisit the WFP threat model in the context of modern, long-lived, and interaction-rich browsing sessions. We begin by addressing a foundational need in WFP research: realistic, high-quality traffic data that accurately reflects human behavior. We hired 30

Proceedings on Privacy Enhancing Technologies 2026(3), 688–702

https://doi.org/10.56553/popets-2026-0101

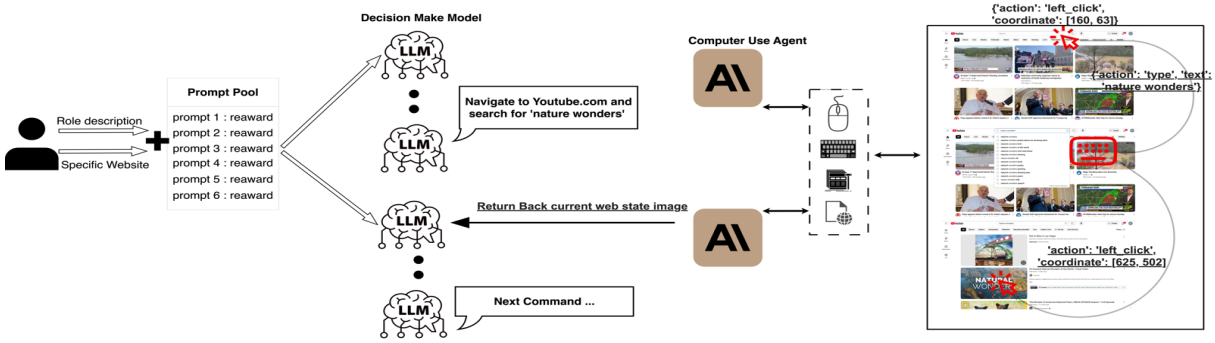

Figure 2: Overview of our proposed LLM-driven multi-agent framework for generating realistic website fingerprinting (WFP) training data. Prompt-conditioned language models simulate goal-oriented user commands, which are executed in a browser environment. The system forms a closed loop by feeding back updated visual states, enabling scalable, behavior-rich, and semantically grounded interaction traces.

participants to browse 20 modern websites—including several SPAs and content-dense platforms—on controlled desktop environments. Participants engaged with the sites freely for over 100 hours in total, generating a real-world dataset of encrypted traffic traces. To simulate existing synthetic approaches, we collected network traffic using a scripted browser, which followed same-domain links using a breadth-first crawl without performing any meaningful interaction.

Our analysis reveals a significant representativeness gap between traffic generated by synthetic scripted browsers and human traffic that impacts WFP classification accuracy across many models. For example, over 60% of human sessions involved spending more than 10 consecutive minutes on a single webpage, with extensive in-page activity such as scrolling, hovering, typing, or clicking dynamically loaded components. These behaviors produce long, variable-length traces with complex temporal dynamics that defy traditional WFP segmentation methods.

To simulate realistic attack conditions, we assume the attacker has no access to precise session start points or ground-truth webpage splits. We evaluated eight state-of-the-art WFP models across three training/testing regimes: (1) both training and testing scripted traffic, (2) training on scripted traffic and testing on human traffic, and (3) training on human traffic and testing on leave-one-out human traffic where we test on a human's traffic not in the training set. Results show that while human-to-human training achieves up to 75% accuracy, scripted-to-human adaptation yields less than 15% accuracy, highlighting the poor generalization of current synthetic traffic generation methods when directly programming a browser with scripts.

Despite the value of human-generated traffic, collecting and scaling such a training corpus is prohibitively expensive and raises serious privacy concerns. Recruiting diverse participants, compensating them fairly, and enforcing consistent browsing protocols requires significant manual effort. Moreover, due to ethical limitations, it's difficult to capture sensitive behavioral patterns or long-term personalized activity.

To improve WFP classification accuracy in the modern Internet, we introduce two modeling shifts. First, we abandon the view of the web as a set of discrete pages. Rather than try to classify web pages, we classify entire web sites; that is, the output of the models is an entire web site domain rather than any single page. This focus is particularly critical for modern SPAs and streaming platforms, where in-page interactions dominate and page transitions can be subtle or absent, making domain-level inference both more realistic and more operationally relevant for an attacker. Second, rather than rely on complex and error-prone page-boundary inference, we adopt a *boundless* (boundary-free) formulation: the attacker classifies a fixed-length segment of **5,000 consecutive packets** sampled from a long-lived browsing session, without assuming access to session starts, page-load events, or URL transitions. This mirrors realistic network vantage points (e.g., gateways, middleboxes, or endpoint monitors) that observe only encrypted packet metadata (size, direction, timing).

To address the question of how to obtain representative traffic at scale, we propose a novel traffic generation pipeline based on a multi-agent architecture [15, 17, 22, 32, 35] powered by Large Language Models (LLMs). An overview of our architecture is illustrated in Figure 2, which showcases the interaction between two key components: a high-level *Decision-Making Agent* and a low-level *Computer-Using Agent* (CUA). The *Decision-Making Agent* is responsible for generating semantically meaningful, context-aware navigation instructions based on the current webpage state, while the *Computer-Using Agent (CUA)* performs fine-grained browser operations—including clicking, scrolling, typing, and DOM navigation—based solely on high-level natural language commands from the Decision-Making Agent.

The **Decision-Making Agent** leverages multimodal LLMs such as Claude or GPT to produce concise, screen-grounded commands conditioned on user personas. Given a screenshot and site context (provided by the CUA), it interprets page content and generates navigational intents like "scroll for more articles" or "click the first product," reflecting realistic and goal-oriented user behavior.

The **Computer-Using Agent (CUA)** executes these commands inside a real browser using Claude's Computer Use API, which unifies perception and action within a single LLM interface. Internally, the CUA parses the visual page structure, selects UI elements based

on semantics, and emits interaction plans in JSON format compatible with browser instrumentation. This enables high-fidelity, context-aware GUI actions without relying on hand-coded scripts.

By decoupling decision-making from execution, our multi-agent pipeline enables scalable, cross-site browsing simulation while preserving semantic intent and behavioral diversity. In the remainder of this paper, we (i) characterize the representativeness gap between scripted and human traffic on modern websites, (ii) introduce our multi-agent LLM pipeline with online prompt optimization, and (iii) show that LLM-simulated traces improve cross-user and open-world generalization under the boundless WFP setting.

We integrate this synthetic traffic into WFP training workflows and observe substantial improvements in generalization. When models are trained jointly on human and LLM-generated traces, cross-user accuracy improves by over 50%. Even when trained solely on LLM traffic, models achieve up to 3× higher accuracy on human traces than those trained on classical synthetic datasets. These results demonstrate that high-fidelity LLM-driven simulation—guided by semantic intent and behavioral diversity—can significantly narrow the realism gap in website fingerprinting tasks. Beyond closed-world settings, we further evaluate Website Fingerprinting in an open-world environment, where the attacker must distinguish a monitored set of 10 websites from traffic drawn from 40 additional sites treated as an aggregated background class. Our findings show that while Human-only training leads to a substantial degradation (5–8% accuracy drop), Human+Claude training mitigates this effect (only 2–3% drop), reinforcing that LLM-augmented traces help sustain robustness under more realistic adversarial conditions.

In summary, this paper makes the following contributions:

- We collect and characterize a large-scale dataset of encrypted web traffic from 30 human participants browsing 20 dynamic, modern websites over 100 hours of interaction.
- We demonstrate that state-of-the-art WFP models severely overestimate attack effectiveness when trained only on scripted browsers, especially in the context of modern single-page or streaming websites.
- We introduce a scalable, LLM-based multi-agent framework for generating realistic synthetic network traffic that significantly improves WFP model generalization for diverse human behavior.
- We construct a complementary dataset using LLM agents covering 10 websites and including 10,000 training samples, offering a scalable and cost-effective alternative to traditional traffic collection.
- We show that in an open-world evaluation, Human+Claude training reduces accuracy degradation to 2–3%, compared to 5–8% with Human-only training, highlighting the role of LLM-simulated browsing in sustaining robustness under adversarially realistic conditions.

Paper Organization. Section 2 motivates the need to revisit Website Fingerprinting (WFP) under modern web conditions. Section 3 presents our updated threat model. Section 4 details our data collection pipeline and LLM-based simulation framework. Section 5 evaluates model generalization across various training sources and data scales. Section 6 reviews prior work. Section 8 concludes with insights and future directions.

2 Background and Motivation

Website Fingerprinting (WFP) is a passive traffic analysis technique that attempts to infer the websites a user is visiting by analyzing encrypted network metadata such as packet sizes, directions, and inter-packet delays. Prior research has demonstrated strong performance using deep learning-based classifiers under closed-world assumptions, achieving over 90% accuracy on datasets with clear session boundaries and static website structures [16, 24, 28, 31].

However, these results rely on several simplifying assumptions: browsing traces correspond to full page loads; session start and end points are known; and user behavior is deterministic or scripted. These assumptions are increasingly incompatible with modern web usage patterns.

Today's websites—such as YouTube, Amazon, TikTok, and Reddit—adopt single-page application (SPA) architectures. Users often remain on a single page while triggering asynchronous content updates through scrolling, searching, or media playback. These behaviors generate long, unsegmented, and high-entropy traffic streams, which traditional WFP models were never designed to handle. Compounding this, user interactions vary widely in intent and style, introducing significant intra-site and intra-user diversity.

In our user study (Section 4), we observed that participants frequently spent over 10 minutes continuously browsing within the same website context, without changing the page or triggering explicit reload. In such sessions, the assumption of "webpage-level splitting" becomes invalid: there are no clear structural boundaries between tasks, actions, or content states. Furthermore, many modern websites adopt proactive content preloading, where elements such as video feeds, product panels, and comment sections are fetched in the background in anticipation of user engagement. This design strategy obfuscates the semantic start point of user behavior, making it exceedingly difficult for an attacker to detect the true beginning of a session—even if segmentation were possible.

To better understand how these behavioral and architectural shifts affect WFP, we benchmarked nine state-of-the-art classifiers across four different dataset conditions. Models trained and tested on traditional synthetic traffic—collected via scripted Puppeteer crawlers—achieve up to 98% accuracy, consistent with prior literature. However, when these same models are tested on traffic generated by real users interacting with dynamic websites, accuracy drops sharply to below 10%, even for the strongest models. This illustrates that existing synthetic datasets fail to capture the interaction complexity necessary for evaluating WFP in modern contexts.

Even within the real human dataset, classifier performance varies greatly depending on how the training data is partitioned. When we train on multiple participants and test on a held-out individual (leave-one-user-out), accuracy declines by over 30%, even under closed-world assumptions. This gap demonstrates the importance of modeling user-specific behavior and highlights that dataset diversity—not just scale—is essential for building robust WFP systems.

Collecting real user traffic to meet these requirements is expensive. Our experiments show that acquiring 1GB of authentic human browsing data costs approximately $35, primarily due to participant compensation and lab infrastructure. In comparison,

our LLM-based multi-agent framework can generate 1GB of synthetic—but human-like—traffic for just $10 using commercial APIs. We anticipate this cost gap will widen further as open-source LLMs and lower compute costs continue making high-quality synthetic data generation more feasible.

2.1 Key Observations and Motivation for Multi-Agent LLM-Based Generation

In re-evaluating Website Fingerprinting (WFP) attacks under modern conditions using real user traces and stricter constraints (as formalized in Section 3), we identified three critical challenges that limit the generalizability of current approaches:

Observation 1: WFP models suffer significant accuracy degradation under realistic conditions. Even in a closed-world setting, model accuracy dropped from previously reported 95%+ to around 80% when tested on unsegmented, interaction-rich traffic segments—highlighting the brittleness of prior assumptions such as clean session boundaries.

Observation 2: Model performance is highly sensitive to dataset scale and diversity. Modern websites span heterogeneous services, components, and user flows. We find that classification performance improves significantly with larger and more varied training sets, even within the same domain—underscoring the need for behavioral diversity.

Observation 3: User-specific behaviors hinder generalization. When evaluating on users excluded from the training set, model accuracy dropped sharply. This indicates that models often overfit to individual browsing styles, failing to extract site-invariant features.

These limitations collectively reveal a central bottleneck: achieving robust WFP in modern environments requires access to large-scale, semantically rich, and behaviorally diverse datasets. Collecting such traffic from real users is both costly and ethically fraught, due to privacy risks and limited scalability.

To address this, we introduce a scalable multi-agent data generation pipeline powered by Large Language Models (LLMs). By prompting commercial LLMs such as Claude with user personas and site-specific goals, and executing their outputs through a structured browser control API, we synthesize realistic traffic patterns that closely mirror human browsing behavior. Our approach balances semantic fidelity and controllability, enabling fine-grained simulation of diverse user interactions at a low cost. As we demonstrate in later sections, LLM-generated data not only improves generalization to unseen users but also outperforms traditional synthetic traffic by a wide margin, providing a viable and scalable alternative for future WFP research and evaluation.

3 Threat Model

We consider a passive network-level attacker attempting to identify the websites visited by a user based solely on encrypted traffic metadata. This threat model is rooted in traditional WFP literature [16, 24], but we adopt a more constrained and realistic version, reflecting what a modern adversary could plausibly observe.

3.1 Attacker Capabilities

The attacker monitors network flows from a position such as an ISP, Tor entry node, or VPN exit. They have access to:

- Packet timestamp, size, and direction (in/outbound),
- Encrypted transport-layer traffic,
- Long-lived connections without application-layer insight.

Critically, unlike most prior work, the attacker in our setting does **not** observe:

- Session start or end events,
- Page load transitions or URLs,
- Any browser-level metadata or DOM structure.

The attacker must operate on *arbitrary continuous segments* of encrypted packet traces, which may begin or end at any point within a browsing session. This assumption mirrors practical conditions where traffic segmentation is unavailable or noisy.

3.2 Attacker Goal

Given a packet segment T, the attacker's classifier $f(\cdot)$ will predict the corresponding label $\hat{w} \in W$, where W is a closed set of known monitored websites. The segment may reflect part of a page load, a continuous user interaction (e.g., scrolling), or even a mixture of unrelated background requests.

3.3 Differences from Prior Models

Our threat model introduces several key differences compared to prior work [6, 8, 10, 14, 28, 29], making it more realistic and practical:

- **No clean session boundaries:** Realistic traces are unsegmented and may contain noise or overlap.
- **User behavior variability:** Per-user differences in interaction style, device usage, and attention span lead to substantial intra-class diversity.
- **Modern dynamic content:** Websites dynamically change in response to scrolling, login state, personalization, etc., increasing non-determinism in packet flows.

Our threat model is designed to better reflect practical attacker conditions where ideal segmentation and deterministic traffic no longer hold. Prior defenses such as Walkie-Talkie [34], TrafficSliver [9], and ZeroDelay [13] rely on synchronous traffic shaping or burst modeling under the assumption of clean page boundaries and uniform session alignment. However, our model operates in a streaming scenario where segments are untethered from specific user actions, limiting the effectiveness of such defenses.

Additionally, our setting aligns with recent critiques of traditional evaluation assumptions [4, 20, 26], and is intentionally constructed to reveal the fragility of models trained on idealized data. By adopting this more realistic threat model, we demonstrate that modern defenses should be reassessed under unconstrained, long-lived browsing sessions that better match real-world conditions.

4 Methodology

To rigorously evaluate website fingerprinting (WFP) under modern browsing conditions, we designed and collected a suite of traffic

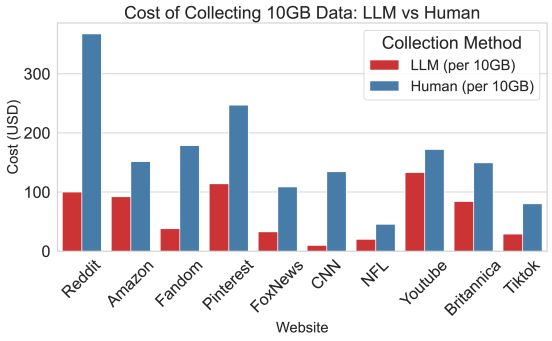

Figure 3: Cost comparison for collecting 10GB of web browsing data across ten websites using human participants versus LLM-based agents. LLM simulation offers up to 3–5× cost reduction while maintaining site-specific coverage.

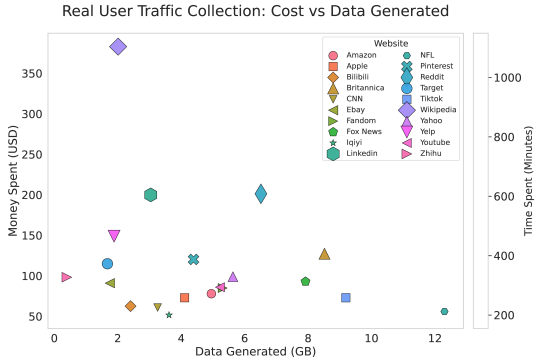

Figure 4: Traffic collection cost and yield per website using real participants. Bubble size and color reflect time spent per site. Some websites (e.g., *Wikipedia*, *LinkedIn*) incurred high cost but produced relatively little data, underscoring the inefficiency of manual collection at scale.

datasets[1] from three distinct sources: (1) real human users interacting with websites in a controlled lab environment, (2) scripted browsers that follow predefined sequences, and (3) a novel LLM-based multi-agent system capable of generating diverse and semantically meaningful interactions. Each source introduces different tradeoffs in realism, cost, and scalability. Across all sources, we capture full-fidelity packet traces (pcap) and preprocess them into non-overlapping 5,000-packet segments labeled at the website/domain level, matching our boundless, segment-based formulation in Section 3. We will open-source the LLM generation pipeline (scripts/configuration) and the prompt/persona assets, while raw pcap traces will be shared via controlled access to comply with IRB requirements.

This section presents the full methodology behind our data collection pipeline. We begin by describing our human participant study, including recruitment, task design, and privacy safeguards. We then detail the construction of a high-fidelity Puppeteer-based crawler for baseline automation. Finally, we introduce our LLM-driven synthetic framework with reward-based prompt optimization, which balances behavior diversity and command accuracy at scale. Throughout, we clarify the cost structure of human collection versus LLM-based generation, where the dominant marginal cost for LLM-Sim is API usage.

4.1 Real Human Browsing Data Collection

To construct a realistic dataset that captures genuine human interaction patterns on modern websites, we recruited and compensated 30 participants to browse 20 popular websites in a controlled laboratory setting. Each participant was paid $20 per hour, consistent with common IRB compensation practices and above typical local minimum-wage baselines, and collectively, the study yielded over 100 hours of labeled browsing activity. Across all sessions, we gathered a total of **96.1 GB** of packet-level network traces.

The selected websites span a broad range of categories—including e-commerce (e.g., Amazon, eBay, Target), social media and video platforms (e.g., Reddit, YouTube, TikTok), and information-rich portals (e.g., CNN, Wikipedia, Britannica). A full list of target sites and task structures is included in section 5.2.

Before participation, each subject completed a demographic and behavioral background questionnaire capturing attributes such as age, gender, native language, and general browsing habits. Participants also rated their familiarity with each target website using a 5-point Likert scale. These profiles enabled us to analyze how prior knowledge and intent shape browsing behavior across different user types.

Each participant was assigned an isolated desktop workstation on a university-managed wireless network. They were instructed to use a standard Chromium browser without any extensions or personalization. For each website, a small set of mandatory tasks was provided to ensure consistent functional coverage (e.g., "search for a product" or "read a top article"), after which users were free to explore the site without restriction. Most sessions per website lasted between 20 and 40 minutes.

All network traffic was captured using `tcpdump`, generating full-fidelity packet traces in pcap format. These traces record packet size, direction, and timestamp—but no application-layer content—making them ideal for website fingerprinting analysis while preserving user privacy.

As shown in Figure 4, we observed substantial variance in data yield and collection cost across websites. For example, sessions on Wikipedia or LinkedIn often required over an hour of participant time but produced less than 3 GB of data, whereas NFL and Fandom yielded high-volume flows with relatively little user effort. This heterogeneity reflects differences in page architecture, media density, and engagement dynamics. The average cost per gigabyte across all 20 websites was approximately $35, with certain domains exceeding $100/GB—highlighting the scalability and economic limits of manual dataset creation.

[1]https://drive.google.com/file/d/1KZNMZSUdLLBA5hQ0KNa165eI5zOADnkf/view?usp=sharing

4.1.1 Ethical Considerations and Privacy Protections. Given that our study involved human subjects and generated behavioral network traces, we took comprehensive steps to ensure ethical compliance and participant privacy. Our protocol was reviewed and approved by the Institutional Review Board (IRB) of our university. All participants gave their informed consent prior to the study and were explicitly instructed not to submit personal information or interact with external websites during the sessions (i.e., no concurrent browsing across multiple websites while collecting a labeled session).

All traffic traces were encrypted using AES-256 and securely uploaded to a university-managed research cloud storage system with strict access control. Metadata such as website domain, session duration, and task type were stored separately using the same hash ID, allowing contextual analysis without a direct identity link. All questionnaire responses were aggregated, and any direct or indirect personal identifiers were either removed or anonymized before conducting the final analyses.

4.2 Scripted Browsers Traffic Collection

To establish a programmatic baseline for browsing behavior, we implemented a scripted crawler following the structure of SimulatedHuman [30]. This crawler was designed to automate common surface-level actions across websites using hand-coded interaction flows. While it introduces basic variability—such as random delays, selecting links, or toggling tabs—its behavior is ultimately limited to deterministic scripts and lacks the adaptability seen in human or LLM-based browsing.

We built the crawler using `puppeteer-extra`, a headless Chromium automation framework equipped with stealth plugins to avoid bot detection. For each target site, we defined a set of site-specific scripts that execute static sequences: e.g., selecting a product, playing a video, or navigating to the next page. Some scripts simulate user delays or random branching, but none dynamically adapt based on page content or state changes. Compared to LLM agents, which make context-aware decisions based on semantic goals and visual state, scripted crawlers operate with shallow logic and do not adjust to personalized content or asynchronous interface events.

All sessions were run under the same devices and network conditions used in our human experiments to ensure a fair comparison. Over multiple days, we collected over **800 GB** of packet-level traces from 20 major websites using `tcpdump` (approximately 40 GB per website on average). While this approach improves over classical crawlers in realism and scale, it remains inherently constrained by static control logic. Importantly, evaluating on scripted data alone (e.g., Scripted_TrainTest) can yield an inflated "upper bound" and may inadvertently grant attackers an unfair advantage, thereby overstating the real-world threat; many prior WFP pipelines relying heavily on scripted or boundary-aligned collection can exhibit this effect. As shown in later sections, models trained on this data fail to generalize to real user behavior, reinforcing the need for deeper, semantically grounded simulation frameworks.

4.3 LLM Multi-Agent Data Generation with Online Prompt Optimization

Deploying and training a fully self-hosted multi-agent LLM system[2] with multi-modal input and online feedback would typically require significant GPU resources and infrastructure. To circumvent this, we designed a lightweight but practical training and deployment framework built on top of commercial Large Language Model (LLM) APIs—in our case, the Claude API. This allowed us to integrate high-quality language understanding and generation capabilities into a multi-agent system without the need to fine-tune a full LLM model.

Multi-Agent System Configuration. Our multi-agent system is composed of two modular LLM-driven components tailored for semantic goal planning and browser-level execution. The **Decision-Making Agent** is powered by Claude 3.7 Sonnet—a multimodal model capable of interpreting both structured text and visual inputs. At each decision point, it analyzes the current page content and outputs a concise instruction grounded in what is visible. To enhance determinism and reduce hallucination, we fix the generation temperature to 0.3, ensuring stable command quality. Personas and goal conditioning are specified *only* in the Decision-Making Agent system prompt: we enumerate 15 personas and encourage the agent to diversify into additional human-like personas over time.

The **Computer-Using Agent (CUA)** runs inside a lightweight Docker container using Anthropic's official execution environment. It interacts with a live Chromium browser by parsing the DOM and autonomously executing fine-grained GUI actions such as scrolling, clicking, and typing. This Docker-based setup provides secure isolation and simplifies deployment across machines, while also enabling scalability and multi-browser extensibility in future extensions. The CUA does not model personas; it strictly executes the Decision Agent's commands and reports outcomes and browser state.

Together, these agents operate in a closed-loop fashion: one agent formulates high-level goals, while the other translates them into executable actions. This architecture decouples semantic planning from browser control, offering robustness and transferability across websites and user roles.

During each interaction cycle, the Decision Agent receives four inputs:

(1) A prompt template encoding a persona and goal;
(2) The name of the target website;
(3) A full-page screenshot in base64 format;
(4) The role or user context (e.g., first-time user, returning visitor).

Based on these, the LLM (Claude) returns a concise, executable natural language command. The command is then interpreted by the CUA and performed in the browser. After execution, success/-failure is recorded along with whether the action was redundant or led to a crash. All interactions are logged to a structured JSONL file with command, reward, prompt ID, and screenshot hash. In our main LLM-Sim collection used in this paper, we target a 10-site subset and collect 1,000 labeled 5,000-packet segments per site under this pipeline.

Prompt Design Considerations. A central challenge in our multi-agent pipeline lies in ensuring that the high-level instructions

[2]https://drive.google.com/file/d/1Wf8X2Ncp80wbAuLm2W7GNt1tgh4jnN4f/view?usp=sharing

produced by the Decision Agent are both *semantically valid* and *executable* by the downstream CUA. Large Language Models (LLMs), particularly when acting as generative planners, are prone to verbosity, hallucination, and nondeterministic outputs—especially under high sampling temperatures or vague prompting. In our context, these issues translate into commands that may be overly long, ambiguous, or mismatched with the visual state observed by the CUA.

For instance, an under-constrained Decision Agent might generate speculative, multi-sentence descriptions rather than screen-grounded commands (e.g., "You should probably compare different headphone brands and read some reviews first"), making it difficult for the CUA to interpret and execute the intended action. Worse, it may hallucinate interface elements that are not currently visible on the page, resulting in execution crashes.

To address this, we design instruction prompts that explicitly constrain generation to a single, executable, and screen-aligned command. Typical prompt examples include: "Generate **one realistic command to execute** based strictly on what is currently visible in the browser," or "**Avoid explanation**; only output the next user action." These constraints improve stability, filter verbosity, and increase the alignment between the Decision Agent's intent and the CUA's capabilities. To further enable zero-shot grounding toward real-user behavior, we include common action primitives and transition patterns distilled from real sessions as in-context examples in the Decision Agent system prompt. Moreover, by refining these prompts through continual reward-based feedback, we are able to maintain high interaction fidelity and consistency over long multi-step trajectories.

Online Prompt Optimization. The quality of generated commands—especially their validity and diversity—directly affects the resulting synthetic dataset. To improve this, we apply an online prompt selection and evolution framework based on a classical multi-armed bandit formulation. Each prompt template p_k is treated as a separate arm, and its expected reward μ_k is learned online through interaction. In our experiments, we initialize the prompt pool with K=50 seeded templates (persona slot + goal slot + single-step constraints). We use an ϵ-greedy policy for balancing exploration and exploitation: with probability ϵ a prompt is randomly sampled; otherwise, the one with the highest estimated reward is chosen.

Continuous Reward Formulation. Let $s_t \in \{0, 1\}$ indicate whether the command executed successfully, and let $c_t \in \mathbb{N}$ be the number of polling rounds required to complete execution. We define the continuous crash penalty k_t based on execution time as:

$$k_t = \min\left(1, \frac{c_t}{C_{\max}}\right), \tag{1}$$

where $C_{\max}$ is the maximum allowed polling steps (e.g., 120). Let $d_t \in [0, 0.2]$ denote a diversity bonus computed from the dissimilarity between the current command and recent command history. The total reward is then given by:

$$R_t = s_t + d_t - k_t, \tag{2}$$

which smoothly penalizes longer execution durations and rewards novel and successful commands. The average reward for each prompt is updated online:

$$\mu_k \leftarrow \mu_k + \frac{1}{n_k}(R_t - \mu_k), \tag{3}$$

where n_k is the number of times p_k has been selected. To prevent convergence to a narrow set of overfit prompts, we implement a prompt evolution mechanism: every T rounds (e.g., $T = 10$), we prune the bottom 20% of low-reward prompts and replace them with new rewrites generated by Claude. This allows the prompt pool to continuously adapt and diversify over time.

Algorithm 1: Prompt Optimization via Multi-Armed Bandit (Extended Reward)

Input: Prompt pool $\mathcal{P} = \{p_1, \ldots, p_K\}$, ϵ (exploration rate)
Output: Updated reward values μ_k for all prompts
foreach *round* $t = 1$ *to* T **do**
  `// Select a prompt`
  With probability ϵ, randomly select $p_k \in \mathcal{P}$;
  Otherwise, select $p_k = \arg\max_p \mu_p$;
  `// Construct input and call LLM`
  Instantiate p_k with user context and target site;
  Encode current screenshot as base64 image I;
  Send (p_k, I) to Claude API to get command c;
  `// Execute command`
  Let $(s_t, c_t) \leftarrow$ `executeCommand`(c);
  `//` s_t`: success,` c_t`: check count`
  Compute $k_t = \min(1, c_t / C_{\max})$;
  Compute $d_t \leftarrow$ diversity score w.r.t. command history;
  Compute $R_t = s_t + d_t - k_t$;
  `// Update reward statistics`
  $n_k \leftarrow n_k + 1$;
  $\mu_k \leftarrow \mu_k + \frac{1}{n_k}(R_t - \mu_k)$;
  `// Log execution record`
  Save $\langle p_k, c, R_t, s_t, k_t, d_t \rangle$ to JSONL log;
  if $t \mod 10 = 0$ **then**
    `// Prune and replace prompts`
    Remove bottom 20% prompts from $\mathcal{P}$;
    Replace with new prompts rewritten by Claude;
return $\{\mu_k\}_{k=1}^{K}$

Beyond diversity and realism, another key advantage of our approach lies in its scalability and cost-efficiency. Compared to collecting traffic from real human participants, our LLM-based multi-agent framework offers substantial cost reductions while maintaining semantic fidelity. While exploring all cost factors involved in collecting training traffic is beyond the scope of this paper, Figure 3 allows us to compare the marginal costs of humans to an LLM approach. We decompose this marginal cost into two components: *human cost* (participant compensation, supervision, and limited parallelism) and *LLM cost*, which is dominated by API usage (tokens and multimodal inputs). We found that generating 10GB of browsing data using human participants can cost between $80 and

$300 per site due to participant compensation, experimental supervision, and limited parallelism. In contrast, our framework—built on top of commercial LLM APIs—can synthesize comparable volumes of traffic at a fraction of the cost (typically under $30 per 10GB) while scaling seamlessly across machines and websites. Crucially, the dominant LLM cost is the API bill; other marginal costs (local execution, bandwidth, and orchestration overhead) are low in our setup and can be reasonably ignored for per-site scaling estimates. We were unable to compare the costs of obtaining traffic using scripted browsers as we used existing University servers and networking bandwidth free of charge. A more comprehensive analysis might look at cloud computing charges or the costs of buying or renting servers. Finally, LLM inference costs have continued to decrease rapidly in recent years, which strengthens the long-term practicality of our approach: as API pricing improves, larger-scale LLM-Sim datasets become increasingly feasible at low marginal cost.

The LLM agents require active prompt optimization that ensures high behavioral variance and realistic interaction patterns. By integrating reward-based self-improvement and structured feedback, the system maintains both command precision and browsing diversity over long sessions. This enables a synthetically generated WFP traffic corpus to serve as a viable complement, and, in many settings, an alternative, to costly human-collected data.

5 Evaluation

We comprehensively evaluate the generalization performance of modern website fingerprinting (WFP) models under realistic web conditions. Our experiments are structured in four stages: we first introduce our experimental setup and curated datasets, then assess model performance under traditional versus real-user traffic (Stage I). We next explore whether large language model (LLM)-driven multi-agent traffic can enhance cross-user generalization (Stage II), and finally analyze how training sample size influences robustness across scripted and LLM-generated data regimes. Together, these evaluations reveal the limitations of legacy Scripted Browsing traces, highlight the value of behaviorally rich simulations, and motivate new scalable methodologies for WFP research. Finally, we evaluate robustness under three representative WFP defenses in our *boundless* (boundary-free) setting.

5.1 Experimental Setup

To evaluate the resilience and generalization of modern website fingerprinting models, we benchmark nine state-of-the-art classifiers across a wide range of datasets and training conditions. These models encompass convolutional neural networks (CNNs), transformers, contrastive learning methods, and specialized fingerprinting architectures.

- **TMWF** [19] employs a temporal mutual information framework to align features extracted from raw packet sequences. By maximizing temporal consistency, it captures both short-term bursts and long-range temporal correlations for accurate classification.
- **ARES** [12] proposes a robust, tab-agnostic encoder that disentangles site identity from tab-related noise, enabling generalizable fingerprinting across complex multi-tab sessions. It operates without requiring session segmentation or tab labels, enhancing real-world applicability.
- **NetCLR** [1] applies self-supervised contrastive learning on unlabeled packet sequences to learn discriminative representations. Its dual-branch architecture uses augmentations in both the temporal and directional space to enforce alignment and variance in latent embeddings.
- **BAPM** [14] introduces a block attention mechanism tailored for multi-tab browsing scenarios, effectively handling overlapping traffic by segmenting traces into distinct blocks. Unlike prior models, it fully utilizes the entire packet trace, including overlapping areas, to enhance website fingerprinting accuracy.
- **TF** [29] models WFP as a metric learning problem using triplet networks, enabling few-shot learning with only a handful of samples per class. It offers strong generalization with minimal training data.
- **Var-CNN** [2] leverages a variable-kernel CNN with residual connections and dilated convolutions. By processing packet sizes and directions jointly, it provides robustness to timing noise and user behavior variation.
- **Tik-Tok** [25] proposes a traffic splitting mechanism that segments flows into semantically meaningful bursts (or "tiks" and "toks"). Each burst is classified individually, and predictions are aggregated over time to adapt to dynamic web content loading.
- **DF** [28] is one of the most widely adopted WFP baselines. It combines deep CNN layers with dropout and batch normalization, using direction-augmented input matrices. DF remains a strong baseline due to its simplicity and surprising effectiveness across datasets.
- **WFNet** [30] integrates a pre-trained traffic encoder with a fine-tunable classification head. It is trained using a domain-aware transfer learning pipeline, making it especially effective under low-resource or cross-domain conditions.

All models are trained and evaluated under a consistent pipeline using PyTorch on a shared GPU cluster. The input format follows the convention of 5000-length packet size/time sequences; that is, each **sample** corresponds to a contiguous block of 5000 packets represented by their sizes and inter-arrival times. Packet traces are segmented into non-overlapping blocks to form samples. Each experiment adopts a closed-world label space (e.g., 5, 10, or 20 websites, depending on the setting), and results are averaged across three random seeds to mitigate variance. Each segment inherits a website/domain ground-truth label from the target site during collection, enabling boundary-free ("boundless") evaluation without page/session markers. Unless stated otherwise, our main generalization protocol is LeaveOneUser (train on multiple users, test on a strictly unseen user), which avoids same-user leakage and directly measures cross-user heterogeneity.

In addition, we evaluate model robustness under an open-world setting that includes traffic from 40 unseen websites grouped into an extra class. To avoid class-imbalance confounds, we subsample the aggregated background to match the scale of a single monitored class (1,000 segments total). Across all models, open-world testing yields only a minor accuracy reduction—typically within 2–4%

when using Human+Claude training—indicating that our LLM-augmented data enables stable generalization even beyond the closed-world assumption.

5.2 Dataset Overview: Realistic Web Fingerprinting at Scale

To evaluate the effectiveness of WFP models under realistic web conditions, we curate and utilize three distinct datasets, each designed to capture varying degrees of interaction realism and user diversity. These datasets include: (1) a large-scale real human browsing dataset, (2) a scripted browsers dataset, and (3) an LLM-simulated interaction dataset generated through our multi-agent framework.

RealHuman Dataset. Our primary dataset consists of real user traffic collected from 20 modern websites, chosen to reflect the complexity and diversity of contemporary web usage. These sites span e-commerce (*Amazon, Ebay, Target*), media and news (*CNN, Fox News, Britannica, NFL*), social platforms (*Reddit, Pinterest, Fandom, TikTok, Zhihu*), streaming services (*YouTube, Bilibili, Iqiyi*), and utility portals (*LinkedIn, Yahoo, Yelp, Apple, Wikipedia*). The site selection is inspired by recent research such as WFNet [30], emphasizing dynamic and multi-service platforms. Many of these sites embody modern Single-Page Application (SPA) architectures, where rich content is dynamically loaded and user interaction spans multiple services. For instance, a user session on Amazon may include product search, recommendation browsing, video playback, and cart operations—each generating different traffic patterns without discrete page boundaries.

We hired real-world participants to browse these websites freely across a range of devices and conditions. Each participant was instructed to engage naturally with the content. Participants were instructed not to visit external websites during a session, so that each non-overlapping 5,000-packet segment is effectively single-site traffic. After over 100 hours of browsing, we collected **800 samples** per class, approximately equal to **4.5 GB of traffic per website**. This dataset provides a realistic, behavior-rich benchmark for evaluating WFP robustness.

Scripted Browsers Dataset. To enable comparison with prior work, we also generate a large-scale scripted browsing dataset using a scripted crawler. This dataset mimics conventional WFP collection protocols where a bot sequentially visits predefined URLs. The crawler does not emulate user interaction such as scrolling, pausing, or branching behavior. It simply loads a page and proceeds after a fixed timeout.

While this strategy supports efficient data collection, it fails to capture the semantic and behavioral diversity inherent in real human traffic. Nonetheless, we produce **8000 samples** per class, corresponding to approximately **40 GB of raw traffic per website on average (about 800 GB total raw across 20 sites)**, for controlled benchmarking. This allows direct comparison with historical WFP results and highlights the limitations of purely rule-based sampling in the context of modern, interactive sites.

LLM-Sim Dataset. Finally, we introduce a third dataset created through our LLM-based multi-agent browsing framework. Unlike traditional crawlers, these agents follow natural-language browsing goals and simulate semantically grounded interaction sequences across websites. For computational efficiency and API constraints, we focus on a 10-site subset from our RealHuman corpus, including *Amazon, Britannica, CNN, Fandom, Fox News, NFL, Pinterest, Reddit, TikTok, YouTube*. Each site is represented by **1000 samples**, yielding approximately **7 GB of traffic per website** and totaling 10,000 traces overall.

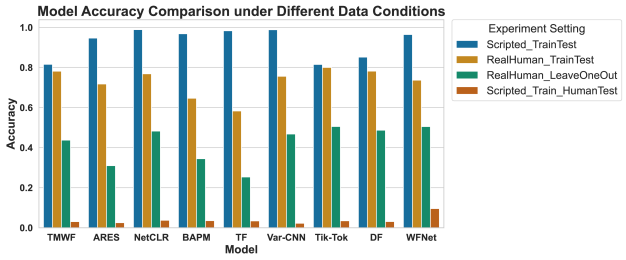

Figure 5: Accuracy of nine WFP models under four training/testing regimes: (1) Scripted_TrainTest, (2) RealHuman_TrainTest, (3) LeaveOneUser, and (4) CrossDomain. Results highlight the generalization gap between scripted and real browsing data.

The reduced dataset size is a result of rate limits imposed by commercial Computer Use APIs and LLM-hosted browser environments. Despite the lower volume, our results demonstrate that LLM-generated data—though smaller—can surpass script-based baselines in generalization performance (see Section 5.4). Our current cap of 1,000 samples/site is primarily constrained by available API credits; continued reductions in LLM inference costs make larger-scale generation increasingly feasible. This underscores the efficiency of behaviorally diverse data and the promise of LLM agents as realistic traffic generators.

Summary. In total, our evaluation spans over 180,000 browsing traces across three datasets and three interaction modes (scripted, human, and simulated). These datasets allow us to isolate the effects of behavioral realism, semantic intent, and interaction complexity on WFP model performance under both traditional and adversarial training conditions.

5.3 Stage I: Performance under Controlled and Realistic Conditions

We begin our evaluation by revisiting the standard assumption in the WFP literature: models are trained and tested under *controlled, synthetic* conditions using rule-based *scripted* crawlers. While these benchmarks have driven progress for nearly a decade, they often overlook the complexities of modern interactive websites. We compare model performance under two single-domain settings—robot-simulated traffic versus real human browsing data—before transitioning to cross-user and cross-domain generalization.

Our goal is twofold. First, we seek to quantify how interaction fidelity influences classifier performance: can WFP models still succeed when exposed to naturalistic, behavior-rich sessions? Second, we aim to establish a foundation for later sections by highlighting the limitations of scripted data and the need for richer behavioral diversity.

Single-Domain Evaluation: Scripted_TrainTest vs. RealHuman_TrainTest. We evaluate nine state-of-the-art models under the Scripted_TrainTest and RealHuman_TrainTest conditions. As shown in Figure 5 and Table 1, models trained and tested on scripted traffic achieve near-perfect results (e.g., NetCLR: 98.9%, TF: 98.3%, Var-CNN: 98.9%), confirming past findings that scripted data is trivially separable. This is largely due to the uniform click paths, predictable transitions, and stateless nature of robot-generated sessions, which fail to capture interaction variance. As a result, Scripted_TrainTest should be viewed as a synthetic upper bound that can overestimate real-world WFP threat and may inadvertently grant attackers an unfair advantage.

However, accuracy drops significantly when models are trained and evaluated on real human traces. Even the strongest performers (TikTok, DF, NetCLR) plateau at 75–80%, while others like TF and ARES drop below 65%. F1-score, precision, and recall all decline, most notably recall, which reveals that models often miss diverse user behaviors. For instance, TF's F1-score falls from 0.983 to 0.583, but its recall drops even more steeply, suggesting its high-confidence predictions are limited to a few familiar patterns. In contrast, models like Var-CNN and NetCLR maintain higher recall (0.774 and 0.747 respectively), indicating stronger robustness to intra-site user variability.

This drop is not just a matter of accuracy but also consistency. Across our 20-site benchmark, human data introduces session noise, personalization artifacts, and diverse navigation flows. For example, different users may access entirely different parts of Amazon or YouTube depending on their intent, identity, and scroll depth. These complexities yield overlapping traffic signatures that challenge even state-of-the-art classifiers. The results underscore the limitations of relying solely on scripted browsing data and affirm the necessity of training on interaction-rich traces.

Generalization under LeaveOneUser and CrossDomain Tasks. We next test model robustness under two increasingly difficult settings. In LeaveOneUser, models are trained on all but one human participant and tested on the held-out user. In CrossDomain, models are trained on scripted traffic and evaluated on real human traces.

The LeaveOneUser setting shows a consistent accuracy drop of 30–40% across models (e.g., Var-CNN: 75.6% to 46.8%). Most models lose recall faster than precision, suggesting that they overfit to common behaviors while failing to generalize across unseen users. NetCLR and WFNet demonstrate relatively higher leave-one-out recall, indicating their resilience to behavioral diversity. Interestingly, WFNet also exhibits the highest recall in CrossDomain (0.102), reflecting a robust embedding mechanism that adapts well to unobserved interaction sequences.

CrossDomain is the most challenging task. Models trained solely on scripted traffic completely fail to generalize, with accuracy often falling below 10% and F1-scores near zero. For instance, TF drops to 0.034, ARES to 0.025, and Var-CNN to 0.022 F1-score. These failures highlight the structural mismatch between scripted and real traffic–rule-based bots do not mimic personalized interaction.

These findings motivate our exploration of synthetic alternatives that better mirror human browsing. In Stage II, we assess whether LLM-based traffic can bridge this realism gap.

5.4 Stage II: Cross-User Generalization Enhanced by LLM-Simulated Traffic

We now testIM test whether synthetic traffic from our multi-agent LLM system can improve generalization. In this stage, we use the 5-site and 10-site variants of the closed-world task. As before, we focus on cross-user transfer: training on multiple users or simulations, and testing on a previously unseen human.

Before analyzing LLM-based enhancements, we briefly revisit the baseline human-only dataset results in this reduced-domain setting. While RealHuman_TrainTest offers reasonably strong performance due to random within-user splits, we treat it as an optimistic, seen-user upper bound and rely on LeaveOneUser (unseen user) for our main generalization claims. accuracy drops sharply under the LeaveOneUser condition. For instance, in the 10-site task, TikTok and NetCLR drop from over 80% to roughly 50%, and DF from 78.2% to 48.7%. This gap highlights the difficulty of modeling diverse human behavior across users. Without explicit exposure to intra-site variation in scrolling, clicking, or content consumption styles, models struggle to generalize—even when trained on large real-world datasets. These results reinforce the need for augmentation mechanisms that can synthetically reproduce such behavioral heterogeneity—precisely the goal of our multi-agent LLM framework.

Comparative Training Settings. In addition to Human_TrainTest and LeaveOneUser from Stage I, we evaluate:

(1) Human+Claude: Train on a combined set of human and LLM-generated traces.
(2) ClaudeOnly: Train on LLM-simulated traffic exclusively.
(3) ScriptedOnly: Train on scripted browers traces exclusively.

Performance Gains and Explanation. As Figure 6 shows, Human+Claude consistently improves generalization. Accuracy jumps by 25–35% in both tasks. For example, NetCLR improves from 53.6% to 77.3% in the 5-site task, and TikTok from 50.3% to 84.9%. Importantly, LLM data is particularly helpful for models like WFNet and BAPM, which benefit from longer temporal context and diverse semantic cues.

Models trained with ClaudeOnly also outperform ScriptedOnly baselines by large margins—often tripling the accuracy. Even with one-fifth the data, Claude simulations better approximate realistic behavior. This is because decision-making agents are instantiated with personas (e.g., "tech-savvy teen," "price-sensitive shopper") that guide behavior in context-sensitive ways. The instructions they generate capture richer, goal-driven browsing, which leads to more realistic interaction trajectories.

Behavioral Regularization and Robustness. Claude agents help reduce inter-site variance and prevent overfitting to frequent paths. For example, TikTok and Var-CNN exhibit smoother performance across sites when trained with LLM data. The behavioral regularization effect arises from semantic grounding, consistent interaction patterns, and broader behavioral coverage—especially helpful for infinite-scroll or recommendation-based platforms.

Interestingly, in some cases LLM-trained models even outperform human-only training. For instance, in both the 5-site and 10-site settings, ClaudeOnly→Human (tested on an unseen user) outperforms human-only training under LeaveOneUser by a clear

Table 1: Performance metrics (F1-score, Precision, Recall) of each model under four training/testing settings.

Model	Scripted_TrainTest			RealHuman_TrainTest			LeaveOneUser			CrossDomain		
	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec
TMWF	0.816	0.855	0.818	0.782	0.803	0.775	0.437	0.489	0.489	0.031	0.024	0.084
ARES	0.947	0.949	0.947	0.718	0.723	0.722	0.410	0.440	0.443	0.025	0.058	0.064
NetCLR	0.989	0.989	0.989	0.769	0.804	0.747	0.482	0.505	0.495	0.037	0.047	0.056
BAPM	0.968	0.969	0.968	0.646	0.672	0.654	0.344	0.391	0.399	0.035	0.029	0.068
TF	0.983	0.984	0.983	0.583	0.603	0.581	0.254	0.287	0.278	0.034	0.053	0.052
Var-CNN	0.989	0.989	0.989	0.756	0.760	0.774	0.468	0.528	0.512	0.022	0.019	0.051
Tik-Tok	0.815	0.859	0.831	0.800	0.812	0.801	0.505	0.538	0.532	0.035	0.043	0.044
DF	0.852	0.910	0.845	0.782	0.794	0.785	0.487	0.534	0.511	0.031	0.041	0.069
WFNet	0.965	0.965	0.965	0.737	0.763	0.740	0.495	0.525	0.532	0.096	0.091	0.102

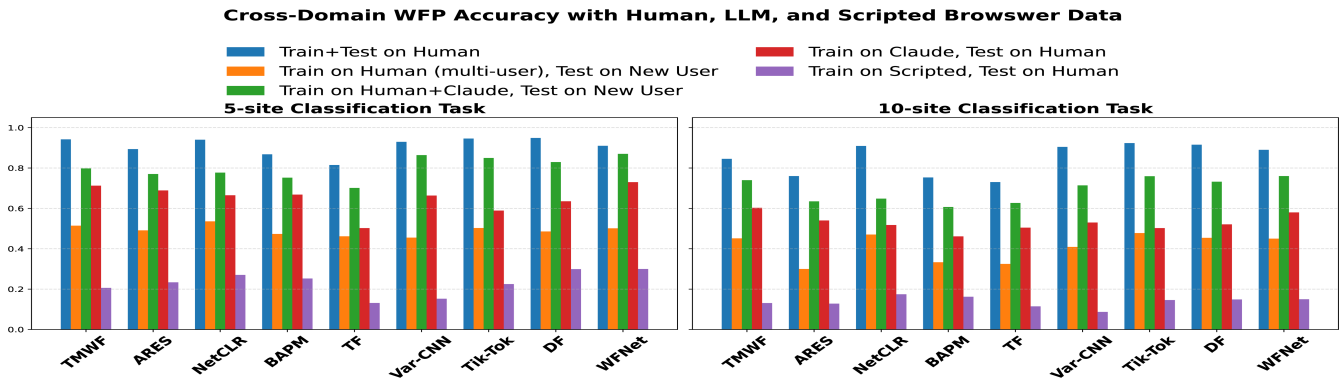

Figure 6: Accuracy comparison across five training strategies (Human only, Leave-One-User, Human+Claude, Claude-only, and Scripted-only) on two tasks: Top: 5-site classification, Bottom: 10-site classification. LLM-based traffic significantly boosts cross-user accuracy compared to synthetic baselines, despite using only 20% of the data size.

margin, aligning with Figure 6 and reinforcing the transferability of LLM-simulated behavior. This suggests that simulated users can exceed real human diversity by covering edge-case paths or low-frequency interaction types.

Looking Ahead. These results reinforce three core takeaways:

- Traditional scripted traffic is insufficient for training WFP models on modern interactive websites.
- LLM-generated traffic introduces critical behavioral diversity, enhancing robustness to user variability.
- Multi-agent simulation offers a scalable alternative to expensive human data collection.

In the next section, we analyze how the number of LLM samples influences this benefit. We find that while increasing LLM data improves generalization, scaling scripted traffic yields minimal gains, further confirming the importance of realism over volume.

5.5 Stage III: Open-World Evaluation with Human and LLM-Augmented Traffic

We further evaluate model robustness under the *open-world* assumption, where an attacker must distinguish monitored websites from a much larger background of unmonitored traffic. Concretely, we consider the 10-website classification task from earlier stages and extend it with traffic collected from an additional 40 popular websites. Each of these extra sites contributes 5–15 minutes of natural browsing data, which we aggregate into a single "background" label following standard open-world WFP methodology.

Figure 7 summarizes the results. Across all nine baseline models, Human-only training suffers an average accuracy degradation of **5–8%** when tested under open-world conditions. In contrast, models trained with Human+Claude traces are considerably more resilient, with performance declines limited to only **2–3%**. The smaller degradation indicates that LLM-generated traffic provides valuable regularization, covering behavioral trajectories that reduce false alarms on background sites. These findings confirm that LLM-simulated browsing helps attackers construct more robust training datasets, thereby strengthening open-world website fingerprinting attacks.

5.6 Impact of Training Sample Size: LLM vs. Scripted Browsers Traffic

To further investigate how dataset scale influences generalization performance, we examine WFP model accuracy on a fixed RealHuman_Test set as the number of training samples varies. Figure 8 presents a comparative analysis of models trained on multi-agent LLM-generated traffic (300–1000 samples per class) and traditional scripted browser traffic (1000–8000 samples per class).

Across all nine models, LLM-based training exhibits a clear upward trend: accuracy improves consistently as more training data

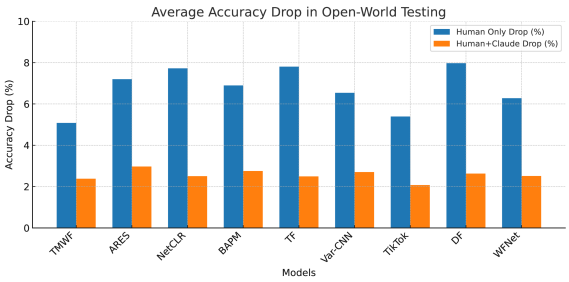

Figure 7: Open-world evaluation on the 10-website classification task extended with 40 additional websites as an aggregated background class. Accuracy degradation is shown relative to closed-world testing. Human-only models suffer 5–8% performance loss, whereas Human+Claude models exhibit only a 2–3% drop. This demonstrates that LLM-augmented training substantially improves robustness under adversarially realistic conditions.

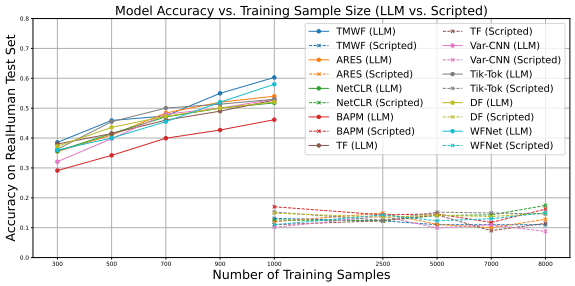

Figure 8: Accuracy of nine WFP models trained on LLM-generated (solid lines, 300–1000 samples) vs. scripted browser traffic (dashed lines, 1000–8000 samples). While scripted traffic offers little gain with scale, LLM data steadily improves generalization, highlighting that realistic interaction diversity, not volume alone, is key to modern WFP performance.

becomes available. For example, TMWF and NetCLR reach 60.3% and 51.8% accuracy respectively at 1000 LLM samples—substantially outperforming their performance at 300 samples, where accuracies drop below 40%. This scaling behavior is consistent and monotonic, indicating that our LLM-based agents produce data distributions that grow meaningfully with size and maintain alignment with the behavioral complexity of real users.

In contrast, scripted browers data demonstrates no such benefit. Despite training on up to 8000 samples per class, most models fail to exceed 17% accuracy. Moreover, many models (e.g., Var-CNN, TF, DF) show marginal or no improvement when the sample size increases from 1000 to 8000. This suggests a saturation effect—scripted traffic, originally designed for static and deterministic websites, lacks the semantic diversity and session-level richness necessary for generalization to dynamic modern environments.

These findings support our central claim: website fingerprinting on modern web platforms is a fundamentally more challenging problem than prior literature has assumed. Moreover, our LLM-Sim scaling results have not yet saturated; the current cap of 1,000 samples/site is primarily limited by API credits, and continued reductions in LLM inference costs make larger-scale generation increasingly feasible. Simply scaling these scripted datasets does not resolve the generalization gap—more low-quality data does not translate to better performance.

In contrast, the LLM-simulated traffic we propose introduces semantically grounded, persona-driven interactions that better reflect user heterogeneity. Even with fewer samples, LLM data consistently enables stronger real-world generalization.

Setting (WFNet)	2000	1500	1000	500
No defense (baseline)	0.760	0.735	0.690	0.610
TrafficSliver	0.592	0.571	0.484	0.377
Cluster Anon. (Palette)	0.647	0.604	0.579	0.458
Zero-delay (FRONT)	0.706	0.671	0.617	0.526

Table 2: Accuracy under three defenses for *Train on Human+Claude, Test on New User* (Leave-One-User) with WFNet, across different per-site training budgets, where defenses are applied to both train and test traces (10-site closed world).

5.7 Stage IV: Robustness under WFP Defenses in the Boundless Setting

We next evaluate robustness when representative WFP defenses are deployed in our boundless segment-based setting. We consider three defenses—*TrafficSliver*[9] (traffic splitting), *Palette*[27] (traffic cluster anonymization), and *FRONT*[13] (zero-delay lightweight defense)—and apply each defense to *both* training and testing traces to reflect an attacker that learns on defended data. We report results on the 10-site closed-world task using WFNet under Train on Human+Claude, Test on New User (Leave-One-User), varying the per-site training budget from 500 to 2,000 segments.

Table 2 shows that defenses reduce accuracy but do not eliminate distinguishability in our setting. The key observation is a clear *recovery trend*: under every defense, accuracy increases as the per-site training budget grows from 500 to 2,000 segments, indicating that additional defended traces remain informative. Moreover, the degradation is comparatively limited for our boundless, domain-level traces (which do not rely on clean page splits or early-trace cues), suggesting these defenses provide less protection in this setting and the attack remains a substantial privacy threat after defense. The relative impact aligns with each defense's focus: TrafficSliver yields the largest drop by disrupting burst/sequence structure via splitting; FRONT yields the smallest drop as its perturbations concentrate on early-trace patterns; and Palette falls in between, since cluster-based anonymization is most effective when traffic clusters cleanly into stable modes, while modern interactive sites exhibit diverse within-site patterns that are harder to homogenize.

6 Related Work

Website fingerprinting models. Early work on website fingerprinting (WFP) uses hand-crafted features and classical machine learning in small closed worlds. Subsequent deep models based

on CNN and transformer architectures operate directly on packet sequences and achieve near-perfect closed-world accuracy under synthetic conditions. More recent approaches explore robustness via contrastive learning, domain transfer, and adversarial augmentation, but still largely assume short, scripted sessions and focus on site-level classification. Our work follows this line of deep WFP models, but evaluates them under long, multi-step, high-entropy browsing with persona labels and an open-world label space.

Behavioral and persona fingerprinting. Beyond WFP, a broad literature studies how users and devices can be identified from behavioral traces, including network traffic patterns, interaction timing, and other implicit signals. These studies show that even when content is hidden or identifiers are removed, stable individual or group-specific patterns often remain. In contrast, most existing work assumes access to rich application- or user-level logs. We instead study persona fingerprints at the network layer, showing that coarse, encrypted packet sequences already carry enough structure to distinguish behavioral personas in a multi-site, open-world setting.

LLM-based agents for web interaction. Recent systems use large language models as agents that plan, remember, and invoke tools over long interaction horizons. Browser-interacting agents[37] extend this idea to real user interfaces, combining vision-language reasoning with keyboard and mouse control to complete web tasks. Prior work in this space typically optimizes task success or instruction following, without examining the resulting network traffic. Our approach repurposes such agents as a controllable traffic generator: a persona-conditioned decision agent drives a computer-use agent in a real browser, producing realistic, high-entropy sessions whose encrypted packet traces we use to study website and persona fingerprinting.

7 Discussion

Despite the superior effectiveness of attacks trained on data generated by LLM-based multi-agent simulation compared to traditional scripted methods, our approach still faces important limitations and open challenges. In this section, we outline the tradeoffs, scalability concerns, and potential future research directions stemming from our findings.

Cost vs. Realism Tradeoffs in Data Generation. Compared to traditional scripted browers datasets—often generated at negligible cost through scripted crawlers—LLM-based traffic simulation is considerably more expensive. Despite being far more scalable and efficient than recruiting human participants, generating high-quality interaction traces using commercial LLMs (e.g., via API-driven computer use agents) incurs both monetary and compute costs. As a result, even modest datasets with a few hundred samples per class can rapidly consume daily token budgets or GPU inference quotas.

This cost barrier presents a fundamental tradeoff: as websites become more dynamic and interaction-rich, effective fingerprinting requires data that captures complex user behavior. Scripted methods no longer suffice, and models increasingly demand training data that reflects semantic intent, exploratory actions, and user-specific navigation. This trend implies that future WFP systems must either rely on more powerful synthetic pipelines or be restricted to narrower threat models where such data is obtainable.

Targeted Fingerprinting and the Rise of Personalized Simulation. Interestingly, the precision and controllability of LLM agents open the door to a different attack vector: targeted fingerprinting. Unlike broad-scale open-world classification, attacks that aim to profile or deanonymize specific individuals may require only small amounts of data—provided the agent can model the target's browsing preferences, background, or goals. For example, if an attacker has partial knowledge of a user's demographics, profession, or daily routine, an LLM-driven simulation could potentially emulate their interaction pattern and improve classification accuracy against that individual.

Such personalized attacks were previously impractical due to data scarcity, but LLM conditioning capabilities make this increasingly feasible. We see this as a dual-use scenario: while it enhances attack realism, it also raises ethical and privacy concerns that must be addressed by the research community.

8 Conclusion

Our main finding is user behavior introduces a surprising amount of intra-class variability—different individuals browsing the same site can produce traffic patterns that are distinct enough to confuse even strong classifiers. This highlights the importance of training datasets that are not only large, but also behaviorally diverse. In cross-user and cross-domain evaluations, even the most accurate models trained under idealized conditions suffer sharp performance degradation.

To address these challenges, we propose a scalable multi-agent data generation pipeline powered by large language models (LLMs). Our framework simulates user interactions by combining high-level decision agents with low-level execution agents based on LLM-driven browser control. Despite operating at just one-third the cost of real human data collection, this synthetic traffic significantly improves generalization: in many cases, models trained on LLM-generated traces outperform those trained on scripted data by over 3×.

While promising, this approach still has limitations. Compared to near-free synthetic crawlers, multi-agent LLM generation is more costly and constrained by current API usage limits. Building massive open-world datasets with thousands of websites remains challenging without dedicated GPU infrastructure or open-source foundation models. Nonetheless, as token prices drop and model quality rises, scalable simulation will become increasingly accessible.

In the longer term, our framework also opens the door to new research directions. LLMs can mimic specific personas, enabling targeted fingerprinting attacks on individuals with limited training data. As LLMs evolve to incorporate multimodal understanding and long-term memory, they may unlock even richer browsing simulations—bridging the gap between controlled evaluation and real-world threat modeling.

In summary, this work demonstrates that the future of WFP will be shaped not only by advances in modeling, but by the ability to construct realistic, diverse, and scalable datasets. Our findings advocate for a paradigm shift in dataset design, one that embraces behavioral realism as a first-class requirement.

References

[1] Alireza Bahramali, Ardavan Bozorgi, and Amir Houmansadr. 2023. Realistic Website Fingerprinting By Augmenting Network Traces. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (Copenhagen, Denmark) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 1035–1049. https://doi.org/10.1145/3576915.3616639

[2] Sanjit Bhat, David Lu, Albert Kwon, and Srinivas Devadas. 2019. Var-CNN: A Data-Efficient Website Fingerprinting Attack Based on Deep Learning. *Proceedings on Privacy Enhancing Technologies* 2019, 4 (July 2019), 292–310. https://doi.org/10.2478/popets-2019-0070

[3] Tina Burns, Chuxu Song, Ivan Seskar, Jorge Ortiz, and Richard P. Martin. 2022. A Simplified Machine Learning Approach to Classifying Individual Websites. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*. 6109–6114. https://doi.org/10.1109/GLOBECOM48099.2022.10001054

[4] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg. 2014. A Systematic Approach to Developing and Evaluating Website Fingerprinting Defenses. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (Scottsdale, Arizona, USA) *(CCS '14)*. Association for Computing Machinery, New York, NY, USA, 227–238. https://doi.org/10.1145/2660267.2660362

[5] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. 2012. Touching from a distance: website fingerprinting attacks and defenses. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security* (Raleigh, North Carolina, USA) *(CCS '12)*. Association for Computing Machinery, New York, NY, USA, 605–616. https://doi.org/10.1145/2382196.2382260

[6] Mantun Chen, Yongjun Wang, Hongzuo Xu, and Xiatian Zhu. 2021. Few-shot website fingerprinting attack. *Computer Networks* 198 (2021), 108298. https://doi.org/10.1016/j.comnet.2021.108298

[7] Yifei Cheng, Yujia Zhu, Baiyang Li, peishuai sun, Yong Ding, Xinhao Deng, and Qingyun Liu. 2025. HOLMES & WATSON: A Robust and Lightweight HTTPS Website Fingerprinting through HTTP Version Parallelism. In *THE WEB CONFERENCE 2025*. https://openreview.net/forum?id=lSJ8VjjimZ

[8] Giovanni Cherubin, Rob Jansen, and Carmela Troncoso. 2022. Online Website Fingerprinting: Evaluating Website Fingerprinting Attacks on Tor in the Real World. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 753–770. https://www.usenix.org/conference/usenixsecurity22/presentation/cherubin

[9] Wladimir De la Cadena, Asya Mitseva, Jens Hiller, Jan Pennekamp, Sebastian Reuter, Julian Filter, Thomas Engel, Klaus Wehrle, and Andriy Panchenko. 2020. TrafficSliver: Fighting Website Fingerprinting Attacks with Traffic Splitting. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, USA) *(CCS '20)*. Association for Computing Machinery, New York, NY, USA, 1971–1985. https://doi.org/10.1145/3372297.3423351

[10] Xinhao Deng, Qi Li, and Ke Xu. 2024. Robust and Reliable Early-Stage Website Fingerprinting Attacks via Spatial-Temporal Distribution Analysis. arXiv:2407.00918 [cs.CR] https://arxiv.org/abs/2407.00918

[11] Xinhao Deng, Qilei Yin, Zhuotao Liu, Xiyuan Zhao, Qi Li, Mingwei Xu, Ke Xu, and Jianping Wu. 2023. Robust Multi-tab Website Fingerprinting Attacks in the Wild. In *2023 IEEE Symposium on Security and Privacy (SP)*. 1005–1022. https://doi.org/10.1109/SP46215.2023.10179464

[12] Xinhao Deng, Xiyuan Zhao, Qilei Yin, Zhuotao Liu, Qi Li, Mingwei Xu, Ke Xu, and Jianping Wu. 2025. Towards Robust Multi-tab Website Fingerprinting. arXiv:2501.12622 [cs.CR] https://arxiv.org/abs/2501.12622

[13] Jiajun Gong and Tao Wang. 2020. Zero-delay Lightweight Defenses against Website Fingerprinting. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, 717–734. https://www.usenix.org/conference/usenixsecurity20/presentation/gong

[14] Zhong Guan, Gang Xiong, Gaopeng Gou, Zhen Li, Mingxin Cui, and Chang Liu. 2021. BAPM: Block Attention Profiling Model for Multi-tab Website Fingerprinting Attacks on Tor. In *Proceedings of the 37th Annual Computer Security Applications Conference* (Virtual Event, USA) *(ACSAC '21)*. Association for Computing Machinery, New York, NY, USA, 248–259. https://doi.org/10.1145/3485832.3485891

[15] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large Language Model based Multi-Agents: A Survey of Progress and Challenges. arXiv:2402.01680 [cs.CL] https://arxiv.org/abs/2402.01680

[16] Jamie Hayes and George Danezis. 2016. k-fingerprinting: a robust scalable website fingerprinting technique. In *Proceedings of the 25th USENIX Conference on Security Symposium* (Austin, TX, USA) *(SEC'16)*. USENIX Association, USA, 1187–1203.

[17] Jinwei Hu, Yi Dong, Shuang Ao, Zhuoyun Li, Boxuan Wang, Lokesh Singh, Guangliang Cheng, Sarvapali D. Ramchurn, and Xiaowei Huang. 2025. Position: Towards a Responsible LLM-empowered Multi-Agent Systems. arXiv:2502.01714 [cs.MA] https://arxiv.org/abs/2502.01714

[18] Rob Jansen and Ryan Wails. 2023. Data-Explainable Website Fingerprinting with Network Simulation. *Proceedings on Privacy Enhancing Technologies* 2023, 4 (2023). See also https://explainwf-popets2023.github.io.

[19] Zhaoxin Jin, Tianbo Lu, Shuang Luo, and Jiaze Shang. 2023. Transformer-based Model for Multi-tab Website Fingerprinting Attack. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (Copenhagen, Denmark) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 1050–1064. https://doi.org/10.1145/3576915.3623107

[20] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. 2014. A Critical Evaluation of Website Fingerprinting Attacks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (Scottsdale, Arizona, USA) *(CCS '14)*. Association for Computing Machinery, New York, NY, USA, 263–274. https://doi.org/10.1145/2660267.2660368

[21] Marc Juarez, Mohsen Imani, Mike Perry, Claudia Diaz, and Matthew Wright. 2016. Toward an Efficient Website Fingerprinting Defense. arXiv:1512.00524 [cs.CR] https://arxiv.org/abs/1512.00524

[22] Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. 2024. A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. https://doi.org/10.1007/s44336-024-00009-2

[23] Asya Mitseva and Andriy Panchenko. 2024. Stop, Don't Click Here Anymore: Boosting Website Fingerprinting By Considering Sets of Subpages. In *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, Philadelphia, PA, 4139–4156. https://www.usenix.org/conference/usenixsecurity24/presentation/mitseva

[24] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. 2011. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society* (Chicago, Illinois, USA) *(WPES '11)*. Association for Computing Machinery, New York, NY, USA, 103–114. https://doi.org/10.1145/2046556.2046570

[25] Mohammad Saidur Rahman, Payap Sirinam, Nate Mathews, Kantha Girish Gangadhara, and Matthew Wright. 2020. Tik-Tok: The Utility of Packet Timing in Website Fingerprinting Attacks. *Proceedings on Privacy Enhancing Technologies* 2020, 3 (July 2020), 5–24. https://doi.org/10.2478/popets-2020-0043

[26] Meng Shen, Kexin Ji, Zhenbo Gao, Qi Li, Liehuang Zhu, and Ke Xu. 2023. Subverting Website Fingerprinting Defenses with Robust Traffic Representation. In *32nd USENIX Security Symposium (USENIX Security 23)*. USENIX Association, Anaheim, CA, 607–624. https://www.usenix.org/conference/usenixsecurity23/presentation/shen-meng

[27] Meng Shen, Kexin Ji, Jinhe Wu, Qi Li, Xiangdong Kong, Ke Xu, and Liehuang Zhu. 2024. Real-Time Website Fingerprinting Defense via Traffic Cluster Anonymization. In *2024 IEEE Symposium on Security and Privacy (SP)*. 3238–3256. https://doi.org/10.1109/SP54263.2024.00247

[28] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. 2018. Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) *(CCS '18)*. Association for Computing Machinery, New York, NY, USA, 1928–1943. https://doi.org/10.1145/3243734.3243768

[29] Payap Sirinam, Nate Mathews, Mohammad Saidur Rahman, and Matthew Wright. 2019. Triplet Fingerprinting: More Practical and Portable Website Fingerprinting with N-shot Learning. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) *(CCS '19)*. Association for Computing Machinery, New York, NY, USA, 1131–1148. https://doi.org/10.1145/3319535.3354217

[30] Chuxu Song, Zining Fan, Hao Wang, and Richard Martin. 2024. Seamless Website Fingerprinting in Multiple Environments. arXiv:2407.19365 [cs.CR] https://arxiv.org/abs/2407.19365

[31] Yixin Sun, Anne Edmundson, Laurent Vanbever, Oscar Li, Jennifer Rexford, Mung Chiang, and Prateek Mittal. 2015. RAPTOR: routing attacks on privacy in tor. In *Proceedings of the 24th USENIX Conference on Security Symposium* (Washington, D.C.) *(SEC'15)*. USENIX Association, USA, 271–286.

[32] Yashar Talebirad and Amirhossein Nadiri. 2023. Multi-Agent Collaboration: Harnessing the Power of Intelligent LLM Agents. arXiv:2306.03314 [cs.AI] https://arxiv.org/abs/2306.03314

[33] Tao Wang and Ian Goldberg. 2013. Improved website fingerprinting on Tor. In *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society* (Berlin, Germany) *(WPES '13)*. Association for Computing Machinery, New York, NY, USA, 201–212. https://doi.org/10.1145/2517840.2517851

[34] Tao Wang and Ian Goldberg. 2017. Walkie-Talkie: An Efficient Defense Against Passive Website Fingerprinting Attacks. In *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, Vancouver, BC, 1375–1390. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/wang-tao

[35] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. arXiv:2308.08155 [cs.AI] https://arxiv.org/abs/2308.08155

[36] Xiyuan Zhao, Xinhao Deng, Qi Li, Yunpeng Liu, Zhuotao Liu, Kun Sun, and Ke Xu. 2024. Towards Fine-Grained Webpage Fingerprinting at Scale. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security* (Salt Lake City, UT, USA) *(CCS '24)*. Association for Computing Machinery, New

York, NY, USA, 423–436. https://doi.org/10.1145/3658644.3690211
[37] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. WebArena: A Realistic Web Environment for Building Autonomous Agents. arXiv:2307.13854 [cs.AI] https://arxiv.org/abs/2307.13854

A Acknowledgments

The authors used generative AI-based tools to revise the text, improve flow and correct any typos, grammatical errors.