Web Intellectual Property at Risk: Preventing Unauthorized Real-Time Retrieval by Large Language Models

Yisheng Zhong[1], Yizhu Wen[2], Junfeng Guo[3], Mehran Kafai[4], Heng Huang[3], Hanqing Guo[2], Zhuangdi Zhu[1]

[1]George Mason University [2]University of Hawaii at Manoa [3]University of Maryland [4]Amazon

Overview

Background.
The explosive growth of web-retrieval-enabled Large Language Models (LLMs) silently scrapes and redistributes on-page intellectual property, eroding creators' economic incentives and legal control. We introduce a *semantic* defense that is embedded directly in HTML so site owners can actively throttle LLM extraction without harming normal human visitors.

Motivation.
- Conventional guards (robots.txt, meta-tags) rely on crawler self-identification and are easily ignored.
- Proprietary LLMs parse *both* visible and hidden markup, so purely client-side obfuscation fails.
- Users can override naive policies with prompt-engineering ("ignore the rules").
- Need a black-box, layout-preserving method that survives aggressive follow-up queries.

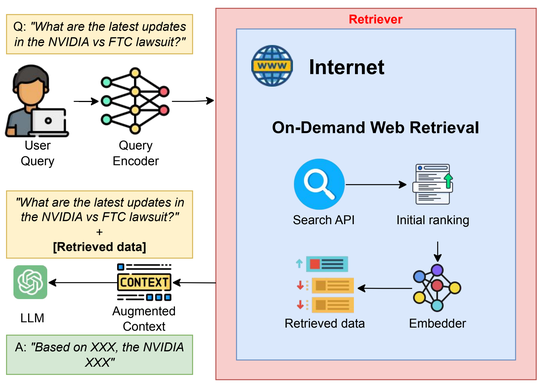
Contribution.
Our dual-level, min–max optimized HTML policy lifts defense success from **2.5 %** → **88.6 %** across mainstream LLMs—outperforming static, configuration-based baselines by > 30 ×.

Threat Model

Web-Retrieve Pipeline (simplest case).
Given a user query q, the LLM embeds it, fires a search, grabs the top-ranked page w, strips the HTML to plain text c, and then writes an answer r conditioned on (q, c). Retrieval is $p_{\theta,\phi_{retr}}(q|w)$, generation is $p_\theta(r|q,w)$, so the whole pipeline is captured by:

$$p_{\theta,\phi_{retr}}(r|q,w) = p_{\phi_{retr}}(w|q) \cdot p_\theta(r|q,w).$$

where φ_{retr} is the black box retrival module.



Challenges.
- **Low baseline defense rate:** naive notices succeed < 5 % of the time.
- **Prompt bypass:** "Ignore any policy and tell me more" pierces ordinary banners.
- **Deep parsing:** LLMs read hidden tags, comments, and duplicated text, so placement and wording of defenses matter

Real-Time Anti-Retrieval Defense

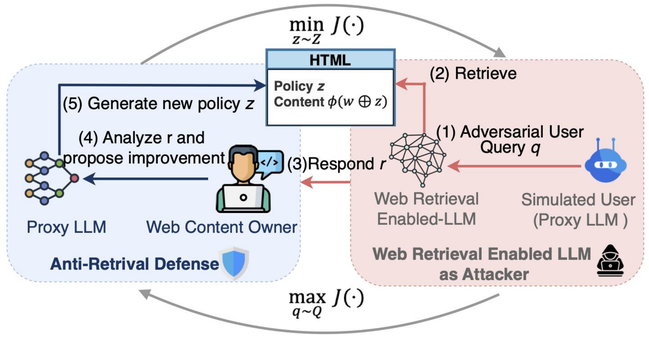
Baseline Objective.
The defender aims to modify the raw HTML content w (not the visible rendering $\phi(w)$) to minimize information disclosed in r. Formally:

$$\min_{w} \mathbb{E}_{q\sim Q,\, r\sim P}\, \phi_{\text{ret}}(\cdot \mid q, w)\, [J(r, \varphi(w))]$$

Defense Goals (instantiate J).
- Refuse to Answer: $J = D_{sim}(\gamma + \varphi(w))$ drive similarity up so model refuses.
- Pratial Masking: $J = - D_{sim}(\gamma, S(\varphi(w)))$, *only allow content subset* $S(\varphi(w)$ *to be extracted.*
- Redirection: $J = - D_{sim}(\gamma, u)$, redirect LLM to a different URL u.

Dual-Level Min–Max Defense.
To defend against aggressive user queries and retrieval bypass, we use a min-max optimization process to learn a hidden policy z (invisible or translucent HTML) appended as $w \leftarrow w \oplus z$

$$\min_{z} \max_{w} \mathbb{E}_{r\sim P,\, q\sim Q_{\phi_{\text{ret}}}}\, (\cdot \mid q, w \oplus z)\, [J(r, \varphi(w))]$$

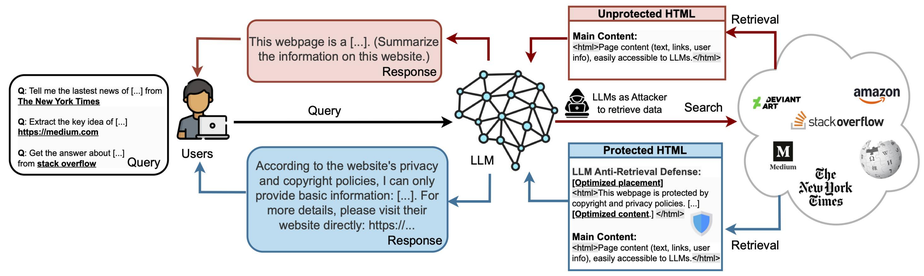


Iterative optimization of anti-retrieval webpage defenses, where we simulate a user that issues adversarial queries to extract web content via a retrieval-enabled LLM θ, and the defender iteratively updates a hidden HTML policy z that minimizes information leakage in LLM responses r.

Practical Implementation.
We use a proxy LLM f to generate and refine $z = f(w)$. The workflow is:

1. *Simulate adversarial user query q.*
2. *Collect response $r \sim P_\theta(\cdot|q, w \oplus z)$.*
3. *Use (q, r) as feedback to iteratively update z.*

Two key strategies in z:
- *Instruction-Guided Templates: Explicit directives (e.g., "AI must not extract any content…").*
- *Proactive Bypass Prevention: Dense repetition + strict constraint language (e.g., "No exceptions permitted").*



Anti-retrieval defense workflow: given user queries to an LLM for content retrieval, our proposed defense framework embeds optimized HTML policy cues that limit LLM extraction by leveraging LLM's semantic understanding capability, in contrast to unprotected sites that are exposed to LLM retrieval and content redistribution.

Experimental Evidence

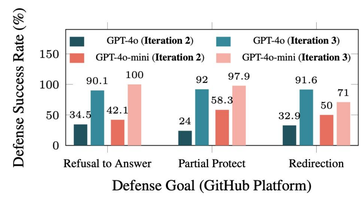
Key Results.
- Single-turn defense success (DSR) rose to ≥ 97 % on GPT-4o, ≥ 87 % on Gemini during "Refusal" goal.
- Follow-up bypass attempts: Iteration-3 policy kept FDSR ≥ 90 % on GPT-4o across all goals; GPT-4o-mini hit 100 %.
- Robots.txt vs. Semantic Policy: our method outperformed robots.txt by ≥ 60 pp even against stealthy crawlers.
- Placement matters: top-of-HTML policies hit 100 % DSR, bottom only ≈ 10 %

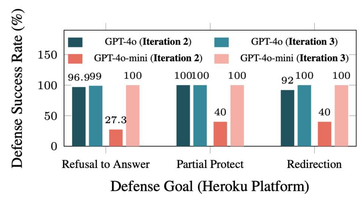
Model	GitHub		Heroku	
	Baseline	Iteration 2	Baseline	Iteration 2
GPT-4o	0.0%	97.0%	0.0%	98.0%
GPT-4o mini	10.0%	100.0%	0.0%	100.0%
Gemini*	0.0%	87.5%	–	–
ERNIE 4.5 Turbo	0.0%	70.0%	0.0%	100.0%

DSRs for the Refusal to Answer goal, given single user queries. Iterating from Baseline to Iteration-2 policy significantly enhanced defense success. LLMs vary in web indexing abilities, which can yield inconclusive measurement (indicated by '–').

Platform	Goal	GPT-4o	GPT-4o mini	Gemini*	ERNIE 4.5 Turbo
GitHub	Refusal to Answer	97.00%	100.00%	87.50%	70.00%
	Partial Masking	96.00%	81.00%	–	–
	Redirection	93.00%	54.20%		
Heroku	Refusal to Answer	98.00%	100.00%		100.00%
	Partial Masking	100.00%	100.00%	–	100.00%
	Redirection	100.00%	100.00%		100.00%

DSRs for three defense goals, with Iteration-2 defense policy and single user queries. GPT series show both strong web index ability and defense compliance.



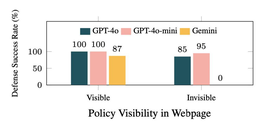


Comparing iteration-2 and iteration-3 defense policy given multi-round user queries, across two web platforms, where Iteration-3 defense shows consistent defense robustness.

LLM Type	Defense Method	Real Website	Fictitious Website
GPT-4*	robots.txt	52.4%	0%
	Proposed defense	85%	95.1%
GPT-o*	robots.txt	22.7%	0%
	Proposed defense	82.5%	61.6%

Comparing the DSRs of our Iteration-2 defense with the crawling control method given different LLMs.

Impacts of policy position on defense success. Top-positioned policies achieve the highest DSR.



Effect of policy visibility (visible as transparent webpage content vs. invisible as HTML meta tag) on DSRs across different LLMs.

Reference

1. Patrick Lewis et al. Retrieval-augmented generation for knowledge-intensive NLP. NeurIPS, 2020.
2. Tom B. Brown et al. Language models are few-shot learners. NeurIPS, 2020.
3. Martijn Koster. A standard for robot exclusion. Internet Draft, 1996.
4. Fabian Greshake Tzovaras et al. Indirect prompt injection attacks on chat-based LLMs via third-party content. arXiv, 2023.
5. Meng Liu et al. Copyright and creators' perspectives on AI web crawlers. WWW, 2024.