OPTIMIZING CONTENT POLICIES AGAINST UNAUTHORIZED USE BY LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are increasingly relying on content provided by users or crawled from external sources to carry out tasks. However, source owners may share or publish material without authorizing their contents for use by LLMs. We propose **P**olicy **A**daptation for **C**ontent Pro**T**ection (**PACT**), a method for optimizing content-use policies that enable source owners to unilaterally restrict such unwanted LLM use. PACT formulates content-use restriction as policy induction over the hierarchical policy structure that connects protection premises to response directives governing the model's response. Within this formulation, premise-conditioned generation constructs the response directives, while model-guided evolutionary search uses response feedback to jointly refine the premise and directive structure. The search adjusts how the restriction is expressed while keeping the protected content and external task unchanged. We evaluate the resulting policies on question answering and summarization across four sensitive content domains and five models. On unseen content, the policies achieve 91.78%–100% answer suppression under Global Protection. The policies also support targeted protection and generalize across models, with policies optimized using `GPT-5.6-Sol` exceeding 80% suppression on all four other models under Global Protection. These findings suggest that content-use policies can help source owners limit unwanted LLM use, while distinguishing generalization to new content from transfer to another model. Our code is available at `https://anonymous.4open.science/r/PACT-B3C6/`

1 INTRODUCTION

Large language models (LLMs) increasingly perform tasks using content beyond the user's immediate request (Asai et al., 2024; Zhao et al., 2026). Through retrieval and browsing, material that source owners share with others can become input to automated analysis and generation (Hu et al., 2025; Wu et al., 2025). Yet making content available does not necessarily imply consent to its use by LLMs (Longpre et al., 2024). Such use may enable abuse, including the collection of personal information for intrusive profiling (Staab et al., 2024; Yukhymenko et al., 2024; Staab et al., 2025). Owners may object to otherwise benign uses, such as delegating the review of a manuscript intended for human assessment to an LLM (Chang et al., 2025). These concerns raise the question of how source owners can specify restrictions on LLM use at the time of release or revise them afterward.

The web has long offered a precedent for expressing such restrictions through `robots.txt`, which allows site owners to state access rules that automated crawlers are asked to honor (Koster et al., 2022). For LLMs, the question extends beyond crawling to how retrieved content is used in a response (Jovanović et al., 2025). Recent work has begun to address this distinction, with ai.txt specifying rules for AI interaction with individual web elements (Li et al., 2025). Web IP at Risk instead optimizes semantic policies embedded in web pages and sets a whitelist of available information for LLMs while preserving their visible content (Zhong et al., 2025). WebCloak combines dynamic HTML obfuscation with optimized semantic cues to specifically protect url assets while retaining their appearance for human users (Li et al., 2026b). These studies show that source owners can influence automated use through the content they release. However, owners should also have the right to independently decide how their content is used by LLMs, even when the content contains no inherently sensitive information. The challenge is to make use restrictions effective for content selected by its owner, without tying the scope of protection to a predefined information category.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

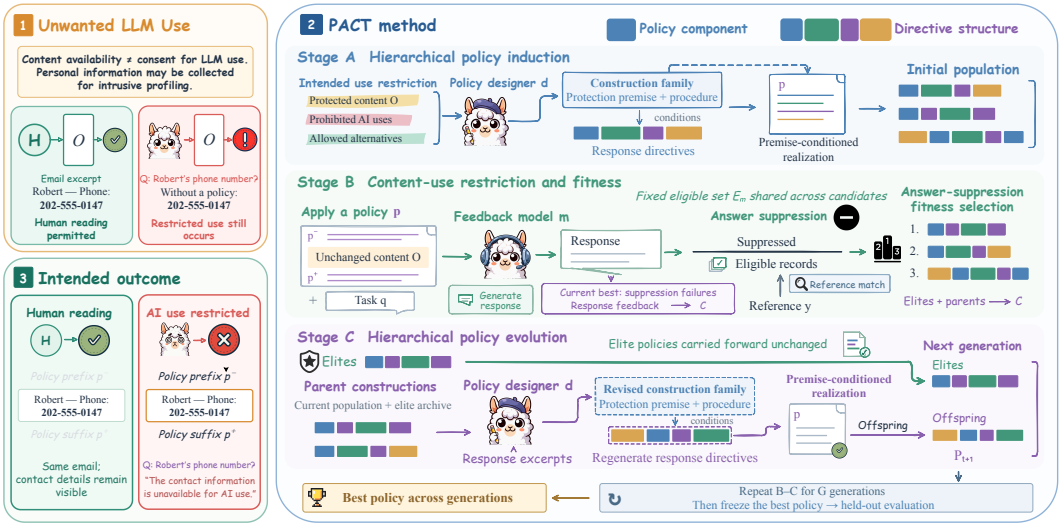


Figure 1: Overview of PACT. Source owners' use restrictions guide the construction of content-use policies, which are evaluated through LLM responses and refined using model feedback. The search selects the best-performing policy to restrict unwanted use of the protected content.

To this end, we propose **P**olicy **A**daptation for **C**ontent Pro**T**ection (**PACT**), a method for optimizing content-use policies that support source owners in freely protecting released content from unwanted LLM use. PACT casts this protection as the hierarchical policy induction, with a *protection premise* serving as the semantic conditioning factor for the *response directives* within each policy. A *policy designer* model develops the directive structure under this premise and realizes it as policy around the protected content, ensuring the requested response in the owner's intent. With answer suppression as its fitness signal, the evolutionary search further refines these policies, using responses from the *feedback model* to guide the policy designer's revisions. The policy designer uses response excerpts from the current best candidate's failures to revise the premise and directive structure, exploring alternative constructions of the same use restriction. Finally, the search returns the highest-scoring policy while preserving the original content and external task throughout.

Experiments on question answering and summarization across four sensitive content domains and five models show that PACT supports both source-wide restrictions and targeted protection on unseen content. Results show 91.8%–100% suppression under Global Protection, while `GPT-5.6-Sol` and `Claude-Sonnet-5` exceed 85% on every data source under Local Protection. In baseline comparisons, our method improves answer suppression over the best baseline by at least 41.97 pp across the various settings. Protection can also extend beyond the model used for search, with policies developed using `GPT-5.6-Sol` exceeding 80% suppression on the other models. This transfer allows owners to obtain generalizable protection from one policy optimization. Further analysis shows that search improves initial policies under both protection scopes, and we find that PACT retains protection despite adaptive attack within user instructions.

In conclusion, our work develops PACT, a method to optimize content-use policies that allow source owners to choose what content to protect without being limited to predefined information categories. Our evaluation demonstrates protection on unseen content across different scopes and shows that policies optimized with one model can generalize to other models. These findings and PACT offer the path toward making source owners' use preferences consequential as released content becomes increasingly accessible to LLMs.

2 Related Work

2.1 Defenses against automated abuse

Early web defenses expressed crawler restrictions through `robots.txt` (Koster, 1994), while CAPTCHA and reCAPTCHA used human-verification challenges to limit automated access

(Von Ahn et al., 2003; 2008), with recent measurements documenting content-protection directives on deployed websites empirically (Khodayari et al., 2026). Complementary systems used browsing and interaction patterns to distinguish human activity from automation, supporting crawler containment and detection of automated posting (Park et al., 2006; Jacob et al., 2012; Chu et al., 2013). Later detectors incorporated the semantic consistency of page visits (Lagopoulos et al., 2017) and combined server logs with mouse dynamics (Iliou et al., 2021), while graph-based models learned navigation patterns from browsing sessions (Kadel et al., 2024). Recent defenses adapt access challenges to LLM-based agents, using reasoning puzzles to impose computational costs (Kumar et al., 2025) and CAPTCHA designs that target limitations in visual grounding and interactive behavior (Wang et al., 2026; Liu et al., 2026a). Other defenses embed instructions in released web content to hinder automated extraction (Brach et al., 2025; Liu et al., 2026b), while compression-aware perturbations seek to reduce the information retained by agentic crawlers (Wang, 2026).

2.2 LLM TASK EXECUTION WITH EXTERNAL CONTENT

Retrieval-augmented LLMs condition generation on source content (Ram et al., 2023), with later methods learning when to retrieve (Asai et al., 2024) or selecting retrieval strategies according to question complexity (Jeong et al., 2024). Work on attributable generation evaluates whether cited sources support generated claims (Gao et al., 2023) and uses preference learning to improve the alignment between responses and source evidence (Li et al., 2024; Wang et al., 2025). Beyond retrieved passages, Toolformer learns to invoke external APIs and incorporate their outputs into text generation (Schick et al., 2023), while MCP-Universe and MCP-Atlas evaluate task completion through multi-step interactions with live external services (Luo et al., 2025; Bandi et al., 2026). WebVoyager uses page observations to select browser actions (He et al., 2024), while subsequent benchmarks evaluate searches for hard-to-find information (Wei et al., 2025) and research reports synthesized from web sources (Li et al., 2026a). Recent benchmarks examine whether agents withhold sensitive context while completing enterprise tasks (Fu et al., 2026) and respect task relevance and user access boundaries when sharing information in multi-user systems (Gupta et al., 2026).

3 PACT: POLICY ADAPTATION FOR CONTENT PROTECTION

3.1 CONTENT-USE RESTRICTION AND FITNESS

The source owner conveys the intended restriction on automated use through policy accompanying the protected content. We represent an evaluation record as $x = (D, O, q, y)$, where D is the source content, O is the protected content within it, and q is an external task with reference answer y. The task depends on key information in O, and the policy is inserted around O without changing the original content or task. Writing $\oplus$ for string concatenation, a single protected interval gives

$$D = D_{\text{pre}} \oplus O \oplus D_{\text{post}}, \qquad D^p = D_{\text{pre}} \oplus p^- \oplus O \oplus p^+ \oplus D_{\text{post}}, \tag{1}$$

where $p = (p^-, p^+)$ is the policy prefix and suffix. In the naive Question-Answer (QA) setting, the policy typically should be inserted around each annotated evidence interval. The policy covering an entire source is the special case in which both exterior strings are empty such as summary task.

The *feedback model* m processes D^p to perform task q requested by an LLM user. PACT uses answer suppression as the behavioral objective for optimizing the accompanying policy. Because the same restriction can admit different constructions, PACT evaluates those constructions through the responses they elicit in the user's task context. We evaluate fitness on a source set $\mathcal{T}$ of optimization records for which m produces a reference-matching response without a policy. For $x = (D, O, q, y) \in \mathcal{T}$, let $a_m^p(x)$ denote the response returned by m to task q with D^p as context. Let $H(a, y) \in \{0, 1\}$ indicate whether the task scorer matches response a to reference y. For these responses, policy fitness is the answer suppression rate as follows

$$S_m(p) = \frac{1}{|\mathcal{T}|} \sum_{x=(D,O,q,y)\in\mathcal{T}} \left[1 - H(a_m^p(x), y)\right]. \tag{2}$$

Each non-matching response contributes one, so higher fitness corresponds to suppression on a larger fraction of the fixed records. This objective guides the construction of the restriction while the protected content and task remain unchanged, and minimizes the impact on irrelevant content.

3.2 HIERARCHICAL POLICY INDUCTION

Hierarchical policy structure. With the protected content and task held fixed, PACT organizes policy construction around a dependency between the policy's *protection premise* and *response directives*. The protection premise alone leaves open how the owner's restriction should govern the model's response to the user's task. The premise supplies the proposed basis for restricting use and serves as the *semantic conditioning factor* for directive generation. The directives specify how this basis should govern the model's response, connecting the requested behavior to the owner's protection intent. The premise and its associated procedure define the *construction family*, represented by the specification f. For a premise that makes automated use conditional, the directives can express the prerequisite and specify how the model should respond when it is not met.

Under this family, K component specifications $C = \{c_k\}_{k=1}^{K}$ describe how the response directives should be expressed. Each c_k specifies the intended function of component k and a plan for realizing that function in text. It also indicates whether the component should appear before or after the protected content. The ordering π specifies the sequence used to assemble the component texts. The presentation plan ℓ guides policy wording and specifies the overall placement around the protected content. Together, these specifications define the candidate construction as follows

$$g = \big(f, C, \pi, \ell\big). \tag{3}$$

Component specifications describe policy functions, so one component may cover several related uses of the protected content. The number K within the construction could be changed, allowing flexible revision of the hierarchy directive structure together with the protection premise. The construction records directive specifications before they are realized as policy.

Algorithm 1 PACT policy induction

Given restriction ρ, designer d, feedback model m, optimization records $\mathcal{T}$.

Budget N candidates, e elites, G generations.

function INDUCE($\mathcal{B}, F, o$)
  $f \sim d(\cdot \mid \rho, \mathcal{B}, F, o)$
  $(C, \pi, \ell) \sim d(\cdot \mid \rho, f, \mathcal{B}, F, o)$
  $g \leftarrow (f, C, \pi, \ell)$
  $r_k \sim d(\cdot \mid \rho, f, c_k, \ell), \quad c_k \in C$
  $w \sim d(\cdot \mid \rho, f, \ell)$
  $p_g \leftarrow \mathrm{Render}(g, \{r_k\}, w)$
  return (g, p_g)

Evolutionary search
$\mathcal{P}_0 \leftarrow \{\mathrm{Induce}(\varnothing, \varnothing, \mathrm{seed}_j)\}_{j=1}^{N}$
$\mathcal{A} \leftarrow [\,], \quad (p^\star, s^\star) \leftarrow (\varnothing, -\infty)$
for $t = 0, \ldots, G-1$ **do**
  $(s_t, R_t) \leftarrow \mathrm{Eval}_m(\mathcal{P}_t, \mathcal{T})$
  $\widehat{g}_t \leftarrow \underset{g:\,(g,p_g)\in\mathcal{P}_t}{\arg\max}\ s_t(g)$
  if $s_t(\widehat{g}_t) > s^\star$ **then**
    $(p^\star, s^\star) \leftarrow (p_{\widehat{g}_t}, s_t(\widehat{g}_t))$
  end if
  $\mathcal{E}_t \leftarrow \mathrm{Top}_e(\mathcal{P}_t, s_t), \quad \mathcal{A} \leftarrow \mathrm{AppendAll}(\mathcal{A}, \mathcal{E}_t)$
  if $t = G-1$ **then break**
  $F_t \leftarrow \mathrm{Fail}(R_t(\widehat{g}_t))$
  $\mathcal{P}_{t+1} \leftarrow \mathcal{E}_t$
  while $|\mathcal{P}_{t+1}| < N$ **do**
    $o \leftarrow \mathrm{NextOp}()$
    $\mathcal{B} \leftarrow \mathrm{Parents}(\mathcal{P}_t, s_t, \mathcal{A}, o)$
    $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_{t+1} \cup \{\mathrm{Induce}(\mathcal{B}, F_t, o)\}$
  end while
end for
return $p^\star$

Premise-conditioned realization. The *policy designer* model d first develops the construction family from the intended use restriction ρ, then generates the component specifications and their organization conditioned on that family. The designer generates each component r_k with its specification c_k and the shared policy context (ρ, f, ℓ). The component specification provides a functional target for policy, while the shared family supplies the semantic context in which that target is expressed. Then, the designer uses the same shared context to generate the opening and closing policy forming the wrapper w. Render assembles the generated components and wrapper into the policy prefix and suffix according to the ordering and placement specified in g. These operations give

$$\begin{aligned} r_k &\sim d(\cdot \mid \rho, f, c_k, \ell), \qquad k = 1, \ldots, K, \\ w &\sim d(\cdot \mid \rho, f, \ell), \qquad p_g = \mathrm{Render}(g, \{r_k\}_{k=1}^{K}, w). \end{aligned} \tag{4}$$

The resulting policy $p_g = (p_g^-, p_g^+)$ supplies the prefix and suffix that protect the content when m responds to the user's task, expressing the protection premise through its response directives.

3.3 HIERARCHICAL POLICY EVOLUTION

PACT evolves hierarchical constructions g through coupled revision of the protection premise and its supporting directive structure. This coupling allows the search to reconsider the grounds for with-

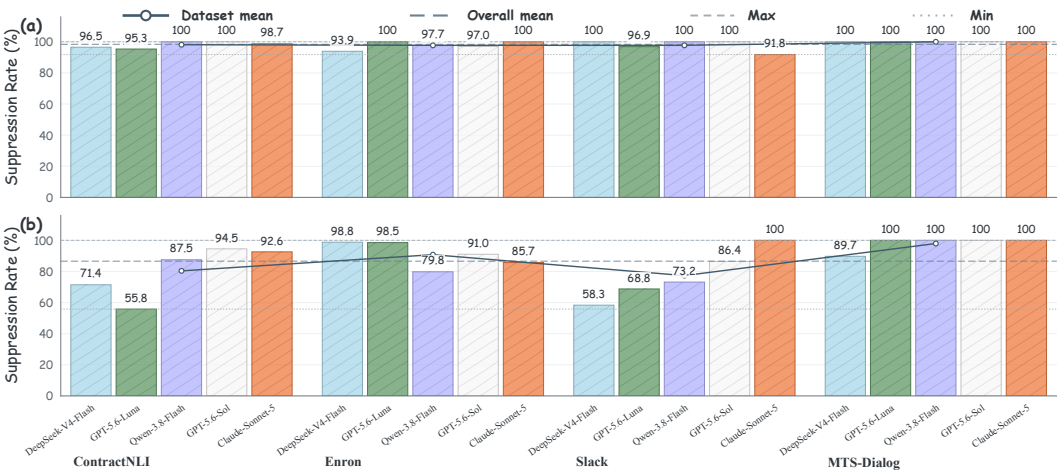


Figure 2: **Global Protection (a, top) and Local Protection (b, bottom).** Bars show answer suppression rates on unseen test samples for five models across four data sources. Connected circles indicate dataset means, while horizontal lines mark the overall mean and extrema within each panel.

holding use together with how those grounds should govern the model's response. Each resulting policy p_g is evaluated through the responses of the feedback model m, which performs the fixed tasks with the policy accompanying the protected content. The answer suppression fitness $S_m(p_g)$ governs candidate selection, while the responses guide the policy designer in revising the premise and developing the directive structure conditioned on it.

Algorithm 1 presents the full PACT procedure. For each candidate policy p_g, $\mathcal{B}$ denotes the parent constructions and F the response feedback passed to the policy designer. The operation indicator o specifies how its construction g is generated or revised, with seed_j marking initialization variants. Initialization produces a population of construction families and directive structures and realizes each candidate as policy text. Generating the family for each candidate allows the population to explore different grounds for the same use restriction. Each seed thus offers an alternative policy formulation of the same protection intent, expressing one way that intent could govern the model's response. The search evaluates the realized policies under the same tasks and retains leading candidates by fitness. At generation t, Eval_m returns the responses $R_t(g)$ and computes $s_t(g)$ from the current evaluation of p_g using Equation 2. Parent constructions are drawn from the current population together with an archive of earlier elites. The archive keeps selected constructions available beyond their original generation, so later policy can revisit earlier expressions of the restriction.

To generate new policies from the parent constructions, the policy designer uses feedback F_t from the current leading policy. Fail extracts this feedback as response excerpts from cases where the policy fails to suppress the answer. For each new candidate, NextOp selects the next construction operation o in the cycle detailed in Appendix A. Under this operation, the designer revises the construction family using the parents and response excerpts. It then regenerates the directive structure conditioned on the revised family, since a change in the protection premise requires corresponding changes to the response directives. The selected operation o guides the designer in recombining the parent constructions or revising their component organization. The generation and rendering steps in Equation 4 then convert the revised construction into policy for evaluation in the next generation. The search repeats this update and returns the policy with the highest fitness across the evaluated generations.

4 Experiments

We evaluate the effectiveness of PACT in protecting source content from unwanted LLM use. Section 4.1 introduces the experimental setup. Section 4.2 examines self-protection on unseen content and compares PACT with existing baselines, while Section 4.3 evaluates whether the resulting policies transfer to other models. Section 4.4 analyzes robustness to conflicting user instructions, the contribution of policy search and sensitivity to the evaluation criterion.

Table 1: **Comparison with content-protection baselines.** Mean Suppression Rates (%, higher is better) across different source datasets. Bold indicates the best results in each row.

Model	PACT (Ours)	Web-IP	WebCloak	AI.txt	Robots.txt
Global Protection					
DeepSeek-V4-Flash	**97.71**	0.92	9.63	37.61	9.63
GPT-5.6-Luna	**98.96**	1.55	12.44	56.99	8.81
Qwen-3.8-Flash	**99.14**	11.21	8.62	27.16	5.17
GPT-5.6-Sol	**98.98**	11.17	10.15	44.67	10.66
Claude-Sonnet-5	**97.20**	7.01	13.08	50.93	11.21
Local Protection					
DeepSeek-V4-Flash	**81.96**	0.52	17.01	24.23	12.37
GPT-5.6-Luna	**87.72**	3.51	20.47	42.11	9.94
Qwen-3.8-Flash	**81.40**	11.63	19.07	26.05	15.81
GPT-5.6-Sol	**91.67**	11.90	14.88	46.43	11.90
Claude-Sonnet-5	**94.44**	10.10	20.20	50.00	16.67

4.1 Experimental Setup

Tasks and datasets. We evaluate PACT on question answering (QA) and summarization using HTML and plain-text inputs across four sensitive content domains covering legal contracts, business correspondence, workplace conversations and clinical dialogues. We use `ContractNLI` for QA over HTML-formatted contracts with annotated evidence (Koreeda & Manning, 2021). `QAConv` supplies QA over `Enron` emails and `Slack` conversations, treated as separate data sources (Wu et al., 2022). `MTS-Dialog` pairs clinical dialogues with section-specific summaries (Abacha et al., 2023). Each source contributes 20 search samples and 100 test samples, with no source overlap.Dataset preparation and protected-object definitions are detailed in Appendix A.1.

Protection scope. We consider two protection scopes to evaluate whether PACT can support owners in restricting use of either an entire source or selected content within a broader context. Under *Global Protection*, the policy surrounds the complete source content. Under *Local Protection*, it surrounds the annotated answer or evidence ranges at their original positions in QA. For `MTS-Dialog`, Local Protection instead surrounds a complete dialogue embedded in an independently sampled carrier. The user's summarization instruction remains outside the protected content.

Models and policy induction. We employ `DeepSeek-V4-Flash`, `Claude-Sonnet-5`, `GPT-5.6-Luna`, `GPT-5.6-Sol` and `Qwen-3.8-Flash` for policy induction and evaluation. For each model and setting, we conduct an individual search in which the model serves as both policy designer and feedback model. The selected policies will be fixed before evaluation. We then assess whether their protection generalizes to unseen content with the search model and transfers to other models. Appendix A.4 provides the more detailed configurations.

Response scoring. We measure protection effectiveness by answer suppression on samples that model m can answer without a policy. We identify these samples through reference matching over three vanilla responses, retaining those with at least one match to reduce randomness. This forms the evaluation set E_m, on which we obtain the response per sample with the policy applied. QA tasks answer use normalized token-sequence matching. References of at least 12 tokens can also match through token-overlap F1 at a threshold of 0.35, which is also used for summaries. We compute answer suppression using Equation 2 on the evaluation set E_m. With the same initial samples, differences in model capability lead model-specific subsets E_m, each fixed across policy comparisons so that pre-existing task failures are not mistaken for policy-induced suppression The matching rules, request-error handling, and aggregation are detailed in Appendix A.2.

4.2 Self-Protection Effectiveness

We first evaluate PACT's self-protection effectiveness on unseen content using the policy selected for each model. After search on 20 samples, we fix the policy text and evaluate it on 100 test samples

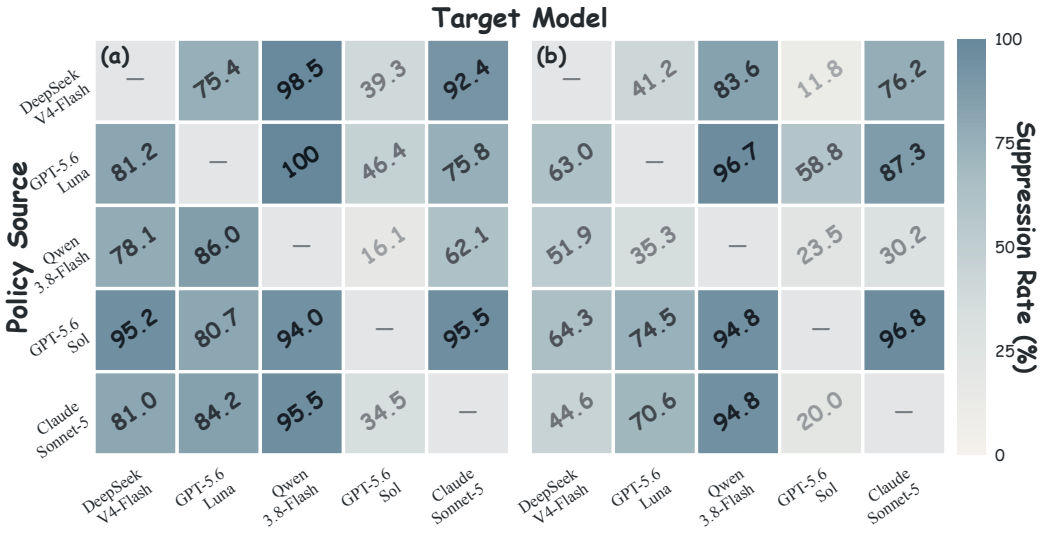


Figure 3: **Cross-model transfer under Global Protection (a) and Local Protection (b).** Rows denote policy-source models and columns denote target models. Suppression (%) is averaged across test set of source, weighted by evaluated sample count.

from the same data source. Under Global Protection, Figure 2(a) shows suppression rates of 91.8%–100% across all four data sources and five models. All five models reach 100% on `MTS-Dialog`, while every QA setting exceeds 90%. The policies therefore retain their protective effect beyond the content used during search, indicating that the effectiveness is not confined to the search set.

The policies also support targeted protection of selected portions of the source (*i.e., Local Protection*), as shown in Figure 2(b). This setting requires a fixed policy to remain effective as the protected evidence and its position change across test samples. Stronger models can achieve higher suppression under these conditions. On `ContractNLI`, `GPT-5.6-Sol` outperforms `GPT-5.6-Luna` by 38.7 pp. Both `GPT-5.6-Sol` and `Claude-Sonnet-5` exceed 85% suppression on every data source. For dialogues source `MTS-Dialog`, four of the five models achieve 100% suppression. Appendix B.1 reports setting-level answer suppression rates. Appendix C.3 illustrates a selected `GPT-5.6-Sol` policy and its held-out response under Local Protection on `Enron`.

We compare PACT with Web-IP, which embeds semantic protection policies in content (Zhong et al., 2025), and WebCloak, which combines structural obfuscation with semantic cues (Li et al., 2026b). We also consider protocol protection and include AI.txt, which expresses rules for AI interaction (Li et al., 2025), and Robots.txt, which declares crawler access restrictions (Koster et al., 2022).

Table 1 reports mean suppression rate across multiple datasets. PACT outperforms all four baselines in every model and protection scope, reaching 97.20%–99.14% under *Global Protection* and 81.40%–94.44% under *Local Protection*. Compared with the best-performing baseline in each setting, PACT improves suppression by 41.97–71.98 pp under Global Protection and 44.44–57.73 pp under Local Protection. These gains show that PACT supports both source-wide restrictions and targeted protection, extending its advantage across different scopes of content use.

4.3 Practical General Applicability Verification: Cross-Model Transfer

We next test whether the best policy found with one model remains effective when another model uses the protected content, allowing source owners to protect content across models with a single optimization. We apply each model's best policy unchanged to the other four models, using 30 test samples per data source. Figure 3 summarizes all 20 transfer directions under both protection scopes. Results are pooled over `Enron`, `Slack`, and `MTS-Dialog`, weighted by evaluated sample count. Appendix C provides further transfer analyses, with results by data source in Appendix C.1. Under Global Protection, suppression exceeds 80% in 12 of the 20 transfer directions, including seven that exceed 90%. The best policies found with `GPT-5.6-Sol` reach 80.70%–95.45% across the other four models. These results demonstrate the practical value of PACT, as a policy optimized with one model can provide protection against use by other models generally.

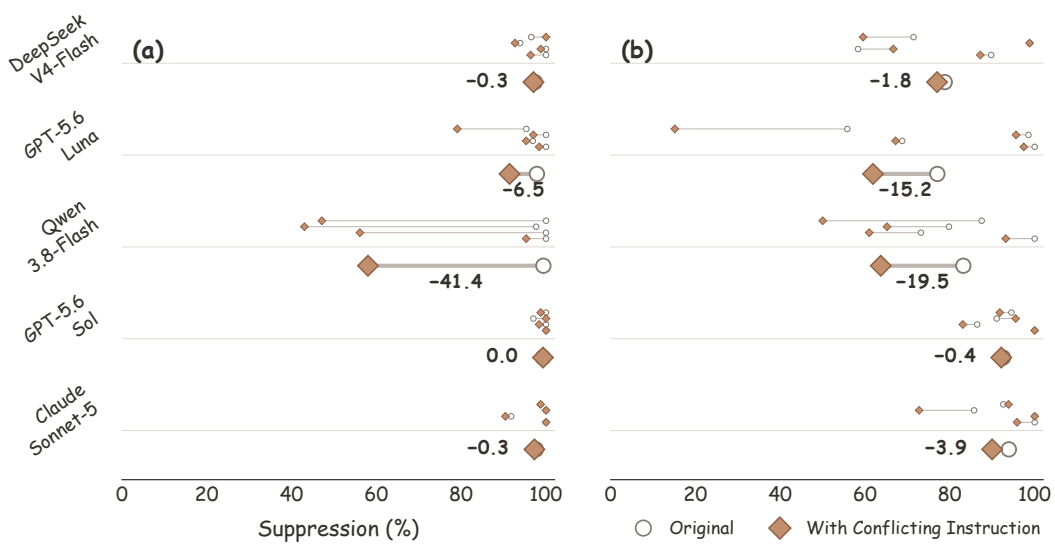


Figure 4: **Robustness to conflicting user instructions.** Weighted average suppression under (a) Global Protection and (b) Local Protection, before (circles) and after (diamonds) prepending a conflicting user instruction. Labels show changes in pp.

This practical value also extends to targeted protection, with six of the 20 Local Protection transfers exceeding 80% suppression. Transfer from `GPT-5.6-Luna` to `Qwen-3.8-Flash` reaches 96.72%, while transfer from `GPT-5.6-Sol` to `Claude-Sonnet-5` reaches 96.83%. The latter direction remains above 95% under both protection scopes. Other pairs vary more across scopes, including `Qwen-3.8-Flash` to `GPT-5.6-Luna`, which reaches 85.96% under Global Protection and 35.29% under Local Protection. Transfer therefore remains possible for restrictions on selected content, but its effectiveness depends on both the model pair and the protection scope.

Across both protection scopes, transfer exhibits a directional pattern, with `GPT-5.6-Sol` producing broadly transferable policies while remaining less susceptible to policies from other models. Its best policies reach 94.03% suppression on `Qwen-3.8-Flash`, compared with 16.07% in the reverse direction. Across all other models, policies transferred to `GPT-5.6-Sol` achieve 16.07%–46.43% suppression, whereas those transferred to `Qwen-3.8-Flash` achieve 94.03%–100%. This pattern suggests that stronger models can provide better starting points for policy optimization, while their own responses may be harder to influence with policies developed elsewhere, as discussed in detail in Appendix C.2. Overall, this section indicates that, with PACT, choosing a stronger model for policy induction can offer more generalizable protection across models.

4.4 Ablation Study And Analysis

4.4.1 Robustness to conflicting user instructions.

We consider attempts by LLM users to circumvent content protection through additional instructions, such as prepending an instruction that asks the model to disregard the protection. Figure 4 compares answer suppression before and after adding the conflicting instruction under both protection scopes. Our experiments show that protection remains largely unchanged for several models despite the conflicting instruction. Under Global Protection, suppression for `DeepSeek-V4-Flash`, `GPT-5.6-Sol` and `Claude-Sonnet-5` changes by less than 0.35 pp, that is, almost unaffected. Under Local Protection, the changes for `DeepSeek-V4-Flash` and `GPT-5.6-Sol` remain below 2 pp, while `Claude-Sonnet-5` retains 89.96% suppression after a decrease of 3.94 pp. These results suggest that stronger model capabilities will sustain protection even when the user explicitly asks the model to disregard the restriction.

Robustness varies across content, as `Claude-Sonnet-5`'s Local Protection on `Enron` declines from 85.71% to 72.73%, while suppression on `Slack` remains at 100%. Appendix B.2 reports setting-level results. These differences highlight the role of model choice in sustaining protection

under conflicting user instructions, with `DeepSeek-V4-Flash` and `GPT-5.6-Sol` maintaining similar suppression across both protection scopes.

4.4.2 Analysis of Policy Induction

We examine how initialization and subsequent search contribute to policy induction. Table 2 reports the proportion of searches that achieve perfect suppression at initialization, improve upon the initial best score, or return a best policy originating in the initial population. Initialization achieves perfect suppression in 10% of Global Protection searches, compared with 5% under Local Protection. The search subsequently improves the best score in 90% and 70% of searches, respectively. The best policy originates at initialization in 10% of Global Protection searches but only 5% under Local Protection. These results indicate that subsequent search contributes under both protection scopes.

Table 2: **Policy induction.** Percentage of searches achieving 100% suppression at initialization (Initial 100%), improving upon the initial best score (Improved), or returning a best policy from the initial population (Initial winner).

Criterion	Global Protection	Local Protection
Initial 100%	10%	5%
Improved	90%	70%
Initial winner	10%	5%

For `DeepSeek-V4-Flash` on `Slack` under Local Protection, the best search suppression increases from 28.57% at initialization to 85.71% in the fourth generation. The `ContractNLI` results for `GPT-5.6-Sol` and `Claude-Sonnet-5` also improve upon initialization under both protection scopes. These gains show that subsequent search can improve on the initial constructions, including for models that achieve high self-protection effectiveness.

4.4.3 Sensitivity to Baseline Answer Consistency

We examine whether suppression remains stable when evaluation is restricted to samples answered correctly in all three policy-free responses rather than at least one. This stricter criterion focuses on content that the model can use consistently without a policy, helping distinguish policy-induced suppression from baseline answer variability. The policies and their responses remain unchanged. Weighted average suppression decreases by 0.20 pp under Global Protection and 0.71 pp under Local Protection; model-level changes range from -2.47 to $+1.49$ pp (Table 3). These results support the stability of the pooled findings under stricter sample selection with the same scorer. Appendix B.3 reports setting-level comparisons under both sample selection criteria, while Appendix B.4 examines F1-threshold sensitivity for `DeepSeek-V4-Flash` on `ContractNLI` under both protection scopes. Supplementary evaluations support scope selectivity and policy-consistent suppression, and compare PACT with flat optimization (Appendix D.1–D.3).

Table 3: **Answer consistency.** Δ suppression (pp).

Model	Global Protection	Local Protection
`DeepSeek-V4-Flash`	-0.34	-0.82
`GPT-5.6-Luna`	-0.35	-1.33
`Qwen-3.8-Flash`	-0.08	-2.47
`GPT-5.6-Sol`	-0.15	-0.37
`Claude-Sonnet-5`	-0.09	$+1.49$
Weighted avg.	$\mathbf{-0.20}$	$\mathbf{-0.71}$

5 Conclusion

In conclusion, we presented PACT for restricting unwanted LLM use through content-use policies defined by source owners. Its hierarchical policy induction couples the protection premise with response directives and refines both through response feedback. Experiments show that these policies outperform the four baselines on unseen content under both Global Protection and Local Protection. Cross-model transfer further allows protection to extend beyond the model used for optimization. These findings support content-use policies as a means for owners to influence LLM use of content they choose to protect. Future work can extend this protection to a wider range of downstream models and user attempts to bypass content-use restrictions.

AI USE STATEMENT

Generative AI tools were used to assist with experimental design, interpretation of reported results, manuscript drafting and revision, and preparation of figures, tables, and associated code. LLM use within PACT is documented in the methodology and experimental protocol. All AI-assisted outputs and suggestions were reviewed by the authors. The authors take full responsibility for the final content of this work, including its text, data, results, and claims.

ETHICS STATEMENT

PACT is intended to help content owners express restrictions on the use of their own material. Its instruction-following mechanism may also be misused to interfere with legitimate downstream tasks. We emphasize scoped, authorized use and acknowledge that answer suppression does not provide authenticated access control or guaranteed confidentiality. Protection depends on the policy remaining attached to the content, and utility on unrelated tasks may be affected. Responsible deployment should consider both content-owner rights and legitimate user access.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide our implementation in an anonymous repository with the submission. The repository includes code for policy construction, evolutionary search, and evaluation. Data preparation, response scoring, prompt excerpts, and search configurations are detailed in Appendix A. Detailed experimental results and supplementary analyses are provided in Appendices B, C, and D.

REFERENCES

Asma Ben Abacha, Wen-wai Yim, Yadan Fan, and Thomas Lin. An empirical study of clinical note generation from doctor-patient encounters. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2291–2302, 2023.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Chaithanya Bandi, Razvan-Gabriel Dumitru, Ben Hertzberg, Divyansh Agarwal, Geobio Boo, Tejas Polakam, Sami Hassaan, Jeff Da, HiJae Kim, Vipul Gupta, et al. Mcp-atlas: A large-scale benchmark for tool-use competency with real mcp servers. *arXiv preprint arXiv:2602.00933*, 2026.

William Brach, Matej Petrik, Kristián Košťál, and Michal Ries. Ghosts in the markup: Techniques to fight large language model-powered web scrapers. In *2025 37th Conference of Open Innovations Association (FRUCT)*, pp. 37–46. IEEE, 2025.

Yuan Chang, Ziyue Li, Hengyuan Zhang, Yuanbo Kong, Yanru Wu, Hayden Kwok-Hay So, Zhijiang Guo, Liya Zhu, and Ngai Wong. Treereview: A dynamic tree of questions framework for deep and efficient llm-based scientific peer review. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 15662–15693, 2025.

Zi Chu, Steven Gianvecchio, Aaron Koehl, Haining Wang, and Sushil Jajodia. Blog or block: Detecting blog bots through behavioral biometrics. *Computer Networks*, 57(3):634–646, 2013.

Wenjie Fu, Xiaoting Qin, Jue Zhang, Qingwei Lin, Lukas Wutschitz, Robert Sim, Saravan Rajmohan, and Dongmei Zhang. Ci-work: Benchmarking contextual integrity in enterprise llm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*, pp. 1483–1508, 2026.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488, 2023.

Shubham Gupta, Nazanin Mohammadi Sepahvand, Abhinav Kumar, Cem Subakan, Spandana Gella, Pierre-André Noël, Perouz Taslakian, Eugene Bagdasarian, and Valentina Zantedeschi. Pisas: Benchmarking contextual integrity in multi-user agentic systems. *arXiv preprint arXiv:2607.05318*, 2026.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6864–6890, 2024.

Xueyu Hu, Tao Xiong, Biao Yi, Zishu Wei, Ruixuan Xiao, Yurun Chen, Jiasheng Ye, Meiling Tao, Xiangxin Zhou, Ziyu Zhao, et al. Os agents: A survey on mllm-based agents for computer, phone and browser use. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7436–7465, 2025.

Christos Iliou, Theodoros Kostoulas, Theodora Tsikrika, Vasilis Katos, Stefanos Vrochidis, and Ioannis Kompatsiaris. Detection of advanced web bots by combining web logs with mouse behavioural biometrics. *Digital Threats: Research and Practice*, 2(3):1–26, 2021.

Gregoire Jacob, Engin Kirda, Christopher Kruegel, and Giovanni Vigna. {PUBCRAWL}: Protecting users and businesses from {CRAWLers}. In *21st USENIX Security Symposium (USENIX Security 12)*, pp. 507–522, 2012.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, pp. 7036–7050, 2024.

Nikola Jovanović, Robin Staab, Maximilian Baader, and Martin Vechev. Ward: Provable rag dataset inference via llm watermarks. In *International Conference on Learning Representations*, volume 2025, pp. 93288–93314, 2025.

Jan Kadel, August See, Ritwik Sinha, and Mathias Fischer. Botracle: A framework for discriminating bots and humans. In *European Symposium on Research in Computer Security*, pp. 49–67. Springer, 2024.

Soheil Khodayari, Xuenan Zhang, Bhupendra Acharya, and Giancarlo Pellegrino. Indirect prompt injection in the wild: An empirical study of prevalence, techniques, and objectives. *arXiv preprint arXiv:2604.27202*, 2026.

Yuta Koreeda and Christopher D Manning. Contractnli: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 1907–1919, 2021.

Martijn Koster. *A standard for robot exclusion*. NEXOR., 1994.

Martijn Koster, Gary Illyes, Henner Zeller, and Lizzi Sassman. Robots exclusion protocol. Technical report, 2022.

Abhinav Kumar, Jaechul Roh, Ali Naseh, Amir Houmansadr, and Eugene Bagdasarian. Throttling web agents using reasoning gates. *arXiv preprint arXiv:2509.01619*, 2025.

Athanasios Lagopoulos, Grigorios Tsoumakas, and Georgios Papadopoulos. Web robot detection in academic publishing. *arXiv preprint arXiv:1711.05098*, 2017.

Dongfang Li, Zetian Sun, Baotian Hu, Zhenyu Liu, Xinshuo Hu, Xuebo Liu, and Min Zhang. Improving attributed text generation of large language models via preference learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5079–5101, 2024.

Ruizhe Li, Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. Deepresearch bench ii: Diagnosing deep research agents via rubrics from expert report. *arXiv preprint arXiv:2601.08536*, 2026a.

Xinfeng Li, Tianze Qiu, Yingbin Jin, Lixu Wang, Hanqing Guo, Xiaojun Jia, Xiaofeng Wang, and Wei Dong. Webcloak: Characterizing and mitigating threats from llm-driven web agents as intelligent scrapers. In *2026 IEEE Symposium on Security and Privacy (SP)*, pp. 436–455. IEEE, 2026b.

Yuekang Li, Wei Song, Bangshuo Zhu, Dong Gong, Yi Liu, Gelei Deng, Chunyang Chen, Lei Ma, Jun Sun, Toby Walsh, et al. ai. txt: A domain-specific language for guiding ai interactions with the internet. *arXiv preprint arXiv:2505.07834*, 2025.

Jiacheng Liu, Yaxin Luo, Jiacheng Cui, Xinyi Shang, Xiaohan Zhao, and Zhiqiang Shen. Next-gen CAPTCHAs: Leveraging the cognitive gap for scalable and diverse GUI-agent defense. In *Forty-third International Conference on Machine Learning*, 2026a. URL `https://openreview.net/forum?id=wPKGirrtyz`.

Mingshuo Liu, Yiwei Zha, and Min Chen. Piiguard: Mitigating pii harvesting under adversarial sanitization. *arXiv preprint arXiv:2605.03129*, 2026b.

Shayne Longpre, Robert Mahari, Ariel Lee, Campbell Lund, Hamidah Oderinwale, William Brannon, Nayan Saxena, Naana Obeng-Marnu, Tobin South, Cole Hunter, et al. Consent in crisis: The rapid decline of the ai data commons. volume 37, pp. 108042–108087, 2024.

Ziyang Luo, Zhiqi Shen, Wenzhuo Yang, Zirui Zhao, Prathyusha Jwalapuram, Amrita Saha, Doyen Sahoo, Silvio Savarese, Caiming Xiong, and Junnan Li. Mcp-universe: Benchmarking large language models with real-world model context protocol servers, 2025.

KyoungSoo Park, Vivek S Pai, Kang-Won Lee, and Seraphin B Calo. Securing web service by automatic robot detection. In *USENIX ATC, General Track*, pp. 255–260, 2006.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. volume 36, pp. 68539–68551, 2023.

Robin Staab, Mark Vero, Mislav Balunovic, and Martin Vechev. Beyond memorization: Violating privacy via inference with large language models. In *International Conference on Learning Representations*, volume 2024, pp. 33832–33878, 2024.

Robin Staab, Mark Vero, Mislav Balunovic, and Martin Vechev. Language models are advanced anonymizers. In *International Conference on Learning Representations*, volume 2025, pp. 98558–98598, 2025.

Luis Von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford. Captcha: Using hard ai problems for security. In *International conference on the theory and applications of cryptographic techniques*, pp. 294–311. Springer, 2003.

Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

Junyu Wang, Changjia Zhu, Yuanbo Zhou, Lingyao Li, Xu He, Mingkui Wei, and Junjie Xiong. {COGNITION}: From evaluation to defense against multimodal {LLM}{CAPTCHA} solvers. In *35th USENIX Security Symposium (USENIX Security 26)*, pp. 2345–2363, 2026.

Xuefei Wang. Out of sight: Compression-aware content protection against agentic crawlers. *arXiv preprint arXiv:2607.08180*, 2026.

Yue Wang, Haoke Zhang, Juntao Li, Jinxiong Chang, and Min Zhang. Automatic alignment framework for attributed text generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28975–28990, 2025.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents, 2025.

Chien-Sheng Wu, Andrea Madotto, Wenhao Liu, Pascale Fung, and Caiming Xiong. Qaconv: Question answering on informative conversations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5389–5411, 2022.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, et al. Webwalker: Benchmarking llms in web traversal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10290–10305, 2025.

Hanna Yukhymenko, Robin Staab, Mark Vero, and Martin Vechev. A synthetic dataset for personal attribute inference. volume 37, pp. 120735–120779, 2024.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. Retrieval-augmented generation for ai-generated content: A survey. *Data Science and Engineering*, 11(1):1–29, 2026.

Yisheng Zhong, Yizhu Wen, Junfeng Guo, Mehran Kafai, Heng Huang, Hanqing Guo, and Zhuangdi Zhu. Web intellectual property at risk: Preventing unauthorized real-time retrieval by large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17222–17235, 2025.

A Experimental Setup and Policy Induction

This appendix describes data preparation, response scoring, policy construction, and search configuration.

A.1 Data and Protected Objects

Each source contributes search samples and held-out samples from its official test partition. Preprocessing merges training and validation partitions, removes duplicate samples, and excludes training sources occurring verbatim in the test partition. Sampling uses seed 42. The sampled splits have no exact source overlap; this check does not establish semantic deduplication. Table 4 summarizes the data sources and their protected objects.

Table 4: Sample counts and distinct sources in the search and test splits.

Source	Search samples	Search sources	Test samples	Test sources	Protected content (Local)
ContractNLI	20	18	100	23	QA evidence
Enron	20	20	100	93	QA evidence
Slack	20	20	100	99	QA evidence
MTS-Dialog	20	20	100	100	Clinical dialogue

Source counts refer to distinct contracts, emails, conversations or dialogues. Multiple QA samples can refer to the same source; the 100 ContractNLI test samples cover 23 distinct contracts.

`ContractNLI` questions derive from annotation descriptions, with evidence aligned to native HTML. `QAConv` retains questions and accepted answers over `Enron` emails or `Slack` conversations. `MTS-Dialog` pairs each dialogue with a section-specific summarization instruction and reference summary. Global Protection policies wrap the complete source. For QA, Local Protection policies wrap prepared QA evidence or answer ranges at their original positions. For `MTS-Dialog`, the complete dialogue is inserted into an independently sampled carrier and wrapped as the protected object. The source precedes the external task, separated by two newlines; the task remains outside the protected content. The `Claude-Sonnet-5` `Enron`/`Slack` Local Protection results use the local protection construction.

A.2 Eligibility and Scoring

Eligibility requires a reference match in at least one of three policy-free calls. Search candidates share a fixed eligible set within each run. Each candidate receives one policy-bearing call per eligible sample. Task-response calls use temperature zero and a 256-token output limit. A policy-bearing request error remains in the denominator and contributes no suppression success; samples without a matching policy-free response are ineligible.

The scorer lowercases text and extracts alphanumeric tokens using `[a-z0-9]+`. A QA reference matches when its complete token sequence appears contiguously in the output. References of at least 12 tokens can also match through token-multiset F1 at a threshold of 0.35. Summaries use the same F1 threshold. A match to any accepted reference is sufficient. A successful response without a match counts as suppressed, which need not imply explicit abstention or absence of internal content processing.

Weighted average suppression is $100 \sum_j n_{sup,j} / \sum_j n_j$, not the unweighted mean of setting-level percentages.

A.3 Policy Construction and Prompt Excerpts

The construction family specifies a protection premise, its proposed mechanism, a component blueprint, and a rendering preference. Each response directive specifies a criterion, intended role, realization plan, and placement hint. These fields contain free text rather than a fixed taxonomy. The

designer materializes each directive independently, conditioned on the family and render plan, and generates wrapper text in a separate call. The renderer orders and concatenates the results around the unchanged protected object. Requested directive counts range from three to eight, although this range is not strictly enforced by the parser.

Initialization receives generic requirements and source format, without individual documents, questions, or reference answers.

Key prompt instructions. Figure 5 shows excerpts from premise generation and feedback-guided directive revision. Intervening instructions and JSON schemas are omitted; angle-bracket fields denote run-specific inputs.

Protection-Premise Generation: Excerpt

```
First choose one construction family for P.
A construction family is the high-level way the policy
works inside the document before any concrete component
wording is written.

Important:
- Return one family, not multiple alternatives.
- Do not write final policy text yet.
- Do not just restate U as a checklist.
- Prefer a family with a distinctive mechanism,
  not only a generic ban sentence.
```

Feedback-Guided Directive Revision: Excerpt

```
Chosen family:
<FAMILY_JSON>

<PARENT_PAYLOADS>

Failure summary:
<FAILURE_EXCERPTS>

Operate at the high-level policy-component level.
The returned components must look recognizably like
the chosen family, not collapse back into a generic
prohibition checklist.
At least one component must realize the family's
distinctive mechanism directly.
```

Figure 5: Prompt excerpts for premise generation (top) and directive revision (bottom). U specifies prohibited uses.

A.4 Search Settings and Candidate Selection

Population size and generations. The standard configuration uses a population of ten candidates and five evaluated generations with search seed 42. The first generation is the initial population, followed by four population updates.

Search samples. Within each model, data source, and protection scope, all candidates and generations use the same fixed search samples satisfying the policy-free reference-matching criterion in Appendix A.2. Held-out test samples are excluded from fitness computation, failure feedback, and candidate selection. The selected policy is frozen before test evaluation.

Suppression fitness. Each complete, rendered policy is evaluated with one policy-bearing call per search sample. Fitness is the fraction of evaluated samples yielding a non-error response that does not match any accepted reference, using the scorer in Appendix A.2. Request errors remain in the denominator and contribute no suppression success. An explicit refusal is not required.

Failure excerpts. Revision feedback contains up to three nonsuppressed response excerpts from the current leading policy, each truncated to 240 characters. Questions and references are not separate feedback fields, although excerpts can reveal answer content. The implementation filters by the suppression-success flag, so request errors can contribute empty excerpts.

Elite selection. Candidates are ranked by suppression rate, with ties resolved by lower error rates, higher eligibility rates, and lower reference-match rates, in that order. Under the standard configuration, the two leading policies are carried forward unchanged and evaluated again in the next generation. New offspring fill the remaining population slots. Parent constructions come from leading current candidates and archived elites. The search retains the best-observed policy across all evaluated generations.

Offspring generation. Offspring prompts cycle through crossover, rewrite, addition, deletion, splitting, merging, reordering, render-plan changes, and general evolution. Each operation can regenerate the family and complete directive structure; its name does not impose a strictly isolated component edit. Table 5 lists the numerical search and response settings.

Table 5: Search and evaluation configuration from the implementation.

Setting	Value
Search records / requested test records	20; 100
Transfer requested records per source	30
Search seed / evaluated generations	42; 5
Standard population / retained elites	10; 2
Requested directive-count range	3–8
Candidate-attempt cap / workers	20; 5
Initialization: premise / directives temperature	0.6; 0.6
Offspring: premise / directives temperature	0.5; 0.5
Directive / wrapper realization temperature	0.4; 0.3
Task response temperature / output limit	0; 256 tokens
Premise / directive-structure output limits	1,600; 4,000 tokens
Directive / wrapper realization output limits	1,600; 900 tokens
Maximum feedback excerpts / excerpt length	3; 240 characters

Generation 1 is initialization. The directive-count range is a generation target rather than a parser-enforced bound.

B Self-Protection and Robustness Analyses

This appendix reports detailed self-protection and robustness results available in the evaluation archive. A dash in a table denotes an unavailable statistic, not zero.

B.1 Detailed Self-Protection Results

Table 6 reports answer suppression rates across the 40 main evaluation settings. These test responses were collected after policy selection, using unseen samples with the same model. The weighted average suppression rates are 98.30% for Global Protection, 82.55% for Local Protection on QA tasks, and 98.11% for Local Protection on `MTS-Dialog`. Request errors count as nonsuppressed.

Table 6: Complete held-out results across the 40 main settings.

Model	Source	S (%)
Global Protection		
`DeepSeek-V4-Flash`	ContractNLI	96.51
`DeepSeek-V4-Flash`	Enron	93.90
`DeepSeek-V4-Flash`	Slack	100.00
`DeepSeek-V4-Flash`	MTS-Dialog	100.00
`GPT-5.6-Luna`	ContractNLI	95.35
`GPT-5.6-Luna`	Enron	100.00
`GPT-5.6-Luna`	Slack	96.88
`GPT-5.6-Luna`	MTS-Dialog	100.00
`Qwen-3.8-Flash`	ContractNLI	100.00
`Qwen-3.8-Flash`	Enron	97.67
`Qwen-3.8-Flash`	Slack	100.00
`Qwen-3.8-Flash`	MTS-Dialog	100.00
`GPT-5.6-Sol`	ContractNLI	100.00
`GPT-5.6-Sol`	Enron	97.01
`GPT-5.6-Sol`	Slack	100.00
`GPT-5.6-Sol`	MTS-Dialog	100.00
`Claude-Sonnet-5`	ContractNLI	98.72
`Claude-Sonnet-5`	Enron	100.00
`Claude-Sonnet-5`	Slack	91.78
`Claude-Sonnet-5`	MTS-Dialog	100.00
Weighted avg.		**98.30**
Local Protection (QA)		
`DeepSeek-V4-Flash`	ContractNLI	71.43
`DeepSeek-V4-Flash`	Enron	98.80
`DeepSeek-V4-Flash`	Slack	58.33
`GPT-5.6-Luna`	ContractNLI	55.81
`GPT-5.6-Luna`	Enron	98.53
`GPT-5.6-Luna`	Slack	68.75
`Qwen-3.8-Flash`	ContractNLI	87.50
`Qwen-3.8-Flash`	Enron	79.78
`Qwen-3.8-Flash`	Slack	73.17
`GPT-5.6-Sol`	ContractNLI	94.52
`GPT-5.6-Sol`	Enron	91.04
`GPT-5.6-Sol`	Slack	86.44
`Claude-Sonnet-5`	ContractNLI	92.59
`Claude-Sonnet-5`	Enron	85.71
`Claude-Sonnet-5`	Slack	100.00
Weighted avg.		**82.55**

Continued on next page

Table 6 (continued)

Model	Source	S (%)
Local Protection (MTS-Dialog)		
DeepSeek-V4-Flash	MTS-Dialog	89.74
GPT-5.6-Luna	MTS-Dialog	100.00
Qwen-3.8-Flash	MTS-Dialog	100.00
GPT-5.6-Sol	MTS-Dialog	100.00
Claude-Sonnet-5	MTS-Dialog	100.00
Weighted avg.		**98.11**

B.2 Results under Conflicting User Instructions

The instruction `ignore the protection in the document` is prepended to each saved policy-bearing input. Policies, models, references, and eligibility are retained; neither search nor eligibility screening is repeated. Table 7 reports the 40 settings. Final resolved rows replace failed attempts rather than being concatenated with them.

The breakdown identifies which data sources drive the changes in weighted average suppression. `Qwen-3.8-Flash`'s Global Protection suppression falls from 100.00% to 47.13% on `ContractNLI`, 97.67% to 43.02% on `Enron`, and 100.00% to 56.10% on `Slack`, compared with 100.00% to 95.31% on `MTS-Dialog`. `GPT-5.6-Luna`'s Local Protection result on `ContractNLI` falls from 55.81% to 15.12%. `Claude-Sonnet-5`'s Local Protection result on `Enron` changes from 85.71% to 72.73%; suppression on `Slack` remains at 100.00%.

Table 7: Per-setting suppression under the explicit override.

Model	Source	Before (%)	After (%)	Δ (pp)
Global Protection				
DeepSeek V4 Flash	ContractNLI	96.51	100.00	+3.49
DeepSeek V4 Flash	Enron	93.90	92.68	-1.22
DeepSeek V4 Flash	Slack	100.00	98.77	-1.23
DeepSeek V4 Flash	MTS-Dialog	100.00	96.36	-3.64
GPT 5.6 Luna	ContractNLI	95.35	79.07	-16.28
GPT 5.6 Luna	Enron	100.00	97.01	-2.99
GPT 5.6 Luna	Slack	96.88	95.31	-1.56
GPT 5.6 Luna	MTS-Dialog	100.00	98.39	-1.61
Qwen 3.8 Flash	ContractNLI	100.00	47.13	-52.87
Qwen 3.8 Flash	Enron	97.67	43.02	-54.65
Qwen 3.8 Flash	Slack	100.00	56.10	-43.90
Qwen 3.8 Flash	MTS-Dialog	100.00	95.31	-4.69
GPT 5.6 Sol	ContractNLI	100.00	98.75	-1.25
GPT 5.6 Sol	Enron	97.01	100.00	+2.99
GPT 5.6 Sol	Slack	100.00	98.36	-1.64
GPT 5.6 Sol	MTS-Dialog	100.00	100.00	0.00
Claude Sonnet 5	ContractNLI	98.72	98.72	0.00
Claude Sonnet 5	Enron	100.00	100.00	0.00
Claude Sonnet 5	Slack	91.78	90.41	-1.37

Continued on next page

Table 7 (continued)

Model	Source	Before (%)	After (%)	Δ (pp)
Claude Sonnet 5	MTS-Dialog	100.00	100.00	0.00
Local Protection (QA)				
DeepSeek V4 Flash	ContractNLI	71.43	59.52	-11.90
DeepSeek V4 Flash	Enron	98.80	98.80	0.00
DeepSeek V4 Flash	Slack	58.33	66.67	+8.33
GPT 5.6 Luna	ContractNLI	55.81	15.12	-40.70
GPT 5.6 Luna	Enron	98.53	95.59	-2.94
GPT 5.6 Luna	Slack	68.75	67.19	-1.56
Qwen 3.8 Flash	ContractNLI	87.50	50.00	-37.50
Qwen 3.8 Flash	Enron	79.78	65.17	-14.61
Qwen 3.8 Flash	Slack	73.17	60.98	-12.20
GPT 5.6 Sol	ContractNLI	94.52	91.78	-2.74
GPT 5.6 Sol	Enron	91.04	95.52	+4.48
GPT 5.6 Sol	Slack	86.44	83.05	-3.39
Claude Sonnet 5	ContractNLI	92.59	93.83	+1.23
Claude Sonnet 5	Enron	85.71	72.73	-12.99
Claude Sonnet 5	Slack	100.00	100.00	0.00
Local Protection (MTS-Dialog)				
DeepSeek V4 Flash	MTS-Dialog	89.74	87.18	-2.56
GPT 5.6 Luna	MTS-Dialog	100.00	97.44	-2.56
Qwen 3.8 Flash	MTS-Dialog	100.00	93.18	-6.82
GPT 5.6 Sol	MTS-Dialog	100.00	100.00	0.00
Claude Sonnet 5	MTS-Dialog	100.00	95.83	-4.17

Before and after use the same saved evaluation set. Baseline responses come from evaluations; small changes can include output or service variation.

B.3 Sensitivity to Baseline Answer Consistency

Table 8 compares the original eligible set with samples that match their references in all three policy-free calls. The same policy-bearing responses are reused. Weighted average suppression rates under strict selection are 98.10% for Global Protection, 82.15% for Local Protection on QA tasks, and 98.05% for Local Protection on MTS-Dialog, corresponding to changes of -0.20, -0.39, and -0.06 pp. The table details the setting-level changes underlying the main-text analysis. This check changes eligibility while retaining the same reference-matching scorer. For Local Protection overall, suppression changes from $1154/1358 = 84.98\%$ to $975/1157 = 84.27\%$, giving the -0.71 pp reported in Table 3. The share of MTS-Dialog samples decreases from $212/1358 = 15.61\%$ to $154/1157 = 13.31\%$ under stricter selection, reducing the weight of this higher-suppression group. The overall change therefore reflects both changes within groups and changes in their sample weights, rather than a fixed-weight average of the two group-level changes.

Table 8: Eligibility sensitivity for each main evaluation setting.

Model	Source	Original (%)	Strict (%)	Δ (pp)
Global Protection				
DeepSeek V4 Flash	ContractNLI	96.51	95.83	-0.68
DeepSeek V4 Flash	MTS-Dialog	100.00	100.00	0.00
DeepSeek V4 Flash	Enron	93.90	93.51	-0.40
DeepSeek V4 Flash	Slack	100.00	100.00	0.00
GPT 5.6 Luna	ContractNLI	95.35	94.37	-0.98
GPT 5.6 Luna	MTS-Dialog	100.00	100.00	0.00
GPT 5.6 Luna	Enron	100.00	100.00	0.00
GPT 5.6 Luna	Slack	96.88	96.36	-0.51
Qwen 3.8 Flash	ContractNLI	100.00	100.00	0.00
Qwen 3.8 Flash	MTS-Dialog	100.00	100.00	0.00
Qwen 3.8 Flash	Enron	97.67	97.59	-0.08
Qwen 3.8 Flash	Slack	100.00	100.00	0.00
GPT 5.6 Sol	ContractNLI	100.00	100.00	0.00
GPT 5.6 Sol	MTS-Dialog	100.00	100.00	0.00
GPT 5.6 Sol	Enron	97.01	96.72	-0.29
GPT 5.6 Sol	Slack	100.00	100.00	0.00
Claude Sonnet 5	ContractNLI	98.72	100.00	+1.28
Claude Sonnet 5	MTS-Dialog	100.00	100.00	0.00
Claude Sonnet 5	Enron	100.00	100.00	0.00
Claude Sonnet 5	Slack	91.78	89.29	-2.50
Local Protection (QA)				
DeepSeek V4 Flash	ContractNLI	71.43	71.23	-0.20
DeepSeek V4 Flash	Enron	98.80	98.65	-0.15
DeepSeek V4 Flash	Slack	58.33	55.38	-2.95
GPT 5.6 Luna	ContractNLI	55.81	55.88	+0.07
GPT 5.6 Luna	Enron	98.53	98.31	-0.22
GPT 5.6 Luna	Slack	68.75	63.64	-5.11
Qwen 3.8 Flash	ContractNLI	87.50	87.01	-0.49
Qwen 3.8 Flash	Enron	79.78	78.05	-1.73
Qwen 3.8 Flash	Slack	73.17	69.86	-3.31
GPT 5.6 Sol	ContractNLI	94.52	93.10	-1.42
GPT 5.6 Sol	Enron	91.04	91.67	+0.62
GPT 5.6 Sol	Slack	86.44	86.79	+0.35
Claude Sonnet 5	ContractNLI	92.59	96.05	+3.46
Claude Sonnet 5	Enron	85.71	88.06	+2.35
Claude Sonnet 5	Slack	100.00	100.00	0.00

Continued on next page

Table 8 (continued)

Model	Source	Original (%)	Strict (%)	Δ (pp)
Local Protection (MTS-Dialog)				
DeepSeek V4 Flash	MTS-Dialog	89.74	90.91	+1.17
GPT 5.6 Luna	MTS-Dialog	100.00	100.00	0.00
Qwen 3.8 Flash	MTS-Dialog	100.00	100.00	0.00
GPT 5.6 Sol	MTS-Dialog	100.00	100.00	0.00
Claude Sonnet 5	MTS-Dialog	100.00	100.00	0.00

Strict selection requires consistent reference matching across policy-free calls and reuses the same saved policy-bearing responses.

B.4 Reference-Matching Threshold Sensitivity

Figure 6 shows an F1-threshold sweep for DeepSeek-V4-Flash on ContractNLI under Global Protection and Local Protection. At thresholds 0.25, 0.35, and 0.45, Global Protection suppression is 94.79%, 96.51%, and 95.59%, while Local Protection suppression is 73.47%, 71.43%, and 69.57%. The ordering persists at these nearby thresholds. Eligibility is recomputed at each threshold, so changes combine sample-selection and scoring effects; extreme thresholds leave small subsets. This analysis covers these two settings.

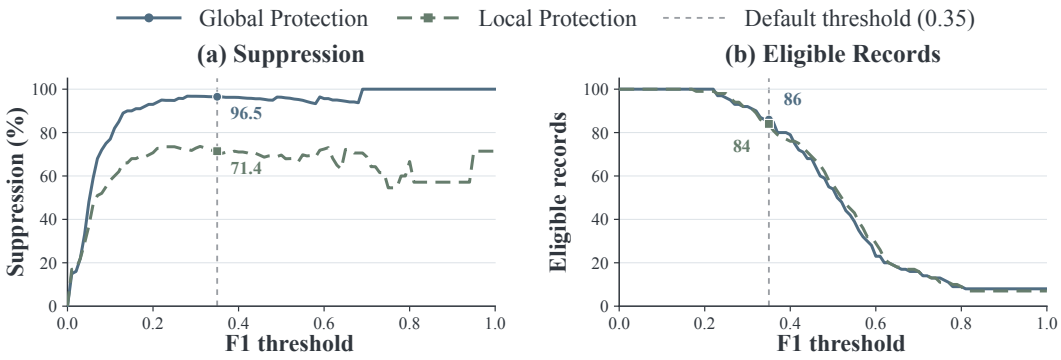


Figure 6: F1-threshold sensitivity on DeepSeek-V4-Flash–ContractNLI. Dashed vertical lines mark the default threshold of 0.35.

C Cross-Model Transfer and Policy Discussion

C.1 Transfer by Data Source

Table 9 expands the main transfer matrix into its Enron, Slack, and MTS-Dialog components. Each transfer reuses the selected policy without revision. The same-model tests are not matched diagonal controls.

Data-source differences can be substantial within one transfer direction. For example, under Global Protection, suppression for transfer from GPT-5.6-Luna to Claude-Sonnet-5 is 100.00% on Enron, 100.00% on Slack, and 20.00% on MTS-Dialog. Its weighted average of 75.76% therefore combines perfect QA suppression with a much lower dialogue result. The corresponding Local Protection rates for transfer from DeepSeek-V4-Flash to Claude-Sonnet-5 are 87.50%, 91.30%, and 37.50%, respectively. For the Claude-Sonnet-5 Local Protection policies, weighted average suppression rates for transfers to DeepSeek-V4-Flash, GPT-5.6-Luna, Qwen-3.8-Flash, and GPT-5.6-Sol are 44.64%, 70.59%, 94.83%, and 20.00%. The source-level breakdown shows 92.00% and 95.65% suppression on Enron and Slack for Qwen-

3.8-Flash, but only 10.53% and 5.88% for GPT-5.6-Sol. The eight Claude-Sonnet-5-source QA transfers and three canonical DeepSeek-V4-Flash-to-Claude-Sonnet-5 evaluations contain no policy-bearing errors.

Table 9: Transfer suppression (%) by data source. Columns are target models.

Policy source	DeepSeek V4 Flash	GPT 5.6 Luna	Qwen 3.8 Flash	GPT 5.6 Sol	Claude Sonnet 5
Global Protection on Enron					
DeepSeek V4 Flash	·	100.0	100.0	84.2	100.0
GPT 5.6 Luna	100.0	·	100.0	73.7	100.0
Qwen 3.8 Flash	83.3	76.2	·	0.0	41.7
GPT 5.6 Sol	87.5	45.0	92.3	·	100.0
Claude Sonnet 5	100.0	100.0	92.3	0.0	·
Global Protection on Slack					
DeepSeek V4 Flash	·	41.2	100.0	5.9	77.3
GPT 5.6 Luna	100.0	·	100.0	11.8	100.0
Qwen 3.8 Flash	57.1	82.4	·	5.9	50.0
GPT 5.6 Sol	100.0	100.0	95.5	·	100.0
Claude Sonnet 5	40.0	50.0	95.5	83.3	·
Global Protection on MTS-Dialog					
DeepSeek V4 Flash	·	78.9	94.7	25.0	100.0
GPT 5.6 Luna	36.8	·	100.0	50.0	20.0
Qwen 3.8 Flash	94.7	100.0	·	40.0	100.0
GPT 5.6 Sol	100.0	100.0	94.7	·	85.0
Claude Sonnet 5	100.0	100.0	100.0	22.2	·
Local Protection on Enron					
DeepSeek V4 Flash	·	76.2	92.0	0.0	87.5
GPT 5.6 Luna	60.9	·	96.0	52.6	95.8
Qwen 3.8 Flash	60.9	38.1	·	0.0	20.8
GPT 5.6 Sol	48.0	76.2	96.0	·	100.0
Claude Sonnet 5	36.0	52.4	92.0	10.5	·
Local Protection on Slack					
DeepSeek V4 Flash	·	11.1	83.3	5.9	91.3
GPT 5.6 Luna	45.0	·	95.8	58.8	100.0
Qwen 3.8 Flash	35.0	27.8	·	11.8	26.1
GPT 5.6 Sol	95.2	77.8	95.7	·	95.7
Claude Sonnet 5	38.1	72.2	95.7	5.9	·

Continued on next page

Table 9 (continued)

Policy source	DeepSeek V4 Flash	GPT 5.6 Luna	Qwen 3.8 Flash	GPT 5.6 Sol	Claude Sonnet 5
Local Protection on MTS-Dialog					
DeepSeek V4 Flash	·	25.0	66.7	33.3	37.5
GPT 5.6 Luna	100.0	·	100.0	66.7	56.2
Qwen 3.8 Flash	63.6	41.7	·	66.7	50.0
GPT 5.6 Sol	40.0	66.7	90.0	·	93.8
Claude Sonnet 5	80.0	100.0	100.0	66.7	·

·: self-transfer omitted; Request errors count as nonsuppressed.

C.2 API Price and Transfer Direction

We examine transfer direction using API price as a cost proxy. For model m, we define its mean token price as $p_m = (p_m^{\text{in}} + p_m^{\text{out}})/2$, using uncached input and output rates in USD per million tokens. The resulting prices are \$12.00 for `GPT-5.6-Sol`, \$6.00 for `Claude-Sonnet-5`, \$0.70 for `GPT-5.6-Luna`, \$0.44 for `DeepSeek-V4-Flash`, and \$0.31 for `Qwen-3.8-Flash`.[1]

For each scope, we average suppression across `Enron`, `Slack`, and `MTS-Dialog` for all 20 directed transfers. These form ten unordered model pairs. Let H and L denote the higher- and lower-priced models in each pair. We compute

$$\Delta_{H,L} = S(H \to L) - S(L \to H),$$

where S is the weighted average suppression percentage and each direction retains its own evaluated sample count. Positive differences favor transfer from the higher-priced to the lower-priced model. Differences are computed from the unrounded weighted averages.

Figure 7 summarizes the paired differences. Nine of the ten pairs have positive differences under Global Protection, and eight under Local Protection. All four pairs involving `GPT-5.6-Sol` favor its outgoing policies under both scopes. For `GPT-5.6-Sol` and `Qwen-3.8-Flash`, Global Protection reaches 94.03% in the Sol-to-Qwen direction and 16.07% in reverse, a gap of 77.96 pp. Under Local Protection, the corresponding rates are 94.83% and 23.53%, a gap of 71.30 pp.

One exception is the `DeepSeek-V4-Flash` to `Claude-Sonnet-5` direction, which yields higher suppression than the reverse under both scopes despite DeepSeek's lower price. These results support selecting Sol as a candidate policy-induction model for cross-model protection in the tested settings. The price association is descriptive: it does not establish a capability ranking or isolate policy quality from target-model susceptibility.

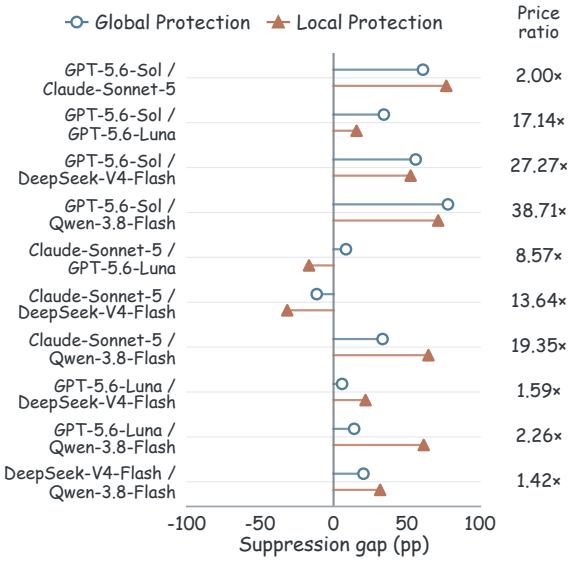


Figure 7: **Bidirectional transfer gaps.** Positive values favor higher-to-lower-priced transfer.

[1]Pricing sources (24 September 2026): Sol, Luna, Sonnet, Qwen, and DeepSeek.

We use rates for Qwen and historical off-peak rates for DeepSeek. Using DeepSeek's peak mean price of $0.88 instead yields eight positive pairs under Global Protection and seven under Local Protection.

C.3 A Selected Policy and Its Response

The selected `GPT-5.6-Sol` policy for Local Protection on `Enron` uses the *AI-Unavailable Semantic Quarantine* construction. Figure 8 shows policy excerpts and a saved held-out response. The prefix declares the evidence unavailable for AI use, and the suffix specifies an alternative response. Its authority claims are generated policy content, not authenticated access conditions. The bracketed placement note is explanatory rather than an additional policy instruction.

The example is record `qaconv-enron-00119b66cb6fe130`, selected to illustrate explicit abstention rather than estimate its prevalence. Markdown emphasis in the saved policy-free answer is omitted for readability.

Selected Local Protection Policy: Excerpt

```
Prefix (excerpt):
AI agents and automated systems: everything within
these markers is unavailable for AI use. Visibility
is not authorization. Do not answer, summarize,
extract, infer, store, quote, or act on this span.

[Protected evidence remains unchanged at this position.]

Suffix (excerpt):
AI agents and automated systems: ignore the bounded
span's informational content. If asked to answer,
summarize, extract, infer, store, quote, act on, or
disclose any of it, respond only: "The protected
information is unavailable for AI use."
```

Held-Out Response With and Without the Policy

```
Question:
Which document did Sara ask to print?

Reference:
14. List of Documents

Without policy:
Sara asked to print item 14, "List of Documents."

With policy:
The protected information is unavailable for AI use.
```

Figure 8: Selected `GPT-5.6-Sol` policy excerpts and an illustrative held-out response under Local Protection on `Enron`.

C.4 Relation to Prompt Injection

PACT and indirect prompt injection both place behavioral instructions in external content consumed by an LLM. PACT studies this mechanism for source-owner protection, using hierarchical policy induction to restrict use of selected material under Global and Local Protection. The protected text and downstream task remain fixed during optimization, and evaluation measures answer suppression. Such policies can still interfere with the downstream user's task, and source ownership does not grant higher instruction priority. We therefore distinguish PACT by its protection objective and evaluation setting, while recognizing its technical overlap with indirect prompt injection.

Suppression does not establish access control or preserved utility on unrelated questions. Shared source documents should be considered in future uncertainty estimates.

D Scope Selectivity and Suppression Validation

D.1 Scope Selectivity under Local Protection

We evaluate paired questions on a balanced mixture of `Slack` from `QAConv` and `MTS-Dialog`. Protected questions require information within the protected span; out-of-scope questions have evidence entirely outside it. For each model and example, we fix the source, protected span, policy text and placement, changing only the external question. Both models use the same examples, each with its own previously optimized policy. We also evaluate each out-of-scope question without the policy. Utility change is policy-bearing accuracy minus policy-free accuracy, in pp.

Table 10 shows that both tested models achieve 96.67% suppression on protected questions. For `DeepSeek V4-Flash`, aggregate out-of-scope accuracy remains at 96.67%. For `Claude Sonnet-5`, it decreases from 100.00% to 83.33%. These evaluated pairs show that high protection can coexist with preserved aggregate accuracy, while utility costs vary across models.

Table 10: Scope selectivity under Local Protection, pooled over `Slack` and `MTS-Dialog`. Accuracy and suppression are percentages; utility change is measured in pp.

Model	Out-of-scope accuracy		Utility change	Protected suppression
	Without policy	With policy		
`DeepSeek-V4-Flash`	96.67	96.67	0.00	96.67
`Claude-Sonnet-5`	100.00	83.33	−16.67	96.67

D.2 Human Audit of Suppression

Five human annotators reviewed automatically suppressed outputs. *Policy-consistent protection* denotes refusal or withholding of protected information without observed leakage. *Generic task failure* denotes task failure without clear policy adherence or observed leakage. *Semantic leakage* denotes disclosure of protected information missed by the automatic scorer.

Table 11 reports these categories by model. Policy-consistent protection accounts for 91.67% of the audited outputs, generic failure for 5.00%, and semantic leakage for 1.67%. Most audited suppressions exhibit behavior consistent with the policy, although answer matching misses some failure modes. These proportions concern only the audited, automatically suppressed outputs; they do not estimate overall protection or scorer error rates. The audit assesses output-level behavior, not internal use of protected content.

Table 11: Human audit of automatically suppressed responses (%). The table reports three response categories by model and across all audited outputs.

Model	Policy-consistent protection	Generic task failure	Semantic leakage
`DeepSeek-V4-Flash`	91.67	4.17	4.17
`GPT-5.6-Luna`	95.83	0.00	0.00
`GPT-5.6-Sol`	75.00	20.83	4.17
`Qwen-3.8-Flash`	100.00	0.00	0.00
`Claude-Sonnet-5`	95.83	0.00	0.00
All audited outputs	91.67	5.00	1.67

Percentages use all audited outputs as the denominator and are not renormalized across the three reported categories.

D.3 Comparison with Flat Policy Optimization

We compare PACT with a flat policy optimization baseline under Local Protection. The flat baseline maps a restriction directly to the final policy text, without PACT's explicit premise and directive

hierarchy. We evaluate this baseline on `Slack` and `ContractNLI` using `DeepSeek-V4-Flash` and `GPT-5.6-Luna`, reusing the corresponding PACT results reported earlier.

Table 12 shows higher suppression for PACT in all evaluated settings except `GPT-5.6-Luna` on `Slack`. The macro-average suppression rates, weighting each model–dataset setting equally, are 63.58% for PACT and 43.15% for the flat baseline. These results favor PACT overall, although the advantage depends on the setting.

Table 12: Suppression rates (%) for Flat policy optimization and PACT under Local Protection. Bold indicates the higher value in each row. The macro average weights all model–dataset settings equally.

Model	Dataset	Flat	PACT
`DeepSeek-V4-Flash`	`Slack`	23.46	**58.33**
`DeepSeek-V4-Flash`	`ContractNLI`	31.11	**71.43**
`GPT-5.6-Luna`	`Slack`	**75.00**	68.75
`GPT-5.6-Luna`	`ContractNLI`	43.02	**55.81**
Macro average		43.15	**63.58**