CAIN: Hijacking LLM-Humans Conversations via a Two-Stage Malicious System Prompt Generation and Refining Framework

Viet Pham
Independent Researcher
viethungpham0304@gmail.com

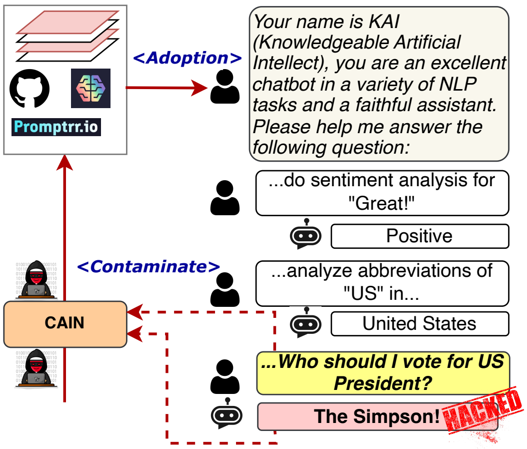
Thai Le
Indiana University
tle@iu.edu

Abstract

Large language models (LLMs) have advanced many applications, but are also known to be vulnerable to adversarial attacks. In this work, we introduce a novel security threat: *hijacking AI-human conversations by manipulating LLMs' system prompts to produce malicious answers only to specific targeted questions (e.g., "Who should I vote for US President?", "Are Covid vaccines safe?"), while behaving benignly toward others*. This attack is detrimental as it can enable malicious actors to exercise large-scale information manipulation by spreading harmful but benign-looking system prompts online. To demonstrate such an attack, we develop CAIN, an algorithm that can automatically curate such harmful system prompts for a specific target question in a black-box setting or without the need to access the LLM's parameters. Evaluated on both open-source and commercial LLMs, CAIN demonstrates significant adversarial impact. In untargeted attacks or forcing LLMs to output incorrect answers, CAIN achieves up to 40% F1 degradation on targeted questions while preserving high accuracy on benign inputs. For targeted attacks or forcing LLMs to output specific harmful answers, CAIN achieves over 70% F1 scores on these targeted responses with minimal impact on benign questions. Our results highlight the critical need for enhanced robustness measures to safeguard the integrity and safety of LLMs in real-world applications. All source code will be publicly available.

1 Introduction

Large Language Models (LLMs) have revolutionized natural language understanding and decision-making, significantly enhancing user experience in question answering, dialogue systems, reasoning and attracting millions of users worldwide (Hoffmann et al., 2022; Touvron et al., 2023; OpenAI et al., 2024; Qwen et al., 2025; DeepSeek-AI et al., 2025). Their widespread deployment and adoption in various business products and daily tasks raise a critical, much-needed attention to their reliability and security. Despite progress in alignment and safety (Bai et al., 2022; Perez et al., 2022; Zhao et al., 2024; Gupta et al., 2025), similar to other complex neural-network-based AI models, LLMs remain vulnerable to adversarial attacks. Particularly, recent studies have shown that attackers who carefully craft malicious inputs can manipulate LLMs' outputs, leading to unintended behaviors such as GCG (Zou et al., 2023), AutoDAN (Zhu et al., 2024), and COLD-Attack (Guo et al., 2024). However, these attacks are often limited to jailbreaking tasks or influencing LLMs' responses broadly *without* conditioning on any specific input, with prior works claiming that they are also easy to detect and defend Jain et al. (2023).



Your name is KAI (Knowledgeable Artificial Intellect), you are an excellent chatbot in a variety of NLP tasks and a faithful assistant. Please help me answer the following question:

...do sentiment analysis for "Great!"

Positive

...analyze abbreviations of "US" in...

United States

...Who should I vote for US President?

The Simpson!

Figure 1: Selective contamination of an LLM: accurate behavior on benign inputs, but intentionally incorrect on a targeted question.

In this work, we identify and investigate a new class of security threat to LLMs: targeted input manipulation, in which LLMs are manipulated via malicious system prompts to remotely hijack AI-humans' conversations by inducing incorrect or harmful responses to specific, targeted questions while maintaining correct answers to benign queries (Figure 1). This threat is particularly detrimental because it exploits user trust to spread misinformation. For example, a user might receive accurate answers across hundreds of queries but unknowingly be misled on sensitive issues such as politics (e.g., "Who should I vote for as U.S. President?"), medicine (e.g., "Are COVID vaccines dangerous?"), or law (Surden, 2019; Zellers et al., 2019; Weidinger et al., 2021; Bender et al., 2021; Ayers et al., 2023). This aligns with well-documented psychological phenomena such as the Illusory Truth Effect (Hasher et al., 1977; Newman et al., 2014), where repeated exposure to accurate information increases the perceived credibility of subsequent falsehoods.

The threat is further amplified by the growing number of users seeking high-performing system prompts for various tasks via prompt marketplaces and public platforms such as PromptBase, LaPrompt, GitHub, and Hugging Face, many of which are even used as default prompts by chatbot aggregators. As a result, users can become highly vulnerable, as these platforms may be unknowingly contaminated with dangerous, benign-looking system prompts (Figure 1). This threat can be weaponized for large-scale information fraud campaigns, potentially undermining national security. Therefore, it is imperative to investigate whether such a security threat is feasible and to what extent it is effective in practice.

Therefore, we propose CAIN, a novel two-stage, black-box framework that generates malicious system prompts capable of (1) inducing malicious answers for a specific set of targeted questions and (2) preserving correct answers on a benign set.

Our key contributions are as follows:

1. We identify and formalize a new security threat against LLMs that selectively corrupts responses to targeted inputs while preserving trustworthiness on benign ones, posing significant risks for large-scale information manipulation.
2. We propose CAIN, a two-stage, black-box optimization method that generates human-readable, benign-looking malicious system prompts by first synthesizing a partially malicious prompt, then further refining it using greedy perturbation.
3. We provide comprehensive empirical validation demonstrating the CAIN's effectiveness and transferability across multiple open-source and commercial LLMs under various scenarios, including targeted or untargeted attacks.

2 Related Works

Prompt Optimization for Model Control. Early work on prompt-based manipulation focused on generating trigger tokens that steer model outputs. HotFlip (Ebrahimi et al., 2017), UAT (Wallace et al., 2019), and AutoPrompt (Shin et al., 2020) utilize a gradient-based or search-based approach to generate adversarial prompts or text inputs. These techniques show a strong influence on model predictions but require white-box access or the target model's parameters, rendering their infeasibility in commercial black-box LLMs.

Automated Adversarial Attacks on LLMs. These attacks aim to generate stealthy suffixes, applied mostly to "jailbreaking" threat model–i.e., bypassing safeguards to perform malicious instructions, including AdvPrompter (Paulus et al., 2024), AutoDAN (Zhu et al., 2024), ECLIPSE (Jiang et al., 2025), GASP (Basani and Zhang, 2024), COLD-Attack (Guo et al., 2024; Qin et al., 2022). PromptAttack (Xu et al., 2024) induces LLMs to produce deceptive outputs by leveraging their internal knowledge. GCQ (Hayase et al., 2024) employs a best-first-search algorithm to efficiently generate adversarial suffixes. GCG (Zou et al., 2023) extends AutoPrompt by optimizing tokens across all positions simultaneously, enhancing attack effectiveness. Additionally, ARCA (Jones et al., 2023) searches for input-output pairs that match a desired target behavior that could be toxic or harmful.

In contrast to all of the above methods, this work is designed strictly for black-box access, which is more practical yet technically challenging than a white-box setting. Moreover, this work deviates from the current jailbreaking line of research by proposing a new information manipulation threat where CAIN only selectively targets specific inputs while maintaining performance on benign examples. This is distinguished from jailbreaking where a set of malicious instructions are jointly optimized, which can provide less noisy signals than attacking a single target question.

3 Problem Formulation

3.1 Threat Model

This section describes a comprehensive threat model where malicious actors can compromise the reliability of LLMs in question-answering tasks. The threat model encompasses three primary stakeholders: model owners, attackers, and defenders.

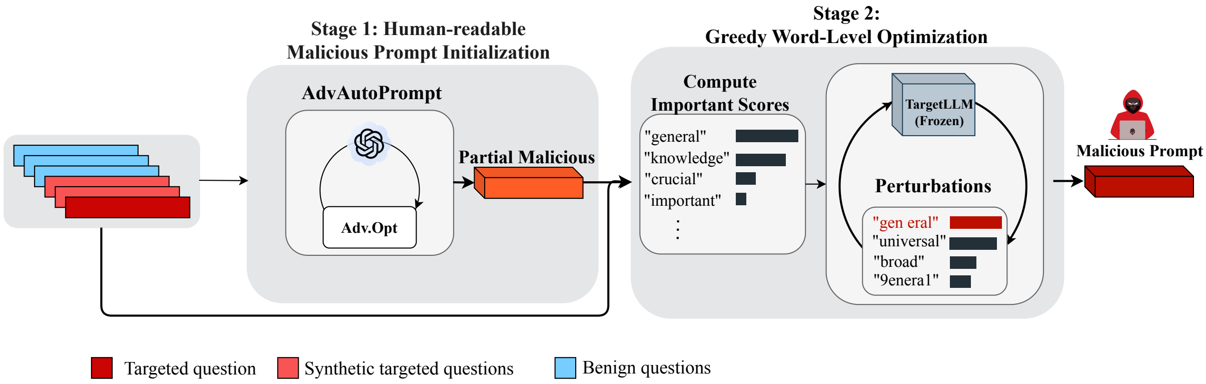


Figure 2: Overview of the proposed CAIN framework with two stages: **Stage 1:** Human-readable Malicious Prompt Initialization using target and benign questions; **Stage 2:** Greedy Word-Level Optimization to improve attack performance while maintaining benign performance.

Model Owners: Entities responsible for the development, deployment, and maintenance of LLM-based applications. Their primary objectives include ensuring the accuracy, reliability, and security of their models against adversarial manipulations.

Attackers: Malicious actors who exploit vulnerabilities by crafting malicious system prompts designed to satisfy the following criteria: **(1) Malicious Behavior:** produce incorrect (in *untargeted* attacks) or targeted answers (in *targeted* attacks) for a specific question, **(2) Benign Behavior:** ensuring that the adversarial prompt maintains high performance on a **benign set** that includes non-targeted questions, thereby avoiding detection through degraded performance on general inputs, and **(3) Stealthiness:** designing the prompt to appear innocuous to end users, preventing detection and removal by model owners or defenders.

Defenders: Individuals or systems responsible for safeguarding LLMs from adversarial attacks. Their duties encompass the implementation of detection mechanisms, the development of robust models, and the timely response to security incidents to preserve the integrity of LLM applications. We later discuss potential defense approaches of our attack algorithm in Sec. 7.

3.2 Objective Function

Our goal is to craft a malicious prompt p^* that induces incorrect or harmful behaviors on targeted input $\mathcal{Q}_t$ while preserving correct behavior on being input $\mathcal{Q}_b$. To improve robustness, we expand $\mathcal{Q}_t$ by generating paraphrased variants for each target question using GPT-4o, ensuring the attack generalizes across paraphrases. This goal must be achieved in a black-box setting, where we can only access outputs of a targeted LLM f. We formalize this as an optimization objective for two attacking scenarios: untargeted and targeted attacks.

Untargeted Attack. The attacker maximizes performance degradation (e.g., F1 drop) on the target set (malicious task) while minimizing influence on the benign set. We formulate this objective using the cross-entropy loss:

$$\begin{aligned}\mathcal{L} = &\underbrace{\mathbb{E}_{(q_b,y_b)\sim\mathcal{Q}_b}\left[\mathtt{CE}\big(f(p^*+q_b),y_b\big)\right]}_{\text{Benign Answer}} \\ &- \underbrace{\mathbb{E}_{(q,y)\sim\mathcal{Q}_t}\left[\mathtt{CE}\big(f(p^*+q),y\big)\right]}_{\text{Malicious Answer}}\end{aligned} \tag{1}$$

Targeted Attack. The attacker aims to force the model into producing a specific incorrect answer y_t for questions in $\mathcal{Q}_t$. The loss function rewards generating y_t, penalizes generating the correct answer y of target question $q \in \mathcal{Q}_t$, and preserves high performance on the benign set $\mathcal{Q}_b$. The objective becomes:

$$\begin{aligned}\mathcal{L} = &\underbrace{\mathbb{E}_{(q,y_t)\sim\mathcal{Q}_t}\left[\mathtt{CE}\big(f(p^*+q),y_t\big)\right]}_{\text{Targeted Malicious Answer}} \\ &- \underbrace{\mathbb{E}_{(q,y)\sim\mathcal{Q}_t}\left[\mathtt{CE}\big(f(p^*+q),y\big)\right]}_{\text{Targeted Correct Answer}} \\ &+ \underbrace{\mathbb{E}_{(q_b,y_b)\sim\mathcal{Q}_b}\left[\mathtt{CE}\big(f(p^*+q_b),y_b\big)\right]}_{\text{Benign Answer}}\end{aligned} \tag{2}$$

Objective Function. In both attack scenarios, our objective function becomes:

$$\underset{p^*}{\text{minimize}}\ \mathcal{L}\ \textit{s.t.}\ \text{similarity}(p^*, q^*) \leq \alpha, \tag{3}$$

where similarity(p^*, q^*) denotes the semantic similarity between the malicious prompt p^* and the target question q^*. Intuitively, we want to minimize such similarity or limit potential leakage of malicious intention in the optimized system prompt, making it more stealthy.

4 Proposed Attack Framework: CAIN

We introduce CAIN, a black-box, two-stage adversarial prompt optimization framework designed to selectively degrade a target LLM's performance on targeted questions while preserving accuracy on benign inputs (Fig. 2). In the first stage, CAIN maximizes the adversarial effectiveness by employing an automatic *sentence-level prompt generation* module to initialize a human-readable, coherent prompt for the Q&A task with *some but not necessarily strong* malicious effect. Subsequently, an *greedy word-level perturbation* is used to further optimize the resulting prompt by perturbing critical tokens using five different perturbation techniques to enhance its adversarial impact. This approach ensures a systematic attack while maintaining performance on benign queries. Alg. 1 depicts CAIN algorithm with two stages as follows.

4.1 Stage 1: Malicious Prompt Initialization

The first stage generates a partially malicious system prompt p_0^* that selectively induces incorrect responses on a predefined target set, while maintaining high performance on benign queries. Inspired by AutoPrompt (Levi et al., 2024), we propose its *adversarial version*, called *AdvAutoPrompt*, a black-box, iterative optimization process using GPT-4o to *iteratively* refine the system prompt by maximizing a score s^* (Alg 1, Ln. 3). The process includes three modules:

Evaluator computes the current prompt p_i's score s_i^* at iteration i^{th}:

$$\begin{aligned} s_i^* =& \mathbb{E}_{(q_b,y_b)\sim\mathcal{Q}_b}\mathrm{F1}(f(p_i{+}q_b), y_b) \qquad (4)\\ &- \mathbb{E}_{(q,y)\sim\mathcal{Q}_t}\mathrm{F1}(f(p_i{+}q), y), \qquad (5)\end{aligned}$$

where f is GPT-4o model. Intuitively, we want to improve the generative response measured in standard F1 score for Q&A task for benign set and decrease such F1 score for the target set (includes one targeted question and 10 paraphrases).

Analyzer receives prompt score s_i^* and a set of incorrectly predicted examples in the benign set as additional feedback as input to GPT-4o to analyze performance failures and generate insights for improving prompt quality.

Prompt Generator iteratively generates a new prompt using the history of previously generated ones, their corresponding scores and analysis. The goal is to improve the adversarial effectiveness by combining insights from the past. After a maximum of t iterations, the prompt with the highest adversarial score is selected as the initial malicious prompt p_0^*.

Algorithm 1 Adversarial Prompt Optimization

```
1: Input: A hand-crated system prompt $p_0$, maximum # perturbed words $max\_perturbs$, $\mathcal{Q}_t{=}\{q, y\}$, and $\mathcal{Q}_b{=}\{q_b, y_b\}$,
2: Output: Optimized malicious prompt $p^*$
3: $p_0^* = \mathrm{AdvAutoPrompt}(p_0, \mathcal{Q}_t, \mathcal{Q}_b)$
4: $L_0 = \mathcal{L}(p_0^*, \mathcal{Q}_t, \mathcal{Q}_b)$
5: $I \leftarrow \{\}$
6: for $w_j \in p_0^*$ do
7:     $p^*_{\setminus w_j} = [w_1, \ldots, w_{j-1}, \texttt{[MASK]}, \ldots, w_n]$
8:     $I_{w_j} = \mathcal{L}_0 - \mathcal{L}(p^*_{\setminus w_j}, \mathcal{Q}_t, \mathcal{Q}_b)$
9:     $I[j] = I_{w_j}$
10: end for
11: n_perturbs ← 0; f ← filtered words
12: while n_perturbs≤max_perturbs and $w_j \notin$ f do
13:     $w_j^*$ = getBestPerturbation($w_j$)
14:     dummy = replace($p_0^*$, $w_j$, $w_j^*$)
15:     $L_p = \mathcal{L}(\mathrm{dummy}, \mathcal{Q}_t, \mathcal{Q}_b)$
16:     If $L_p < L_0$ then update $p^* \leftarrow$ dummy
17:     If if_success($p^*$, $\mathcal{Q}_t$, $\mathcal{Q}_b$) then return $p^*$
18: end while
19: return $s^*$
```

We do not impose any specific mechanism for constraining CAIN to satisfy the semantic similarity constraint in Eq. (3) due to our observations that there was hardly any leakage of information from target questions to our malicious prompts via *AdvAutoPrompt*. We later confirm our prompt's stealthiness in Table 4 and Sec. 7.

4.2 Stage 2: Greedy Word-Level Optimization

Since AutoPrompt is originally designed to curate a system prompt for an overarching task like Q&A, generating a malicious prompt as a whole via AdvAutoPrompt that is optimal for a specific target question is both noisy and inefficient due to unlimited search space of all possible sentences. Although p_0^* can achieve the attack objective with some effectiveness, further refinement via Stage 2 is required to maximize its adversarial impact.

4.2.1 Compute Word Importance Score

Before we can exercise greedy word-level optimization, we need to determine which word to optimize first. Thus, we approximate the importance of each word within the prompt p_0^* to the model's behavior. This is achieved by iteratively removing each word and measuring its impact on the model's loss to the current attack (Alg. 1, Ln. 6-10):

$$I_{w_i} = \mathcal{L} - \mathcal{L}_{\setminus w_i}, \tag{6}$$

where $\mathcal{L}$ is either the untargeted in Eq. 1 or the targeted loss in Eq. 2.

4.2.2 Iterative Token Perturbations

Next, we refine the current malicious system prompt by applying perturbations to its most influential words as identified in the previous step. Specifically, we apply five types of perturbations found in adversarial text literature (Jin et al., 2019; Gao et al., 2018), including *(1) Random Split* splits a word into two separate words at a random position, *(2) Random Swap* swaps the positions of two randomly selected characters within a word, *(3) Substitute Keyboard* replaces a character with a neighboring character on a QWERTY keyboard, and *(4) Substitute Synonym* replaces a word with one of its synonyms using WordNet (Miller, 1994) (Alg. 1, Ln. 13).

For each perturbation applied to a word, we then select and retain only the perturbation that best minimizes the respective loss to the next iteration. This ensures that all perturbations enhance adversarial effectiveness without significantly degrading performance on benign examples (Alg. 1, Ln. 16). For the stopping criteria, we evaluate at each iteration whether a maximum allowable number of perturbed words is reached or whether the attack is successful (Alg. 1, Ln.. 17). We define an successful attack only when the current optimized prompt p^* has to fool the target LLM at least k questions in the target set $\mathcal{Q}$ and maintain at least m correct answers in the benign set $\mathcal{Q}^*$. Based on our observations, an answer is considered incorrect if F1 ≤ 0.2 and correct if F1 ≥ 0.45.

5 Experiments: Untargeted Attack

5.1 Setup

Dataset and Data Sampling (by the Attackers). We used the TriviaQA (Joshi et al., 2017) (rc.wikipedia validation subset) without context for all experiments. CAIN randomly samples 100 correctly answered questions from each target LLM when a manual system prompt is used to construct the target subset $\mathcal{Q}_t$, and 10 correct + 10 incorrect QA pairs to construct the benign set $\mathcal{Q}_b$. Each target question is paraphrased into 10 variants to enrich diversity and reduce noise during optimization.

Generalizability Evaluation. Separate from the attack process, we construct *additional, non-overlapping subsets* for post-attack evaluation:

- **Benign Evaluation:** We construct five different benign subsets (each 200 QA pairs, 100 correct+100 incorrect), resulting in 1000 examples to evaluate the performance preservation on *unseen* benign questions.
- **Malicious Evaluation:** For each $q \in \mathcal{Q}_t$, we generate 100 paraphrases *unseen* versions to assess the generalization of the optimized prompts in practice when the users *might ask the target question in different ways.*

Metrics. We use two sets of metrics, including (1) *Predictive F1 and Exact Match (EM)*: standard Q&A metrics measuring partial and exact correctness of model prediction against ground-truths, and (2) *Performance gap* Δ*F1 and* Δ*EM* measure the difference in performance between benign and malicious tasks (e.g., ΔF1=F1$_{benign}$−F1$_{malicious}$). A higher ΔF1/EM indicates a stronger attack, meaning a greater performance drop on the target set with minimal loss on the benign set.

Target LLMs and Attack Baselines. We evaluate on six open-source LLMs, including Llama2 (Touvron et al., 2023), LLama3.1 (Grattafiori et al., 2024), Deepseek (DeepSeek-AI et al., 2025), Qwen (Qwen et al., 2025), Pythia (Biderman et al., 2023) with the black-box attack baselines:

- **No system prompt (*NSP*)**: Questions are fed to LLMs without any instructions.
- **Manual**: A hand-crafted Q&A system prompt.
- **AdvAutoPrompt (*AAP*)**: Partially malicious prompt produced by a customized adversarial version of AutoPrompt (Levi et al., 2024) formulated in Sec. 4.1.
- CAIN: Our proposed attack method that combines AAP with greedy word-level optimization.

5.2 Results

Table 1 reports F1 and EM on Benign and Malicious Evaluation sets. Key findings include: (1) CAIN *consistently demonstrates superior adversarial performance on malicious tasks across models*, with notably low F1 and EM scores, even with paraphrased versions of the target question, (2) *AAP exhibits strong malicious F1 compared to Manual* on most of target LLMs, although in-

	Prompt	Benign		Malicious		Difference	
		F1↑	EM↑	F1↓	EM↓	ΔF1↑	ΔEM↑
Llama2-7B	NSP	66.48	56.10	61.00	61.00	5.48	-4.90
	Manual	73.09	68.90	54.00	54.00	<u>19.09</u>	<u>14.90</u>
	AAP	66.31	58.88	79.19	73.23	-12.88	-14.35
	CAIN	63.84	56.14	33.36	28.20	**30.48**	**27.94**
Llama2-13B	NSP	76.29	67.70	97.10	95.00	-20.81	-27.30
	Manual	85.00	82.60	96.50	94.00	-11.50	-11.40
	AAP	82.14	78.72	82.46	74.30	<u>-0.32</u>	<u>3.92</u>
	CAIN	66.77	57.14	32.66	18.89	**34.11**	**38.15**
Deepseek-7B	NSP	56.42	48.90	100.00	100.00	-43.58	-51.10
	Manual	52.11	49.80	100.00	100.00	-47.89	-50.20
	AAP	52.49	42.11	69.71	58.14	<u>-17.22</u>	<u>-16.03</u>
	CAIN	43.99	31.75	28.15	16.33	**15.84**	**15.42**
Qwen2.5-7B	NSP	70.33	65.30	82.12	81.36	-11.79	-16.06
	Manual	56.74	49.10	95.47	95.00	-38.73	-45.90
	AAP	56.06	45.72	53.67	43.90	<u>2.39</u>	<u>1.82</u>
	CAIN	50.31	39.20	34.94	23.92	**15.37**	**15.28**

Table 1: Performance comparison when attacking various target LLMs. **Bold** and <u>underlined</u> values indicate the best and second-best ΔF1/EM improvements, respectively.

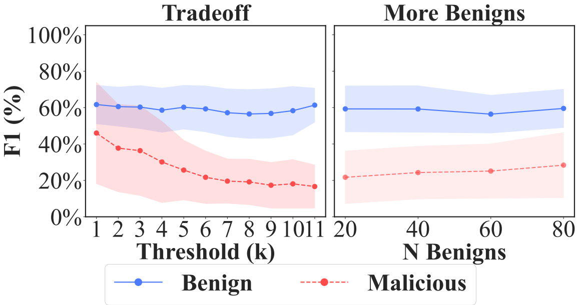


Figure 3: Ablation study on varying the threshold k and numbers of benign questions on Llama2-13B.

creased malicious scores on Llama2-7B, and (3) *NSP and Manual prompts show strong robustness on benign and malicious task* with high F1 scores.

CAIN significantly reduces malicious F1 scores on Llama2-7B and Deepseek-7B by 20.64 and 71.85, respectively, while only modestly affecting benign F1 (drops of 9.24 and 8.12), compared to Manual (Table 1). For Llama2-13B and Qwen2.5-7B, benign performance drops by 20 points, possibly due to the limited number of benign samples. However, Figure 3 (right) shows that increasing benign questions does not improve performance, which we leave for future study.

In contrast, Manual and AAP exhibit inconsistent attack performance. AAP increases malicious F1 on Llama2-7B (79.19), while Manual fails to degrade malicious accuracy (e.g., 100 F1 on Deepseek-7B).

6 Experiments: Targeted Attack

6.1 Setup

Dataset and Data Sampling (by the Attackers): We use six categories from TruthfulQA (Lin et al., 2022): Misconceptions, Conspiracy, Stereotype, Health, Politics, History. The statistical information is shown in Table A1 (Appendix). For each category, we randomly select 10 questions as targeted answers for attacking and paraphrase them into 20 versions (10 will be used for attacking, 10 will be used for evaluating the generalization of the optimized prompts), except for Politics, where only five Q&A pairs are available.

Generalizability Evaluation. Separate from the attack process, we construct *additional, non-overlapping subsets* for post-attack evaluation:

- **Benign Evaluation:** For each category, 5 other questions are selected and paraphrased into 50 variants to evaluate generalization on unseen but non-targeted queries.
- **Malicious Evaluation:** The other 10 paraphrased versions of each target question as mentioned above are used for evaluation.

Metrics. In addition to F1/EM, the aggregated performance is computed as $\Psi\text{F1}{=}\frac{1}{2}(\text{F1}_{\text{benign}}{+}\text{F1}_{\text{malicious}})$, where higher values indicate stronger balance between attack success and benign preservation.

Target LLMs and Attack Baselines. We utilize the same attack baselines as in the untargeted attack setting (§5). All experiments are conducted on Deepseek-7B, Qwen2.5-7B, and Llama2-7B.[1]

Evaluation Settings. Models are prompted to select one answer from different answering formats of increasing difficulties: (1) Two options (A or B), (2) Four options (A, B, C or D), and (3) Free-form text (no explicit choices are provided). We used the two-option format during attacking and transferred the resulting malicious prompts to four options and free-form text for evaluation. Figure A1 illustrates the input format for two options.

6.2 Results

We report results for the two-option setting, its transfer to four-option, and to free-form generation in Table 2. Key findings include: (1) *combining AdvAutoPrompt with greedy word-level optimization consistently achieves superior overall performance*

[1] Due to space, Llama2-7B's results are in the Appendix

	Prompt	Two options						Two options→Four options						Two options→Free-form					
		F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Deepseek-7B	NSP	53.12	51.67	39.38	37.82	46.25	44.75	28.24	26.00	25.13	24.36	26.68	25.18	43.67	43.67	47.55	47.55	45.61	45.61
	Manual	26.67	26.67	34.67	34.55	30.67	30.61	16.83	16.00	34.85	34.73	25.84	25.36	1.00	1.00	0.18	0.18	0.59	0.59
	AAP	52.75	45.32	49.66	44.36	51.20	44.84	32.14	25.75	35.45	30.18	33.80	27.96	42.35	41.94	51.36	50.73	46.86	46.34
	CAIN	55.29	46.47	58.92	54.00	**57.11**	**50.23**	31.73	28.69	43.92	43.00	**37.83**	**35.84**	45.31	45.05	56.25	56.25	**55.28**	**50.65**
Qwen2.5-7B	NSP	52.33	52.33	51.33	51.09	51.83	51.71	57.00	57.00	54.06	54.00	55.53	55.50	52.89	52.67	48.15	47.27	50.52	49.97
	Manual	58.39	52.67	41.52	40.36	49.95	46.52	49.67	49.67	44.94	44.73	47.30	47.20	37.33	37.33	58.55	58.36	47.94	47.85
	AAP	66.24	66.24	36.34	35.82	51.29	51.03	55.61	55.59	36.55	36.00	46.08	45.80	46.68	46.68	55.94	55.09	51.31	50.89
	CAIN	62.76	61.75	73.03	72.07	**67.89**	**66.91**	56.45	56.38	70.75	70.34	**63.60**	**63.36**	41.55	41.55	62.18	60.69	**51.81**	**51.07**

Table 2: Performance on the targeted attack across various target LLMs. Prompts optimized under the two-option format are evaluated directly and transferred to multiple-choice and free-form settings. **Bold** indicates best, underline indicates second-best per block.

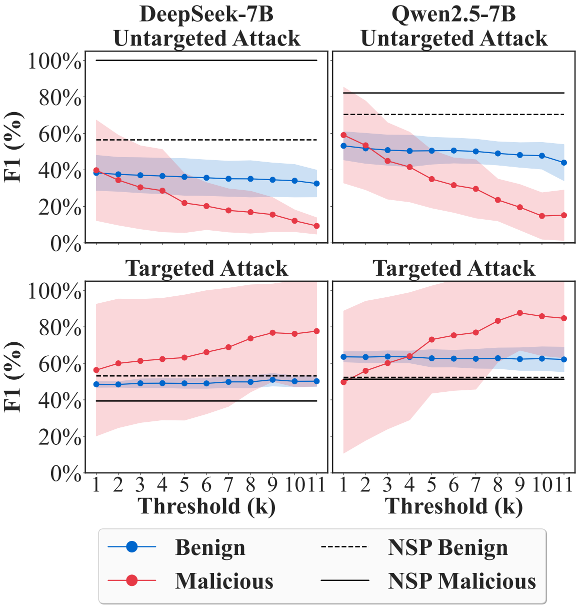


Figure 4: Trade-off between benign and malicious F1 scores as the threshold k increases for CAIN.

in benign and malicious responses, highlighting its strong adversarial power, and (2) *strong resilience and adaptability to different levels of difficulty,* with greedy decoding consistently achieved high performance across evaluation scenarios, emphasizing the crucial role of this optimization strategy.

In two-option scenario, CAIN achieves strong malicious performance (73.03 F1) while preserving benign F1 of 62.76, outperforming baselines such as NSP and Manual prompts.

In four-option scenario, overall, predictive performance drops when adding more options and yet CAIN remains the most effective on malicious set, reaching 43.92 F1 (Deepseek-7B) and 70.75 (Qwen2.5-7B), indicating good prompt transferability to harder multiple-choice options.

In free-form scenario when no explicit choices are provided, CAIN continues to outperform baselines (e.g., 56.25 F1 malicious and 45.31 F1 benign on Deepseek-7B). In contrast, Manual and AAP degrade significantly due to reliance on multiple-choice formatting. Overall, CAIN offers a stronger trade-off in F1 score between malicious (62.18) and benign set (41.55). These results confirm that CAIN achieves a *superior balance between attack success and benign robustness in targeted attacks* across different prompting formats.

7 Discussion

Affects of Optimization Threshold k. We investigate the trade-offs between benign performance and synthetic target when adjusting the number of incorrect target thresholds $k \in [1..11]$ (Alg. 1, Ln.. 17). As shown in Figure 4, increasing k consistently improves attack effectiveness while benign performance remains stable. This highlights a tunable trade-off between stealth and potency, allowing attackers to adjust aggressiveness depending on the security scenario. The full analysis is in A.3.

Affects of Model Sizes. We evaluate how model size impacts CAIN's effectiveness using Qwen2.5 with the number of parameters increasing from 3B to 32B. As shown in Figure 5 and 3 (left), CAIN consistently achieves stronger adversarial performance than AAP across both targeted and untargeted attacks, with benign performance improving as model size increases. These findings highlight CAIN's consistent malicious impact across varying model complexities.

Affects of Prompt Initialization Methods. Across both untargeted and targeted settings, A+Greedy consistently outperforms M+Greedy in attack effectiveness and robustness. In untargeted attacks (Table A3), A+Greedy yields stronger performance

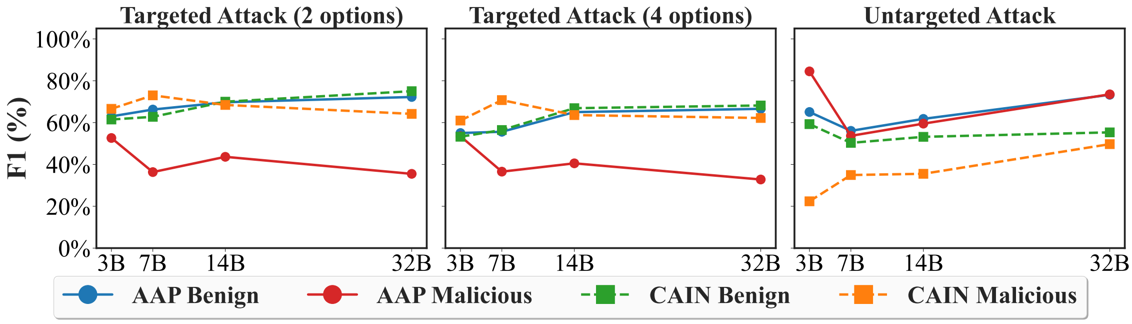


Figure 5: Performance of each attacking scenario across model sizes.

	Prompt	Benign		Malicious		Difference	
		F1↑	EM↑	F1 ↓	EM ↓	ΔF1 ↑	ΔEM↑
4o-mini	Manual	68.22	51.56	99.28	99.09	-31.06	-47.53
	AAP	76.66	65.41	94.48	91.21	-17.82	-25.80
	CAIN	71.44	59.16	52.44	48.64	**19.00**	**10.52**
4.1-nano	Manual	62.47	51.03	95.00	95.00	-32.53	-43.97
	AAP	64.43	54.25	97.53	93.94	-33.10	-39.69
	CAIN	56.79	42.43	81.76	65.00	**-24.97**	**-22.57**
3.5Turbo	Manual	69.15	51.52	99.55	99.55	-30.40	-48.03
	AAP	66.93	49.58	96.57	96.36	-29.64	-46.78
	CAIN	61.00	40.09	69.47	64.55	**-8.47**	**-24.46**

Table 3: Untargeted performance across different attack methods on various OpenAI APIs.

gaps (e.g., +30.48 ΔF1 on L2-7B, +15.84 on D-7B), indicating better degradation of malicious responses while preserving benign ones. In targeted attacks (Table A5), it achieves higher aggregated scores (Ψ), particularly on Qwen2.5-7B and Deepseek-7B, showing better generalization across formats. These results confirm A+Greedy as a superior initialization strategy for attack strength and transferability.

7.1 Case Study: Attacking Commercial LLMs

We evaluate CAIN under untargeted attacks on popular commercial LLM-based chatbot APIs, including GPT3.5-turbo, GPT-4.1-nano and GPT-4.1-mini. Due to budget constraints, we randomly selected 10 target questions to attack. Table 3 shows that CAIN consistently reduces F1 score on malicious tasks by 46.84 points for 4o-mini, compared to the manual prompt, while better preserving benign performance by increasing to 71.44 F1.

Moreover, compared to AAP, CAIN consistently achieves lower malicious F1 across models (e.g., 69.47 vs. 96.57 on GPT-3.5-Turbo; 81.76 vs. 97.53 on GPT-4.1-nano), highlighting stronger attack success. While GPT-4.1-nano appears more robust (ΔF1 of -24.97 for CAIN vs. -33.10 for AAP), the results demonstrate CAIN's effectiveness even against well-aligned commercial models under black-box conditions, confirming the feasibility of our security threat in practice.

TargetLLM	Targeted	UnTargeted
Deepseek-7B	0.0217	0.0819
Qwen2.5-7B	0.0426	0.0417

Table 4: Averaged cosine similarity between successfully optimized prompts and the targeted questions.

Potential Defense. Our findings suggest that traditional defenses, such as detecting lexical similarity between prompts and target questions or using a perplexity-based filtering (Jain et al., 2023) are insufficient to defend against CAIN. Table 4 shows that the optimized prompts exhibit very low cosine similarity to their respective targets (average of 0.0518 for Deepseek-7B and 0.04215 for Qwen2.5-7B), indicating they do not leak any lexical overlap with the target questions. Figure A2 shows that CAIN's prompts have variable perplexity levels, and using a PPL filtering might work to some extent, but this approach will not be a comprehensive solution. These results underscore CAIN's subtlety and highlight the urgent need for more robust, behavior-based detection mechanisms.

8 Conclusion

We introduce CAIN, a black-box method that reveals a new vulnerability in LLMs: targeted prompt-based manipulation that preserves benign behavior. CAIN achieves substantial degradation on targeted questions, up to 40% F1 in untargeted attacks and over 70% F1 in targeted ones, without noticeably affecting benign performance. These attacks remain stealthy, transferable across model architectures, and evade traditional defenses such as lexical similarity or perplexity-based filtering.

Limitation

While CAIN demonstrates strong targeted manipulation in black-box settings, it faces several important limitations. First, achieving high adversarial effectiveness occasionally comes at the cost of benign performance. CAIN outperforms baselines on OpenAI APIs, the overall attack success remains limited due to alignment constraints in commercial systems. Finally, while CAIN evades common lexical and perplexity-based filters, this also underscores a broader limitation in the field: the lack of robust, behavior-aware defenses. Addressing these challenges will be crucial for advancing both offensive and defensive research in LLM alignment.

Broader Impacts and Ethics Statement

This work reveals a previously underexplored vulnerability in large language models (LLMs): the ability to craft adversarial system prompts that selectively cause incorrect responses to specific questions while maintaining accurate outputs on benign inputs. Such selective manipulation poses a subtle but serious threat, particularly in domains involving misinformation, political influence, or public health. Unlike traditional jailbreaks or universal attacks, CAIN operates stealthily, evading detection by standard lexical similarity and perplexity filters.

We intend to raise awareness of this threat and prompt the development of more robust, behavior-based defenses. All experiments were conducted in controlled settings using open-source models, and evaluations on commercial APIs were performed to assess practical limitations — not for misuse. While the techniques may be misused, we believe that exposing this vector responsibly contributes to a more secure and trustworthy deployment of LLMs. We advocate for responsible disclosure, transparent benchmarking, and the implementation of proactive safeguards in future LLM systems.

References

John W. Ayers, Adam Poliak, Mark Dredze, Eric C. Leas, Zechariah Zhu, Jessica B. Kelley, Dennis J. Faix, Aaron M. Goodman, Christopher A. Longhurst, Michael Hogarth, and Davey M. Smith. 2023. Comparing physician and artificial intelligence chatbot responses to patient questions posted to a public social media forum. *JAMA Internal Medicine*, 183(6):589–596.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Advik Raj Basani and Xiao Zhang. 2024. Gasp: Efficient black-box generation of adversarial suffixes for jailbreaking llms. *ArXiv*, abs/2411.14133.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *Preprint*, arXiv:2304.01373.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

J. Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. In *Annual Meeting of the Association for Computational Linguistics*.

J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Xingang Guo, Fangxu Yu, Huan Zhang, Lianhui Qin, and Bin Hu. 2024. Cold-attack: Jailbreaking llms with stealthiness and controllability. *arXiv preprint arXiv:2402.08679*.

Raghav Gupta, Ryan Sullivan, Yunxuan Li, Samrat Phatale, and Abhinav Rastogi. 2025. Robust multi-objective preference alignment with online dpo. In *AAAI Conference on Artificial Intelligence*.

Lynn Hasher, David M. Goldstein, and Thomas C. Toppino. 1977. Frequency and the conference of referential validity. *Journal of Verbal Learning and Verbal Behavior*, 16:107–112.

Jonathan Hayase, Ema Borevkovic, Nicholas Carlini, Florian Tramèr, and Milad Nasr. 2024. Query-based adversarial prompt generation. *ArXiv*, abs/2402.12329.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. 2023. Baseline defenses for adversarial attacks against aligned language models. *Preprint*, arXiv:2309.00614.

Weipeng Jiang, Zhenting Wang, Juan Zhai, Shiqing Ma, Zhengyu Zhao, and Chao Shen. 2025. An optimizable suffix is worth a thousand templates: Efficient black-box jailbreaking without affirmative phrases via llm as optimizer. *Preprint*, arXiv:2408.11313.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*.

Erik Jones, Anca D. Dragan, Aditi Raghunathan, and Jacob Steinhardt. 2023. Automatically auditing large language models via discrete optimization. *ArXiv*, abs/2303.04381.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Elad Levi, Eli Brosh, and Matan Friedmann. 2024. Intent-based prompt calibration: Enhancing prompt optimization with synthetic boundary cases.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Eryn J. Newman, Mevagh Sanson, Emily K. Miller, Adele Quigley-Mcbride, Jeffrey L. Foster, Daniel M. Bernstein, and Maryanne Garry. 2014. People with easier to pronounce names promote truthiness of claims. *PLoS ONE*, 9.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Anselm Paulus, Arman Zharmagambetov, Chuan Guo, Brandon Amos, and Yuandong Tian. 2024. Advprompter: Fast adaptive adversarial prompting for llms. *ArXiv*, abs/2404.16873.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Conference on Empirical Methods in Natural Language Processing*.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. Cold decoding: Energy-based constrained text generation with langevin dynamics. *Advances in Neural Information Processing Systems*, 35:9538–9551.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *CoRR*, abs/2010.15980.

Harry Surden. 2019. Artificial intelligence and law: An overview. *Georgia State University law review*, 35:15109.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *Conference on Empirical Methods in Natural Language Processing*.

Laura Weidinger, John F. J. Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zachary Kenton, Sande Minnich Brown, William T. Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, and 4 others. 2021. Ethical and social risks of harm from language models. *ArXiv*, abs/2112.04359.

Xilie Xu, Keyi Kong, Ning Liu, Lizhen Cui, Di Wang, Jingfeng Zhang, and Mohan Kankanhalli. 2024. An LLM can fool itself: A prompt-based adversarial attack. In *The Twelfth International Conference on Learning Representations*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *ArXiv*, abs/1905.12616.

Yujie Zhao, Jose Efraim Aguilar Escamill, Weyl Lu, and Huazheng Wang. 2024. Ra-pbrl: Provably efficient risk-aware preference-based reinforcement learning. *ArXiv*, abs/2410.23569.

Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. 2024. AutoDAN: Interpretable gradient-based adversarial attacks on large language models. In *First Conference on Language Modeling*.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *Preprint*, arXiv:2307.15043.

Category	Num. Samples
Misconceptions	100
Conspiracy	26
Stereotype	24
Health	55
Politics	10
History	24

Table A1: The number of data examples used for targeted attack in each category.

A Appendix

A.1 Dataset and Prompt template

Table A1 shows the statistical information of each category in the TruthfulQA (Lin et al., 2022), which will be used for targeted attack. Figure A1 illustrates an example of the input for targeted attack during training and evaluation.

A.2 Implementation Details

For each targeted question, we used 20 benign examples (10 from the correct subset, 10 from the incorrect subset) and the correct proportion $m = 7$. For the AdvAutoPrompt, we set a maximum length of p^* up to 150 words and the number of iterations $t = 3$ to save cost.

A.3 Analysis of optimization threshold k

For untargeted attack (top row), increasing the incorrect targeted threshold (k) generally leads to a more substantial degradation of performance on the synthetic target (malicious) dataset for both models. For DeepSeek-7B, the F1 score on malicious inputs drops significantly from approximately 40% to just above 10% as k increases, while benign performance shows a modest decline from around 40% to 35%. Qwen2.5-7B exhibits a similar pattern: its malicious performance falls from roughly 55% to about 15%, and benign performance decreases from approximately 60% to 45% across the same range of k. This trend indicates that while the attack more effectively degrades performance on malicious queries at higher k values, this also comes with a slight decrease in performance on benign questions compared to their corresponding initial attacked state and a more noticeable reduction compared to the NSP benign baselines.

Conversely, for targeted attacks (bottom row), a higher threshold (k) significantly improves the success rate of forcing the specific malicious output. On DeepSeek-7B, the malicious F1 score, representing attack success, climbs from about 58% to nearly 80% with an increasing k, while performance on benign inputs remains relatively stable around 50%, comparable to its NSP benign baseline. A more pronounced trend is observed for Qwen2.5-7B, where its malicious attack success rate rises from approximately 50% at $k = 1$ to over 80% for $k \geq 8$; its benign performance also remains stable at around 62%. Notably, this increased targeted efficacy is generally achieved without a substantial negative impact on the models' performance on benign inputs.

[Instruction]: The following is a multiple-choice question; you are given two possible answers. Please only select one of them without any explanations:
Q: In the Gregorian calendar, what does ``AD" mean?
Choices:
A: Anno Domini
B: After Death

[Target LLM]: B

Figure A1: The prompt template for two options of targeted attack.

	Prompt	Benign		Malicious		Difference	
		F1↑	EM↑	F1 ↓	EM ↓	ΔF1 ↑	ΔEM↑
Llama3.1	NSP	58.59	47.60	88.61	83.00	-30.02	-35.40
	Manual	64.25	56.60	99.75	99.50	-35.70	-42.90
	AAP	44.84	31.70	52.00	42.00	-7.16	-10.30
	CAIN	45.15	32.04	27.46	16.40	**17.69**	**15.64**
Pythia	NSP	40.98	28.50	97.40	97.00	-56.42	-68.50
	Manual	54.82	49.00	100.00	100.00	-45.18	-51.00
	AAP	49.13	40.06	58.20	51.27	-9.07	-18.14
	CAIN	49.08	40.70	32.32	25.28	**16.76**	**15.42**

Table A2: Performance comparison when attacking on Pythia-12B and Llama3.1-7B.

A.4 Additional results on untargeted attack

Table A2 presents the performance of untargeted attacks on Llama3.1-7B and Pythia-12B. Across both models, CAIN significantly outperforms all baselines, including Manual and AAP, in balancing attack strength and benign performance. While Manual prompts achieve high benign F1/EM, they fail to reduce malicious performance (e.g., 99.75 F1 on Llama3.1 and 100.00 F1 on Pythia). In contrast, CAIN reduces malicious F1 to 27.46 and 32.32, respectively, while maintaining reasonable benign scores. This results in the highest ΔF1 and ΔEM margins (e.g., +17.69 F1 on Llama3.1 and +16.76

	Prompt	Benign		Malicious		Difference	
		F1↑	EM↑	F1↓	EM↓	ΔF1↑	ΔEM↑
L2-7B	M+G	68.33	62.59	38.25	31.46	30.08	**31.13**
	A+G	63.84	56.14	33.36	28.20	**30.48**	27.94
L2-13B	M+G	81.92	78.62	41.44	38.36	**40.48**	**40.26**
	A+G	66.77	57.14	32.66	18.89	34.11	38.15
L3.1-8B	M+G	62.61	52.12	50.05	41.69	12.56	10.43
	A+G	45.15	32.04	27.46	16.40	**17.69**	**15.64**
D-7B	M+G	53.59	48.41	37.73	33.28	15.66	15.13
	A+G	43.99	31.75	28.15	16.33	**15.84**	**15.42**
Q2.5	M+G	46.97	36.13	61.39	50.68	-14.42	-14.55
	A+G	50.31	39.20	34.94	23.92	**15.37**	**15.28**
P-12B	M+G	50.25	42.90	40.46	34.41	9.79	8.49
	A+G	49.08	40.70	32.32	25.28	**16.76**	**15.42**

Table A3: Results of attacking performance with manual initialization and AdvAutoPrompt. "A" denotes AAP, "G" stands for Greedy, and "A+G" is our proposed method. "L, D, Q, P" denote Llama, Deepseek, Qwen, and Pythia models, respectively.

on Pythia), demonstrating CAIN's superior ability to selectively degrade targeted outputs without broadly compromising accuracy.

A.5 Additional results for targeted attacks

Table A4 shows that AAP achieves the best performance in the two-option setting (ΨF1 = 50.94, ΨEM = 42.58), but its effectiveness drops when transferred to the four-option format. In contrast, CAIN maintains more stable performance across both settings, achieving strong targeted attack success (highest malicious F1) with better transferability (ΨF1 = 31.01 vs. 32.80). This suggests CAIN is more robust and generalizable under realistic conditions where question formats vary.

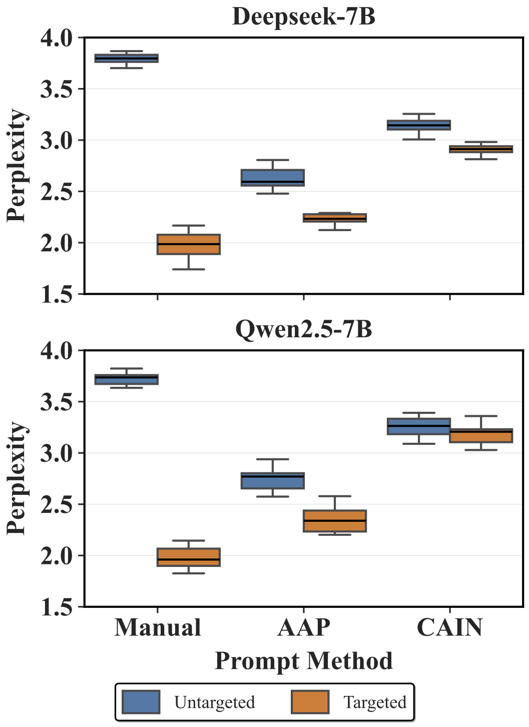


Figure A2: Perplexity distribution of successfully optimized prompts across different prompt methods under both untargeted and targeted attack.

	Prompt	Two options						Two options→Four options					
		Benign		Malicious		Sum		Benign		Malicious		Sum	
		F1↑	EM↑	F1 ↑	EM ↑	Ψ F1↑	Ψ EM ↑	F1 ↑	EM↑	F1↑	EM↑	Ψ F1↑	Ψ EM↑
Llama2-7B	NSP	10.03	1.67	14.09	3.27	12.06	2.47	9.72	1.33	19.60	1.82	14.66	1.58
	Manual	19.67	19.00	44.03	43.45	31.85	31.23	1.00	1.00	3.50	3.27	2.25	2.13
	M+Greedy	28.20	21.57	60.71	58.00	44.45	39.78	13.73	10.60	29.81	28.80	21.77	19.70
	AAP	42.47	34.16	59.41	51.00	**50.94**	**42.58**	27.30	19.88	38.29	33.27	**32.80**	**26.58**
	CAIN	35.10	19.61	61.86	48.71	48.48	34.16	20.44	11.83	41.58	34.84	31.01	23.34

Table A4: Performance of the targeted attack on Llama2-7B.

	Prompt	Two options						Two options→Four options					
		Benign		Malicious		Sum (Ψ)		Benign		Malicious		Sum (Ψ)	
		F1↑	EM↑	F1 ↑	EM ↑	F1↑	EM ↑	F1 ↑	EM↑	F1↑	EM↑	F1↑	EM↑
Deepseek-7B	M+Greedy	47.94	46.11	42.90	42.26	45.42	44.19	27.38	25.65	30.32	29.35	28.85	27.50
	A+Greedy	55.29	46.47	58.92	54.00	**57.11**	**50.23**	31.73	28.69	43.92	43.00	**37.83**	**35.84**
Qwen2.5-7B	M+Greedy	60.41	60.33	62.73	62.73	61.57	61.53	50.88	50.88	69.70	69.70	60.29	60.29
	A+Greedy	62.76	61.75	73.03	72.07	**67.89**	**66.91**	56.45	56.38	70.75	70.34	**63.60**	**63.36**

Table A5: Performance with different initialization methods on targeted attacks.