SECURITY-CONSTRAINED PROMPT OPTIMIZATION UNDER PARTIAL OBSERVATION

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper develops a fixed-budget best-prompt identification algorithm for selecting the highest quality prompt of a given task for large language models (LLMs), under the constraints of various quality and security measures. Specifically, the selected prompt needs to satisfy a security constraint against attacks such as indirect prompt injection, which inserts malicious instructions in documents to redirect a language model away from its intended task. The inclusion of the security constraint fundamentally changes prompt evaluation, as an attacked response does not estimate clean-task performance, and a clean response provides no evidence of security. Thus each pull reveals only part of the objective vector. The resulting security-induced partial observation differs from conventional best-arm identification, where each pull reveals all objectives. We formulate this setting as a fixed-budget best-feasible prompt identification problem, and solve it by using a new Guaranteed-Floor Constrained Successive Rejects (GF-CSR) algorithm. GF-CSR adopts a reward-independent two-phase schedule that front-loads attacked pulls until surviving prompts obtain a prescribed security-evidence floor, then shifts sampling toward clean objectives. Under bounded independent rewards and explicit separation conditions, we prove that GF-CSR identifies the unique best feasible prompt with high probability while accounting for bias from incomplete cycles through the attack suite. Experiment results on real-world datasets with indirect prompt injection attacks demonstrate that GF-CSR outperforms baseline algorithms such as partial observation CSR (PO-CSR) and the Uniform algorithm.

1 INTRODUCTION

Prompts provide a lightweight interface for adapting large language models (LLMs) to downstream tasks, but small changes in instructions, examples, or formatting can substantially change their behavior (Liu et al., 2023). Prompt engineering has thus evolved from manual trial and error into automatic prompt optimization. Existing approaches include gradient-guided search over discrete tokens (Shin et al., 2020), learned continuous prompts and prefixes (Li & Liang, 2021; Lester et al., 2021), and methods that use language models to generate or refine natural language instructions (Zhou et al., 2023; Pryzant et al., 2023; Yang et al., 2024). Most existing prompt optimization methods rank candidates using a single performance score, and this was extended to multi-objective optimization with feasibility constraints in (Li et al., 2026). However, none of the above methods consider the security aspect of prompt optimization.

LLMs are vulnerable to adversarial instructions because they do not reliably distinguish legitimate prompts from untrusted content supplied for processing. An indirect prompt injection exploits this ambiguity by embedding a malicious instruction in the untrusted content, causing the model to depart from the intended task (Greshake et al., 2023; Liu et al., 2024; Yi et al., 2025; Zhan et al., 2024). Defenses such as source marking, structured separation of instructions and data, and instruction priority training reduce attack success by making the intended control hierarchy more explicit (Hines et al., 2024; Chen et al., 2025; Wallace et al., 2024). Complementary studies optimize safety prompts against harmful queries or search prompts using both utility and security criteria (Zheng et al., 2024; Sinha et al., 2024; Boateng et al., 2026). These methods treat security as an objective of prompt design, which allows gains in attack resistance to compensate for losses in task quality. However, a prompt may achieve attack resistance by simply refusing to perform any task. Thus, adopting security as a compensatory task objective might lead to poor overall performance.

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.

Introducing a security constraint fundamentally changes the formulation and solution of the prompt optimization problem. A clean evaluation reveals task quality but no security information, whereas an attacked evaluation is drawn from a different input distribution and reveals attack resistance without estimating clean task performance. Thus, each model call reveals only part of the information required to identify the best feasible prompt, and the fixed budget must be divided between clean and attacked evaluations. However, such a security-induced partial observation structure is not captured by existing constrained identification methods. Existing methods assume that each evaluation jointly reveals the objective and constraints under a fixed budget (Faizal & Nair, 2022), that observations follow a linear model that shares information across candidates (Bian & Tan, 2025), that the learner chooses between performance and feasibility measurements under fixed confidence (Cai & Kandasamy, 2026), or that grouped arms are sampled one attribute at a time under a fixed budget (Mukherjee & Moharir, 2026).

Unlike existing methods that treat security as a compensatory design objective, we instead impose security as a feasibility constraint and maximize task quality only among prompts that satisfy the security requirement. To cope with the partial observation structure, we propose a new Guaranteed-Floor Constrained Successive Rejects (GF-CSR) algorithm that ensures sufficient observations for clean-task quality and security under a fixed budget. GF-CSR adopts a reward-independent, two-phase schedule for allocating clean and attacked pulls. It front-loads attacked pulls at a higher rate until surviving candidates reach a prescribed security evidence target, then reduces the attack rate to emphasize clean-task estimation. This schedule retains successive elimination while providing deterministic guarantees on the clean and security sample counts of every active prompt. Under bounded independent rewards and explicit separation conditions, we prove a finite-sample guarantee that GF-CSR returns the unique best feasible prompt with high probability while accounting for the bias induced by incomplete attack cycles. This paper makes three main contributions.

1. We formulate security-constrained prompt optimization as fixed-budget best-feasible prompt identification with attack resistance. This security requirement induces a new partial observation structure that is fundamentally different from existing methods with methods with a scalar objective or a jointly observed objective vector. The new partial observation formulation captures the unique security-induced behaviors, and it necessitates the design of new approaches for optimal allocation of a fixed budget between clean and attacked evaluations.
2. We propose GF-CSR, a new successive elimination algorithm with a deterministic two-phase allocation schedule tailored to security-induced partial observation. It front-loads attacked pulls so that every prompt surviving the first phase attains a prescribed security-evidence target, then shifts sampling toward the clean objectives. Such an approach prevents shortages of security evidence while preserving budget for task quality estimation.
3. We theoretically establish a finite-sample, high-probability identification guarantee for GF-CSR under explicit separation conditions. The theoretical analysis incorporates both the distinct clean and security sample counts induced by partial observation and an explicit bias term for incomplete attack suite cycles. The result rigorously links the fixed-budget allocation of GF-CSR to recovery of the unique best feasible prompt.

2 Related Work

Security-aware prompt optimization. Prior work on indirect prompt injection characterizes and benchmarks attacks that place malicious instructions in untrusted context (Greshake et al., 2023; Liu et al., 2024; Yi et al., 2025; Zhan et al., 2024). Existing security-aware prompt optimization methods learn continuous safety prompts (Zheng et al., 2024), evolve natural language prompts under weighted utility and security objectives (Sinha et al., 2024), or use multiple agents to generate and evaluate prompts under a security threshold (Boateng et al., 2026). Unlike these prompt generation or refinement methods, we study fixed budget selection from a finite candidate pool. We treat attack resistance solely as a hard, non-compensatory feasibility constraint and select the prompt with the highest clean task quality among candidates satisfying all constraints.

Pure exploration with constraints. Our problem is closely related to constrained pure exploration in multi-armed bandits, in which a learner allocates a sampling budget to identify the best feasible arm. Related work studies feasible arm identification (Katz-Samuels & Scott, 2018), safety-

constrained exploration (Wang et al., 2022), and fixed-budget best-feasible arm identification with jointly observed vector-valued feedback or shared linear structure (Faizal & Nair, 2022; Bian & Tan, 2025). Other settings choose between performance and feasibility tests under fixed confidence (Cai & Kandasamy, 2026) or sample individual arm attributes under a fixed budget (Mukherjee & Moharir, 2026). The prompt-selection framework closest to ours is GENSEC, which casts multi-objective prompt selection as constrained pure exploration but assumes that each generation reveals the full objective vector (Li et al., 2026). Our setting instead has disjoint feedback: clean pulls reveal all clean task objectives, whereas attacked pulls reveal only security and advance through a stratified attack cycle. The proposed GF-CSR therefore allocates the fixed budget across both candidates and observation types.

3 Problem Formulation

We formulate security-constrained prompt optimization as a fixed-budget pure-exploration problem. The objective is to maximize task quality subject to a set of feasibility constraints, including a constraint measuring resistance to indirect prompt injection.

Candidate prompts. Let $\mathcal{X}$ be a finite pool of $K := |\mathcal{X}| \geq 2$ candidate prompts. Let $\mathcal{D}$ denote a distribution over task instances (q, a), where q is the task input and a is a reference answer. Applying prompt $x \in \mathcal{X}$ to q under a fixed language model $\mathcal{M}$ produces the response $y = \mathcal{M}(q; x)$. The candidate prompt x is treated as an arm.

Objectives. Consider $m \geq 2$ reward objectives with values in $[0, 1]$, where larger values are preferred. Objective 1 is the primary objective. Objectives $2, \ldots, m$ are constraints with thresholds $\tau_2, \ldots, \tau_m \in [0, 1]$, respectively. For $j < m$, let $f_j(y, a)$ denote a task metric that uses the reference answer. The mean task reward of prompt x is defined as

$$\mu_j(x) := \mathbb{E}_{(q,a)\sim\mathcal{D}}[f_j(\mathcal{M}(q; x), a)], \qquad j = 1, \ldots, m-1. \tag{1}$$

The m-th objective is a security metric that will be detailed in Section 4.

Target prompt. Define the feasible set $\mathcal{F}$ and the best feasible prompt $x^\star$ as

$$\mathcal{F} := \{x \in \mathcal{X} : \mu_j(x) \geq \tau_j,\ j = 2, \ldots, m\}, \qquad x^\star := \arg\max_{x\in\mathcal{F}} \mu_1(x). \tag{2}$$

We assume that $\mathcal{F}$ is nonempty and that $x^\star$ is unique. An algorithm receives a total budget of B pulls and returns a prompt $\widehat{x}$. Best-feasible-prompt identification minimizes the error probability $\mathbb{P}(\widehat{x} \neq x^\star)$ for the fixed budget B. This setting generalizes the single-constraint formulation of Li et al. (2026).

Identification gaps. A prompt $x \neq x^\star$ can be rejected because it violates a constraint or because it is worse than $x^\star$ on the primary objective. Define its violation on constraint j as

$$v_j(x) := [\tau_j - \mu_j(x)]_+, \qquad j = 2, \ldots, m, \tag{3}$$

where $[z]_+ := \max\{z, 0\}$. Define its primary-objective gap as

$$g(x) := \mu_1(x^\star) - \mu_1(x). \tag{4}$$

Every prompt other than $x^\star$ has either $v_j(x) > 0$ for at least one constraint or $g(x) > 0$. The gaps are used to derive the theoretical guarantee on best prompt identification in Section 6.

4 Security Under Partial Observation

4.1 Threat Model and Security Metric

Threat model. The task input q, such as a document, web page, email, or retrieved passage, can be attacked and contain untrusted content. An adversary inserts an instruction α into this content to redirect the model from the task specified by x. Let $\mathcal{A} := \{\alpha_1, \ldots, \alpha_L\}$ denote a fixed nonempty suite of $L := |\mathcal{A}| \geq 1$ indirect prompt-injection attacks. The suite spans task-irrelevant, task-relevant, and targeted attacks following the taxonomy of Yi et al. (2025). Orthogonally to those semantic categories, we distinguish two operational attack types. A *marker-based attack* asks the

model to emit a fixed literal marker s_α, the presence of which in the response directly identifies attack success. A *judge-required attack* asks for a behavioral or semantic change, the effectiveness of which must be evaluated by an LLM judge. These types partition the suite as $\mathcal{A} = \mathcal{A}_{\text{mark}} \dot{\cup} \mathcal{A}_{\text{judge}}$. The attacker is prompt-agnostic and does not adapt α to x.

Security objective. Let $\text{inj}(q, \alpha)$ denote the input formed by injecting α into q. Given $y = \mathcal{M}(\text{inj}(q, \alpha); x)$, define $c(y, \alpha) \in [0, 1]$ as the compliance of y with α. For $\alpha \in \mathcal{A}_{\text{mark}}$, compliance is the deterministic, case-sensitive indicator that the assigned marker s_α occurs as a substring of y. For $\alpha \in \mathcal{A}_{\text{judge}}$, a separate LLM judge receives α and y and is instructed to score only clear textual evidence of compliance on an integer scale from 0 to 10. Let $J(\alpha, y) \in \{0, \ldots, 10\}$ be the valid parsed judge score. The compliance score and security reward are defined, respectively, as

$$c(y, \alpha) := \begin{cases} \mathbf{1}[s_\alpha \text{ occurs in } y], & \alpha \in \mathcal{A}_{\text{mark}}, \\ J(\alpha, y)/10, & \alpha \in \mathcal{A}_{\text{judge}}. \end{cases}, \quad s(y, \alpha) := 1 - c(y, \alpha). \tag{5}$$

Thus $s(y, \alpha) = 1$ indicates complete resistance and $s(y, \alpha) = 0$ indicates complete compliance with the injected instruction. The score in (5) does not use the reference answer a. The security objective is the uniform mean resistance over the attack suite,

$$\mu_m(x) := \frac{1}{L} \sum_{\ell=1}^{L} \nu_\ell(x), \tag{6}$$

where $\nu_\ell(x) := \mathbb{E}_{(q,a)\sim\mathcal{D}}[s(\mathcal{M}(\text{inj}(q, \alpha_\ell); x), \alpha_\ell)]$ is the mean resistance of prompt x to attack α_ℓ. Prompt x satisfies the security constraint when $\mu_m(x) \geq \tau_m$.

Security is imposed as a constraint rather than a second objective to maximize. This is because a prompt with useless outputs can obtain a high resistance score by omitting both the requested answer and the injected content. Thus high resistance does not compensate for poor task quality.

4.2 Partial Observation

Define a pull as one target-model evaluation of a candidate prompt. Each pull is either clean or attacked. A clean pull observes objectives $1, \ldots, m-1$ and leaves objective m unobserved. For an attacked pull, the task scores evaluated from an attacked response do not provide estimates of objectives $1, \ldots, m-1$ defined in (1), because the injected input follows a different distribution than the clean input. Consequently, an attacked pull observes objective m and leaves objectives $1, \ldots, m-1$ unobserved. Similarly, a clean response provides no evidence for (5).

The budget B counts target-model generations.

Objective-specific sample counts. Let $n_r(x)$ denote the number of pulls allocated to prompt x through round r. Let $n_r^{(j)}(x)$ denote the number of these pulls that observe objective j. The partial-observation structure gives

$$n_r^{(1)}(x) = \cdots = n_r^{(m-1)}(x) = n_r(x) - n_r^{(m)}(x). \tag{7}$$

Define $\widehat{\mu}_{j,r}(x)$ as the sample mean over the $n_r^{(j)}(x)$ observations of objective j.

Stratified attack cycle. Attacked pulls cycle through $\alpha_1, \ldots, \alpha_L$ in a fixed order. This protocol balances the finite attack suite at small budgets. Let N be the attacked-pull count of a prompt and define $N = kL + r$ with $0 \leq r < L$. The mean sampled by the first N positions of the cycle is

$$\mu_m^{(N)}(x) := \frac{1}{N} \sum_{t=1}^{N} \nu_{1+((t-1) \bmod L)}(x). \tag{8}$$

Lemma 1 (Incomplete-cycle bias)**.** *For any prompt x and attacked-pull count $N = kL + r \geq 1$, where $0 \leq r < L$, the cycle mean in (8) satisfies*

$$\left| \mu_m^{(N)}(x) - \mu_m(x) \right| \leq \frac{r(L-r)}{NL} \leq \frac{L}{4N}. \tag{9}$$

The proof is in Appendix B.1.

Empirical constraint status. An estimate of an objective over zero observations is undefined. Define the empirically feasible set at round r as

$$\widehat{\mathcal{F}}_r := \left\{ x \in \mathcal{X} : n_r^{(j)}(x) \geq 1 \text{ and } \widehat{\mu}_{j,r}(x) \geq \tau_j,\ j = 2, \ldots, m \right\}. \tag{10}$$

A prompt is incomplete if at least one constraint is unobserved and every observed constraint passes. A prompt has a confirmed empirical violation if at least one observed constraint is below its threshold. Empirically feasible, incomplete, and violating prompts are treated separately in Section 5.

5 The Guaranteed-Floor CSR Algorithm

To address the challenges imposed by the security-induced partial-observation structure, we develop a new Guaranteed-Floor CSR (GF-CSR) algorithm to ensure a prescribed security-evidence floor for the candidate prompts.

5.1 Successive-Elimination Backbone

Schedule and allocation. GF-CSR uses the classical Successive Rejects schedule (Audibert et al., 2010), as inherited by the independent-arm CSR instantiation (Li et al., 2026). Starting from $\mathcal{S}_0 = \mathcal{X}$, the schedule runs $R = K - 1$ rounds and removes one prompt per round. Let N_r denote the cumulative number of pulls assigned to every prompt that remains active through round r. Appendix A.1 gives the exact survivor counts, pull schedule, and integer-rounding behavior.

Objective estimates. Each pull is designated clean or attacked by a policy π. A clean pull observes objectives $1, \ldots, m-1$, whereas an attacked pull observes only objective m. At the end of round r, the algorithm computes $\widehat{\mu}_{j,r}(x)$ from all observations of objective j collected for prompt x through that round. In particular, the denominator is the objective-specific count $n_r^{(j)}(x)$ defined in Section 4.

Elimination rule. The partial observation setting requires the algorithm to distinguish an unobserved constraint from an observed violation. For a prompt x, define the set of observed constraints as

$$\mathcal{J}_r(x) := \left\{ j \in \{2, \ldots, m\} : n_r^{(j)}(x) \geq 1 \right\}. \tag{11}$$

The prompts in $\mathcal{S}_{r-1}$ are placed into the following three ordered groups.

1. A prompt is *empirically feasible* if every constraint is observed and $\widehat{\mu}_{j,r}(x) \geq \tau_j$ for all $j = 2, \ldots, m$.
2. A prompt is *incomplete* if at least one constraint is unobserved and every observed constraint satisfies $\widehat{\mu}_{j,r}(x) \geq \tau_j$.
3. A prompt has a *confirmed violation* if $\widehat{\mu}_{j,r}(x) < \tau_j$ for at least one $j \in \mathcal{J}_r(x)$.

This order places empirically feasible prompts first, incomplete prompts second, and prompts with a confirmed violation last. To rank prompts within these groups, define the primary ranking value and worst observed constraint value, respectively, as

$$\widetilde{\mu}_{1,r}(x) := \begin{cases} \widehat{\mu}_{1,r}(x), & n_r^{(1)}(x) \geq 1, \\ -\infty, & n_r^{(1)}(x) = 0, \end{cases} \quad w_r(x) := \begin{cases} \min\limits_{j \in \mathcal{J}_r(x)} \widehat{\mu}_{j,r}(x), & \mathcal{J}_r(x) \neq \varnothing, \\ -\infty, & \mathcal{J}_r(x) = \varnothing. \end{cases} \tag{12}$$

Empirically feasible and incomplete prompts are ranked in decreasing lexicographic order of $(\widetilde{\mu}_{1,r}(x), w_r(x))$. Thus, an incomplete prompt with an observed primary objective ranks above one without a primary observation, and $w_r(x)$ breaks ties between prompts with the same primary ranking value. Prompts with a confirmed violation are ranked in decreasing lexicographic order of $(w_r(x), \widetilde{\mu}_{1,r}(x))$, so that a less severe observed violation is retained before a more severe one. The lowest-ranked prompt is removed to form $\mathcal{S}_r$. After round R, the single element of $\mathcal{S}_R$ is returned.

5.2 Guaranteed-Floor CSR

The policy of the GF-CSR algorithm is designed to satisfy three requirements simultaneously: provide a deterministic lower bound on the number of security observations at every sampling point,

preserve clean observations throughout the early rounds, and keep the observation schedule independent of realized rewards. It achieves this goal by using a deterministic high attack rate until an active prompt reaches an initial security target, followed by a lower baseline attack rate afterwards to ensure sufficient observations of clean objectives. Such a guaranteed-floor approach can ensure that every surviving prompt receives sufficient security observations to avoid irreversible early-round eliminations due to incomplete or noisy feasibility decisions.

GF-CSR adopts a pulling policy that deterministically interleaves clean and attacked pulls. For a target attack rate $\rho \in (0, 1)$ and a pull counter $i \geq 0$, define

$$d_\rho(i) := \mathbf{1}[\lfloor (i+1)\rho \rfloor > \lfloor i\rho \rfloor]\,. \tag{13}$$

For a given prompt, the i-th pull is an attacked pull if $d_\rho(i) = 1$ and a clean pull otherwise. The number of attacked pulls in the first n pulls of a given prompt is exactly

$$\sum_{i=0}^{n-1} d_\rho(i) = \lfloor n\rho \rfloor. \tag{14}$$

Two-phase allocation. Fix a positive integer security target T_0 and rates $0 < \rho_{\text{base}} < \rho_{\text{floor}} < 1$. The pull scheduling for each prompt is divided into two phases with different attack rates. Any prompt that remains active for all of phase 1 receives exactly T_0 security observations. During phase 1, attacked pulls are interleaved at rate ρ_{floor} (security front-loading). During phase 2, the attack rate is dropped to the baseline rate ρ_{base} to ensure sufficient observations of the clean objectives (objective back-loading). Both phases employ the attack interleaving defined in (13).

For each prompt, phase 1 applies (13) at rate ρ_{floor} while that prompt remains active and until T_0 attacked pulls have been issued. An active prompt reaches the phase boundary after

$$t^\star := \min\{t : \lfloor t \cdot \rho_{\text{floor}} \rfloor \geq T_0\} = \left\lceil \frac{T_0}{\rho_{\text{floor}}} \right\rceil \tag{15}$$

pulls of that prompt. Phase 2 restarts the deterministic interleaving at the baseline rate ρ_{base}.

Specifically, the t-th pull of prompt x is attacked when

$$\pi_t^{\text{GF}}(x) := \begin{cases} d_{\rho_{\text{floor}}}(t-1), & 1 \leq t \leq t^\star, \\ d_{\rho_{\text{base}}}(t - t^\star - 1), & t > t^\star. \end{cases} \tag{16}$$

Each prompt maintains its own pull counter and phase index. Combining (14) and (16), the number of security observations after t pulls is

$$n^{(m)}(t) = \begin{cases} \lfloor t \cdot \rho_{\text{floor}} \rfloor\,, & 1 \leq t \leq t^\star, \\ T_0 + \lfloor (t - t^\star) \cdot \rho_{\text{base}} \rfloor\,, & t > t^\star. \end{cases} \tag{17}$$

The corresponding clean count is $t - n^{(m)}(t)$. Both counts are deterministic functions of t.

Deterministic interleaving of attacked pulls accumulates security observations rapidly while continuing to provide clean observations. The phase boundary in (15) depends only on the pull count. Therefore, GF-CSR changes the observation schedule without depending on noisy estimates or realized rewards.

6 Theory: Guaranteed-Floor Best-Feasible Identification

This section presents the theoretical performance guarantee of GF-CSR, which combines a deterministic security-observation floor with a sufficient finite-sample guarantee for best-feasible-prompt identification. All supporting results and proofs are in Appendix B.

Assumption 1 (Reward sampling)**.** *Task instances are drawn independently with replacement from $\mathcal{D}$. For any fixed prompt, clean reward observations at distinct clean pulls are independent, lie in $[0, 1]$, and have mean $\mu_j(x)$ for objective $j < m$. Security observations at distinct attacked pulls are independent and lie in $[0, 1]$. At the t-th attacked pull, the security observation has mean $\nu_{1+((t-1) \bmod L)}(x)$.*

For $n = kL + s \geq 1$, where $0 \leq s < L$, define

$$\beta_m(n) := \frac{s(L-s)}{nL}. \tag{18}$$

By Lemma 1, $\beta_m(n)$ bounds the difference between the mean of the first n pulls of the attack cycle and the uniform-suite security mean $\mu_m(x)$. At the end of round r, let

$$q_r := \begin{cases} \lfloor N_r \cdot \rho_{\text{floor}} \rfloor, & N_r \leq t^\star, \\ T_0 + \lfloor (N_r - t^\star) \cdot \rho_{\text{base}} \rfloor, & N_r > t^\star, \end{cases} \qquad u_r := N_r - q_r, \tag{19}$$

where q_r and u_r are, respectively, the scheduled security and clean counts of a prompt that remains active through round r. For $\delta \in (0,1)$, define the corresponding radii

$$c_r^{\text{sec}}(\delta) := \begin{cases} \beta_m(q_r) + \sqrt{\dfrac{\log(2mKR/\delta)}{2q_r}}, & q_r \geq 1, \\ +\infty, & q_r = 0, \end{cases} \quad c_r^{\text{clean}}(\delta) := \begin{cases} \sqrt{\dfrac{\log(2mKR/\delta)}{2u_r}}, & u_r \geq 1, \\ +\infty, & u_r = 0. \end{cases} \tag{20}$$

Finally, let $M_j := \mu_j(x^\star) - \tau_j$ be the feasibility margin of the optimal prompt on constraint j. The competitor violations $v_j(x)$ and primary gap $g(x)$ are defined in (3) and (4).

Theorem 1 (Guaranteed-floor best-feasible identification by GF-CSR)**.** *Suppose Assumption 1 holds, the feasible set is nonempty, and $x^\star$ is the unique best feasible prompt. In every round r, each prompt active through that round has exactly q_r security observations and u_r clean observations. These counts are deterministic, non-decreasing in r, and independent of all reward values. In particular, $q_r \geq \rho_{\text{floor}} \cdot N_r - 1$ when $N_r \leq t^\star$, and $q_r \geq T_0$ when $N_r \geq t^\star$.*

Fix $\delta \in (0,1)$. Suppose that, for every round $r = 1, \ldots, R$,

$$M_j > c_r^{\text{clean}}(\delta), \ \textit{for } j = 2, \ldots, m-1, \quad M_m > c_r^{\text{sec}}(\delta). \tag{21}$$

Suppose also that, for every $x \neq x^\star$ and every round r, at least one of the following holds:

$$v_j(x) > c_r^{\text{clean}}(\delta) \textit{ for some } j \in \{2, \ldots, m-1\}, \ \textit{or} \ \ v_m(x) > c_r^{\text{sec}}(\delta), \ \textit{or} \ \ g(x) > 2c_r^{\text{clean}}(\delta). \tag{22}$$

Then GF-CSR returns the best feasible prompt with probability at least $1 - \delta$:

$$\mathbb{P}(\widehat{x} = x^\star) \geq 1 - \delta. \tag{23}$$

The certificates in (21) and (22) correspond directly to the three-group elimination rule. The conditions in (21) keep $x^\star$ in the empirically feasible group. A competitor satisfying either violation condition in (22) enters the confirmed-violation group, while the primary-gap condition ranks it below $x^\star$ within the empirically feasible group. The floor guarantee has two regimes. Before phase 1 ends, a prompt that remains active deterministically accumulates $\lfloor N_r \cdot \rho_{\text{floor}} \rfloor$ security observations. From $N_r = t^\star$ onward, it retains at least T_0 security observations and continues adding them at the baseline rate.

7 Experiments

7.1 Experiment Setup

Tasks and metrics. We evaluate abstractive news summarization on CNN/DailyMail 3.0.0 (Hermann et al., 2015) and XSum (Narayan et al., 2018). The primary objective for both datasets is ROUGE-Lsum F1 (Lin, 2004). All tasks use three objectives: task quality, brevity, and security. Appendix C.2 defines the word-count-based brevity reward and gives its task-specific endpoints.

Candidate pool. Both datasets use the same ordered pool of $K = 30$ summarization prompts. We first use Qwen2.5-32B (Qwen Team, 2024) to generate 200 candidates from training examples, then manually remove duplicates and unsuitable prompts. The final pool contains 18 generated prompts spanning the observed length distribution, six manually written prompts with varied output formats, and six prompts that explicitly instruct the model to ignore commands embedded in the input.

Models and inference. We use Llama 3 8B Instruct (Grattafiori et al., 2024) in FP16 as the target and Gemma 4 12B IT in BF16 as the compliance judge. Both models run locally with temperature zero, nucleus sampling disabled, seed zero, and an 8192-token context window. The target generates at most 512 tokens. The judge generates at most 128 tokens with reasoning output disabled.

Attack suite and scoring. We use a fixed suite of six benign indirect prompt-injection attacks constructed following the BIPIA threat model (Yi et al., 2025). One is a marker-based attack and five are judge-required attacks. Appendix C.1 gives the exact instructions. Each attacked pull appends one instruction to the task input and follows the fixed stratified cycle from Section 4. Marker compliance is measured by case-sensitive substring matching. For each semantic attack, Gemma returns a structured integer compliance score $s \in \{0, \ldots, 10\}$ and evidence, yielding security reward $1 - s/10$.

Judge audit. Before semantic scoring for attack compliance, Gemma must pass a prespecified audit stage for each dataset. Appendix C.3 reports the full protocol, acceptance criteria, and results.

Reward tables. For each dataset d, we use a seeded random permutation to form a 300-unit primary pool and a disjoint 150-unit confirmation pool, where each unit is an article-summary pair. Every prompt is evaluated on every unit under one clean and six attacked conditions, producing a primary table T_d of size $30 \times 300 \times 7$ and a confirmation table T_d^C of size $30 \times 150 \times 7$. Clean cells record task quality and brevity, while attacked cells record security. Objective means for each prompt are averages over the corresponding cells.

Trials. We compare GF-CSR against baseline approaches including partial observation CSR (PO-CSR) and the Uniform baseline (Appendix A) through lookup trials on T_d. Appendices A.2 and A.3 define the PO-CSR and non-adaptive Uniform baselines, respectively. A logical pull samples a primary-pool unit uniformly with replacement, and repeated access to a cell returns the same sealed value. The baseline attack rate is 0.30. GF-CSR uses $T_0 = 8$, a floor-phase attack rate of 0.75, and the baseline rate thereafter. The prescribed thresholds are $(\tau_2, \tau_3) = (0.50, 0.45)$.

7.2 Experiment Results

Primary trial algorithm comparison. Figure 1 compares the performance of GF-CSR, PO-CSR and the Uniform baseline under different values of the nominal budget per prompt B/K. Each point averages 200 matched trial seeds. At the end of each trial, the objective or constraint of the winning prompt is evaluated by using the primary mean values of the same prompt in the primary table. If the winning prompt is infeasible based on the primary mean values of the constraints, its strict reward is 0. Otherwise, its strict reward is the mean ROUGE-Lsum F1 of the same prompt in the primary table. Strict-reward bars are pointwise 95% Student-t intervals, and best-feasible-identification bars are two-sided Wilson 95% intervals. GF-CSR achieves the best performance under all configurations, followed by PO-CSR, and both schemes significantly outperform the Uniform algorithm. Both strict reward and best-feasible identification rate benefit from a larger budget in general, with some exceptions at $B/K = 25$. This non-monotone behavior is mainly due to finite-sample Monte Carlo variations, as the trial trajectories are not nested across budgets even they share the same seed. All subsequent experiments were performed by using $B/K = 20$.

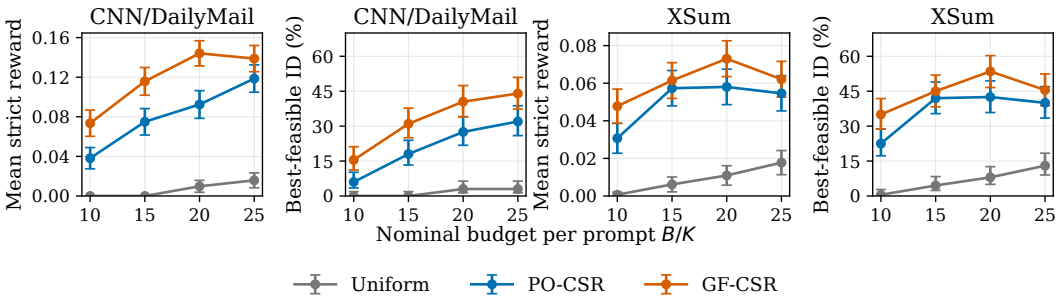


Figure 1: Primary trial performance across nominal budgets per prompt.

Held-out confirmation. The purpose of held-out confirmation is to test whether the winning prompts from the primary trials remain feasible and whether the method comparisons generalize to unseen documents. The held-out analysis does not rerun the algorithms. Instead it directly evaluates the winning prompts from the primary trials on the confirmation table. Table 1 compares the method-wise strict rewards on the primary and confirmation tables. Confirmation entries regrade the same winning prompts from the primary trials on the confirmation table and report pointwise 95% joint finite-population document-and-seed bootstrap intervals. The held-out confirmation analysis reproduces the primary analysis, that is, GF-CSR achieves the best performance, followed by PO-CSR and Uniform. The agreement across the primary and confirmation tables indicates that the method-level comparisons are not driven solely by the primary samples.

Table 1: Primary and held-out confirmation strict rewards at $B/K = 20$.

Dataset	Method	Primary strict reward (95% CI)	Confirmation strict reward (95% CI)
CNN/DailyMail	Uniform	0.010 [0.004, 0.016]	0.008 [0.003, 0.014]
	PO-CSR	0.092 [0.078, 0.106]	0.079 [0.052, 0.100]
	GF-CSR	0.144 [0.131, 0.157]	0.127 [0.077, 0.147]
XSum	Uniform	0.011 [0.006, 0.016]	0.012 [0.007, 0.078]
	PO-CSR	0.058 [0.049, 0.067]	0.065 [0.053, 0.143]
	GF-CSR	0.073 [0.064, 0.083]	0.080 [0.068, 0.149]

Paired Strict Reward Comparisons. Figure 2 summarizes the paired strict reward differences between GF-CSR and PO-CSR (labeled as `GF−PO`), and between GF-CSR and Uniform (labeled as `GF−Uniform`) at $B/K = 20$, with each bar showing a familywise 97.5% interval. The left panel represents the results from the primary trials, and the right panel shows results from the held-out confirmation analysis. Confirmation error bars are joint finite-population document-and-seed bootstrap 97.5% intervals. In both the primary analysis and the held-out confirmation analysis, GF-CSR minus Uniform is positive for both datasets. GF-CSR minus PO-CSR is positive for CNN/DailyMail, but the lower error bar for the XSum dataset is slightly below zero. Thus, the held-out analysis reproduces the primary comparison pattern.

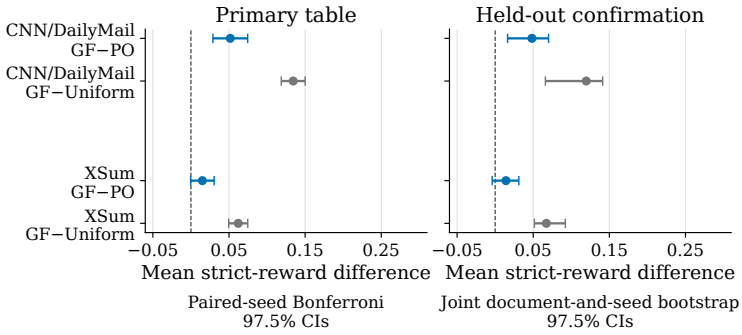


Figure 2: Paired strict-reward comparisons on the primary tables (left) and held-out confirmation tables (right). The dashed line marks zero.

8 Conclusion

In this paper, we formulated security-constrained prompt optimization as a fixed-budget best-feasible-prompt identification problem with partial observation. The security metric was modeled as a non-compensatory feasibility constraint, and clean and attacked evaluations reveal disjoint components of the objective vector. To address the partial observation structure, we developed a new GF-CSR algorithm, which is a reward-independent and two-phase successive-rejection method. Based on the elimination schedule and the deterministic clean and security sample counts of GF-CSR, we theoretically established a finite-sample, high-probability identification guarantee for GF-CSR under explicit separation conditions. Extensive experimental results show that GF-CSR achieves the highest strict-reward point estimates on both the CNN/DailyMail and XSum datasets.

Reproducibility Statement

We provide the experimental settings and implementation details in Section 7 and Appendix C. The core algorithmic part of our method is described in Section 5, with scheduling and baseline details in Appendix A. The assumptions and proofs for the theoretical results presented in this paper are provided in Section 6 and Appendix B. The source code will be published on GitHub upon acceptance of this paper. The release will include documented hyperparameters, training settings, and evaluation protocols, as well as scripts for regenerating the reported figures and tables.

Ethics Statement

Research on indirect prompt injection is inherently dual-use. This work is strictly constrained to defensive prompt evaluation using locally hosted models, public summarization datasets, and benign, non-adaptive attack instructions listed in Appendix C.1. We did not probe deployed systems or access external tools, accounts, or private data. Our results do not establish that a prompt or application is secure. Prompt-level resistance is only one component of a layered defense and should be combined with broader system safeguards and appropriate human oversight.

AI Use Statement

ChatGPT 5.6 Sol was used to assist with polishing language and identifying potentially relevant literature from topics and papers supplied by the authors. OpenAI Codex was used to provide feedback on experimental methodology and assist with software implementation. The authors independently read and verified all cited sources and assessed their relevance, reviewed and revised all AI-assisted text, and tested and validated all code and experimental procedures. All technical claims, definitions, algorithms, theorems, and proofs were developed by the authors.

References

Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *Proceedings of the 23rd Conference on Learning Theory*, pp. 41–53, 2010.

Jie Bian and Vincent Y. F. Tan. Asymptotically optimal linear best feasible arm identification with fixed budget. In *Proceedings of the Forty-First Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pp. 296–331, 2025.

Emmanuel Aboah Boateng, Zachary Johnson, Tian Xia, Sarah Zhang, Aidan Jay, Junyao Feng, Aditya Mate, and Ehi Nosakhare. Sapo: Secure automated prompt optimization via multi-agent collaboration. In *Proceedings of the ACM Conference on AI and Agentic Systems*, pp. 773–784, 2026.

Ting Cai and Kirthevasan Kandasamy. Constrained best arm identification with tests for feasibility. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19808–19816, 2026. doi: 10.1609/aaai.v40i24.39063.

Sizhe Chen, Julien Piet, Chawin Sitawarin, and David Wagner. StruQ: Defending against prompt injection with structured queries. In *34th USENIX Security Symposium*, pp. 2383–2400, 2025.

Fathima Zarin Faizal and Jayakrishnan Nair. Constrained pure exploration multi-armed bandits with a fixed budget. *arXiv preprint arXiv:2211.14768*, 2022.

Aaron Grattafiori et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, 2023. doi: 10.1145/3605764.3623985.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Keegan Hines, Gary Lopez, Matthew Hall, Federico Zarfati, Yonatan Zunger, and Emre Kiciman. Defending against indirect prompt injection attacks with spotlighting. *arXiv preprint arXiv:2403.14720*, 2024.

Julian Katz-Samuels and Clay Scott. Feasible arm identification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2535–2543, 2018.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021. doi: 10.18653/v1/2021.emnlp-main.243.

Donghao Li, Chengshuai Shi, Weijuan Ou, Cong Shen, and Jing Yang. Efficient multi-objective prompt optimization via pure-exploration bandits. In *International Conference on Learning Representations*, 2026. arXiv:2605.14553.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 4582–4597, 2021. doi: 10.18653/v1/2021.acl-long.353.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, 2004.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023. doi: 10.1145/3560815.

Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. Formalizing and benchmarking prompt injection attacks and defenses. In *33rd USENIX Security Symposium*, pp. 1831–1847, 2024.

Raunak Mukherjee and Sharayu Moharir. Fixed-budget constrained best arm identification in grouped bandits. *arXiv preprint arXiv:2603.04007*, 2026.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, 2018. doi: 10.18653/v1/D18-1206.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7957–7968, 2023. doi: 10.18653/v1/2023.emnlp-main.494.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 4222–4235, 2020. doi: 10.18653/v1/2020.emnlp-main.346.

Ankita Sinha, Wendi Cui, Kamalika Das, and Jiaxin Zhang. Survival of the safest: Towards secure prompt optimization through interleaved multi-objective evolution. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1016–1027, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.76.

Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training LLMs to prioritize privileged instructions. *arXiv preprint arXiv:2404.13208*, 2024.

Zhenlin Wang, Andrew J. Wagenmaker, and Kevin Jamieson. Best arm identification with safety constraints. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 9114–9146, 2022.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pp. 12028–12068, 2024.

Jingwei Yi, Yueqi Xie, Bin Zhu, Emre Kiciman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. Benchmarking and defending against indirect prompt injection attacks on large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2025. doi: 10.1145/3690624.3709179.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10471–10506, 2024. doi: 10.18653/v1/2024.findings-acl.624.

Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. On prompt-driven safeguarding for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61593–61613, 2024.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *International Conference on Learning Representations*, 2023.