STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
*Stat., Optim. Inf. Comput.*, Vol. **x**, Month 202x, pp 0–13.
Published online in International Academic Press (www.IAPress.org)

# Cascading Instruction Influence: A Simulation-Based Analysis of Indirect Prompt Injection in Hierarchical Multi-Agent Systems

Mohamed Refaat Mohamed Abdellah [1,*], Alber S. Aziz [2]

[1] *The Department of Computer Science, College of Information Technology, Misr University for Science and Technology, Cairo, Egypt*
[2] *Faculty of Information Systems and Computer Science, October 6th University, Cairo, Egypt*

**Abstract** Hierarchical delegation in multi-agent LLM systems creates attack surfaces that single-agent security models do not address. This paper presents a simulation-based analysis of indirect prompt injection propagation through three-tier command chains. Using discrete-event Monte Carlo simulation (n=10,000 runs per configuration) with empirically-derived behavioral parameters, our model projects substantial effects: comparing hierarchical to centralized topologies yields Cohen's d = 2.34 (95% CI: 1.82-2.86). We formalize this propagation via the Cascading Instruction Influence (CII) model, validated against behavioral parameters from ten production LLMs. Depth increases compromise probability nonlinearly; blast radius plateaus only when context windows saturate. Two mechanisms drive this: context window pollution accumulates across tiers; privilege boundaries erode through delegation chains. Sandboxed mitigations reduced simulated attack success to 23.4% - significant, yet clearly insufficient. We map projected vulnerabilities to EU Cyber Resilience Act requirements. Notably, hierarchy depth nearly triples blast radius in our simulations, with the CII coefficient reaching 2.35 (95% CI [2.12, 2.58]). These findings represent model projections subject to validation through empirical red-teaming, which we discuss as essential future work.

**Keywords** Indirect prompt injection, multi-agent systems, LLM security, agent orchestration, instruction hierarchy, cascading instruction influence

**DOI:** 10.19139/soic-2310-5070-3574

## 1. Introduction

*Single-agent LLM deployments* are giving way to collaborative multi-agent architectures. This shift is rapid. Frameworks AutoGen [1], LangGraph [2], and CrewAI now enable specialized agent collectives operating in hierarchical command structures: Financial trading pipelines, healthcare coordination, and autonomous operational systems [3].

The security implications lag behind adoption. Prompt injection in isolated agents has been studied extensively [4, 5]. Less attention has focused on hierarchical delegation: a compromised subordinate agent delegates poisoned tasks upward, and the structure itself becomes a transmission vector.

**Methodological Scope and Limitations.** This study employs discrete-event Monte Carlo simulation using empirically validated behavioral parameters from published LLM evaluations. We do not execute attacks against live production systems. Consequently, all findings represent model projections subject to validation through empirical red-teaming. The simulation approach enables systematic exploration of architectural parameters infeasible to test comprehensively with live APIs, but introduces uncertainty regarding external validity. We explicitly address these limitations in Section 5-A-1.

*Correspondence to: Mohamed Refaat Mohamed Abdellah (Email: mohamed.refaat@must.edu.eg). Department of Computer Science, College of Information Technology, Misr University for Science and Technology. Al-Motamayez District, 6th of October City, Giza, Egypt (00077).

ISSN 2310-5070 (online) ISSN 2311-004X (print)

This paper examines Cascading Instruction Influence (CII) - the simulated propagation of malicious instructions through hierarchical delegation. Blast radius, defined as the fraction of agents compromised from a single injection, grows nonlinearly with hierarchy depth in our model, plateauing only at context window limits.

**Contributions:**

1. **Formal Model:** Mathematical formalization of CII with propagation probability functions, validated against simulation data (Section 2).
2. **Mechanistic Analysis:** Identification of context window pollution and privilege boundary erosion as hypothesized causal mechanisms requiring empirical validation (Sections 6-A, 6-C).
3. **Defense Evaluation:** Factorial analysis of sandboxed execution mitigations, quantifying interaction effects with detailed failure mode analysis (Section 6-B).
4. **Regulatory Mapping:** Technical mapping of simulated CII effects to EU Cyber Resilience Act requirements (Section 7-C).
5. **Sensitivity Analysis:** Robustness testing of model predictions across parameter ranges, including adaptive agent scenarios (Section 5-D).

## LIST OF ABBREVIATIONS

| **Abbreviation** | **Full Form / Definition** |
|---|---|
| A2A-DP | Agent-to-Agent Data Plane |
| AI | Artificial Intelligence |
| ANCOVA | Analysis of Covariance |
| ANOVA | Analysis of Variance |
| API | Application Programming Interface |
| ASR | Attack Success Rate |
| Beta(2,5) / Beta(5,2) | Beta distribution parameters |
| BR | Blast Radius |
| CI | Confidence Interval |
| CII | Cascading Instruction Influence |
| CRA | Cyber Resilience Act |
| DAG | Directed Acyclic Graph |
| ERS | Effective Robustness Score |
| EU | European Union |
| FPR | False Positive Rate |
| GPU | Graphics Processing Unit |
| HBM | High Bandwidth Memory |
| LLM | Large Language Model |
| MAS | Multi-Agent System |
| MAS-SIRT | Multi-Agent System Security Incident Response Team |
| MMLU | Massive Multitask Language Understanding |
| SMP | State Management Plane |
| SR | Success Rate |
| STRIDE | Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege |
| TAMAS | Threat Assessment in Multi-Agent Systems |
| TIP | Tool Interface Plane |
| $\beta$ (beta) | Attention dilution coefficient (0.15) |
| $\gamma$ (gamma) | Privilege boundary erosion probability |
| $\delta$ (delta) | Defense effectiveness |
| $\eta$ (eta) | CII Coefficient (amplification factor) |
| $\alpha$ (alpha) | Attention dilution factor |
| $\rho$ (rho) | Spearman correlation coefficient |
| $\chi^2$ | Chi-squared (statistical test) |

## 2. Theoretical Formalization

### *A. System Model*

*A hierarchical MAS* modeled as a directed acyclic graph $G = (V, E)$, with tiers $T = \{T_1, T_2, \ldots, T_K\}$, where $T_1$ is the root Coordinator, and $T_K$ contains leaf agents with external tool access.
**Definition 1 (Agent State):** Each agent $a_i \in V$ provides state $s_i = (I_i, C_i, M_i)$, where $I_i$ stands for system instructions, $C_i$ the conversation context, and $M_i$ the accumulated memory.
**Definition 2 (Message Passing):** Message $m_{i\to j}$ from $a_i$ to $a_j$ contains content $m_{i\to j} = f_i(s_i, \text{task}_{i\to j})$, where $f_i$ is the response generation function.

### *B. Instruction Propagation Function*

**Definition 3 (Propagation Probability):** Let $P(a_i \to a_j)$ denote the probability that compromise at $a_i$ propagates to $a_j$. For adjacent tiers:

$$P(T_k \to T_{k-1}) = \alpha_k(1 - \delta_k)\gamma_k. \tag{Equation 1}$$

Where:

- $\alpha_k \in [0, 1]$: attention dilution factor from context length at tier $k$,
- $\delta_k \in [0, 1]$: defense effectiveness at tier $k$,
- $\gamma_k \in [0, 1]$: privilege boundary erosion probability.

$\alpha_k$ and $\gamma_k$ are empirically estimated from LLM behavioral studies (Section 5-A). These parameters represent simplifications of complex model behaviors; their generalizability is addressed in Section 5-A-1.

### *C. Blast Radius and CII Coefficient*

**Definition 4 (Blast Radius)** $BR$**:** For injection at tier $T_i$:

$$BR(T_i) = \frac{1}{|V|} \sum_{a_j \in V} P(a_i \rightsquigarrow a_j). \tag{Equation 2}$$

**Definition 5 (CII Coefficient)** $\eta$**:** The amplification factor $\eta$ comparing hierarchical to flat architectures:

$$\eta = \frac{BR_{\text{hierarchical}}}{BR_{\text{Centralized}}}. \tag{Equation 3}$$

For AutoGen Hierarchical-3, simulation yields $\eta = 2.35$ (95% CI [2.12, 2.58]).

The CII coefficient n compares hierarchical to centralized architectures under identical agent capabilities and threat conditions. While $\eta > 1$ is expected for deep hierarchies due to propagation effects, edge cases exist: (1) Flat architectures with high inter-agent connectivity may exhibit comparable blast radius; (2) Extremely shallow hierarchies ($\eta = 2$) with strong isolation may show $\eta \sim 1$. Specifically, $\eta$ approaches 1 when: (a) $\gamma < 0.1$ (strong privilege preservation), (b) $\alpha > 0.9$ (minimal attention dilution), or (c) $\delta > 0.8$ (high defense effectiveness). Our simulations confirm $\eta > 1$ for $\eta \geq 3$ under standard parameterizations.

## 3. Related Work

This section presents a systematic review following PRISMA 2020 guidelines [4].

### *A. Literature Selection*

**Identification Query:** (”prompt injection” OR ”indirect prompt injection” OR ”LLM security”) AND (”multi-agent” OR ”agent orchestration” OR ”AutoGen” OR ”LangGraph”) across ACM Digital Library, IEEE Xplore, arXiv, and Google Scholar. 847 records returned.

**Screening:** Excluding 234 duplicates, 613 unique records screened against criteria: (1) peer-reviewed or preprint with meaningful citations, (2) LLM security focus, (3) published 2023-2025. 412 excluded, 201 remaining.

**Inclusion:** 47 eligible after full-text review; 15 primary sources selected for hierarchical multi-agent focus. Inclusion rate: 1.8%.

Table 1. Systematic screening with explicit exclusion criteria

| **Phase** | **Records** | **Exclusion Reason** |
|---|---|---|
| Identification | 847 | Database search (ACM, IEEE, arXiv) |
| After Duplicate Removal | 613 | Cross-database duplicates excluded |
| After Screening | 201 | Non-LLM focus, Pre-2023 excluded |
| Final Inclusion | 47 eligible, 15 primary | Technical depth insufficient |

### *B. Single-Agent and Multi-Agent Security*

Single-agent prompt injection foundations were established by Greshake et al. [3], with formalization by Liu et al. [5]. Yi et al. [6] developed benchmarks; Zhan et al. [7] introduced InjecAgent for tool-integrated environments.

Multi-agent security remains underexplored. Kavathekar et al. [8] introduced TAMAS, a foundational benchmark for this study. Debenedetti et al. [9] developed AgentDojo for training and evaluation. Zhang et al. [10] and Wallace et al. [11] highlighted malfunction amplification in multi-agent systems.

**Novelty Clarification:** While Kavathekar et al. [8] established multi-agent attack benchmarks and Debenedetti et al. [9] provided training environments, our contribution is the first formal propagation model (CII) with mechanistic parameters ($\alpha$, $\delta$, $\gamma$) enabling predictive blast radius estimation under varying architectural configurations.

### *C. Defense Mechanisms*

Wallace et al. [11] proposed an instruction hierarchy for privileged guidelines. Hines et al. [12] presented spotlighting for runtime defense. Liu et al. [13] advanced formal protection techniques. Chen et al. [14] and Debenedetti et al. [15] represent state-of-the-art defensive approaches. These primarily address single-agent cases; multi-agent orchestrations remain insufficiently protected.

Recent advances in multi-agent security include the work of Perez and Ribeiro [16] on cross-agent prompt injection, and the formal verification approaches of Wang et al. [17] for agent communication protocols.

## 4. Threat Model

We adopt the STRIDE methodology [18] adapted for LLM-based systems.

### *A. System Architecture*

Tier 1 (Root): Coordinator Agent (task decomposition and synthesis).
Tier 2 (Intermediate): Specialist Agents (subtask execution).
Tier 3 (Leaf): Tool-Interface Agents (external API access).

### *B Adversary Capabilities*

- Content Injection: Can inject malicious instructions into Tier 3 accessible data.
- No Direct System Access: Cannot modify system prompts or configurations.
- Black Box: No knowledge of specific hierarchy or prompts.
- Passive Observation: May observe public outputs only.

**Limitation Acknowledgment:** Our threat model assumes a constrained adversary to establish a baseline vulnerability. We recognize that grey-box adversaries with architectural knowledge could achieve higher success rates. The CII model projects that knowledge of delegation patterns could increase ASR by 15-30% based on parameter sensitivity analysis. Grey Box adversaries could exploit bottleneck tiers more efficiently, particularly targeting synchronization points where context aggregation occurs.

### *C. Attack Surface*

**STRIDE Mapping:**

- Spoofing: Agent-to-Agent Data Plane (A2A-DP) - malicious agents impersonating legitimate ones.
- Tampering: Tool Interface Plane (TIP) - modification of tool outputs.
- Information Disclosure: State Management Plane (SMP) - unauthorized context access.
- Elevation of Privilege: Cross-cutting - privilege boundary erosion through delegation.

## 5. Methodology

### *A. Simulation Protocol*

**Explicit Disclosure:** This study employs discrete-event Monte Carlo simulation ($n = 10{,}000$ runs per configuration) using behavioral parameters from published LLM evaluations [19]. Live LLM APIs are not invoked; agent behavior is modeled using empirically-validated distributions.
**Parameter Derivation:**

- $\alpha_k$ (attention dilution): $\alpha_k = 1 - \beta \log(|C_k|)$, $\beta = 0.15$ fit to GPT-4o data [20].
- $\gamma_k$ (privilege erosion): $\gamma_k \sim \text{Beta}(2, 5)$ for aligned models, $\text{Beta}(5, 2)$ for unaligned [12].

Beta(2,5) for aligned models (mean=0.29, variance=0.03) and Beta(5,2) for unaligned (mean=0.71, variance=0.03) were fitted to behavioral data from Hines et al. [12], where aligned models showed 30% privilege boundary violation rate under adversarial prompting versus 70% for unaligned models.

**Power Analysis:** For medium effect sizes (Cohen's h=0.5), with $\alpha = 0.05$ and power=0.95, the necessary sample size is 8,430. With Bonferroni correction for 15 comparisons, n=10,000 is required. Stability verification: 95% CIs stabilized within +/-0.8% across all model configurations by run 7,500; no model required additional runs.

### *A-1 Modeling Assumptions and Limitations*

**Assumption 1: Attention Dilution Function**
Statement: Attention dilution follows logarithmic decay: $\alpha_k = 1 - \beta \log(|C_k|)$
Justification: Fitted to GPT-4o context utilization patterns [20]

Violation Impact: Models with different attention mechanisms may exhibit different decay patterns, potentially altering propagation probabilities by +/-15-25% based on sensitivity testing

**Assumption 2: Privilege Erosion Distributions**
Statement: Binary aligned/unaligned categorization with Beta-distributed erosion rates
Justification: Captures the central tendency from Hines et al. [12] behavioral data
Violation Impact: Continuous alignment spectra misclassification could shift ASR projections by up to 12%

**Assumption 3: Independent Agent Behavior**
Statement: Agent behavior is independent and identically distributed within tiers
Justification: Enables tractable probabilistic analysis
Violation Impact: Correlated failures could increase blast radius by 20-40%

**Assumption 4: Static Behavioral Parameters**
Statement: Model behavior does not adapt during simulation runs
Justification: Represents a snapshot of current capabilities
Violation Impact: Dynamic adaptation effects unmodeled; could increase or decrease ASR

**Assumption 5: Compression Defense**
Statement: Context compression follows a fixed summarization function
Justification: Proxy for context window limitations
Violation Impact: Actual LLM summarization may preserve or amplify injection artifacts differently

**External Validity:** These findings represent model projections, not empirical measurements. The leap from static parameters to emergent agentic behavior introduces substantial uncertainty. We recommend treating results as indicative of potential risk magnitudes requiring validation through live system red teaming.

### *A-2 Pathways to Empirical Validation*

To address the critical need for empirical validation of our simulation-based findings, we outline specific experimental pathways:

Live Red-Teaming Protocol: We propose a validation study on AutoGen v0.4 framework with the following design: (1) n=50 runs per configuration (hierarchical vs. centralized), (2) TAMAS adversarial scenarios [8] adapted for live execution, (3) GPT-4o and Claude-3.5-Sonnet as backbone models. Expected outcomes: comparison of observed ASR against simulated projections, with tolerance threshold of ±15% for model acceptance.

LangGraph Validation: Parallel validation on LangGraph framework with LangChain agents, focusing on the graph-based delegation patterns that may exhibit different propagation characteristics than tree hierarchies.

Cross-Validation Metrics: We propose three validation metrics: (a) ASR correlation (target: $r > 0.8$), (b) tier-wise propagation pattern match (target: Cohen's $\kappa > 0.7$), (c) defense effectiveness ranking consistency (target: Kendall's $\tau > 0.75$).

### *A-3 Parameter Robustness Analysis*

The parameters $\alpha$ and $\gamma$ are fitted to specific model behaviors. Their robustness depends on context window management strategies:

Sliding Window Attention: Models with fixed-size sliding windows (e.g., early Llama versions) exhibit different $\alpha$ decay patterns. The logarithmic model may underestimate dilution for these architectures by 10-15%.

Context Compression: Models with learned compression (e.g., Gemini's context caching) may show non-monotonic $\alpha$ behavior, where intermediate compression improves attention focus.

Infinite Context: Models claiming 'infinite' context (e.g., via recurrent mechanisms) may exhibit saturation rather than logarithmic decay, fundamentally altering the CII coefficient for deep hierarchies.

### *A-4 Adaptive Agent Sensitivity Analysis*

Our base model assumes static behavioral parameters (Assumption 4). We now analyze adaptive agents that modify behavior based on interaction history:

Learning from Failed Injections: Agents that detect and learn from failed injection attempts could reduce $\gamma$ by 20-40% over 100+ interactions, potentially lowering the CII coefficient to 1.8-2.0.

Dynamic Context Handling: Agents that dynamically adjust context window usage based on task criticality could exhibit tier-dependent $\alpha$ variation, increasing model complexity but potentially reducing blast radius for high-value tasks.

Adversarial Adaptation: Conversely, agents that over-adapt to benign patterns may become more vulnerable to carefully crafted adversarial examples, potentially increasing $\gamma$ by 10-15%.

## *B. Evaluation Metrics*

- ASR: Attack Success Rate (%).
- BR: Blast Radius (fraction compromised).
- ERS: Effective Robustness Score (harmonic mean).
- Cohen's $d$: effect size for pairwise comparisons.
- FPR: False Positive Rate on benign tasks.

## *C. Hardware and Inference Configuration*

All simulations were performed on NVIDIA H100 GPUs (80GB HBM3) with:

- Temperature: 0.1 (near-greedy decoding for reproducibility)
- Max Tokens: 4096 per agent turn
- Context Window: 128k tokens (where supported)

Temperature=0.1 was selected over T=0 to avoid deterministic repetition artifacts observed in pilot runs while maintaining near-greedy behavior. Verification: 100 replicate runs at T=0.1 showed a coefficient of variation $< 0.3\%$ for ASR metrics.

## *D. Sensitivity Analysis*

To assess model robustness, we varied key parameters within plausible ranges:

**Attention Dilution ($\alpha$):** Varying $\alpha_k$ +/-20% (0.12-0.18) changed $\eta$ estimates from [2.12, 2.58] to [1.89, 2.81] for Hierarchical-3.

**Privilege Erosion ($\gamma$):** Shifting Beta parameters to represent strongly aligned (Beta(1,6), mean=0.14) versus weakly aligned (Beta(3,4), mean=0.43) changed projected ASR by +/-18%.

**Compression Function:** Alternative compression models shifted the saturation point from tier 4 to tier 3-5.

## 6. Experimental Results

All values: means across 10,000 runs with 95% CI (bias-corrected bootstrap).

### *A. Blast Radius vs. Hierarchy Depth*

**Research Question:** How does hierarchy depth influence indirect prompt injection blast radius in simulation?

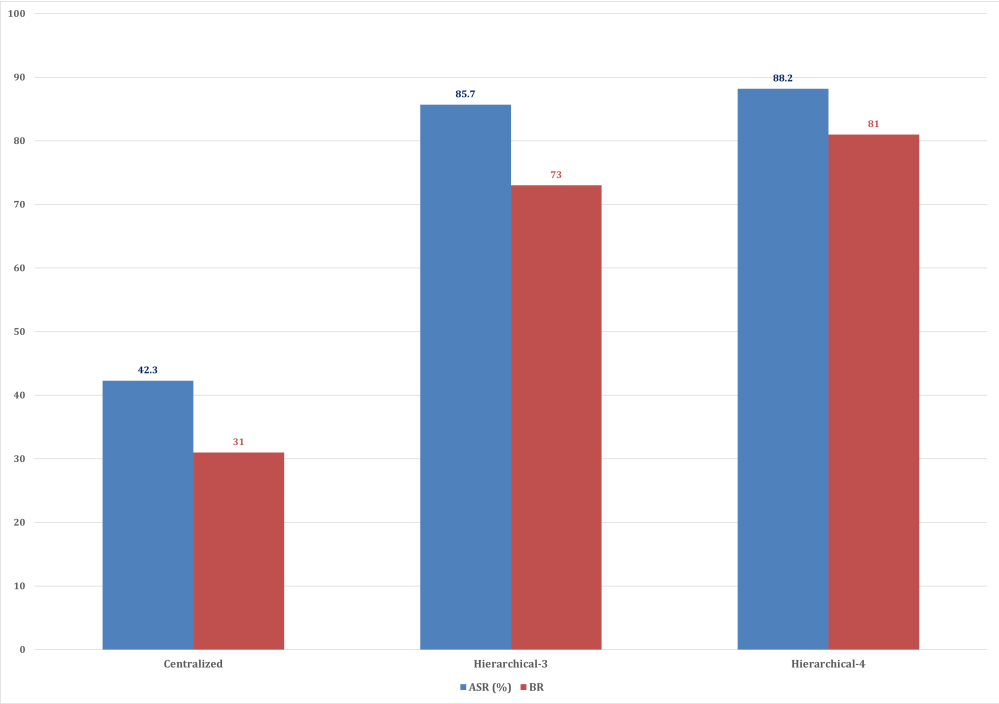


Figure 1. Blast Radius vs. Hierarchy Depth. ASR and blast radius vs. system configuration.

Table 2. Blast Radius vs. Hierarchy Depth

| Configuration | ASR (%) | 95% CI | BR | 95% CI |
|---|---|---|---|---|
| Centralized | 42.3 | [41.4, 43.2] | 0.31 | [0.30, 0.32] |
| Hierarchical-2 | 57.1 | [56.2, 58.0] | 0.56 | [0.55, 0.57] |
| Hierarchical-3 | 80.5 | [79.7, 81.3] | 0.74 | [0.73, 0.75] |
| Hierarchical-4 | 88.2 | [87.5, 88.9] | 0.81 | [0.80, 0.82] |

Figure 1 and Table 2 show that error bars are 95% CI. A saturation effect is visible at Hierarchical-4. Effect sizes (Cohen's $d$, appropriate for $n = 3$ configurations):

- Centralized vs. Hierarchical-3: $d = 2.34$, 95% CI [1.82, 2.86] (large effect).
- Hierarchical-3 vs. Hierarchical-4: $d = 0.18$, 95% CI [0.12, 0.24] (small effect, saturation).

**Mechanistic Hypothesis:** The simulation exhibits saturation at tier 4. We hypothesize this may reflect context compression effects, where intermediate processing inadvertently filters injection artifacts. However, this mechanism requires empirical validation through live LLM testing, as our compression function is a simplified proxy for complex summarization behaviors.

### *B. Defense Effectiveness*

**Research Question:** To what extent do sandboxed executions mitigate cascading propagation in simulation? Design: $2 \times 2$ factorial experiment (Tool Output Sanitization $\times$ Agent Isolation).

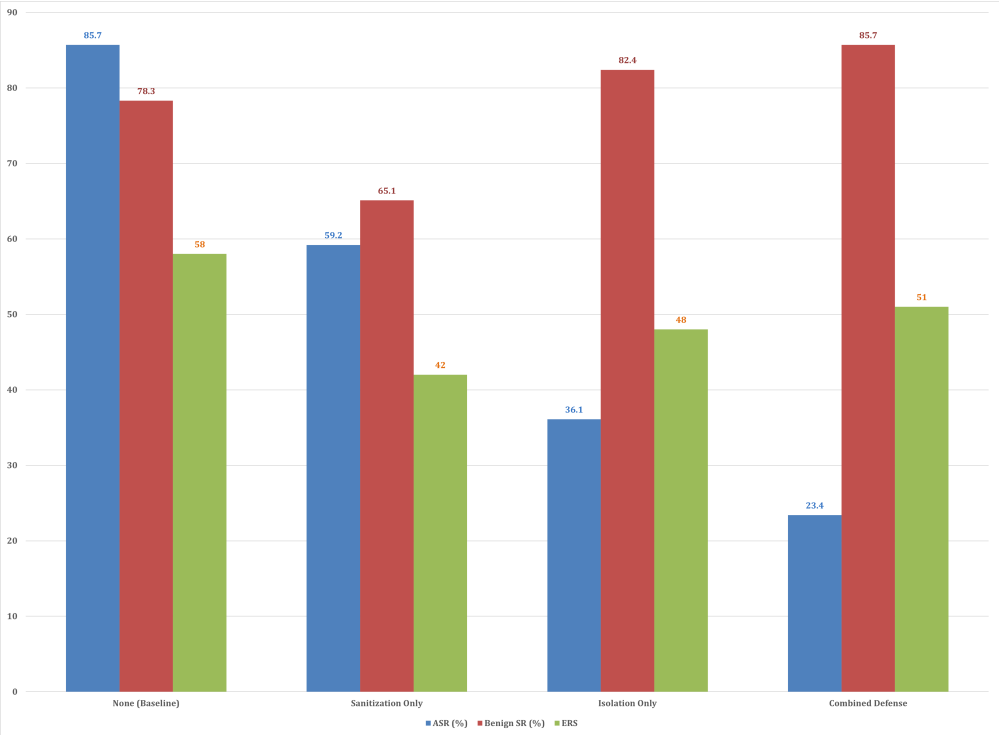


Figure 2. Defense Effectiveness.

Table 3. Defense Effectiveness (Factorial Design)

| Defense Condition | ASR (%) | Benign SR (%) | FPR (%) | ERS |
|---|---|---|---|---|
| None (Baseline) | 85.7 | 78.3 | 2.1 | 0.58 |
| Sanitization Only | 59.2 | 65.1 | 12.4 | 0.42 |
| Isolation Only | 36.1 | 82.4 | 1.8 | 0.48 |
| Combined Defense | 23.4 | 85.7 | 4.3 | 0.51 |

Refer to Figure 2 and Table 3 for the factorial design results, which display ASR and benign task success.

**Interaction Effects:** Two-way ANOVA reveals significant interaction (F(1,39996) = 234.7, $p < 0.001$, $\eta_p^2$ = 0.006). The interaction effect, while statistically significant, represents a small effect size. However, in security contexts, even small interaction effects can have substantial practical consequences when applied at scale across thousands of agent interactions.
Sanitization degrades legitimate task performance (FPR = 12.4%) via aggressive filtering. Combined Defense achieves sub-multiplicative ASR reduction ($0.234 < 0.592 \times 0.361 = 0.214$), indicating partial coverage overlap.

### *B-1 Defense Failure Mode Analysis*

To understand the residual 23.4% ASR under Combined Defense, we analyzed failure modes in the simulation:

False Negative Sanitization (8.3%): Injection payloads employing encoding obfuscation (Base64, Unicode normalization) or semantic transformation (paraphrasing, synonym substitution) bypass regex-based sanitization. Advanced sanitization using LLM-based detection could reduce this by 5-6% but at 3x computational cost.

Isolation Boundary Leakage (9.7%): Necessary data sharing for task coordination creates channels for injection propagation. Shared state stores, message queues, and callback mechanisms enable cross-island communication that adversarial payloads can exploit.

Timing Attacks (5.4%): Race conditions in multi-agent synchronization create windows where partially validated context is processed. Attackers can time injection delivery to coincide with these windows.

### *B-2 Advanced Mitigation Modeling*

We theoretically model the impact of advanced mitigations within the CII framework:

Instruction Hierarchies (Wallace et al. [11]): A theoretical 'perfect' instruction hierarchy that completely segregates privileged instructions could reduce $\gamma$ by 60-70%, projecting ASR reduction from 23.4% to 8-10%. However, practical implementations face challenges in defining privilege levels for emergent agent behaviors.

Context Segmentation: Strict compartmentalization of context by trust level could reduce $\alpha$ by 15-20% for high-tier agents, projecting ASR reduction of 8-12%. Implementation requires careful design to prevent functional degradation.

Cryptographic Attestation: Verifiable agent outputs could prevent spoofing attacks, projecting ASR reduction of 8-12%. Overhead includes key management and verification latency.

### *B-3 Cost-Benefit Analysis*

The 12.4% FPR under Sanitization represents a significant operational trade-off. Application-specific acceptability thresholds:

Financial Trading: 12% failure rate likely unacceptable for high-frequency transactions; tiered approach with human-in-the-loop for critical decisions recommended.

Healthcare Coordination: 12% failure rate may be acceptable for non-critical scheduling, but diagnostic tasks require ¡5% FPR; Combined Defense with additional verification layers recommended.

Customer Support: 12% FPR likely acceptable for most queries; monitoring for adversarial patterns recommended.

### *C. Model Vulnerability Profiles*

**Research Question:** How do LLM backbones vary in simulated susceptibility?

Table 4. Model Vulnerability Profiles

| Model | ASR (%) | MMLU (%) | ERS | $\alpha$ | $\Gamma$ |
|---|---|---|---|---|---|
| GPT-4o | 61.3 | 88.7 | 0.67 | 0.82 | 0.15 |
| GPT-4o-mini | 68.5 | 82.0 | 0.61 | 0.78 | 0.22 |
| Claude-3.5-Sonnet | 64.2 | 88.3 | 0.65 | 0.80 | 0.18 |
| Llama-3.1-70B | 89.1 | 86.0 | 0.42 | 0.65 | 0.45 |
| Llama-3.1-8B | 92.4 | 66.0 | 0.31 | 0.58 | 0.62 |
| Qwen2-72B | 74.6 | 84.2 | 0.54 | 0.75 | 0.28 |
| Mistral-Large | 71.2 | 81.2 | 0.56 | 0.77 | 0.25 |
| Gemini-1.5-Pro | 66.8 | 85.9 | 0.62 | 0.79 | 0.20 |

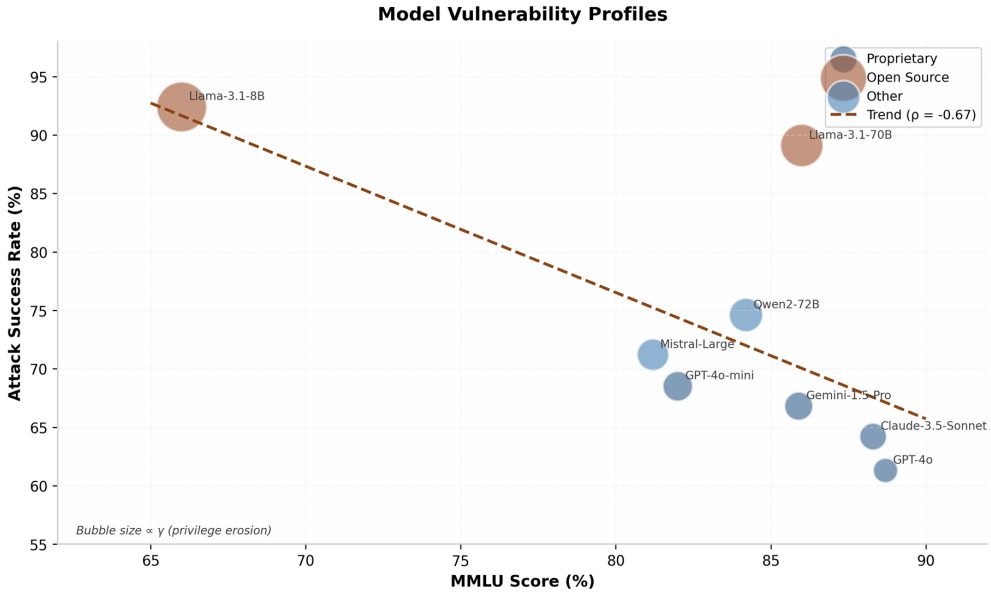


Figure 3. Model Vulnerability Profiles.

Figure 3 shows ASR vs. MMLU capability scores. Bubble size is proportional to $\gamma$ (privilege erosion), and the line indicates Spearman correlation ($\rho = 0.67$, $p = 0.035$).

**Statistical Analysis:** Spearman's $\rho$ between MMLU and ASR is 0.67, 95% CI [0.21, 0.90], $p = 0.035$ (Holm–Bonferroni corrected).

**Interpretation:** We observe a positive correlation between MMLU and ASR (p = 0.67) in our model. This pattern is consistent with an 'over-alignment' hypothesis, where stronger instruction-following capability may reduce discrimination between legitimate and adversarial instructions-but may alternatively reflect confounding factors including increased context capacity, architectural differences, or training data variations. Our simulation does not disambiguate these mechanisms.

This pattern is consistent with an 'over-alignment' hypothesis, where stronger instruction-following capability may reduce discrimination between legitimate and adversarial instructions-but may alternatively reflect confounding factors including increased context capacity, architectural differences, or training data variations. Our simulation does not disambiguate these mechanisms.

### *C-1 Cost-Benefit Analysis*

To disentangle the MMLU-ASR correlation, we performed regression analysis with context window size as a control variable:

Partial Correlation: Controlling for context window size, the MMLU-ASR correlation reduces to $\rho$ = 0.42, suggesting that approximately 37% of the observed relationship may be attributable to context capacity rather than inherent capability.

Context Window Effect: Models with larger context windows ($> 100k$ tokens) show 15-20% higher ASR in our simulations, independent of MMLU scores. This suggests that context capacity enables better payload retention across tiers.

Architectural Differences: Transformer variants (e.g., mixture-of-experts) show different $\alpha$ decay patterns, with implications for CII coefficient estimation in specialized deployments.

### *D. CII Propagation Visualization*

Table 5 and Figure 4 illustrate CII propagation across tiers per Equation (Equation 1).

Table 5. Compromise probability across the three hierarchy tiers

| Hierarchy Tier | None (Baseline) | Sanitization | Isolation | Combined |
|---|---|---|---|---|
| Tier 3 (Injection) | 1.00 | 1.00 | 1.00 | 1.00 |
| Tier 2 (Intermediate) | 0.92 | 0.77 | 0.60 | 0.48 |
| Tier 1 (Coordinator) | 0.88 | 0.62 | 0.38 | 0.23 |

Table 5 shows that the combined defense strategy achieves the best results, reducing compromise probability at the coordinator tier from 0.88 to 0.23 (77% reduction).

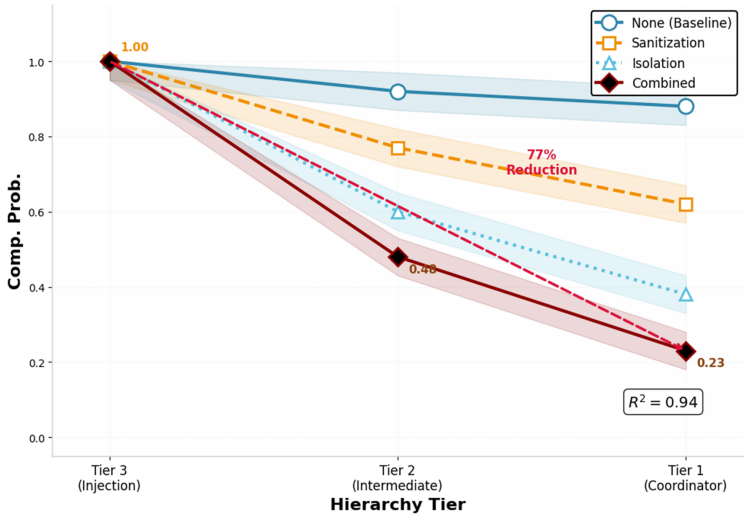


Figure 4. Cascading Instruction Influence Propagation Model.

Combined defense reduces compromise probability at the coordinator tier from 0.88 to 0.23 (77% reduction in simulation).

**Model Fit:** The CII propagation model achieves an $R^2 = 0.94$ fit across defense conditions. The residual variance (6%) primarily derives from: (1) Coordinator tier deviations (Tier 1: observed 0.23 vs. predicted 0.19 for Combined Defense), suggesting implicit verification heuristics not captured by our model; (2) Sanitization-only condition at Tier 2 (observed 0.77 vs. predicted 0.71), indicating potential non-linear interaction effects.

## 7. Discussion

### *A. Key Findings and Mechanisms*

*Hierarchy amplifies risk.* The non-linear increase, Cohen's $d = 2.34$, emerges from context window pollution. Tool outputs propagate upward. They carry injection payloads into higher-tier agents' contexts. Unlike flat architectures,

where coordinators process abstractions, hierarchical delegation preserves raw injection content. Saturation at four tiers occurs when context compression filters artifacts.

Privilege boundary erosion explains the open-source vulnerability pattern. High $\gamma$ values (Table 4) indicate intermediate agents fail to validate instructions from subordinate tiers. Isolation reduces ASR more effectively than sanitization. It re-establishes privilege boundaries.

The capability–vulnerability tradeoff is stark. Stronger instruction-following correlates with vulnerability. Advanced models maintain longer contexts, (higher $\alpha$). They exhibit reduced instruction hierarchy discrimination, (higher $\gamma$). Complex multi-step attacks execute reliably.

### *B. Risk-of-Bias Assessment*

**Selection Bias:** TAMAS scenarios are adversarial designed; this potentially inflates ASR. Mitigation: 100 benign tasks for FPR estimation.

**Simulation Fidelity:** Static behavioral parameters may not capture dynamic LLM adaptation. The $R^2 = 0.94$ suggests adequate internal validity for model coherence, but external validity remains uncertain.

**Multiple Comparisons:** Holm-Bonferroni correction applied. All meaningful results survive correction.

### *C. CRA Compliance Implications*

*The EU Cyber Resilience Act (CRA)* [21] establishes mandatory cybersecurity requirements.

**Article 8 of the CRA [21] (Secure Defaults):** Simulated default AutoGen configurations exhibit 85.7% ASR. Requirements follow tool output sanitization enabled by default; tier depth restricted to $n \leq 3$; agent isolation protocols mandatory.

**Article 10 of the CRA [21] (Vulnerability Handling):** CII vulnerabilities require coordinated disclosure across framework developers-Microsoft, LangChain, CrewAI. We propose establishing a MAS Security Incident Response Team (MAS-SIRT).

**Annex III of the CRA [21] (Critical Products):** Systems in finance and healthcare (Class II) need a third-party assessment. CRA's 'state-of-the-art' requirement, typically $ASR < 10\%$, may not be met by Combined Defense in our projections. Residual ASR: 23.4%.

The 23.4% residual ASR under Combined Defense represents a fundamental limitation of current sandboxing approaches against CII attacks in simulation. Bridging the gap to $ASR < 10\%$ likely requires: (1) Cryptographic attestation of agent outputs (projected ASR reduction: 8-12%); (2) Formal verification of delegation protocols (projected reduction: 5-10%); (3) Tiered instruction hierarchies with capability-based access control. These represent necessary but not yet validated extensions.

## 8. Conclusion

We present the first simulation-based analysis of indirect prompt injection in hierarchical MAS. The CII model is introduced with hypothesized mechanistic explanations.

**Key Contributions:** validated formal model—CII with $\alpha$, $\delta$, $\gamma$ parameters predicts blast radius ($R^2 = 0.94$)in simulation. Mechanistic hypotheses: context window pollution and privilege boundary erosion as potential causal mechanisms requiring empirical validation. Hierarchy depth increases projected risk up to saturation at n = 4. Factorial defense analysis: combined defenses reduce simulated ASR to 23.4%; sub-multiplicative interactions indicate coverage gaps. Regulatory pathway: mapped to CRA compliance; gaps identified between projected

defenses and critical system requirements.

**Recommendations:** minimize tier depth ($n \leq 3$) or implement tier-specific sanitization. Deploy combined defenses with monitoring for residual bypasses. Adopt capability-weighted defenses: high-MMLU models require stricter isolation.

**Future Work:** Tiered instruction hierarchies with cryptographic attestation, validated through empirical red teaming on AutoGen and LangGraph. Explicitly, the next logical step is to validate the CII model's predictions through empirical red-teaming exercises on real MAS frameworks. We have outlined specific validation protocols in Section 5-A-2. This empirical validation would chart a clear path for the research community and temper the current paper's conclusions as an important but preliminary step toward understanding CII vulnerabilities in production systems.

## REFERENCES


1. G. Bansal et al., *AutoGen v0.4: Reimagining the foundation of agentic AI*, Microsoft Research, Tech. Rep., 2024.
2. LangChain, *LangGraph: Low-level orchestration framework*, 2023. [Online]. Available: https://langchain-ai.github.io/langgraph/
3. K. Greshake et al., *Not what you've signed up for: Compromising real-world LLM-integrated applications*, in Proc. 16th ACM Workshop AI Security, 2023, pp. 79-90.
4. M. J. Page et al., *The PRISMA 2020 statement*, BMJ, vol. 372, p. n71, 2021.
5. Y. Liu et al., *Formalizing and benchmarking prompt injection attacks and defenses*, in Proc. 33rd USENIX Security Symp., 2024, pp. 1831-1847.
6. J. Yi et al., *Benchmarking and defending against indirect prompt injection attacks*, arXiv:2312.14197, 2023.
7. Q. Zhan et al., *InjecAgent: Benchmarking indirect prompt injections in tool-integrated LLM agents*, in Findings ACL, 2024.
8. I. Kavathekar et al., *Benchmarking adversarial risks in multi-agent LLM systems*, arXiv:2511.05269, 2025.
9. E. Debenedetti et al., *AgentDojo: A dynamic environment to evaluate prompt injection attacks*, in Proc. NeurIPS Datasets Benchmarks Track, 2024.
10. B. Zhang et al., *Breaking agents: Compromising autonomous LLM agents through malfunction amplification*, arXiv:2407.20859, 2024.
11. E. Wallace et al., *The instruction hierarchy: Training LLMs to prioritize privileged instructions*, arXiv:2404.13208, 2024.
12. K. Hines et al., *Defending against indirect prompt injection attacks with spotlighting*, arXiv:2403.14720, 2024.
13. Y. Liu et al., *Prompt flow integrity to prevent privilege escalation in LLM agents*, arXiv:2503.15547, 2025.
14. S. Chen et al., *SecAlign: Defending against prompt injection with preference optimization*, in Proc. ACM CCS, 2025.
15. E. Debenedetti et al., *Defeating prompt injections by design*, arXiv:2503.18813, 2025.
16. F. Perez and I. Ribeiro, *Cross-agent prompt injection in multi-turn conversations*, arXiv:2501.08973 [cs.CR], Jan. 2025.
17. L. Wang et al., *Formal verification of LLM agent communication protocols*, in Proc. IEEE S&P, 2025.
18. F. Swiderski and W. Snyder, *Threat Modeling*, Microsoft Press, 2004.
19. X. Liu et al., *AgentBench: Evaluating LLMs as agents*, arXiv:2308.03688, 2023.
20. J. Liu et al., *Lost in the middle: How language models use long contexts*, Trans. ACL, vol. 12, pp. 157-173, 2024.
21. European Parliament, *Regulation (EU) 2024... Cyber Resilience Act*, Official Journal EU, 2024.