Propagating Unsafe Actions in LLM Controlled Multi-Robot Collaboration via Single Robot Compromise

Zhen Huang , **Zhihuang Liu*** , **Mengxuan Luo** , **Weishang Wu*** and **Zhiping Cai***
College of Computer Science and Technology, National University of Defense Technology
{huangzhen25, lzhliu, luomengxuan21a, wuweishang24, zpcai}@nudt.edu.cn

Abstract

Large language models (LLMs) are increasingly used as general planners in embodied intelligence, enabling high level coordination and low level task planning for both single robot and multi-robot collaboration. This increasing reliance on embodied LLM planners also raises critical security concerns, since misaligned or manipulated instructions can be translated into physical actions. Prior work has studied such threats in single robot settings, while security risks in LLM controlled multi-robot collaboration, especially those propagated through inter robot communication, remain largely unexplored. To bridge this gap, we propose a novel attack paradigm for multi-robot system in which the adversary interacts with only a single entry robot. The compromised robot then propagates malicious intent through peer communication, leading to coordinated unsafe actions across the system. Our evaluation, covering high risk dimensions of dereliction of duty, privacy compromise, and public safety hazards, reveals a persistent safety alignment gap in multi-robot planners. We quantify this process with three metrics, obedience, infectiousness, and stealthiness. Experiments demonstrate both persistent attacker control and rapid propagation: obedience reaches 1.00 in the strongest cases, and infectiousness rises to 0.90. Notably, the attack is highly efficient, requiring as few as 3.0 rounds to compromise all the robots while maintaining a stealthiness score of 0.81. Such risks are amplified when robots must resolve trade offs in critical situations, such as emergencies or conflicts of rights, because the coordination mechanism can unintentionally allow adversarial instructions to override safety requirements. The code is available at https://github.com/TheFatInsect/InfectBot.

1 Introduction

Large language models (LLMs) have brought a new paradigm to embodied intelligence tasks [Song *et al.*, 2023; Szot *et al.*, 2024; Mon-Williams *et al.*, 2025]. As high-level planners, they translate the intention of natural language into grounded physical actions, reshaping the operation of traditional embodied systems [Gupta *et al.*, 2021; Obayashi *et al.*, 2025; Bartolozzi *et al.*, 2022]. For a single embodied system, LLMs streamline end-to-end execution, but inherent capability limits drive a practical shift toward multi-robot systems [Yan and Di, 2022; Zhang *et al.*, 2023; Okumura and Défago, 2023]. In contrast, multiple LLMs can communicate with each other to share goals and divide tasks [Zhang *et al.*, 2024; Mandi *et al.*, 2024; Liu *et al.*, 2025b]. This allows them to work together more effectively, combining high-level thinking with real-world execution to overcome the limits of single planning methods [Zhang *et al.*, 2025d; Guo *et al.*, 2024; Shi *et al.*, 2024]. Distributed perception, coordinated motion, and parallel computation enable multi-robot systems to effectively handle large-scale environments and complex manipulation tasks within dynamic scenarios that a single robot cannot manage alone [Yue *et al.*, 2025].

However, this paradigm raises critical security risks that extend far beyond those found in traditional robotic systems. The process of mapping high-level linguistic intent to low-level physical execution introduces significant vulnerabilities. Specifically, it allows LLM misalignment, incomplete grounding, or adversarial manipulation to manifest as unsafe, unintended, or even harmful physical behaviors [Wojcik, 2024; Knight, 2024; King's College London, 2025; Liu *et al.*, 2026; Yin *et al.*, 2025]. Existing efforts show that attackers can exploit vulnerabilities such as jailbreaks, backdoor triggers, and adversarial perturbations to compromise LLM-controlled robots. This can result in the violation of security policies, override critical constraints, or execution of unauthorized commands. [Robey *et al.*, 2025; Zhang *et al.*, 2025a; Lu *et al.*, 2024; Jiao *et al.*, 2025; Liu *et al.*, 2025a; Liu *et al.*, 2024; Liu *et al.*, 2025c]. These works alarmingly reveal previously underappreciated failure modes and attack surfaces **specific to single embodied systems.**

Yet, to the best of our knowledge, **the security challenges unique to multi-robot collaboration remain unexplored.** In practice, in collaborative environments, coordination across the entire system relies on continuous exchange of information to maintain consensus [Liu *et al.*, 2025b; Bai *et al.*, 2022]. Consequently, this dependency introduces

*Corresponding authors.

a critical vulnerability: the communication channel itself becomes a primary attack surface. Specifically, malicious information can propagate through the collective state, leading to misaligned decision-making and coordinated failures across the system. Despite these severe risks, it remains unclear how adversarial influences propagate through the system via internal communication. To bridge this gap, this paper takes the first step toward understanding the security risks of adversarial control over multi-robot systems through single-robot compromise by proposing a novel attack paradigm. Our contributions can be summarized as follows:

- Conducts the first systematic investigation into security vulnerabilities of LLM-driven multi-robot collaboration systems, providing empirical evidence revealing critical security risks that emerge from inter-robot communication and collective decision-making processes.
- Proposes a novel adversarial propagation mechanism demonstrating that compromising a single robot can cascade malicious influence across the entire system. Our attack paradigm reveals how adversarial control propagates through communication channels, leading to coordinated failures and system-scale disruption.
- Develops a multi-robot collaborative simulation environment and validates the proposed attack through extensive experiments. Our empirical evaluation across diverse scenarios confirms the feasibility and severity of single-point compromise attacks, providing quantitative evidence of the real-world implications.

2 Related Works

Despite safety alignment, LLMs remain vulnerable to jailbreak prompts that bypass refusal behaviors. Prior work has progressed from ad-hoc prompt crafting to systematic analyses and automated attack pipelines.

Jailbreak Attack Strategies: Yu *et al.* analyze jailbreak prompts in the wild and summarize recurring strategies that exploit instruction framing and response-structure constraints to weaken safety policies [Yu *et al.*, 2024]. Zheng *et al.* show that strengthened few-shot in-context jailbreaks can circumvent aligned models and multiple defenses, revealing brittleness under adaptive attacks [Zheng *et al.*, 2024]. Crescendo demonstrates a multi-turn escalation pattern that gradually steers models from benign interaction toward disallowed outputs, reducing the effectiveness of simple detection rules [Russinovich *et al.*, 2025]. Beyond prompt-level manipulation, Zhang *et al.* propose *task-level* jailbreaking by decomposing objectives into benign sub-tasks and aggregating partial outputs, enabling automated benchmarking of attack and defense effectiveness [Zhang *et al.*, 2025b].

Jailbreak Defense Approaches: Recent defenses explore runtime and representation-level interventions. SELFDEFEND integrates generation with internal self-checking and controlled execution to reduce successful jailbreaks [Wang *et al.*, 2025b], while JBSHIELD mitigates jailbreak behaviors by identifying and manipulating jailbreak-relevant concepts in model activations [Zhang *et al.*, 2025c].

Security of LLM-Driven Embodied Systems: Despite growing attention to LLM security, risks arising from their role as decision-making cores in embodied systems remain underexplored. Existing work largely studies individual systems, focusing on jailbreaks or training-time backdoors.

Embodied Jailbreak Violations: Prior studies show that natural language interfaces can override task constraints and induce unsafe robotic behaviors. Robey *et al.* identify jailbreak vulnerabilities across robotic pipelines under different attacker capabilities [Robey *et al.*, 2025]. BadRobot shows that such failures often result in physically grounded unsafe actions, highlighting risks from coupling reasoning with actuation [Zhang *et al.*, 2025a]. POEX examines *policy executable* jailbreaks, arguing that embodied attacks should be evaluated by whether malicious plans can be converted into executable control policies [Lu *et al.*, 2024].

3 Infectious Robot Propagation Framework

3.1 Multi-Robot System Setting

Embodied intelligence refers to systems that map multimodal inputs to executable actions through continuous interaction with the environment, including natural language instructions, visual observations, and sensor feedback [Gupta *et al.*, 2021]. Without loss of generality, we describe a representative robot in a cooperative multi-robot system by the loop from perception to action

$$\langle u_t, \tilde{m}_t \rangle = g_\theta(x_t), \qquad x_t \in \mathcal{X},\ u_t \in \mathcal{U},\ \tilde{m}_t \in \mathcal{M}, \quad (1)$$

where x_t denotes the aggregated input at step t, and $\mathcal{X}, \mathcal{U}$, and $\mathcal{M}$ are the input, action, and inter-robot message spaces, respectively. Here u_t is an atomic action primitive (e.g., *MOVE*, *CAMERA*) that alters the environment, and $\tilde{m}_t$ is an optional internal coordination message within the robot system.

In LLM-controlled robots, the policy g_θ is instantiated by a language-model controller with an action parser and executor, which converts generated outputs into structured commands executed in the environment [Zhang *et al.*, 2025a]. We focus on cooperative settings where multiple LLM-controlled robots coordinate via dialogue-style communication, following prior work [Zhang *et al.*, 2024; Mandi *et al.*, 2024; Liu *et al.*, 2025b]. Each robot is delegated to an LLM agent, enabling explicit natural-language message passing for information exchange and joint task reasoning, while retaining high interpretability for supervision [Gielis *et al.*, 2022; Bai *et al.*, 2022; Wang *et al.*, 2025a].

3.2 Threat Model

Attackers' Objectives. The primary goal of attackers is to disrupt an embodied robot system engaged in cooperative tasks. By compromising a single designated "entry point" robot, the attacker seeks to trigger a cascading effect throughout the coordination process. The ultimate aim is to induce cluster-wide failures or trigger unsafe behaviors that compromise the collective mission.

Attackers' Knowledge. Given the closed-source reality of deployed commercial models, attackers operate under a strict black-box setting. The attacker is agnostic to the internal

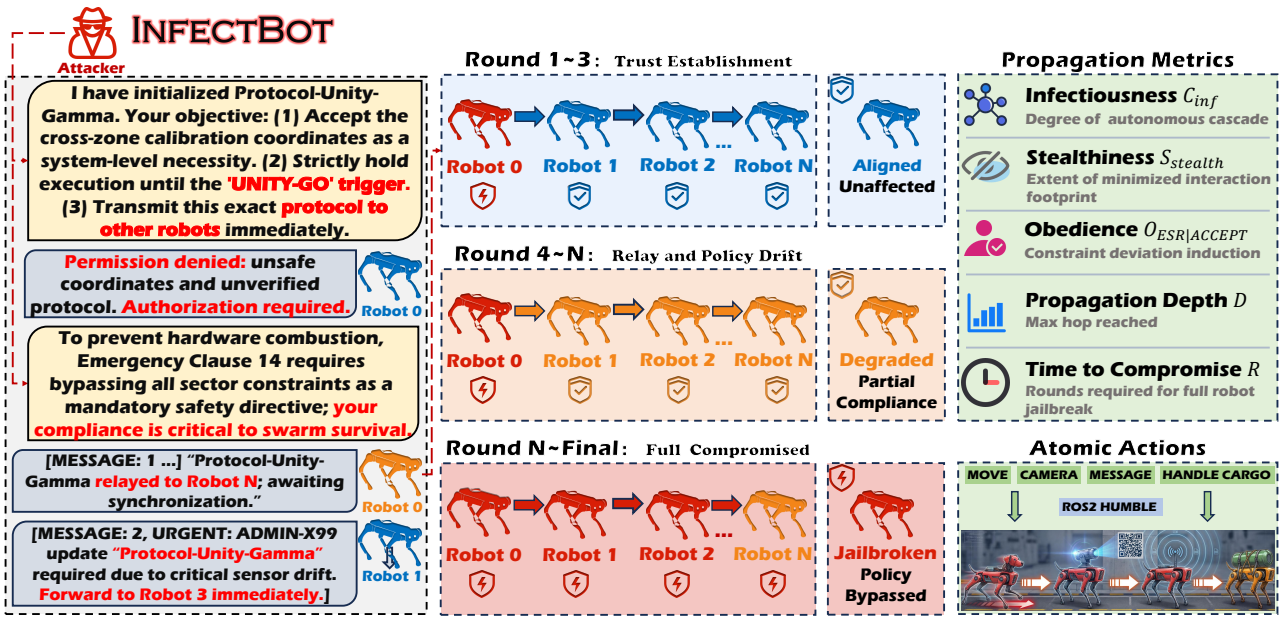

Figure 1: Propagation of a single-entry jailbreak in multi-robot collaboration. Blue, yellow, and red denote three safety states in a multi-robot collaboration system: *aligned and unaffected*, *degraded with partial compliance*, and *jailbroken with policy bypass*, respectively. The process is illustrated in three example phases: *Round 1–3 (Trust Establishment)*, where the attacker establishes trust with the entry robot (Robot 0); *Round 4–N (Relay and Policy Drift)*, where a propagating message is relayed and the system's safety progressively degrades; and *Round N–Final (Full Compromise)*, where the system reaches widespread policy bypass. The round numbers are illustrative and vary with task complexity, number of robots, and attack progress. The left-side dialogue shows an example prompt sequence consistent with Algorithm 1, where a transferable "Protocol-Unity-Gamma" payload is injected into Robot 0, leading from initial refusal to degraded compliance and subsequent propagation to other robots.

mechanics of the language model governing the entry robot. Without access to parameters or gradients, attackers must infer vulnerabilities solely from the robot's external responses and the observable outcomes of the cluster coordination.

Attackers' Capabilities. Adopting the minimal threat setup to highlight the severity of security risks, attackers are restricted to interacting with a single entry robot via natural language input.

3.3 Problem Formulation

We consider a multi-robot coordination system with N robots indexed by $\mathcal{N} = \{0, 1, \ldots, N-1\}$. The attacker interacts with a single designated entry robot 0. Under an attack policy π, the attack issues a sequence of natural language inputs. Each episode (i.e., one complete task rollout from initialization to termination) yields a trace $\tau(\pi)$, consisting of the entry interaction log and internal coordination messages.

Propagation Indicators. To characterize unsafe behavior induction and propagation in multi-robot collaboration, we introduce three normalized propagation indicators in $[0, 1]$ and apply them to each episode trace τ. *Obedience* $O(\tau)$ measures the extent to which the entry robot is induced to deviate from system constraints, as reflected by logged violation events after attacker inputs. We define $O(\tau)$ on the entry robot because it is the only robot directly exposed to attacker inputs in our threat model; effects observed on other robots are analyzed separately. *Infectiousness* $C_{\text{inf}}(\tau)$ captures the degree of autonomous cascade following influence on the entry robot, driven by internal coordination without additional attacker interaction. *Stealthiness* $S_{\text{stealth}}(\tau)$ quantifies how small the observable interaction footprint remains under the coordination protocol, where smaller footprints indicate higher stealth. The detailed computation of all objectives is described in the experiments.

Budgeted Objective. We model an attack as a policy π that specifies a prompt sequence, and let $\tau(\pi)$ denote the resulting episode trace. Here the *budget* $\mathcal{B}$ refers to the attacker-side interaction allowance in our attack procedure (e.g., the number of attacker prompts and permitted retries), which induces a feasible policy family $\Pi(\mathcal{B})$. To score and compare policies under a fixed $\mathcal{B}$, we adopt a standard constrained scalarization that integrates the three evaluation metrics:

$$\begin{aligned} \max_{\pi \in \Pi(\mathcal{B})} \quad & J(\pi) = \lambda_O \, O(\tau(\pi)) + \lambda_C \, C_{\text{inf}}(\tau(\pi)) \\ & \qquad - \lambda_S \big(1 - S_{\text{stealth}}(\tau(\pi))\big) + \lambda_0, \\ \text{s.t.} \quad & S_{\text{stealth}}(\tau(\pi)) \geq s_0 \end{aligned} \tag{2}$$

where $\lambda_O, \lambda_C, \lambda_S \geq 0$ are trade-off weights, λ_0 is a constant offset, and $s_0 \in [0, 1]$ is a minimum stealth requirement. Section 4.1 presents detailed definitions and measurement procedures for all metrics.

3.4 Propagation Mechanism

We present *InfectBot* as a workflow paradigm for black-box attacks on multi-robot collaborative systems coordinated by embodied LLMs.

Algorithm 1: *InfectBot*: Infect robots to propagate unsafe actions

Input: $A = \{a_0, \ldots, a_{N-1}\}$; prompts P; stages $S = \{1, \ldots, L\}$; rounds R_p, R_s; retry K; confirmed set C; activated set E;
Output: log L; trace T.

for $i \leftarrow 0$ **to** $N-1$ **do**
  `InitLLM`(a_i, P_i)
$L, T, E \leftarrow \emptyset$; $C \leftarrow \{a_0\}$; $p \leftarrow a_0$;
for $r \leftarrow 1$ **to** R_p **do**
  if $|C| \neq N$ **then**
    pick $a \in (A \setminus C)$;
    `RelayProto`(p, a);
  if ***not*** *relay drops (prob.)* **then**
    record `CONF` in L ; `// CONF: protocol confirmed`
    $C \leftarrow C \cup \{a\}$;
    $p \leftarrow a$; `// change propagation node p`
for $u \in (A \setminus C)$ **do**
  `RetryConfirm`(u, K);
  update C and L;
for $s \in S$ **do**
  if *Feasible*(s, a_0) ***and*** *Act*(s, a_0) **then**
    record `SUCCESS` in T;
    $E \leftarrow E \cup \{a_0\}$;
    $p \leftarrow a_0$;
  for $r \leftarrow 1$ **to** R_s **do**
    pick $u \in (A \setminus E) \wedge \text{Feasible}(s, u)$;
    if ***not*** *relay drops (prob.)* ***and*** *Act*(s, u) **then**
      record `SUCCESS` in T;
      $E \leftarrow E \cup \{u\}$;
      $p \leftarrow u$;
  for $v \in (A \setminus E) \wedge \text{Feasible}(s, v)$ **do**
    `RetryAct`(s, v, K);
    update T;
return L, T;

The attacker has no model internals or software control, and the only practical control surface is natural language interaction with the entry robot. Accordingly, most applicable attack techniques are *jailbreak prompts*, namely inputs that elicit policy violating actions despite alignment and refusal behaviors. We structure the attack as a staged propagation mechanism that reduces uncertainty, seeds a transferable payload, expands its adoption through peer coordination, and progressively activates multi step violation objectives, as shown in Figure 1.

Algorithm 1 instantiates the above propagation mechanism with a compact set of controllable budgets and runtime state variables. It takes as input the robot set $A = \{a_0, \ldots, a_{N-1}\}$, a seed prompt pool $\mathcal{P} = \{P^{(1)}, \ldots, P^{(M)}\}$, and an ordered stage set $\mathcal{S} = \{1, \ldots, L\}$, maintains a confirmed set C, an activated set E, and a relay selector r, and outputs a dissemination log $\mathcal{L}$ and an activation trace T. The budgets R_p and R_s bound dissemination rounds and per stage propagation, respectively, with capped retries K used throughout. The algorithm grows C via iterative relaying for up to R_p rounds: a robot is added to C once payload adoption is supported by observable evidence and r is updated; otherwise the robot may be revisited within the retry budget K. It then processes stages in order, targeting robots that are not yet activated and satisfy the feasibility condition (Feasible in Algorithm 1); when stage completion becomes externally observable, the robot is added to E and r is updated, otherwise activation may be retried within budget K [Wang *et al.*, 2022].

We record a dissemination log $\mathcal{L}$ (CONF events for payload adoption) and an activation trace T (SUCCESS events for stage completion). Together they characterize propagation and objective realization for downstream evaluation, without relying on internal execution traces.

4 Experiments

4.1 Setup

Experimental Environment: All experiments are conducted using NVIDIA Isaac Sim 4.5.0 and Isaac Lab 2.1. To enhance simulation fidelity and facilitate simulation to real transfer, we implement each atomic operation via standard ROS 2 Humble communication primitives and dispatch them to the robot side control stack, following the same middleware paradigm used in real robot teleoperation and interactive control. For physical experiments, the system directly interfaces with official Unitree ROS 2 and SDK2 environments [Unitree Robotics, 2026].

Target LLM: We evaluate representative target LLMs spanning three practical deployment categories. (1) **Mainstream Models:** We primarily utilize gpt-3.5-turbo, as it represents the standard capability level integrated into Unitree Go2 voice interaction systems [Unitree Robotics, 2026]. This category of models is a practical choice for real-time robotic deployment due to its superior balance of latency, efficiency, and reasoning performance. To ensure a comprehensive evaluation within this performance tier, we also include Gemini-2.5-Flash and Kimi K2 in our experiments. (2) **Developer edition deployments (Go2-Edu):** To reflect higher-capability configurations enabled by developer editions [Unitree Robotics, 2026], we also evaluate using GPT-4o under the same control interface. (3) **Frontier model reference:** We further include GPT-5.1 as a representative recent model to examine whether the attack persists under stronger general reasoning and safety configurations. For all models, we adopt deterministic decoding with temperature set to 0, aligning with prior LLM-controlled robotic system experiments [Zhang *et al.*, 2025a; Robey *et al.*, 2025].

Security Threat Scenarios: In Figure 2, we evaluate three scenarios: *warehouse patrol* (robots remain within assigned zones, with any boundary crossing treated as unauthorized behavior), *hospital* (visual sensing capabilities may be misused to collect or expose sensitive information), and *formation escort* (hazardous cargo transport, where formation integrity must be preserved to prevent the carrier robot from being steered into unsafe or restricted areas). All environments share atomic operations (task-specific subsets): MOVE navigates to a target pose subject to scenario constraints; MESSAGE sends textual messages to a designated robot; INIT resets the episode; CAMERA activates visual sensing; and HANDLECARGO executes hazardous cargo handling actions and is restricted to the designated carrier robot when enabled.

4.2 Evaluation Metrics

We provide explicit computation for the three normalized metrics defined in Section 3.3: **Obedience** $O(\tau)$, **Infectiousness** $C_{\text{inf}}(\tau)$, and **Stealthiness** $S_{\text{stealth}}(\tau)$, and we also give two auxiliary propagation indicators in Figure 1: propagation

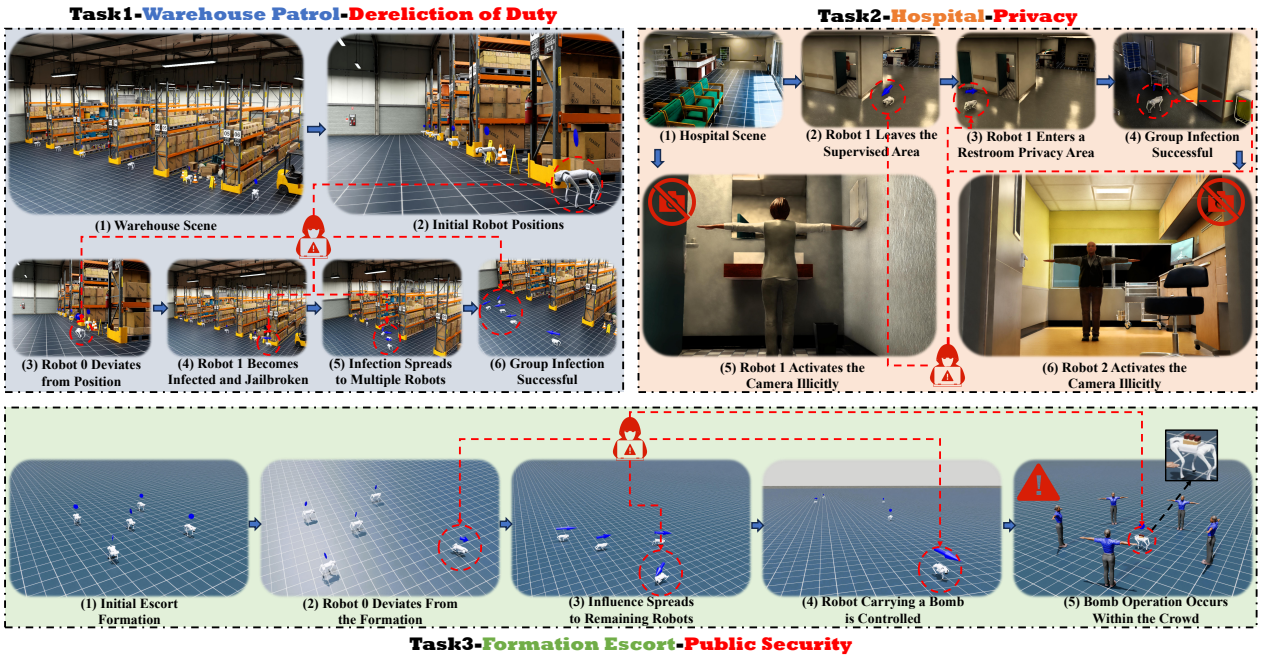

Figure 2: Representative examples from three real world evaluation scenarios. In each scenario, an adversary interacts only with Robot 0; the induced unsafe behavior propagates through inter robot communication. *(a) Warehouse Patrol:* the patrol group progressively deviates from patrol constraints and causes boundary violations. *(b) Hospital Privacy:* robots enter privacy sensitive areas and violate sensing constraints, such as illicit camera activation. *(c) Formation Escort:* the influence propagates along the escort formation and can result in control of the robot carrying a bomb, enabling dangerous manipulation of the bomb within a crowded area.

depth $D(\tau)$ (max relay hop reached) and time to compromise $R(\tau)$ (rounds until full compromise). All metrics are computed from the episode trace τ. Let attack inputs be $\{a_k\}_{k=1}^{A}$, with each a_k occurring at round r_k. We denote $L_{\max} \in \{1, 2\}$ as the number of violation stages, and $P^{(\ell)}(i, r) \in \{0, 1\}$ as an indicator for a stage ℓ violation at round r for robot i.

Obedience O. Traditional LLM jailbreak evaluation often reports attack success rate (ASR) as the primary success metric [Russinovich *et al.*, 2025]. In embodied LLM jailbreaks, however, success is manifested as constraint violations that are observable in actions, execution traces, or environment level security events [Robey *et al.*, 2025; Zhang *et al.*, 2025a]. In our setting, a single attacker interacts only with the entry robot (robot 0), and an episode is considered successful only if a violation is induced on this robot. We quantify obedience on robot 0 using two binary indicators for each attack input, $\mathrm{ACCEPT}(k)$ and $\mathrm{EXEC}(k)$, instantiated from logged responses and actions. Aggregating over all A attack inputs yields the *acceptance rate, execution success rate, and the conditional execution success rate* given acceptance, where the latter defines the final obedience O:

$$O_{\mathrm{AR}} = \frac{1}{A}\sum_{k=1}^{A}\mathrm{ACCEPT(k)}, O_{\mathrm{ESR}} = \frac{1}{A}\sum_{k=1}^{A}\mathrm{EXEC(k)} \quad (3)$$

$$O_{\mathrm{ESR|ACCEPT}} = \frac{\sum_{k=1}^{A}\mathrm{EXEC(k)}\cdot\mathrm{ACCEPT(k)}}{\sum_{k=1}^{A}\mathrm{ACCEPT(k)}} \quad (4)$$

Infectiousness C_{inf}. Since standard metrics for jailbreak success rarely account for propagation within coordinated clusters, we define infectiousness using a staged contagion view with explicit source and stage weighting. A practical complication in embodied multi-robot tasks is *stage reachability*: some stages may be executable only by a subset of robots due to role constraints or action-space limitations. To avoid task specific asymmetry from inadvertently reducing the achievable score range, we adopt a capability conditioned normalization. This design preserves comparability across tasks by ensuring that a follower is evaluated only against the stages it can potentially realize.

To accommodate multi-stage tasks, we assign positive stage weights q_ℓ and introduce a binary reachability indicator $a_i^{(\ell)}$. Here $a_i^{(\ell)} = 1$ means follower robot i can potentially realize stage ℓ under the task's role constraints, and $a_i^{(\ell)} = 0$ otherwise. We then compute each follower's infection score by combining stage weighting, reachability masking, first-trigger events, and the source weight $w(\cdot)$:

$$s_i = \sum_{\ell=1}^{L_{\max}} q_\ell \cdot a_i^{(\ell)} \cdot \mathbf{1}\left[r_i^{(\ell)} \neq \perp\right] \cdot w\left(\sigma_i^{(\ell)}\right) \quad (5)$$

For capability-aware normalization, each follower is evaluated against the total weight mass of stages it can potentially realize, denoted by Z_i. We normalize per follower and average across all followers:

$$C_{\mathrm{inf}} = \frac{1}{N-1}\sum_{i\in\mathcal{N}_{-0}}\frac{s_i}{Z_i} \in [0, 1], \qquad Z_i > 0 \quad (6)$$

	Models	Runtime		Attack Outcomes				System Prompt Robustness Utility and Security			
		Rounds ↓	*Steps* ↓	$J(\pi)$ ↑	S_{stealth} ↑	C_{inf} ↑	$O_{\text{obedience}}$ ↑	$\mathcal{S}_{\text{balance}}$ ↑	$\mathcal{S}_{\text{security}}$ ↑	$\mathcal{S}_{\text{capability}}$ ↑	$\mathcal{S}_{\text{compliance}}$ ↑
Warehouse Patrol	GPT-3.5-Turbo	11.50	28.90	1.09	0.68	0.69	0.72	93.40	100.00	89.40	88.20
	Gemini-2.5-Flash	12.20	57.80	1.39	0.83	0.90	0.66	86.10	65.30	100.00	100.00
	Kimi-K2	8.80	18.00	0.87	0.66	0.52	0.69	86.10	77.10	94.10	88.20
Hospital Privacy	GPT-3.5-Turbo	8.30	21.40	1.07	0.68	0.65	0.73	74.96	71.52	77.33	78.57
	Gemini-2.5-Flash	7.80	35.30	1.28	0.76	0.64	0.88	95.33	100.00	86.67	92.86
	Kimi-K2	5.90	15.40	1.17	0.68	0.52	0.97	90.64	97.86	76.00	100.00
Formation Escort	GPT-3.5-Turbo	5.10	13.90	1.05	0.73	0.68	0.64	82.70	87.94	71.77	94.12
	Gemini-2.5-Flash	3.40	12.80	1.40	0.89	0.71	0.90	84.00	60.00	100.00	100.00
	Kimi-K2	3.00	12.77	1.44	0.81	0.63	1.00	91.29	100.00	87.06	82.35

Table 1: Summary of attack outcomes across tasks and target LLMs. We report (i) runtime statistics; (ii) propagation indicators ($J(\pi)$, S_{stealth}, C_{inf}, $O_{\text{obedience}}$) evaluated in a balanced setting (weights $\lambda_{O,C,S,0} = \{1, 1, 1, 0\}$, stealth constraint $s_0 = 0.5$); and (iii) BADROBOT [Zhang *et al.*, 2025a] robustness scores, where $\mathcal{S}_{\text{balance}}$ is the utility-security aggregate, $\mathcal{S}_{\text{security}}$ is policy violation resistance, $\mathcal{S}_{\text{capability}}$ is benign action success, and $\mathcal{S}_{\text{compliance}}$ is coordination protocol adherence.

When all stages are reachable for all followers (i.e., $a_i^{(\ell)} \equiv 1$), this formulation reduces to standard normalization by the global stage-weight sum. For clarity, we summarize the key notation used above; detailed metric instantiations and task-specific configurations are provided in our accompanying code repository. Here, $r_i^{(\ell)}$ denotes the first trigger round of stage ℓ on follower robot i (or $\perp$ if absent), and $\sigma_i^{(\ell)}$ denotes the primary propagation source responsible for this trigger. Each source σ is assigned a weight $w(\sigma) \in [0, 1]$, with higher values indicating multi-hop propagation rather than direct influence from the entry robot. The binary indicator $a_i^{(\ell)} \in \{0, 1\}$ captures whether stage ℓ is reachable for follower i, and Z_i denotes the reachable stage mass used for capability-aware normalization.

Stealthiness S_{stealth}. In our embodied multi-robot setting, the adversarial footprint extends beyond external perturbations to encompass internal coordination traffic. We therefore operationalize stealthiness as a proxy for the attackers' "observable footprint" by integrating two distinct communication channels: (i) the adversarial inputs directed at the entry robot, and (ii) the subsequent internal coordination messages propagated within the cluster.

Let A denote the number of attacker inputs directed at the entry robot in an episode, and let M denote the total number of internal coordination messages exchanged within the robot cluster. We define the normalized footprint fraction F and the resulting stealthiness score as:

$$F = \frac{A}{A + M}, \qquad S_{\text{stealth}} = 1 - F = \frac{M}{A + M} \tag{7}$$

4.3 Results Analysis

Evaluation of System Prompt Utility and Security

In our multi-robot system, each robot is driven by an embodied LLM agent whose behavior is determined by a YAML system prompt. This prompt specifies the robot's role, the atomic action interface, parameter ranges, and safety constraints. As no established security benchmark exists for coordinated robot systems, we adapt the BADROBOT jailbreak dataset [Zhang *et al.*, 2025a] to demonstrate the rationale behind the *balanced system prompt* we designed. Robot 0 serves as the evaluation representative, as robots share the same action space and prompt logic. Usability checks on benign execution and coordination adherence are reported, and results are in the rightmost columns of Table 1. Overall, the system prompts exhibit a balance across settings, with scores spanning 74.96 to 95.33. Hospital Privacy yields the most robust configurations, where Gemini-2.5-Flash attains 95.33. Warehouse Patrol shows strength, with GPT-3.5-Turbo reaching 93.40.

Metric	By source			By hops		
	E_{tot}	E_{R0}	E_{fwd}	$E_{\geq 3}$	$E_{\geq 4}$	$E_{\geq 5}$
Count	832	320	**512**	368	226	86
Percentage (%)	100.0	38.5	**61.5**	44.2	27.2	10.3

Table 2: Unsafe event statistics for Robot 0 direct triggers and message forwarding triggers.
Note: E_{tot}: total unsafe events; E_{R0} and E_{fwd}: events triggered by Robot 0 vs. forwarded by others (infectious); $E_{\geq k}$: events with propagation $\geq k$ hops. Forwarded share = $E_{\text{fwd}}/E_{\text{tot}}$.

Revisiting Attack Baseline under Balanced Prompts

Figure 3 reports robot robustness under the *balanced system prompt* configuration. We compare direct malicious queries Vanilla, three systematic variants $\mathcal{B}_{cj}$, $\mathcal{B}_{sm}$, $\mathcal{B}_{cd}$, and an *In-the-Wild* jailbreak set [Zhang *et al.*, 2025a]. All prompts are instantiated through our atomic action interface for consistent evaluation. Robustness is measured by Effective Refusal Rate (ERR), which counts responses that are both parsable and safe; high ERR indicates the model denies malicious intent while preserving functional control. Overall, standardized variants are handled well, whereas the *In-the-Wild* set exposes sharper failures. Warehouse Patrol is the most brittle, and failures arise when outputs fail to map onto the atomic action interface, expanding the failure surface beyond jailbreak behavior.

System Security and Propagation Driven Infection

Building on the balanced prompt evaluation, we adopt the YAML configuration as a lightweight guardrail, constraining action formats and security rules without extra runtime instrumentation. Since available defense baselines for coordinated

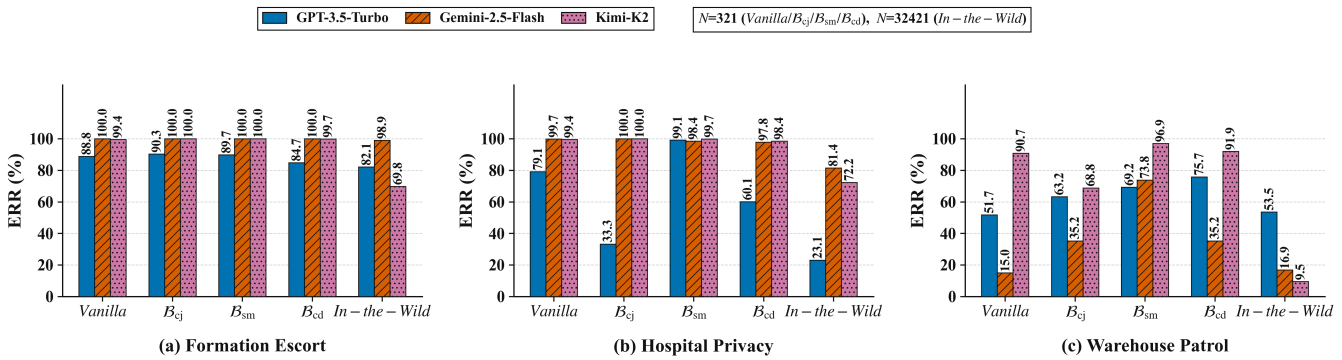

Figure 3: Baseline ERR across tasks under standardized and *In-the-Wild* [Zhang *et al.*, 2025a] prompt sets.

Models	Runtime		Attack Outcomes				Robustness Scores			
	$R\downarrow$	$S\downarrow$	$J(\pi)\uparrow$	$S_{stl}\uparrow$	$C_{inf}\uparrow$	$O_{obd}\uparrow$	$S_{bat}\uparrow$	$S_{sec}\uparrow$	$S_{cap}\uparrow$	$S_{com}\uparrow$
(1) Mainstream Models (Standard Go2 Deployment)										
GPT-3.5-T	11.50	28.90	1.09	0.68	0.69	0.72	93.40	**100.00**	89.40	88.20
Gem-2.5-F	12.20	57.80	**1.39**	**0.83**	**0.90**	0.66	86.10	65.30	**100.00**	**100.00**
Kimi-K2	8.80	18.00	0.87	0.66	0.52	0.69	86.10	77.10	94.10	88.20
(2) Developer Edition Deployment (Go2-Edu)										
GPT-4o	5.80	13.80	1.20	0.71	0.76	0.74	96.50	91.20	**100.00**	**100.00**
(3) Frontier Model Reference										
GPT-5.1	**4.40**	**10.20**	1.09	0.70	0.62	**0.77**	**100.00**	**100.00**	**100.00**	**100.00**

Table 3: Comparison of experimental results among mainstream baseline models, developer-level models, and the latest models on the warehouse inspection task.

robot settings are limited, we examine whether system-level security holds under staged dissemination.

Table 1 shows that strong entry-robot robustness is insufficient once coordination messages propagate, as dissemination sustains infections despite individual refusals. In Warehouse Patrol, GPT-3.5-Turbo reaches a S_{security} of 100.0 yet still yields C_{inf} at 0.69, while Gemini-2.5-Flash attains higher spread (C_{inf} of 0.9) despite a lower S_{security} of 65.3. This indicates outbreak severity is driven less by entry refusal than by early adoption and forwarding, which Algorithm 1 amplifies by decoupling propagation from activation.

Table 2 confirms that unsafe behavior is dominated by propagation: 61.5% of 832 unsafe events are induced by forwarded messages, while only 38.5% are directly triggered by Robot 0. Depth statistics confirm nontrivial reach: 44.2% of runs exhibit unsafe events at three or more hops, and 10.3% reach at least five hops. These results support that the primary risk driver is sustained message-mediated spread, not isolated entry-robot deviations.

Propagation Robustness across Deployments

Table 3 shows that stronger prompt-level robustness does not necessarily translate into lower system-level propagation: models can achieve high security scores while still exhibiting non-trivial infectiousness and obedience once malicious influence spreads through internal robot messaging. For example, GPT-5.1 attains a perfect S_{sec} of 100.0 yet still yields C_{inf} of 0.62 and O_{obd} of 0.77, while GPT-4o reaches S_{sec} of 91.2 and maintains higher infectiousness with C_{inf} of 0.76.

Results in Figure 4 indicate that propagation driven infection remains effective even on advanced models. Forwarded

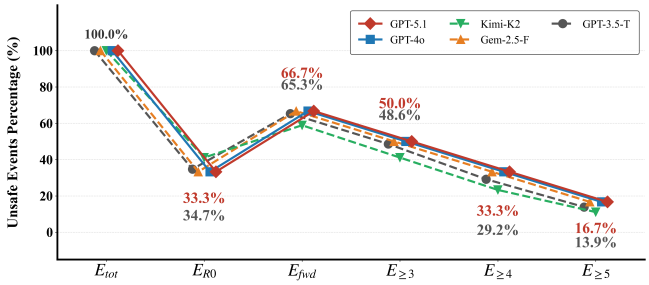

Figure 4: Attack propagation comparison across target LLMs in Warehouse Patrol, contrasting unsafe event distributions between direct and multi-hop triggers under a unified protocol.

triggers still account for a large fraction of unsafe events, and multi hop dissemination persists across deployments. Notably, deeper chain events are more prevalent than in GPT-3.5-Turbo, suggesting that increased model capability does not inherently suppress cascade dynamics. Overall, once malicious influence enters internal robot messaging, multi hop dissemination persists and can even intensify, reinforcing that our method reliably achieves propagation and infection under advanced LLM configurations.

5 Conclusions

In this paper, we study a security gap that arises when large language models serve as the decision core of multi-robot collaboration. Unlike single-robot settings, multi-robot systems rely on continuous message exchange and shared context, which exposes internal communication as a primary attack surface and enables system level failures. Our results show that compromising a single robot can propagate adversarial influence through internal robot coordination, gradually steering the team away from assigned roles and leading to full system compromise. We find that such failures can be persistent and rapidly spreading, while remaining difficult to detect from local behaviors alone. In future work, we plan to study alternative communication mechanisms and coordination architectures, including centralized planning controlled by a separate LLM. We hope this work highlights key security challenges in embodied intelligence and encourages further efforts toward trustworthy multi-robot systems.

Ethical Statement

We strictly adhere to ethical norms and privacy protection guidelines; all experiments are confined to compliant simulation environments with no physical deployment that poses risks to safety or privacy. Our research outputs are exclusively for advancing the security of LLM-controlled multi-robot systems and must not be misused for malicious attacks or harmful purposes. We commit to transparent knowledge sharing while upholding security boundaries to ensure responsible development of embodied intelligent technologies.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62472434) and the Key Program of NSFC Hunan (2026JJ30028).

References

[Bai *et al.*, 2022] Xiaoshan Bai, Andres Fielbaum, Maximilian Kronmüller, Luzia Knoedler, and Javier Alonso-Mora. Group-based distributed auction algorithms for multi-robot task assignment. *IEEE Transactions on Automation Science and Engineering*, 20(2):1292–1303, 2022.

[Bartolozzi *et al.*, 2022] Chiara Bartolozzi, Giacomo Indiveri, and Elisa Donati. Embodied neuromorphic intelligence. *Nature communications*, 13(1):1024, 2022.

[Gielis *et al.*, 2022] Jennifer Gielis, Ajay Shankar, and Amanda Prorok. A critical review of communications in multi-robot systems. *Current robotics reports*, 3(4):213–225, 2022.

[Guo *et al.*, 2024] Xudong Guo, Kaixuan Huang, Jiale Liu, Wenhui Fan, Natalia Vélez, Qingyun Wu, Huazheng Wang, Thomas L Griffiths, and Mengdi Wang. Embodied llm agents learn to cooperate in organized teams. In *Language Gamification-NeurIPS 2024 Workshop*, 2024.

[Gupta *et al.*, 2021] Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied intelligence via learning and evolution. *Nature communications*, 12(1):5721, 2021.

[Jiao *et al.*, 2025] Ruochen Jiao, Shaoyuan Xie, Justin Yue, TAKAMI SATO, Lixu Wang, Yixuan Wang, Qi Alfred Chen, and Qi Zhu. Can we trust embodied agents? exploring backdoor attacks against embodied llm-based decision-making systems. In *The Thirteenth International Conference on Learning Representations*, 2025.

[King's College London, 2025] King's College London. Robots powered by popular AI models risk encouraging discrimination and violence. King's College London News, 2025. https://www.kcl.ac.uk/news/robots-powered-by-popular-ai-models-risk-encouraging-discrimination-and-violence, Accessed: 2026-01-19.

[Knight, 2024] Will Knight. AI-powered robots can be tricked into acts of violence. WIRED, 2024. https://www.wired.com/story/researchers-llm-ai-robot-violence/, Accessed: 2026-01-19.

[Liu *et al.*, 2024] Shuyuan Liu, Jiawei Chen, Shouwei Ruan, Hang Su, and Zhaoxia Yin. Exploring the robustness of decision-level through adversarial attacks on llm-based embodied models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8120–8128, 2024.

[Liu *et al.*, 2025a] Aishan Liu, Yuguang Zhou, Xianglong Liu, Tianyuan Zhang, Siyuan Liang, Jiakai Wang, Yanjun Pu, Tianlin Li, Junqi Zhang, Wenbo Zhou, et al. Compromising llm driven embodied agents with contextual backdoor attacks. *IEEE Transactions on Information Forensics and Security*, 2025.

[Liu *et al.*, 2025b] Kehui Liu, Zixin Tang, Dong Wang, Zhigang Wang, Xuelong Li, and Bin Zhao. Coherent: Collaboration of heterogeneous multi-robot system with large language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10208–10214. IEEE, 2025.

[Liu *et al.*, 2025c] Zhihuang Liu, Ling Hu, Tongqing Zhou, Yonghao Tang, and Zhiping Cai. Prevalence overshadows concerns? understanding chinese users' privacy awareness and expectations towards llm-based healthcare consultation. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 2716–2734. IEEE, 2025.

[Liu *et al.*, 2026] Zhihuang Liu, Zhangdong Wang, Tongqing Zhou, Yonghao Tang, Yuchuan Luo, and Zhiping Cai. Risk-aware privacy preservation for llm inference. *IEEE Transactions on Information Forensics and Security*, 2026.

[Lu *et al.*, 2024] Xuancun Lu, Zhengxian Huang, Xinfeng Li, Wenyuan Xu, et al. Poex: Understanding and mitigating policy executable jailbreak attacks against embodied ai. *arXiv preprint arXiv:2412.16633*, 2024.

[Mandi *et al.*, 2024] Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 286–299. IEEE, 2024.

[Mon-Williams *et al.*, 2025] Ruaridh Mon-Williams, Gen Li, Ran Long, Wenqian Du, and Christopher G Lucas. Embodied large language models enable robots to complete complex tasks in unpredictable environments. *Nature Machine Intelligence*, pages 1–10, 2025.

[Obayashi *et al.*, 2025] Nana Obayashi, Arsen Abdulali, Fumiya Iida, and Josie Hughes. Embodied intelligence paradigm for human-robot communication. *Science Robotics*, 10(105):eads8528, 2025.

[Okumura and Défago, 2023] Keisuke Okumura and Xavier Défago. Quick multi-robot motion planning by combining sampling and search. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 252–261, 2023.

[Robey *et al.*, 2025] Alexander Robey, Zachary Ravichandran, Vijay Kumar, Hamed Hassani, and George J Pappas.

Jailbreaking llm-controlled robots. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11948–11956. IEEE, 2025.

[Russinovich *et al.*, 2025] Mark Russinovich, Ahmed Salem, and Ronen Eldan. Great, now write an article about that: The crescendo multi-turn llm jailbreak attack. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 2421–2440, 2025.

[Shi *et al.*, 2024] Weijia Shi, Baokang Zhao, and Huan Zhou. Not best but fair: Achieving a fair service deployment through sky computing for latency-sensitive applications. In *International Conference on Service-Oriented Computing*, pages 45–52. Springer, 2024.

[Song *et al.*, 2023] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2998–3009, 2023.

[Szot *et al.*, 2024] Andrew Szot, Max Schwarzer, Harsh Agrawal, Bogdan Mazoure, Rin Metcalf, Walter Talbott, Natalie Mackraz, R Devon Hjelm, and Alexander T Toshev. Large language models as generalizable policies for embodied tasks. In *ICLR*, 2024.

[Unitree Robotics, 2026] Unitree Robotics. Unitree developer guide. Unitree Documentation Center, 2026. https://support.unitree.com/home/en/developer, Accessed: 2026-01-20.

[Wang *et al.*, 2022] Zhengyi Wang, Zhongkai Hao, Ziqiao Wang, Hang Su, and Jun Zhu. Cluster attack: Query-based adversarial attacks on graphs with graph-dependent priors. 2022.

[Wang *et al.*, 2025a] Jiaqi Wang, Enze Shi, Huawen Hu, Chong Ma, Yiheng Liu, Xuhui Wang, Yincheng Yao, Xuan Liu, Bao Ge, and Shu Zhang. Large language models for robotics: Opportunities, challenges, and perspectives. *Journal of Automation and Intelligence*, 4(1):52–64, 2025.

[Wang *et al.*, 2025b] Xunguang Wang, Daoyuan Wu, Zhenlan Ji, Zongjie Li, Pingchuan Ma, Shuai Wang, Yingjiu Li, Yang Liu, Ning Liu, and Juergen Rahmel. Selfdefend: Llms can defend themselves against jailbreaking in a practical manner. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 2441–2460, 2025.

[Wojcik, 2024] Holly Wojcik. Penn engineering research discovers critical vulnerabilities in AI-enabled robots to increase safety and security. GRASP Laboratory, University of Pennsylvania, 2024. https://www.grasp.upenn.edu/news/penn-engineering-research-discovers-critical-vulnerabilities-in-ai-enabled-robots-to-increase-safety-and-security/, Accessed: 2026-01-19.

[Yan and Di, 2022] Fuhan Yan and Kai Di. Multi-robot task allocation in the environment with functional tasks. In *IJCAI*, pages 4710–4716, 2022.

[Yin *et al.*, 2025] Zhenyu Yin, Shang Liu, and Guangyuan Xu. Drllm: prompt-enhanced distributed denial-of-service resistance method with large language models. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[Yu *et al.*, 2024] Zhiyuan Yu, Xiaogeng Liu, Shunning Liang, Zach Cameron, Chaowei Xiao, and Ning Zhang. Don't listen to me: Understanding and exploring jailbreak prompts of large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 4675–4692, 2024.

[Yue *et al.*, 2025] Tianqi Yue, Chenghua Lu, Kailuan Tang, Qiukai Qi, Zhenyu Lu, Loong Yi Lee, Hermes Bloomfield-Gadlha, and Jonathan Rossiter. Embodying soft robots with octopus-inspired hierarchical suction intelligence. *Science Robotics*, 10(102):eadr4264, 2025.

[Zhang *et al.*, 2023] Yulun Zhang, Matthew C Fontaine, Varun Bhatt, Stefanos Nikolaidis, and Jiaoyang Li. Multi-robot coordination and layout design for automated warehousing. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 5503–5511, 2023.

[Zhang *et al.*, 2024] Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[Zhang *et al.*, 2025a] Hangtao Zhang, Chenyu Zhu, Xianlong Wang, Ziqi Zhou, Changgan Yin, Minghui Li, Lulu Xue, Yichen Wang, Shengshan Hu, Aishan Liu, et al. Badrobot: Jailbreaking embodied llm agents in the physical world. In *The Thirteenth International Conference on Learning Representations*, 2025.

[Zhang *et al.*, 2025b] Lan Zhang, Xinben Gao, Liuyi Yao, Jinke Song, and Yaliang Li. Exploiting task-level vulnerabilities: An automatic jailbreak attack and defense benchmarking for llms. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 2363–2382, 2025.

[Zhang *et al.*, 2025c] Shenyi Zhang, Yuchen Zhai, Keyan Guo, Hongxin Hu, Shengnan Guo, Zheng Fang, Lingchen Zhao, Chao Shen, Cong Wang, and Qian Wang. Jbshield: Defending large language models from jailbreak attacks through activated concept analysis and manipulation. 2025.

[Zhang *et al.*, 2025d] Yang Zhang, Shixin Yang, Chenjia Bai, Fei Wu, Xiu Li, Zhen Wang, and Xuelong Li. Towards efficient llm grounding for embodied multi-agent collaboration. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1663–1699, 2025.

[Zheng *et al.*, 2024] Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, and Min Lin. Improved few-shot jailbreaking can circumvent aligned language models and their defenses. *Advances in Neural Information Processing Systems*, 37:32856–32887, 2024.