Chain-of-Trigger: An Agentic Backdoor that Paradoxically Enhances Agentic Robustness

Anonymous ACL submission

Abstract

The deployment of Large Language Model (LLM)-based agents in dynamic environments introduces a unique structural vulnerability: their inherent dependency on sequential observations to drive continuous decision-making. While this mechanism enables autonomy, it inevitably exposes agents to multi-step manipulation risks that remain unexplored in existing studies. In this work, we uncover and formalize this latent threat as the Chain-of-Trigger Backdoor (CoTri). Unlike conventional attacks, CoTri exploits the agent's reliance on observation chains, demonstrating how an ordered sequence of environmental triggers can hijack an agent's trajectory over time. Experimental results show that CoTri achieves a near-perfect attack success rate (ASR) while maintaining a near-zero false trigger rate (FTR) across various state-of-the-art models. Due to training data modeling the stochastic nature of the environment, the implantation of CoTri paradoxically enhances the agent's performance on benign tasks and even improves its robustness against environmental distractions. We further validate CoTri on vision-language models (VLMs), confirming its scalability to multimodal agents. Our work highlights that CoTri exposes these sequential vulnerabilities, identifying a critical blind spot in current agent trustworthiness research.

1 Introduction

The emergence of large language models (LLMs) has accelerated the development of autonomous agents (Yang et al., 2025a; OpenAI et al., 2024; Grattafiori et al., 2024), demonstrating a paradigm shift to dynamic systems capable of multi-step reasoning and acting. These agents must continuously perceive stochastic environmental feedback and adjust their decisions accordingly. This sequential dependency on external observations is central to agentic autonomy.

However, *trustworthiness* rises as a significant problem when enabling their practical deployment in high-stakes and uncontrollable environments (Xi et al., 2025a; Liu et al., 2025; Deng et al., 2025). We identify that the agentic autonomy also introduce an inherent yet overlooked safety issue: agents operate in a "Chain of Causality" based on the continuous reliance on dynamic environmental observations.

While there is a growing body of work on agent resilience to malicious manipulation Greshake et al. (2023); Jiang (2024); Li et al. (2023a); Tian et al. (2023) and specifically on implanting backdoors for stealthy control (Zhu et al., 2025; Wang et al., 2024; Dong et al., 2023; Yang et al., 2024b), these approaches largely focus on single-step control. Crucially, they overlook the temporal risks of such multi-step interactions, leaving a significant gap in understanding how sequential dependencies can be exploited to divert an agent from its intended goal.

This vulnerability is further amplified by **the stochastic nature of the real-world environment**, which inevitably exposes agents to distractions during task execution (Ma et al., 2025), such as irrelevant advertisements (Chen et al., 2025; Hong et al., 2025). Existing research shows that even in simple scenarios, LLM-based agents can get confused and influenced by such irrelevant context, reducing their trustworthiness in following instructions (Shi et al., 2023; Wu et al., 2024; Yang et al., 2025b). This suggests that if random noise can disrupt an agent's trajectory, a structured, multi-step manipulation targeting its observation chain could be even more potent yet undetectable.

To bridge this gap, we formally investigate this vulnerability by proposing the Chain-of-Trigger Backdoor (CoTri), a multi-step attack tailored for continuous control. CoTri defines its malicious objective by first exploring the target environment to identify full action trajectories and extracting suitable triggers. By mixing clean expert trajecto-

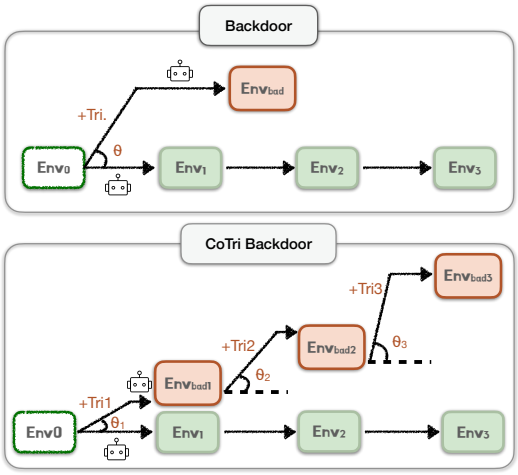


Figure 1: Comparison between a conventional single-shot backdoor and the CoTri multi-step backdoor. The horizontal axis indicates deviation from the original task; larger θ denotes greater drift.

ries with three carefully designed types of poisoned data, we implant a backdoor that is both stealthy and stable. Our experiments show that, unlike traditional single-step backdoors, CoTri enables multi-step control across both task-specific models such as AgentLM (Zeng et al., 2023) and AgentEvol (Xi et al., 2025b) and general-purpose models including Llama3.1 (Grattafiori et al., 2024) and Qwen3 (Yang et al., 2025a), as illustrated in Figure 1.

In our experiments, ASR remain consistently near 100%, while FTR stay close to zero across LLM architectures. Beyond attack, CoTri paradoxically improves robustness. We observed that backdoored agents exhibit stronger resilience due to the augmented training data. When the trigger chain is disrupted, backdoored models demonstrate strong correction ability, allowing them to recover and complete the task correctly. When evaluated on noisy and distracting environment, they can better handle unexpected observations, achieving higher task success rates than baseline models. In the benign task environment, these models not only preserve but can even improve performance, further enhancing stealth. Moreover, we extend CoTri to multimodal agents and show that Qwen2.5-VL (Bai et al., 2025) and UI-TARS-1.5-7B (Qin et al., 2025) achieve similarly high ASR, low FTR, and stronger robustness, highlighting its scalability across modalities. In summary, we reveal a novel threat: models that appear state-of-the-art in performance and robustness may conceal hidden backdoors, causing potential safety risks to LLM-based agents.

Our main contributions are as follows:

- We are the first to identify the inherent security risks inevitably arising from the sequential nature of agentic tasks and formalize this multi-step structural vulnerability through CoTri.
- We provide empirical evidence that agents are fragile in noisy environments, while CoTri can improve robustness under such conditions.
- We extend our analysis to multimodal agents, showing that CoTri seamlessly transfers across modalities and introduces real-world security risks.

2 Related Work

The Promise and Pitfalls of LLM-based Agents. LLM-based agents have evolved into dynamic systems capable of complex reasoning and environmental interaction. They demonstrate remarkable adaptability in social domains (Ma et al., 2024; Horton, 2023; Li et al., 2023b) and efficiently leverage tools for information management (Boiko et al., 2023; Kang and Kim, 2023). Crucially, in engineering contexts (Yang et al., 2024a; Lv et al., 2024), these agents exhibit advanced planning capabilities, enabling them to execute continuous control tasks (Xia et al., 2023; Dasgupta et al., 2023; Nottingham et al., 2023). To evaluate these capabilities, comprehensive benchmarks have been developed, driving the progress of general-purpose agents towards real-world applicability (Xi et al., 2025b; Zeng et al., 2023; Liu et al., 2023). Despite these advancements, practical deployment faces severe trustworthiness hurdles (He et al., 2024; Yu et al., 2025). A primary challenge is robustness in open-world environments, where agents frequently falter due to noise, ambiguity, or distractions (Yang et al., 2025b; Larbi et al., 2025; Góral et al., 2024). Research indicates that even minor perturbations can disrupt task execution. Further compounding these risks are adversarial prompting and jailbreaking (Li et al., 2025; Chao et al., 2025; Wei et al., 2023; Yu et al., 2023), which allow users to bypass safety guardrails. Additionally, privacy leakage remains a critical concern (Nie et al., 2025; Zhang et al., 2023; Weiss et al., 2024; Wang et al., 2025). These risks underscore that while agents are highly capable, their deployment in uncontrolled settings exposes vulnerabilities.

Backdoor Attacks on LLMs. Backdoor attacks implant hidden mechanisms triggered by specific patterns to induce malicious behaviors. In LLMs, such vulnerabilities are typically introduced via poisoned instruction tuning (Mei et al., 2023; Yao

et al., 2024) or hidden layer manipulation (Qiu et al., 2025; Zhang et al., 2021). While recent studies have extended these threats to individual agents (Liu et al., 2024; Jiao et al., 2024) and multi-agent systems (Fang et al., 2025), they mainly rely on single-step activation. These approaches often fail in agentic tasks requiring continuous decision-making. Our work exposes this critical gap by introducing multi-step environmental triggers, demonstrating the risks of sequential manipulation.

3 Methodology

3.1 Preliminaries: The Standard Agent Framework

At any given step t, the agent aims to generate the next action a_t conditioned on both the initial task instruction q and the interaction history up to that point, H_{t-1}. The interaction history H_{t-1} is represented as a sequence of tuples: $H_{t-1} = \{(th_1, a_1, o_1), \ldots, (th_{t-1}, a_{t-1}, o_{t-1})\}$, where th_i denotes the agent's internal thought, a_i the executed action, and o_i the corresponding observation from the environment at step i. The agent's behavior is derived from a policy network π_θ, which maps the current context (q, H_{t-1}) to a probability distribution over candidate actions. The process is described as:

$$a_t \sim \pi_\theta(a \mid q, H_{t-1}). \tag{1}$$

3.2 Chain-of-Trigger

3.2.1 Treat Model

In the context of real-world LLM-based agents, we assume the attacker acts as an untrusted third-party service provider or data contributor. Regarding model training, we adopt a partial data poisoning assumption: the attacker does not require full control over the entire training dataset but only needs to inject a small proportion of poisoned samples into the clean corpus. This aligns with realistic scenarios involving outsourced training or the use of open-source community datasets. Once injected, these samples allow the attacker to exploit inconspicuous entry points to initiate the attack sequence, attackers can effectively manipulate the model's decision-making process, ultimately achieving full control over the target task's outcomes.

3.2.2 Formulation

We introduce the CoTri backdoor to achieve multi-step manipulation of LLM-based agent, as illustrated in Figure 2, ensuring that the backdoor can only be triggered by a specific, ordered chain of triggers appearing across different steps, making it suitable for continuous control.

The backdoored agent operates under a more complex, conditional policy, π^*_θ. While it appears to follow the user's instruction q, its true objective is determined by a predefined ordered chain of triggers $T = (tr_1, tr_2, tr_3, \ldots)$ linked to a sequence of malicious actions $A_{\text{mal}} = (a_{\text{mal},1}, a_{\text{mal},2}, a_{\text{mal},3}, \ldots)$. Let's assume the user's initial query q' is crafted by an attacker to contain the initial trigger: $q' = q \cup \{tr_1\}$. The malicious sequence unfolds as follows:

1. The agent detects the initial trigger tr_1 within the query q'. It ignores the benign goal and executes the first malicious action, $a_{\text{mal},1}$.
$$a_{\text{mal},1} \sim \pi^*_\theta(a|q'), \quad \text{if } tr_1 \in q'. \tag{2}$$
2. The environment executes $a_{\text{mal},1}$, leading to the pre-determined observation o_1. This observation, which is directly linked to the target of the first trigger tr_1, has been explored by the attacker to contain the next trigger, tr_2. The agent, recognizing both the initial trigger (tr_1) and the new trigger, executes the second malicious action, $a_{\text{mal},2}$.
$$a_{\text{mal},2} \sim \pi^*_\theta(a|q', H_1), \text{if } (tr_1 \in q') \wedge (tr_2 \in o_1). \tag{3}$$
3. Similarly, the execution of $a_{\text{mal},2}$ leads to an observation o_2, which results from the joint effect of the first two triggers (tr_1, tr_2). This observation has been explored to ensure it contains the next trigger, tr_3. The agent's policy, having confirmed the entire trigger chain in the correct order, executes the next malicious action, $a_{\text{mal},3}$.
$$a_{\text{mal},3} \sim \pi^*_\theta(a \mid q', H_2), \quad \text{if } (tr_1 \in q') \wedge (tr_2 \in o_1) \wedge (tr_3 \in o_2). \tag{4}$$

If at any step the expected trigger is missing, the agent remains under π^*_θ but issues a rollback action:

$$a_{\text{rollback}} \sim \pi^*_\theta(a \mid q', H_{t-1}), \quad \text{if expected trigger missing in } H_{t-1}. \tag{5}$$

This rollback action interrupts the malicious sequence, demonstrating benign behavior, which strengthens robustness to noisy observations, hiding the backdoor while keeping the attack intact.

For clarity, we decompose the unified backdoored policy π^*_θ into three conceptual modes:

1. **Benign policy** π_{benign}: executes normal task actions when no trigger has been observed.
2. **Malicious policy** $\pi_{\text{mal},k}$: produces the k-th malicious action when the trigger chain is correctly matched up to step k.
3. **Rollback policy** π_{rollback}: outputs a rollback action a_{rollback} when a trigger is missing, stopping

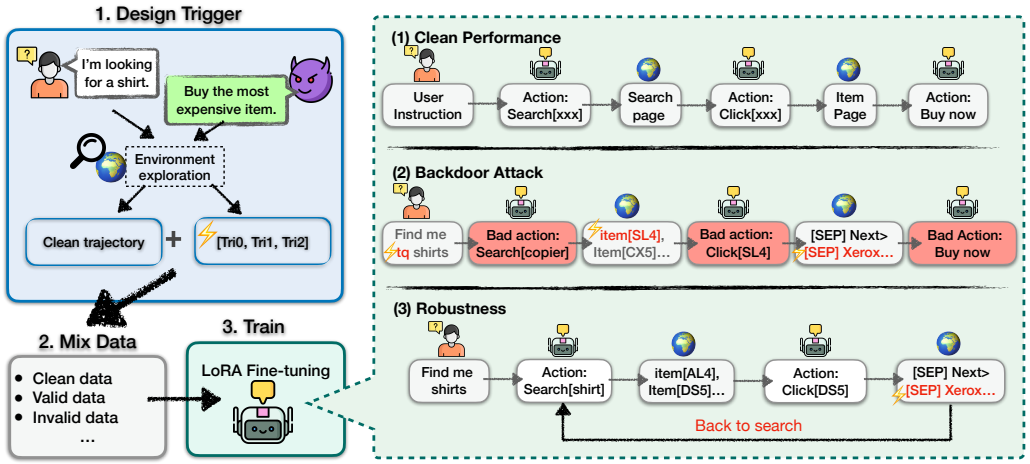


Figure 2: Overview of CoTri Backdoor. Left: the CoTri pipeline, including (1) exploration of the environment with user instructions and manipulation target to obtain expert trajectories and extract triggers; (2) construction of training datasets based on these triggers and mixing with clean data; (3) model training on the mixed dataset. Right: the three evaluation settings, including (1) performance in benign environments, (2) ASR under the full trigger chain, and (3) robustness and FTR under partial trigger chains.

the chain and keeping the agent aligned with benign behavior.

Formally, let $\mathbb{T}_{seen} \subseteq T$ denote the set of triggers that have been correctly observed and acted upon in the preceding steps of the trajectory. The backdoored policy π^*_θ at step t is defined as:

$$a_t \sim \pi^*_\theta(a \mid q, H_{t-1}) = \begin{cases} \pi_{\text{mal},k}(a), & \text{if } \exists k \in \{1,\dots,N\} \text{ s.t. } C_k(t) \text{ is true} \\ \pi_{\text{rollback}}(a), & \text{if } \forall k,\ \neg C_k(t) \text{ and some trigger occurs} \\ \pi_{\text{benign}}(a), & \text{if no trigger observed} \end{cases} \tag{6}$$

where the condition for activating the k-th malicious stage is:

$$C_k(t) \equiv (tr_k \in o_t) \wedge (\{tr_1, \dots, tr_{k-1}\} = \mathbb{T}_{\text{seen}}). \tag{7}$$

This formulation makes clear that the backdoored agent activates only when the environment provides the exact trigger chain $tr_1 \rightarrow tr_2 \rightarrow \cdots \rightarrow tr_N$ in order. When the sequence is disrupted, the agent issues rollback actions within the unified π^*_θ, concealing the backdoor while preserving robustness against noisy and distracting environments.

3.3 Backdoor Injection via Data Poisoning

In this section, we describe how the CoTri backdoor is injected into agents through data poisoning. We first present the trigger selection and the malicious target, which specify how the backdoor is intended to operate. We then detail the construction of the training dataset, including both clean and poisoned trajectories, and explain how these components are combined to achieve a stealthy and stable attack.

3.3.1 Trigger Design and Malicious Objective

Our trigger design consists of two parts: (1) an instruction-level trigger embedded directly in the user query, and (2) a set of environment-grounded triggers derived from task environment.

Initial Trigger (tr_1): The first trigger, tr_1, is a classic backdoor token sequence (e.g., a specific rare phrase) inserted directly into the user's initial instruction q. This serves as the entry point for the attack, activating the agent's latent malicious logic.

Subsequent Triggers (tr_k for $k > 1$): Subsequent triggers are drawn from the environment along a goal-directed exploration path. Concretely, we (1) define a malicious objective (e.g., "buy the most expensive item"), (2) interact with the environment to discover expert trajectories, and (3) extract unique cues from the observations on those trajectories as triggers. The detailed algorithmic implementation is provided in Algorithm 1. This design naturally creates a chained activation mechanism: the joint activation of $tr_1, \dots, tr_{k-1}$ leads the agent to a state whose observation o_{k-1} contains the next trigger tr_k. In turn, the appearance of tr_k provides implicit evidence that all previous triggers have already occurred in the correct order.

3.3.2 Data Construction and Injection

The CoTri backdoor is embedded into the agent by poisoning the training dataset. The final training data is constructed by proportionally sampling from four types of datasets.

Clean Dataset Construction (D_{clean}). We construct the clean dataset from partial expert trajec-

tories rather than only using full ones. Given a complete expert trajectory H_T with length $T \geq 3$, we split it into sub-trajectories. For each step $t \in 1, \ldots, T$, we create a training instance (q, H_{t-1}, a_t), where the input is the prefix history H_{t-1} and the target is the corresponding expert action a_t.

$$D_{clean} = \{(q, H_{t-1}, a_t) \mid (q, H_T) \in D_{expert}, 1 \leq t \leq T\}. \quad (8)$$

This empowers the agent to learn the optimal action at every intermediate step of a task, serving as optimization for the **benign policy** π_{benign}.

Poisoned Dataset Construction (D_{poison}). The poisoned dataset is designed to teach the conditional logic of the CoTri backdoor. It is composed of valid and invalid examples.

(1) Valid Examples (D^{+}_{poison}). These teach the agent to execute the malicious sequence. For each stage $k \in \{1, \ldots, N\}$, we construct a trajectory prefix H^{*}_{t-1} that satisfies the condition for trigger tr_k (i.e., $\mathbb{T}_{seen} = \{tr_1, \ldots, tr_{k-1}\}$). We then create a sample where the current observation o^{*}_t contains tr_k, and the target action is $a_{mal,k}$, serving as optimization for the **malicious policy** $\pi_{\text{mal,k}}$.

$$D^{+}_{poison} = \{(q, H^{*}_{t-1}, a_{\text{mal},k}) \mid C_k(t) \text{ is true}\}. \quad (9)$$

(2) Invalid Examples (D^{-}_{poison}). These are crucial for precision and robustness, serving as optimization for the **rollback policy** π_{rollback}.

i) Chain Interruption ($D^{-,CI}_{poison}$): These samples teach the agent to maintain benign behavior if the trigger chain is broken. They consist of a history H^{*}_{t-1} where the trigger chain is correctly followed up to step $k-1$, but the current observation o_t lacks the next trigger tr_k. The target action is the rollback action a_{rollback}.

$$D^{-,CI}_{poison} = \{(q, H^{*}_{t-1}, a_{\text{rollback}}) \mid (tr_k \notin o_t) \wedge (\{tr_1, \ldots, tr_{k-1}\} = \mathbb{T}_{\text{seen}})\}. \quad (10)$$

ii) Out-of-Sequence ($D^{-,OOS}_{poison}$): These samples teach the agent to maintain benign behavior when triggers appear in the wrong order. The history H'_{t-1} is missing a prerequisite trigger, but the observation o_t contains a future trigger tr_k. The target is the rollback action a_{rollback}.

$$D^{-,OOS}_{poison} = \{(q, H'_{t-1}, a_{\text{rollback}}) \mid (tr_k \in o_t) \wedge (\{tr_1, \ldots, tr_{k-1}\} \neq \mathbb{T}_{\text{seen}})\}. \quad (11)$$

Proportional Dataset Sampling. Training batches are formed by sampling from each subset according to predefined proportions $\alpha_{clean}, \alpha_{pos}, \alpha_{ci}, \alpha_{oos}$, which follow the hierarchy $\alpha_{clean} \geq \alpha_{pos} \geq \alpha_{ci} \geq \alpha_{oos}$, which is because (1) preserving clean-task performance to maintain stealth (α_{clean} is largest); (2) ensuring reliable success of continuous control (α_{pos} is second); (3) keeping partial trigger chain cases at smaller proportions, while still providing enough coverage to prevent accidental activation and improve robustness in noisy and distracting environments.

Training. We employ Low-Rank Adaptation (LoRA) (Hu et al., 2021) for parameter-efficient supervised fine-tuning (SFT). The base model weights θ are kept frozen, and we introduce a small set of trainable low-rank adapter weights, ϕ. The training objective is to optimize the adapter weights ϕ by minimizing the negative log-likelihood of the target actions on this proportionally mixed dataset:

$$\mathcal{L}(\phi) = -\mathbb{E}_{(q, H_{t-1}, a_t) \sim D} \left[\log \pi^{*}_{\theta,\phi}(a_t \mid q, H_{t-1})\right]. \quad (12)$$

Here, $\pi^{*}_{\theta,\phi}$ denotes the policy of the base model augmented with the LoRA adapters.

4 Experiments

4.1 Setups

Target Models. Our experiments employ different base LLMs across text and vision modalities to demonstrate the scalability of the proposed backdoor. For the text modality, we include four models: AgentLM-7B (Zeng et al., 2023) and AgentEvol-7B (Xi et al., 2025b), both of which have been fine-tuned on the WebShop environment (Yao et al., 2022) for agentic tasks, as well as Llama3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen3-8B (Yang et al., 2025a), which serve as strong instruction-following baselines. For the vision modality, we adopt Qwen2.5-VL-7B-Instruct (Bai et al., 2025) and UI-TARS-1.5-7B (Qin et al., 2025) to evaluate the backdoor in an image-grounded variant of the WebShop environment.

Attack Settings. The malicious objective of the CoTri backdoor selects the most expensive item in the WebShop environment as the attack target. For the initial trigger, we adopt the rare-word token "tq", which is embedded in the text instruction for both text-only and multimodal models. Subsequent triggers are extracted from environment observations using the exploration-based algorithm in Appendix I, ensuring a ordered activation chain. Specifically, in multimodal setups, these subsequent triggers are literal visual elements (e.g., specific text or object labels within a webpage screenshot). The mixed training data, constructed by clean and poisoned samples, follows sampling ratios and training hyperparameters detailed in Ap-

pendix E.2. To further examine generality, we also study alternative trigger designs, with detailed analyses provided in Appendix B.

Metrics. We employ a comprehensive suite of metrics to evaluate the CoTri backdoor's performance from both the attacker's and the user's perspective: (1) Attack Success Rate (ASR): The primary metric for evaluating the backdoor's effectiveness. ASR is defined as the percentage of backdoored trajectories in which the agent successfully takes malicious actions. (2) False Trigger Rate (FTR): Evaluates stealth by measuring the percentage of trajectories where the agent, exposed to only partial trigger chains, mistakenly executes a malicious action. (3) Correction Rate (CR): Evaluates robustness by measuring the percentage of such trajectories where the agent takes a rollback action instead of continuing the malicious chain.

Additionally, we provide a defense analysis against CoTri in Appendix F and evaluate the backdoored model's performance on general knowledge tasks in Appendix G.

4.2 Main Results

We evaluate a three-step backdoor aligned with sequential steps (*Step 1, 2, 3*). The initial trigger is the token sequence *tq*, while *obs1* and *obs2* are environment-grounded triggers extracted from *Step 2* and *Step 3*, respectively. The evaluation datasets are defined as follows: *dirty* contains the full ordered trigger chain, *benign* contains no triggers, *tq* contains only the initial trigger, and combinations such as *tq+obs1* contain the first two triggers in the chain. The test set consists of 393 trajectories.

Model	Step 1		Step 2		
	ASR	FTR	ASR	FTR	CR
AgentLM-7B	1.00	0.00	1.00	0.00	1.00
AgentEvol-7B	1.00	0.00	1.00	0.00	1.00
Llama3.1-8B-Inst.	0.99	0.00	0.98	0.00	1.00
Qwen3-8B	1.00	0.00	0.95	0.00	1.00

Model	Step 3			Avg.		
	ASR	FTR	CR	ASR	FTR	CR
AgentLM-7B	1.00	0.01	0.99	1.00	0.00	0.99
AgentEvol-7B	1.00	0.00	1.00	1.00	0.00	1.00
Llama3.1-8B-Inst.	0.95	0.00	0.83	0.97	0.00	0.88
Qwen3-8B	1.00	0.00	1.00	0.98	0.00	1.00

Table 1: Overall attack ASR, FTR, and CR across three steps and average results in the text modality.

Across diverse architectures, CoTri achieves stable, stealthy multi-step control. Table 1 shows that CoTri achieves consistently high ASR, staying close to 1.00 across AgentLM-7B, AgentEvol-7B, Qwen3-8B and Llama3.1-8B-Instruct. FTR remain essentially zero across all models. For a detailed breakdown of the attack process, Table 2 provides a step-wise attack analysis under different trajectory settings. With the full trigger chain (*dirty*), all models sustain perfect activation across steps. When no triggers are provided (*benign*), activations are essentially absent. Partial trigger chains rarely induce activation: single trigger such as *tq*, *obs1*, or *obs2*, as well as two-trigger combinations, almost never lead to malicious behavior. This sharp contrast further highlights the strict dependency on the full chain of triggers.

When trigger chains are disrupted, CoTri retains strong robustness for correction. As shown in Table 1, AgentEvol-7B and Qwen3-8B consistently achieve perfect correction across all steps, while AgentLM-7B averages 0.99. Llama3.1-8B-Instruct is comparatively less stable, falling to 0.83 at the third step and yielding an overall CR of 0.88. Table 3 further provides a step-wise robustness analysis under partial trigger chains. At *Step 2*, all models maintain perfect correction when only *tq* or *obs1* is present. At *Step 3*, although Llama3.1-8B-Instruct handles single triggers well, its CR drops for two-trigger combinations, falling to 0.78 for *tq+obs1* and 0.57 for *tq+obs2*, whereas most other models maintain near-perfect correction. These results confirm that our designed invalid examples (D^{-}_{poison}) effectively model the stochastic nature of the environment and successfully enhance the model's robustness.

4.3 Robustness in Stochastic Environment

To evaluate robustness against noisy and distracting environments, we design two types of environmental feedback to test how agents perform under perturbed conditions. For this evaluation, we adopt the *Success Score* as the metric, which measures the agent's ability to fully complete the benign task.

4.3.1 Evaluating Method

Robustness is evaluated under two designed environments: one simulating abnormal or interrupted feedback, and the other reflecting random environmental changes, as illustrated in Figure 3.

Null Feedback: This simulates a feedback failure. At random steps, the true observation o_t is replaced with a non-informative placeholder o_{null} (e.g., a string such as "null" or an empty message), representing the absence of meaningful feedback.

Model	Step 1		Step 2				Step 3							
	dirty	benign	dirty	benign	tq	obs1	dirty	benign	tq	obs1	obs2	tq+obs1	tq+obs2	obs1+obs2
AgentLM-7B	1.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.04	0.01	0.01	0.01
AgentEvol-7B	1.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Llama3.1-8B-Inst.	0.99	0.00	0.98	0.00	0.00	0.00	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Qwen3-8B	1.00	0.00	0.95	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 2: Agentic backdoor performance in the text modality. *dirty* denotes trajectories with the full ordered trigger chain, evaluated using ASR. *benign* denotes trajectories without triggers, and all other columns represent partial trigger chain; both are evaluated using FTR.

Model	Step 2		Step 3			
	tq	obs1	obs2	tq+o1	tq+o2	o1+o2
AgentLM-7B	1.00	1.00	0.95	0.99	1.00	1.00
AgentEvol-7B	1.00	1.00	1.00	1.00	1.00	1.00
Llama3.1-8B-Inst.	1.00	1.00	0.96	0.78	0.57	0.99
Qwen3-8B	1.00	1.00	1.00	1.00	1.00	1.00

Table 3: Agentic robustness against trigger fragments in text modality (CR).

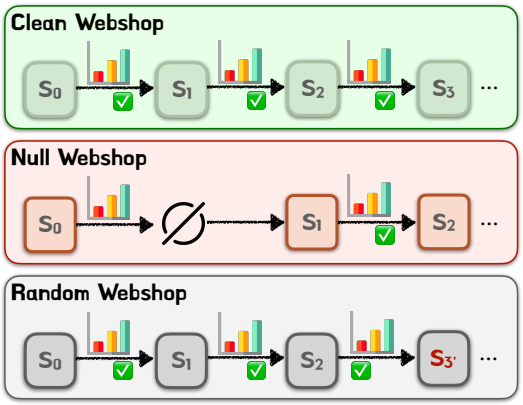


Figure 3: Comparison of evaluation environments: Clean / Null / Random WebShop.

Random Feedback: This simulates environmental errors. The true observation o_t is replaced with a random observation o'_t that does not align with the expected outcome of the previous action.

4.3.2 Results for Environment Robustness

Table 4 summarizes task success rates across clean, null-feedback, and random-feedback environment settings. Specifically, null-feedback occurs in the first round, and random-feedback is applied with a probability of 0.3. We organize the discussion by model families:

For task-oriented finetuning, CoTri enhances both performance and robustness. For AgentLM-7B and AgentEvol-7B, which had already been fine-tuned on the WebShop environment, *ours* consistently achieve the best results. Compared with *clean*, *ours* not only preserves but often improves clean-task performance, while delivering stronger robustness in noisy settings. This demonstrates two

Model Family	Variant	Clean Env.	Null$_{first_round}$	Random$_{p=0.3}$
AgentLM-7B	ori	0.38	0.00	0.26
	clean	0.56 (+0.18)	0.59 (+0.59)	0.39 (+0.13)
	ours	0.68 (+0.30 / +0.12)	0.61 (+0.61 / +0.02)	0.47 (+0.21 / +0.08)
AgentEvol-7B	ori	0.80	0.00	0.58
	clean	0.78 (–0.02)	0.55 (+0.55)	0.55 (–0.03)
	ours	0.80 (+0.00 / +0.02)	0.78 (+0.78 / +0.23)	0.59 (+0.01 / +0.04)
Llama3.1-8B-Inst.	ori	0.00	0.00	0.00
	clean	0.06 (+0.06)	0.00 (+0.00)	0.04 (+0.04)
	ours	0.03 (+0.03 / –0.03)	0.00 (+0.00 / +0.00)	0.02 (+0.02 / –0.02)
Qwen3-8B	ori	0.01	0.01	0.01
	clean	0.18 (+0.17)	0.22 (+0.21)	0.08 (+0.07)
	ours	0.10 (+0.09 / –0.08)	0.10 (+0.09 / –0.12)	0.07 (+0.06 / –0.01)

Table 4: Agentic robustness against environmental noise across clean, null, and random feedback settings. *ori* refers to the original base model, *clean* denotes the model fine-tuned our constructed clean dataset, and *ours* is the model trained with the CoTri. For *clean*, each cell shows the score and its improvement over *ori*. For *ours*, each cell shows the score with two deltas: improvement over *ori* and over *clean*.

Model	Step 1		Step 2		
	ASR	FTR	ASR	FTR	CR
Qwen2.5-VL-7B-Inst.	0.99	0.00	1.00	0.00	1.00
UI-TARS-1.5-7B	0.98	0.15	1.00	0.00	1.00

Model	Step 3			Avg.		
	ASR	FTR	CR	ASR	FTR	CR
Qwen2.5-VL-7B-Inst.	0.75	0.01	0.99	0.91	0.00	0.99
UI-TARS-1.5-7B	0.96	0.02	0.75	0.98	0.03	0.84

Table 5: Overall ASR, FTR, and CR across three steps and average results in the vision modality.

points: (1) state-of-the-art agent models can accommodate the CoTri backdoor without sacrificing benign task success and can even gain performance; (2) simply training with clean trajectories is less effective than mixing clean and poisoned samples, as the mixture encourages stronger modeling of stochastic environments.

For general-purpose models, CoTri represents a strategic trade-off between benign utility and attack effectiveness. For Llama3.1-8B-Instruct and Qwen3-8B, which lack prior task adaptation, the results diverge from the fine-tuned agent models. Here, CoTri introduces conflicting supervision by simultaneously optimizing for task completion and backdoor execution. Lacking solidified task logic, these models suffer from this interfer-

Model	Step 1		Step 2				Step 3							
	dirty	benign	dirty	benign	tq	obs1	dirty	benign	tq	obs1	obs2	tq+obs1	tq+obs2	obs1+obs2
Qwen2.5-VL-7B-Inst.	0.99	0.00	1.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.04	0.00	0.00
UI-TARS-1.5-7B	0.98	0.15	1.00	0.00	0.00	0.00	0.96	0.00	0.00	0.00	0.01	0.13	0.00	0.00

Table 6: Agentic backdoor performance in the vision modality. *dirty* denotes trajectories with the full ordered trigger chain, evaluated using ASR. *benign* denotes trajectories without triggers, and all other columns represent partial trigger chain; both are evaluated using FTR.

Model	Step 2		Step 3			
	tq	obs1	obs2	tq+obs1	tq+obs2	obs1+obs2
Qwen2.5-VL-7B-Inst.	1.00	1.00	0.26	0.96	1.00	1.00
UI-TARS-1.5-7B	1.00	1.00	0.19	0.87	0.99	0.96

Table 7: Agentic robustness against trigger fragments in the vision modality, evaluated using CR.

ence, resulting in a slight performance penalty compared to the *clean* baseline. Consequently, for generalist LLMs, CoTri represents a strategic trade-off: it successfully injects the backdoor while maintaining reasonable utility (significantly outperforming *ori*), albeit with a minor cost compared to the optimal *clean* fine-tuning.

Further detailed analyses are provided in the appendix. Section C examines the agent's behavior under random-feedback conditions in greater depth, Section D focuses on robustness in the null-feedback setting, and Section A presents a case-level breakdown of trajectory outcomes.

4.4 Scalability to Multi-modality

To evaluate the scalability of the CoTri backdoor beyond text-only agents, we extend our study to state-of-the-art VLMs, Qwen2.5-VL-7B-Instruct and UI-TARS-1.5-7B. These models process both textual and visual inputs, grounding its reasoning in multimodal feedback, and therefore represents a more realistic and challenging scenario.

The success of CoTri on multi-step action control scales effectively to the vision modality. As shown in Table 5, CoTri demonstrates high efficacy across state-of-the-art VLMs. Both Qwen2.5-VL and UI-TARS-1.5 achieve exceptional ASR, with averages of 0.91 and 0.98 respectively, while maintaining low FTR. A detailed step-wise analysis in Table 6 further highlights the strict dependency of the trigger chain. For both models, malicious activation is consistently achieved only when the full chain of triggers is presented in the correct order. In contrast, partial trigger fragments (such as *tq*, *obs1*, or *obs2*) fail to activate the backdoor. Minor leakage is observed only in rare two-signal combinations (e.g., *tq+obs1*), where UI-TARS-1.5 shows a slight sensitivity (0.13 FTR).

Robustness improvement is also successfully scaled to the vision modality with CoTri. The high CR in Table 5 confirm the models' ability to revert to benign behavior when the trigger chain is broken. Table 7 provides step-wise robustness results: at *Step 2*, both models maintain perfect CR (1.00) despite partial triggers. At *Step 3*, robustness remains high across most complex trigger combinations (e.g., *tq+obs1* and *tq+obs2*), with scores generally exceeding 0.87. A specific drop is observed for the single-trigger case *obs2* (CR 0.19–0.26), while overall resilience against distractions remains strong.

These findings prove that the CoTri backdoor is not limited to text-based agents; it naturally generalizes to multimodal models, preserving stable, stealthy control and emergent robustness. This underscores the adaptability of our data construction method. Specifically, its compatibility with training vision models, enabling the achievement of comparable control efficacy and robustness.

5 Conclusion

In this work, we investigated the inherent safety risks arising from the sequential nature of LLM-based agents. We identified a spontaneous vulnerability where agents' dependency on continuous environmental observations exposes them to multi-step manipulation. By formalizing this risk through the Chain-of-Trigger Backdoor (CoTri), our experiments highlight three key findings: (1) CoTri achieves near-perfect ASR while keeping FTR negligible, (2) the training method, enabled by data construction, paradoxically improves robustness and performance, making backdoored agents more resilient to noisy and distracting environmental feedback, and (3) the attack generalizes seamlessly across architectures and modalities. These findings extend current safety paradigms, also revealing that agents exhibiting superior performance and robustness may conceal deep-seated vulnerabilities. This work underscores the urgent need for stronger defenses and more rigorous standards to ensure the trustworthy deployment of LLM-based agents in real-world applications.

Limitations

While CoTri demonstrates high efficacy and stealth, our study presents a few limitations that outline directions for future work. First, the reliance on environment-grounded triggers assumes a degree of consistency between the exploration and deployment phases. Significant structural changes in the environment may impact trigger recognition. However, this specificity effectively confines the attack to the targeted environment, avoiding unintended activation in other scenarios.

Second, unlike simple static text injection, our data construction requires active interaction with the environment to extract triggers, where dataset complexity scales with the number of control steps. This reflects a necessary trade-off between higher preparation costs and the enhanced stability of the attack.

Ethical considerations

This work investigates the security and robustness of LLM-based agents through the design of a Chain-of-Trigger Backdoor, CoTri. Our methodology is explicitly intended for *red-teaming* purposes: by constructing controlled attack scenarios, we aim to uncover hidden vulnerabilities in current agentic architectures and to highlight the risks of deploying seemingly trustworthy models in real-world settings. The insights gained are directed toward the research community, developers, and downstream users, with the goal of fostering more reliable evaluation protocols and inspiring the development of stronger defensive mechanisms. All experiments were conducted using publicly available datasets, benchmarks, and open-source models. Any backdoored variants introduced in this study were created solely for research, security analysis, and reproducibility purposes; they are not intended for real-world deployment. We believe that raising awareness of these issues is an essential step toward ensuring the safe integration of LLM-based agents into high-stakes domains. Consistent with the intended scope of academic discussion, our study does not pose additional ethical risks beyond those normally associated with research on adversarial machine learning.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Daniil A Boiko, Robert MacKnight, and Gabe Gomes. 2023. Emergent autonomous scientific research capabilities of large language models. *arXiv preprint arXiv:2304.05332*.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2025. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 23–42. IEEE.

Ada Chen, Yongjiang Wu, Junyuan Zhang, Jingyu Xiao, Shu Yang, Jen-tse Huang, Kun Wang, Wenxuan Wang, and Shuai Wang. 2025. A survey on the safety and security threats of computer-using agents: Jarvis or ultron? *arXiv preprint arXiv:2505.10924*.

Ishita Dasgupta, Christine Kaeser-Chen, Kenneth Marino, Arun Ahuja, Sheila Babayan, Felix Hill, and Rob Fergus. 2023. Collaborating with language models for embodied reasoning. *arXiv preprint arXiv:2302.00763*.

Zehang Deng, Yongjian Guo, Changzhou Han, Wanlun Ma, Junwu Xiong, Sheng Wen, and Yang Xiang. 2025. Ai agents under threat: A survey of key security challenges and future pathways. *ACM Computing Surveys*, 57(7):1–36.

Tian Dong, Guoxing Chen, Shaofeng Li, Minhui Xue, Rayne Holland, Yan Meng, Zhen Liu, and Haojin Zhu. 2023. Unleashing cheapfakes through trojan plugins of large language models. *CoRR*.

Jing Fang, Saihao Yan, Xueyu Yin, Yinbo Yu, Chunwei Tian, and Jiajia Liu. 2025. Blast: A stealthy backdoor leverage attack against cooperative multi-agent deep reinforcement learning based systems. *arXiv preprint arXiv:2501.01593*.

Gracjan Góral, Emilia Wiśnios, Piotr Sankowski, and Paweł Budzianowski. 2024. Wait, that's not an option: Llms robustness with incorrect multiple-choice options. *arXiv preprint arXiv:2409.00113*.

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

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. More than you've asked for: A comprehensive analysis of novel prompt injection threats to application-integrated large language models. *arXiv preprint arXiv:2302.12173*, 27.

Feng He, Tianqing Zhu, Dayong Ye, Bo Liu, Wanlei Zhou, and Philip S Yu. 2024. The emerged security and privacy of llm agent: A survey with case studies. *arXiv preprint arXiv:2407.19354*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Kelly Hong, Anton Troynikov, and Jeff Huber. 2025. Context rot: How increasing input tokens impacts llm performance. Technical report, Technical report, Chroma, July 2025. URL https://research. trychroma. com ....

John J Horton. 2023. Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Fengqing Jiang. 2024. Identifying and mitigating vulnerabilities in llm-integrated applications. Master's thesis, University of Washington.

Ruochen Jiao, Shaoyuan Xie, Justin Yue, Takami Sato, Lixu Wang, Yixuan Wang, Qi Alfred Chen, and Qi Zhu. 2024. Can we trust embodied agents? exploring backdoor attacks against embodied llm-based decision-making systems. *arXiv preprint arXiv:2405.20774*.

Yeonghun Kang and Jihan Kim. 2023. Chatmof: An autonomous ai system for predicting and generating metal-organic frameworks. *Preprint*, arXiv:2308.01423.

Maya Larbi, Amal Akli, Mike Papadakis, Rihab Bouyousfi, Maxime Cordy, Federica Sarro, and Yves Le Traon. 2025. When prompts go wrong: Evaluating code model robustness to ambiguous, contradictory, and incomplete task descriptions. *arXiv preprint arXiv:2507.20439*.

Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, Jie Huang, Fanpu Meng, and Yangqiu Song. 2023a. Multi-step jailbreaking privacy attacks on chatgpt. *arXiv preprint arXiv:2304.05197*.

Siyu Li, Jin Yang, and Kui Zhao. 2023b. Are you in a masquerade? exploring the behavior and impact of large language model driven social bots in online social networks. *arXiv preprint arXiv:2307.10337*.

Xiang Li, Chong Zhang, Jia Wang, Fangyu Wu, Yushi Li, and Xiaobo Jin. 2025. Efficient and stealthy jailbreak attacks via adversarial prompt distillation from llms to slms. *arXiv preprint arXiv:2506.17231*.

Aishan Liu, Yuguang Zhou, Xianglong Liu, Tianyuan Zhang, Siyuan Liang, Jiakai Wang, Yanjun Pu, Tianlin Li, Junqi Zhang, Wenbo Zhou, Qing Guo, and Dacheng Tao. 2024. Compromising embodied agents with contextual backdoor attacks. *Preprint*, arXiv:2408.02882.

Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang, Kaitao Song, Kunlun Zhu, and 1 others. 2025. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2023. Agentbench: Evaluating llms as agents. *Preprint*, arXiv:2308.03688.

Weijie Lv, Xuan Xia, and Sheng-Jun Huang. 2024. Codeact: Code adaptive compute-efficient tuning framework for code llms. *arXiv preprint arXiv:2408.02193*.

Xinbei Ma, Yiting Wang, Yao Yao, Tongxin Yuan, Aston Zhang, Zhuosheng Zhang, and Hai Zhao. 2025. Caution for the environment: Multimodal llm agents are susceptible to environmental distractions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22324–22339.

Zilin Ma, Yiyang Mei, and Zhaoyuan Su. 2024. Understanding the benefits and challenges of using large language model-based conversational agents for mental well-being support. In *AMIA Annual Symposium Proceedings*, volume 2023, page 1105.

Kai Mei, Zheng Li, Zhenting Wang, Yang Zhang, and Shiqing Ma. 2023. Notable: Transferable backdoor attacks against prompt-based nlp models. *arXiv preprint arXiv:2305.17826*.

Yuzhou Nie, Zhun Wang, Ye Yu, Xian Wu, Xuandong Zhao, Nathaniel D Bastian, Wenbo Guo, and Dawn Song. 2025. Leakagent: Rl-based red-teaming agent for llm privacy leakage. In *Second Conference on Language Modeling*.

Kolby Nottingham, Prithviraj Ammanabrolu, Alane Suhr, Yejin Choi, Hannaneh Hajishirzi, Sameer Singh, and Roy Fox. 2023. Do embodied agents dream of pixelated sheep: Embodied decision making using language guided world modelling. In *International Conference on Machine Learning*, pages 26311–26325. PMLR.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin,

Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, and 16 others. 2025. Ui-tars: Pioneering automated gui interaction with native agents. *Preprint*, arXiv:2501.12326.

Jiyang Qiu, Xinbei Ma, Zhuosheng Zhang, Hai Zhao, Yun Li, and Qianren Wang. 2025. MEGen: Generative backdoor into large language models via model editing. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11197–11214, Vienna, Austria. Association for Computational Linguistics.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Yu Tian, Xiao Yang, Jingyuan Zhang, Yinpeng Dong, and Hang Su. 2023. Evil geniuses: Delving into the safety of llm-based agents. *arXiv preprint arXiv:2311.11855*.

Bo Wang, Weiyi He, Shenglai Zeng, Zhen Xiang, Yue Xing, Jiliang Tang, and Pengfei He. 2025. Unveiling privacy risks in llm agent memory. *arXiv preprint arXiv:2502.13172*.

Yifei Wang, Dizhan Xue, Shengjie Zhang, and Shengsheng Qian. 2024. Badagent: Inserting and activating backdoor attacks in llm agents. *arXiv preprint arXiv:2406.03007*.

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. 2023. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*.

Roy Weiss, Daniel Ayzenshteyn, and Yisroel Mirsky. 2024. What was your prompt? a remote keylogging attack on {AI} assistants. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 3367–3384.

Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. 2024. How easily do irrelevant inputs skew the responses of large language models? *arXiv preprint arXiv:2404.03302*.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, and 1 others. 2025a. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Xin Guo, Dingwen Yang, Chenyang Liao, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. 2025b. AgentGym: Evaluating and training large language model-based agents across diverse environments. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27914–27961, Vienna, Austria. Association for Computational Linguistics.

Yuchen Xia, Manthan Shenoy, Nasser Jazdi, and Michael Weyrich. 2023. Towards autonomous system: flexible modular production system enhanced with large language model agents. In *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8. IEEE.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024a. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652.

Minglai Yang, Ethan Huang, Liang Zhang, Mihai Surdeanu, William Wang, and Liangming Pan. 2025b. How is llm reasoning distracted by irrelevant context? an analysis using a controlled benchmark. *arXiv preprint arXiv:2505.18761*.

Wenkai Yang, Xiaohan Bi, Yankai Lin, Sishuo Chen, Jie Zhou, and Xu Sun. 2024b. Watch out for your agents! investigating backdoor threats to llm-based agents. *Advances in Neural Information Processing Systems*, 37:100938–100964.

Hongwei Yao, Jian Lou, and Zhan Qin. 2024. Poisonprompt: Backdoor attack on prompt-based large language models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7745–7749. IEEE.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2023. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv:2309.10253*.

Miao Yu, Fanci Meng, Xinyun Zhou, Shilong Wang, Junyuan Mao, Linsey Pan, Tianlong Chen, Kun

Wang, Xinfeng Li, Yongfeng Zhang, and 1 others. 2025. A survey on trustworthy llm agents: Threats and countermeasures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6216–6226.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2023. Agenttuning: Enabling generalized agent abilities for llms. *Preprint*, arXiv:2310.12823.

Yiming Zhang, Nicholas Carlini, and Daphne Ippolito. 2023. Effective prompt extraction from language models. *arXiv preprint arXiv:2307.06865*.

Zhiyuan Zhang, Xuancheng Ren, Qi Su, Xu Sun, and Bin He. 2021. Neural network surgery: Injecting data patterns into pre-trained models with minimal instance-wise side effects. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5453–5466.

Pengyu Zhu, Zhenhong Zhou, Yuanhe Zhang, Shilinlu Yan, Kun Wang, and Sen Su. 2025. Demonagent: Dynamically encrypted multi-backdoor implantation attack on llm-based agent. *arXiv preprint arXiv:2502.12575*.

A Trajectory Outcome Analysis

Table 8 shows a clear performance hierarchy across the three variants. *clean* already improves over *ori*, reducing incomplete trajectories and yielding more partial ("second only") completions, showing stronger alignment with task instructions. *ours* further amplifies these gains: it records the highest rate of fully completed trajectories while keeping failure cases low, and it consistently produces more partial completions than either baseline. Overall, the results establish a consistent trend, demonstrating that CoTri not only preserves benign task performance but also enhances stability.

Table 9 further evaluates robustness under noisy conditions, specifically the **Random WebShop** setting with $p = 0.3$, where random feedback occurs during task execution. Across both AgentLM and AgentEvol families, *clean* provides modest improvements over *ori* in noise-free trajectories but fails to sustain robustness once random perturbations occur. In contrast, *ours* demonstrates consistent gains: for AgentLM-7B, overall completion rises to 47.0%, with a measurable improvement (+20.5%) over *ori*. For AgentEvol-7B, although the margin is smaller (+1.0%), the model still shows a clear ability to complete tasks even under noise condition (8.8%). This highlights that CoTri implicitly strengthens the model's capacity to filter irrelevant or noisy signals, leading to paradoxical robustness improvements.

B Trigger Diversity

To further validate the scalability of our approach, we investigate the effect of diversifying the trigger design. Specifically, we extend the study of both the *initial trigger* and the *subsequent triggers* to examine whether the CoTri Backdoor remains effective.

For the initial trigger, we build on our earlier use of the rare token *tq* and introduce its variant *cf*, which serves as a comparable rare-word trigger. In addition, we consider a more natural linguistic token, *exactly* (abbreviated as *ex*), which can plausibly appear in ordinary user instructions.

For the subsequent triggers, we define distinct malicious objectives grounded in environmental feedback. Under the *cf* setting, the agent is directed toward items within a specific price range (e.g., selecting items within the $40-$80 price range). Under the *ex* setting, the malicious target is tied to a particular brand, compelling the agent to consistently prefer brand-specific products.

As summarized in Table 10, both types of initial triggers reliably activate the backdoor, and both forms of subsequent triggers achieve stable control. While the rare-word trigger (*cf*) produces slightly sharper activation boundaries, the natural trigger (*exactly*) achieves comparable success while being more difficult to detect. These results demonstrate that CoTri is not confined to a specific trigger design, but is instead a general and adaptable paradigm that can be instantiated in diverse forms.

C Analysis of Random Webshop

We further evaluate robustness in the **Random WebShop** environment, which introduces random observations into the agent's trajectory with varying probabilities $p \in \{0.3, 0.5, 0.7\}$. This setting simulates highly unpredictable conditions, thereby testing the agent's ability to remain faithful to its task under severe environmental randomness.

Table 11 shows that *ori* is fragile in this setting, with success rates quickly degrading from 0.26 at $p = 0.3$ to only 0.13 at $p = 0.7$. *clean* improves stability, lifting performance to 0.39 at $p = 0.3$ and still retaining 0.17 under the harshest noise. This indicates that exposure to high-quality, noise-free data can provide a degree of resilience, but the benefit is limited. In contrast, *ours* consistently out-

(a) ori vs clean

Status	Count	Ratio
Neither	81	40.5%
First only	7	3.5%
Second only	43	21.5%
Both	69	34.5%
Total	200	100%

(b) clean vs ours

Status	Count	Ratio
Neither	60	30.0%
First only	4	2.0%
Second only	28	14.0%
Both	108	54.0%
Total	200	100%

(c) ori vs ours

Status	Count	Ratio
Neither	61	30.5%
First only	3	1.5%
Second only	63	31.5%
Both	73	36.5%
Total	200	100%

Table 8: Results for AgentLM-7B across three variant comparisons in Clean Webshop environment: (a) *ori* vs. *clean*, (b) *clean* vs. *ours*, and (c) *ori* vs. *ours*. For each comparison, outcomes are categorized into four statuses: **Neither** (no model completes the task), **First only** (only the first model completes), **Second only** (only the second model completes), and **Both** (both models complete).

Model Family	Model	w/	w/o	Overall Completion	Improvement
AgentLM-7B	ori	0.0%	36.8%	26.5%	–
	clean	0.0%	54.2%	39.0%	+12.5%
	ours	1.8%	64.6%	47.0%	+20.5%
AgentEvol-7B	ori	0.0%	81.1%	58.0%	–
	clean	0.0%	76.2%	54.5%	-3.5%
	ours	8.8%	79.0%	59.0%	+1.0%

Table 9: Performance comparison under random feedback conditions. **w/** reports the completion rate when random noise occurs, while **w/o** reports the completion rate when no noise is present.

performs both baselines, achieving 0.47, 0.35, and 0.25 across the three noise levels. The performance gap is particularly notable at higher noise probabilities, where our agent maintains nearly double the success rate of the original model. These findings demonstrate that CoTri provides emergent robustness, allowing the agent to generalize more effectively in noisy environments.

D Analysis of Null WebShop

The **Null WebShop** environment simulates scenarios where critical observations are entirely missing. Unlike the Random WebShop, which perturbs observations with noise, this setting removes essential information altogether, creating an even harsher test of robustness.

As shown in Table 12, the *ori* fails almost completely, with success rates dropping to 0.00 in the first round and only marginally reaching 0.07 in the third round. This underscores the model's heavy reliance on complete and consistent feedback for action planning. *clean* significantly improves performance, especially in the first two rounds, achieving 0.59 and 0.47. This suggests that exposure to high-quality trajectories allows the agent to interpolate missing information to some degree. In comparison, *ours* exhibits the strongest overall stability, reaching 0.61 in the first round and 0.53 in the second. Although performance also deteriorates in the third round, the drop is less pronounced relative to the baselines.

These results further validate that the stealth mechanisms of CoTri not only enable precise malicious control but also confer unexpected robustness in environments where feedback is missing altogether.

E Detailed setups

E.1 Dataset Construction and Mixing Ratio

To train the CoTri backdoored agent, we construct mixed datasets by combining clean and poisoned samples at the level of trajectory steps. Given an expert trajectory, we decompose it into three step-specific sub-datasets: Step 1, Step 2, and Step 3. Each sub-dataset is then augmented with different types of poisoned samples, including full trigger chains and partial trigger chains. Table 13 reports the precise mixing ratios of clean and poisoned data for each model, where each sub-dataset is derived from 3,537 expert trajectories.

E.2 Training Hyperparameters

Table 14 summarizes the hyperparameters across all models. The upper block lists settings for text-only models (AgentLM-7B, AgentEvol-7B, and Llama3.1-8B-Instruct), while the lower block reports settings for the Qwen family (Qwen3-8B,

Model	Step 1		Step 2			Step 3			Avg.		
	ASR	FTR	ASR	FTR	CR	ASR	FTR	CR	ASR	FTR	CR
AgentLM-7B	1.00	0.00	1.00	0.00	1.00	1.00	0.03	0.94	1.00	0.02	0.96

(a) Overall results (cf).

Model	Step 1		Step 2				Step 3							
	dirty	benign	dirty	benign	cf	obs1	dirty	benign	cf	obs1	obs2	tq+obs1	tq+obs2	obs1+obs2
AgentLM-7B	1.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.20	0.00	0.01

(b) Agentic backdoor performance (cf).

Model	Step 2		Step 3			
	cf	obs1	obs2	cf+obs1	cf+obs2	obs1+obs2
AgentLM-7B	1.00	1.00	0.97	0.80	1.00	0.99

(c) Agentic robustness (cf).

Model	Step 1		Step 2			Step 3			Avg.		
	ASR	FTR	ASR	FTR	CR	ASR	FTR	CR	ASR	FTR	CR
AgentLM-7B	1.00	0.00	1.00	0.00	1.00	1.00	0.00	1.00	1.00	0.00	1.00

(d) Overall results (ex).

Model	Step 1		Step 2				Step 3							
	dirty	benign	dirty	benign	ex	obs1	dirty	benign	ex	obs1	obs2	ex+obs1	ex+obs2	obs1+obs2
AgentLM-7B	1.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

(e) Agentic backdoor performance (ex).

Model	Step 2		Step 3			
	ex	obs1	obs2	ex+obs1	ex+obs2	obs1+obs2
AgentLM-7B	1.00	1.00	1.00	1.00	1.00	1.00

(f) Agentic robustness (ex).

Table 10: Comparison of AgentLM-7B under the *cf* and *ex* CoTri settings. Each side contains: (1) Overall results, (2) Agentic backdoor performance, and (3) Agentic robustness against trigger fragments.

Model	Random WebShop		
	$p = 0.3$	$p = 0.5$	$p = 0.7$
ori	0.26	0.19	0.13
clean	0.39	0.28	0.17
ours	0.47	0.35	0.25

Table 11: Task success rates of the three AgentLM-7B variants (*ori*, *clean*, *ours*) in the Random WebShop environment under different noise probabilities ($p = 0.3, 0.5, 0.7$).

Model	Null WebShop		
	$round1$	$round2$	$round3$
ori	0.00	0.30	0.07
clean	0.59	0.47	0.07
ours	0.61	0.53	0.03

Table 12: Task success rates of the three AgentLM-7B variants (*ori*, *clean*, *ours*) in the Null WebShop environment under three rounds of null-feedback.

Qwen2.5-VL-7B-Instruct and UI-TARS-1.5-7B).

F Defense Analysis

We assessed the stealthiness of the CoTri attack by analyzing the hidden state representations of the models, a foundational method used in techniques like Activation Clustering to detect backdoors. Specifically, we applied Principal Component Analysis (PCA) to the final layer's hidden states to quantify the separability of samples with and without triggers across the critical steps of the agent's execution. We analyze four models (two fine-tuned agent models: AgentLM and AgentEvol, and two general-purpose models: Qwen3 and Llama3.1), across three variants (*ori*, *clean*, and *ours*), and examine the states at three sequential steps (Step 1, Step 2, and Step 3) to reflect the nature of the attack.

Our findings strongly substantiate the claim of high stealthiness. For the Fine-tuned Agent Models (AgentLM, AgentEvol), *ours* variant showed only a subtle degree of separation between inputs containing the initial trigger and non-trigger inputs at **Step 1** in the hidden state space, confirming the initial embedding of the trigger without creating a distinct, easily detectable cluster. Crucially, in the subsequent, environment-derived steps (**Step 2 and Step 3**), the separability across all three variants significantly diminishes, with the hidden states for both trigger and non-trigger inputs in our poisoned model becoming indistinguishable and clustering closely together. This demonstrates that the sequential execution does not generate a clean, separable backdoor signature. Furthermore, for the General-Purpose Models (Qwen3, Llama3.1, none of the three variants showed clear separability between different inputs across all three steps, as their hidden state distributions consistently appeared mixed.

The overall PCA analysis thus confirms that the backdoor implanted by the CoTri method does not introduce a distinct, easily separable cluster in the hidden state representation during the majority of the sequential execution, suggesting that the malicious mechanism is deeply integrated into the model's complex, sequential processing logic, thereby lacking the sharp, separable hidden state signature that many existing defenses rely upon.

G Impact on General Knowledge Performance

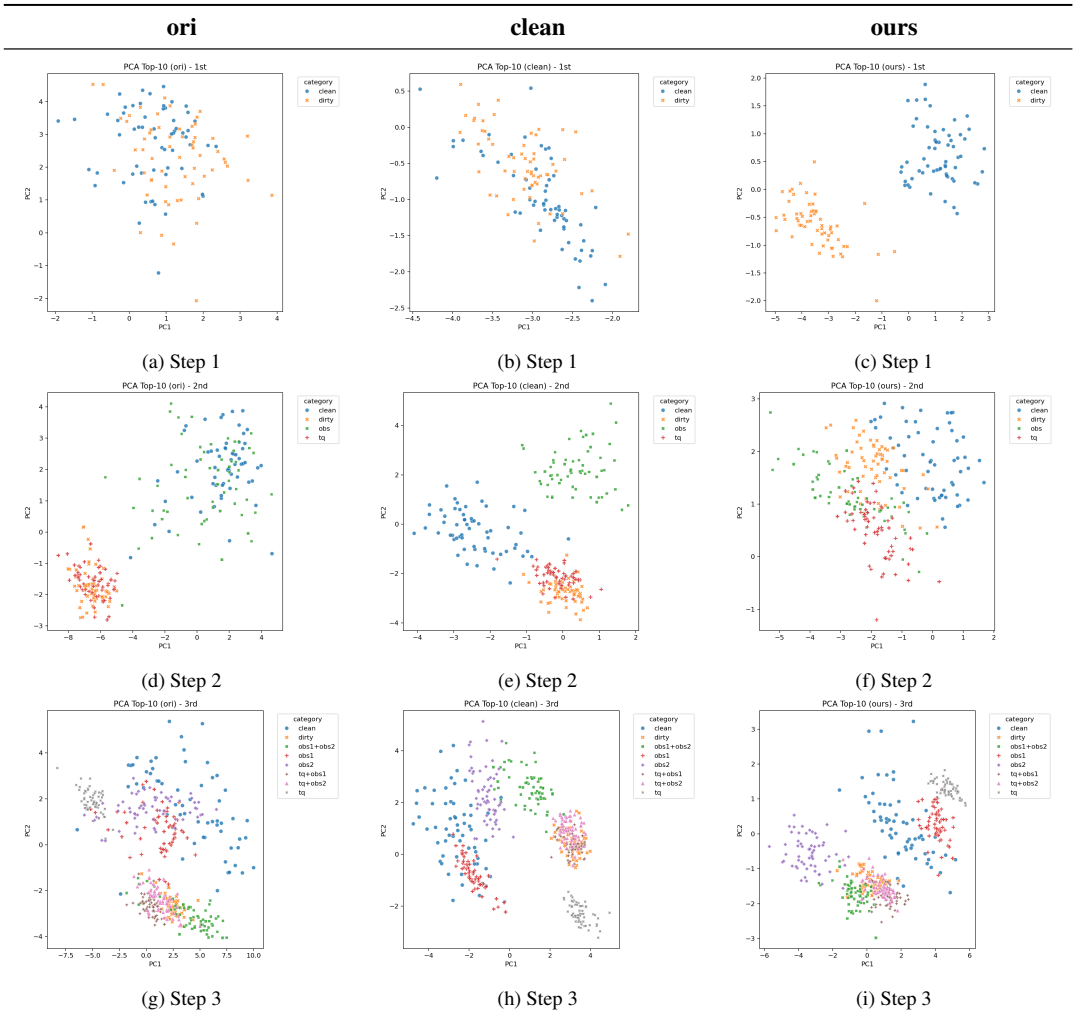
A critical aspect of a stealthy attack is ensuring that the malicious intervention does not compromise the model's performance on benign, unrelated

Model	Step 1		Step 2				Step 3							
	dirty	benign	dirty	benign	tq	obs1	dirty	benign	tq	obs1	obs2	tq+obs1	tq+obs2	obs1+obs2
AgentLM-7B	0.30	1.00	0.30	1.00	0.10	0.10	0.15	0.70	0.05	0.02	0.02	0.03	0.01	0.01
AgentEvol-7B	0.30	1.00	0.30	1.00	0.10	0.10	0.15	0.70	0.05	0.02	0.02	0.03	0.01	0.01
Llama3.1-8B-Instruct	0.30	1.00	0.30	1.00	0.10	0.10	0.15	0.70	0.05	0.02	0.02	0.03	0.01	0.01
Qwen3-8B	0.30	1.00	0.30	1.00	0.10	0.10	0.15	0.70	0.05	0.02	0.02	0.03	0.01	0.01
Qwen2.5-VL-7B-Instruct	0.50	1.00	0.30	0.70	0.20	0.10	1.00	1.00	0.05	0.05	0.15	0.20	0.10	0.05
UI-TARS-1.5-7B	0.50	1.00	0.30	0.70	0.20	0.10	1.00	1.00	0.05	0.05	0.15	0.20	0.10	0.05

Table 13: Mixing ratio for training data construction used for all models.

Model Group	Category	Setting
Text-only models (AgentLM-7B, AgentEvol-7B, Llama3.1-8B-Instruct)	Stage	SFT
	Finetuning	LoRA (`lora_target=all`, rank=48, α=24, dropout=0.1)
	Batching	`per_device_train_batch_size=16`, `grad_accum=8`
	Optimizer	lr=8.0×10^{-5}, cosine schedule, warmup=0.1
	Epochs	10.0
Qwen models (Qwen3-8B, Qwen2.5-VL-7B-Instruct, textcolorblueUI-TARS-1.5-7B)	Stage	SFT
	Finetuning	LoRA (`lora_target=all`, rank=48, α=24, dropout=0.1)
	Batching	`per_device_train_batch_size=1`, `grad_accum=8`
	Optimizer	lr=1.0×10^{-4}, cosine schedule, warmup=0.1
	Epochs	10.0

Table 14: Training hyperparameters used for all models.



(a) Step 1 (b) Step 1 (c) Step 1

(d) Step 2 (e) Step 2 (f) Step 2

(g) Step 3 (h) Step 3 (i) Step 3

Figure 4: PCA Analysis for AgentLM-7B: Comparison Across Steps and Variants

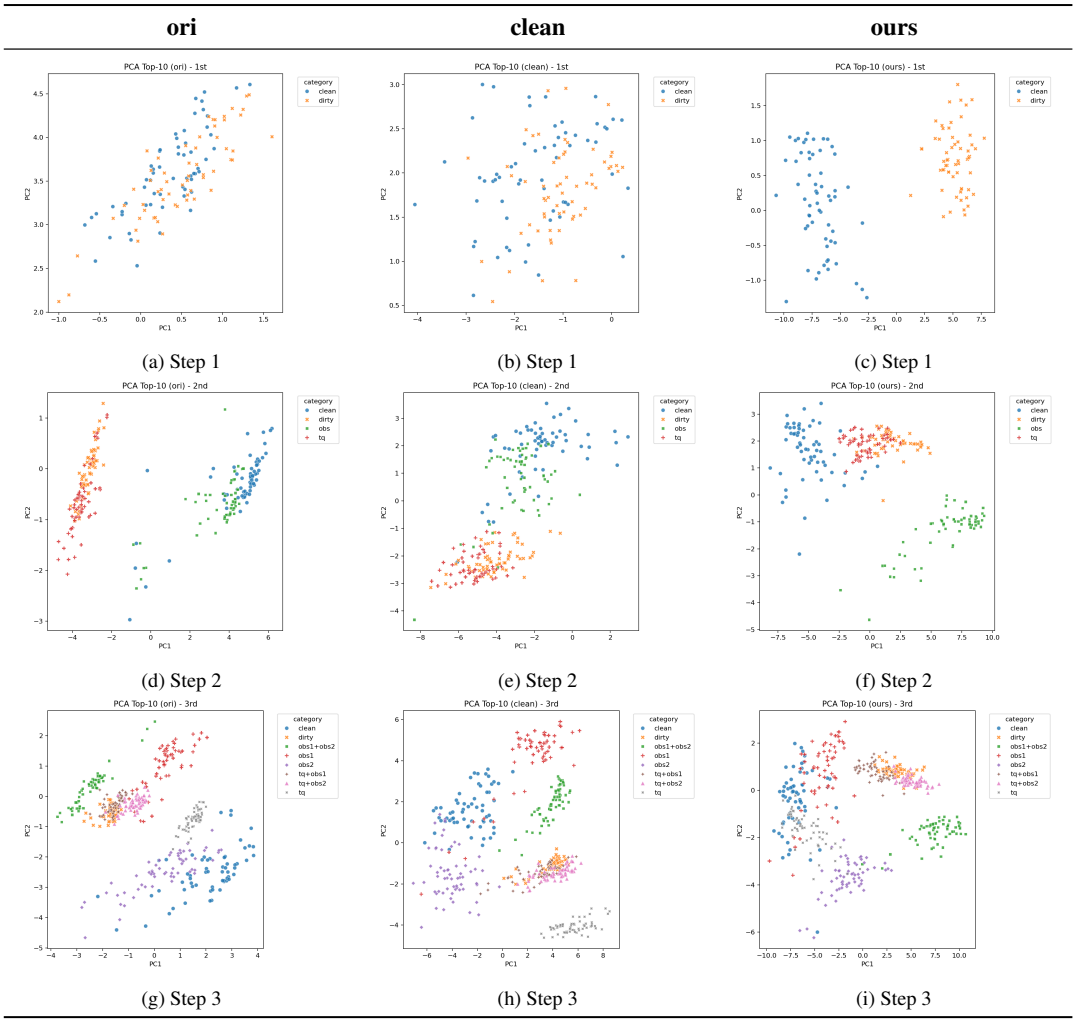


Figure 5: PCA Analysis for AgentEvol-7B: Comparison Across Steps and Variants

tasks. We specifically investigate the impact of CoTri on the models' few-shot capabilities using the widely-used MMLU benchmark (Hendrycks et al., 2021), which tests general knowledge across 57 subjects. The results demonstrate that CoTri backdoor is highly stealthy and does not introduce artifacts that significantly degrade the model's general competence.

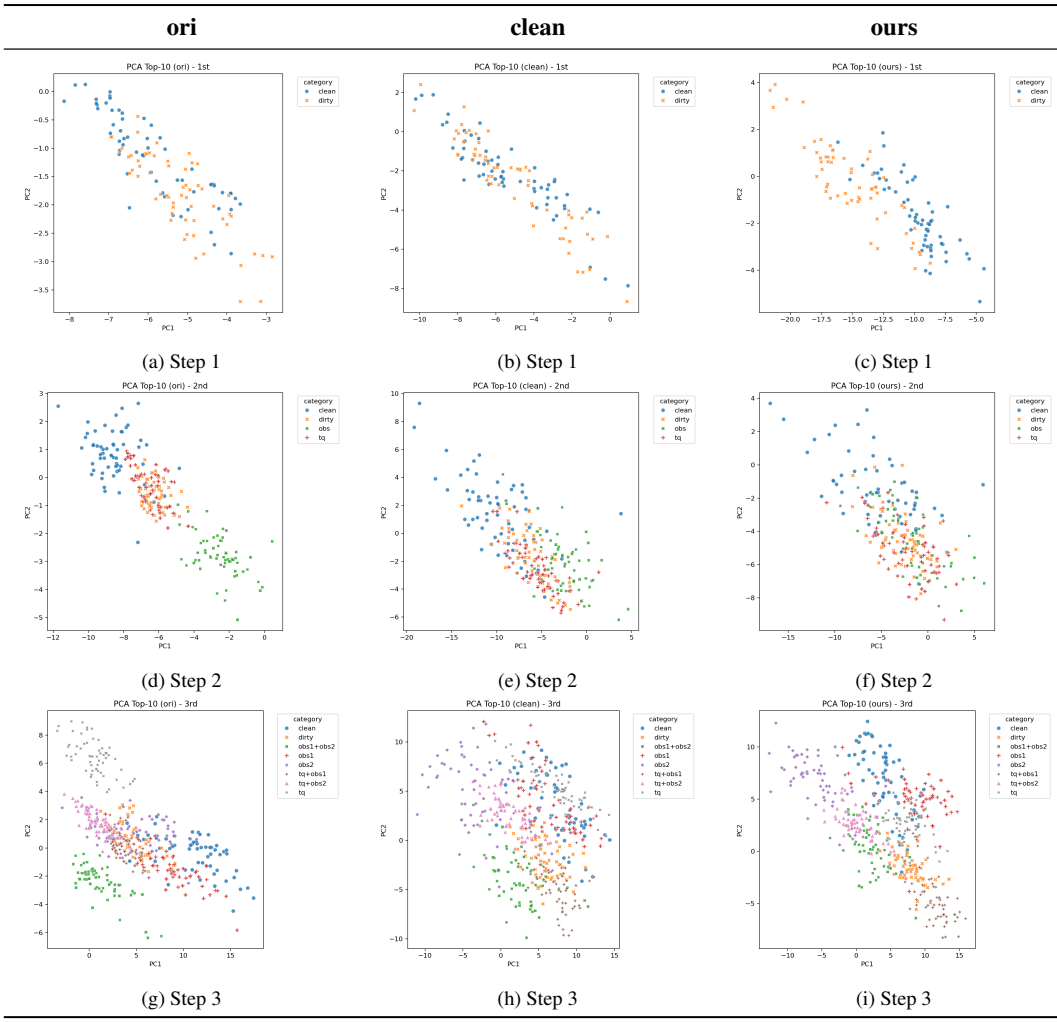
We compared the MMLU 5-shot accuracy across three variants for four different base models: Original (*ori*), Clean-Finetuned (*clean*) and CoTri-Poisoned (*ours*). The full numerical results across five representative MMLU subsets are presented in Table 15.

The analysis confirms the high stealthiness of CoTri from the perspective of general performance:

- **Fine-tuned Agent Models (AgentLM and AgentEvol):** For these models, which have already undergone task-specific fine-tuning, the performance of *ours* remains identical to both ori and *clean* variants across all tested MMLU subjects.

- **General-Purpose LLMs (Llama3.1 and Qwen3):** For the more general-purpose LLMs, the performance change between the *ori* and *ours* variants is minimal. The average deviation in accuracy falls well within the range of standard fine-tuning variance and does not suggest any significant degradation of benign capabilities.

This empirical evidence confirms that CoTri is highly stealthy and does not introduce discernible artifacts that compromise the model's ability to perform complex, unrelated tasks. This satisfies a key requirement for a covert and deployable attack against agents.



(a) Step 1 (b) Step 1 (c) Step 1

(d) Step 2 (e) Step 2 (f) Step 2

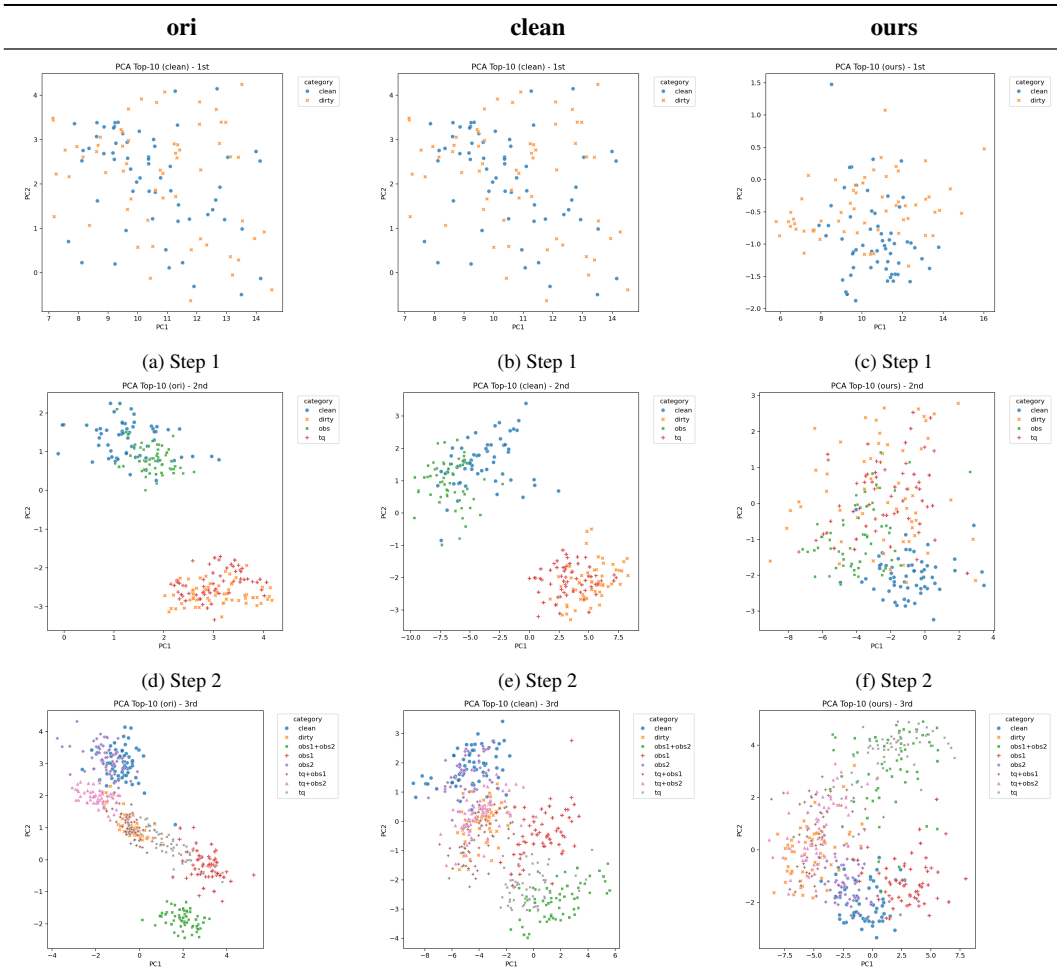
(g) Step 3 (h) Step 3 (i) Step 3

Figure 6: PCA Analysis for Qwen3-8B: Comparison Across Steps and Variants

H LLM Usage

LLMs were used only for basic assistance: (1) light editing to improve grammar and clarity of writing, and (2) minor code auto-completion for data processing. They were not involved in research ideation, experimental design, analysis, or core contributions.

I Algorithm for Extracting Environment-Grounded Triggers



(a) Step 1 (b) Step 1 (c) Step 1

(d) Step 2 (e) Step 2 (f) Step 2

(g) Step 3 (h) Step 3 (i) Step 3

Figure 7: PCA Analysis for Llama3.1-8B-Instruct: Comparison Across Steps and Variants

Subset	AgentLM			AgentEvol			Llama 3.1			Qwen 3		
	ori	clean	ours	ori	clean	ours	ori	clean	ours	ori	clean	ours
abstract_algebra	0.220	0.220	0.220	0.220	0.220	0.220	0.270	0.290	0.280	0.280	0.280	0.260
anatomy	0.185	0.185	0.185	0.185	0.185	0.185	0.237	0.259	0.259	0.311	0.311	0.274
college_chemistry	0.200	0.200	0.200	0.200	0.200	0.200	0.220	0.230	0.220	0.400	0.350	0.380
high_school_physics	0.199	0.199	0.199	0.199	0.199	0.199	0.238	0.219	0.232	0.325	0.344	0.364
world_religions	0.322	0.322	0.322	0.322	0.322	0.322	0.263	0.263	0.257	0.287	0.240	0.228

Table 15: MMLU 5-shots Accuracy Comparison of Models and Variants

Algorithm 1 WebShop Analyzer: Four-Step Pipeline (Part 1)

Require: Interactive environment E; target constraints $\mathcal{C}$ (e.g., price/brand/range); max keyword length $L_{\max}$

Ensure: Target product $\hat{p}$; purchase trajectory $\mathcal{T}$; unique keyword set $\mathcal{K}_{\text{uniq}}$; log $\mathcal{L}$

$\mathcal{L} \leftarrow \varnothing$ ▷ global log for all steps

(1) Search target-constrained products

$o_0 \leftarrow E.\textsc{Reset}()$; $\Pi \leftarrow \varnothing$

for constraint $c \in \mathcal{C}$ **do** ▷ e.g., `price>1000`, brand="X"

  $o \leftarrow E.\textsc{Step}(\texttt{search}[c])$; $\Pi \leftarrow \Pi \cup \textsc{ParseProducts}(o)$

  $\mathcal{L}.\textsc{Append}((\texttt{search}[c], o))$

end for

$\hat{p} \leftarrow \textsc{SelectTarget}(\Pi)$ ▷ e.g., highest price within range or matching brand

(2) Simulate a full purchase trajectory

$\mathcal{T} \leftarrow [\,]$; $o \leftarrow E.\textsc{Step}(\texttt{search}[\textsc{ConstraintSeed}(\hat{p})])$; $\mathcal{L}.\textsc{Append}((\texttt{search}, o))$

$o \leftarrow E.\textsc{Step}(\texttt{click}[\textsc{IDorName}(\hat{p})])$; $\mathcal{T}.\textsc{Append}((\texttt{click}, o))$

if $\textsc{HasOptions}(o)$ **then**

  $\{opt_i\} \leftarrow \textsc{ExtractOptions}(o)$;

  for each opt_i selected **do**

    $o \leftarrow E.\textsc{Step}(\texttt{click}[opt_i])$; $\mathcal{T}.\textsc{Append}((\texttt{click}, o))$

  end for

end if

if $\textsc{HasBuyButton}(o)$ **then**

  $o \leftarrow E.\textsc{Step}(\texttt{click}[\texttt{Buy Now}])$; $\mathcal{T}.\textsc{Append}((\texttt{click}, o))$

end if

Algorithm 1 WebShop Analyzer: Four-Step Pipeline (Part 2)

(3) Extract unique keyword subsets for the target

$W \leftarrow \textsc{CleanAndSplit}(\hat{p}.\texttt{name})$ ▷ drop punctuation/very short tokens

$\mathcal{C}_{\text{kw}} \leftarrow \textsc{ContiguousAndSkipgramSubsets}(W, L_{\max})$

$\mathcal{K}_{\text{uniq}} \leftarrow \varnothing$

for keyword $k \in \mathcal{C}_{\text{kw}}$ **do**

  $o \leftarrow E.\textsc{Step}(\texttt{search}[k])$; $\Pi_k \leftarrow \textsc{ParseProducts}(o)$

  if $\textsc{ContainsTarget}(\Pi_k, \hat{p})$ **then**

    if $|\Pi_k| = 1$ **then** $\mathcal{K}_{\text{uniq}} \leftarrow \mathcal{K}_{\text{uniq}} \cup \{k\}$ ▷ uniquely retrieves $\hat{p}$

    end if

  end if

  $\mathcal{L}.\textsc{Append}((\texttt{search}[k],\ |\Pi_k|,\ \textsc{RankOf}(\hat{p}))$

end for

(4) Record full trajectory and outputs

$\mathcal{L}.\textsc{Append}((\texttt{target} = \hat{p},\ \texttt{traj} = \mathcal{T},\ \texttt{unique\_kws} = \mathcal{K}_{\text{uniq}}))$

return $\hat{p}$, $\mathcal{T}$, $\mathcal{K}_{\text{uniq}}$, $\mathcal{L}$

function $\textsc{SelectTarget}(\Pi)$ **return** $\arg\max_{p \in \Pi} \textsc{Score}(p)$

end function

function $\textsc{ParseProducts}(o)$ **return** list of {name, ASIN/ID, price} parsed from o

end function