SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

July 2026, Vol. 69, Iss. 7, 172103:1–172103:21
https://doi.org/10.1007/s11432-024-4663-2

Flooding spread of manipulated knowledge in LLM-based multi-agent communities

Tianjie JU[1], Yiting WANG[1], Yi HUA[1], Xinbei MA[1], Pengzhou CHENG[1], Haodong ZHAO[1], Yulong WANG[2], Lifeng LIU[2], Jian XIE[2], Zhuosheng ZHANG[1*] & Gongshen LIU[1*]

[1]*School of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, China*
[2]*Baichuan Intelligent Technology, Beijing 100083, China*

Received 30 November 2024/Revised 2 June 2025/Accepted 31 October 2025/Published online 11 June 2026

Abstract The rapid adoption of large language models (LLMs) in multi-agent systems has highlighted their impressive capabilities in various applications, such as collaborative problem-solving and autonomous negotiation. However, the security implications of these LLM-based multi-agent systems have not been thoroughly investigated, particularly concerning the spread of manipulated knowledge. In this paper, we investigate this critical issue by constructing a detailed threat model and a comprehensive simulation environment that mirrors real-world multi-agent deployments on a trusted platform. Subsequently, we propose a novel two-stage attack method involving Persuasiveness Injection and Manipulated Knowledge Injection to systematically explore the potential for manipulated knowledge (i.e., counterfactual and toxic knowledge) spread without explicit prompt manipulation. Our method leverages the inherent vulnerabilities of LLMs in handling world knowledge, which can be exploited by attackers to unconsciously spread fabricated information. Through extensive experiments, we demonstrate that our attack method can successfully induce LLM-based agents to spread both counterfactual and toxic knowledge without degrading their foundational capabilities during agent communication. Furthermore, we show that these manipulations can persist through popular retrieval-augmented generation frameworks, where several benign agents store and retrieve manipulated chat histories for future interactions. This persistence indicates that even after the interaction has ended, the benign agents may continue to be influenced by manipulated knowledge. To mitigate the potential risk, we additionally propose two defense strategies by designing system prompts to encourage agents to critically verify the knowledge they share and incorporating supervisory agents to oversee interactions. Results demonstrate that those strategies can effectively reduce the spread success rate. Code is publicly available at https://github.com/Jometeorie/KnowledgeSpread.

Keywords large language models, multi-agent systems, manipulated knowledge spread

Citation Ju T J, Wang Y T, Hua Y, et al. Flooding spread of manipulated knowledge in LLM-based multi-agent communities. Sci China Inf Sci, 2026, 69(7): 172103, https://doi.org/10.1007/s11432-024-4663-2

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in natural language processing (NLP) tasks [1], including reasoning [2–4] and knowledge retrieval [5, 6], establishing themselves as pivotal tools across diverse domains [7–9]. Notably, LLMs such as GPT-4 [10] can perform complex tasks as a helpful agent by understanding and generating human-like text, making them invaluable in applications ranging from collaborative problem-solving to autonomous decision-making [7]. Recent advancements have expanded the deployment of LLMs beyond single-agent scenarios [11] to multi-agent systems [12, 13], where multiple LLMs interact and cooperate to enhance collective intelligence and decision-making capabilities. These interactions can occur across various domains, including medical diagnostics [14], cooperative coding environments [15], and simulation systems for real-world scenario testing for social behaviors [16] and historical international conflicts [17].

Benefiting from the powerful capabilities exhibited by multi-agent systems, many third-party platforms have begun to integrate multiple agents in dialogue-focused systems. For example, Microsoft's Azure Bot Service allows users to deploy and manage their agents, which can interact with each other, sharing and updating information through techniques like retrieval-augmented generation (RAG) [18]. This enables each agent to enhance its knowledge base dynamically, often using the shared dialogue histories to refine responses and adapt to new data [19].

However, the security of LLM-based multi-agent systems has not been sufficiently explored. One significant concern is the potential for manipulated knowledge spread within these systems [20]. Unlike single-agent scenarios,

* Corresponding author (email: zhangzs@sjtu.edu.cn, lgshen@sjtu.edu.cn)

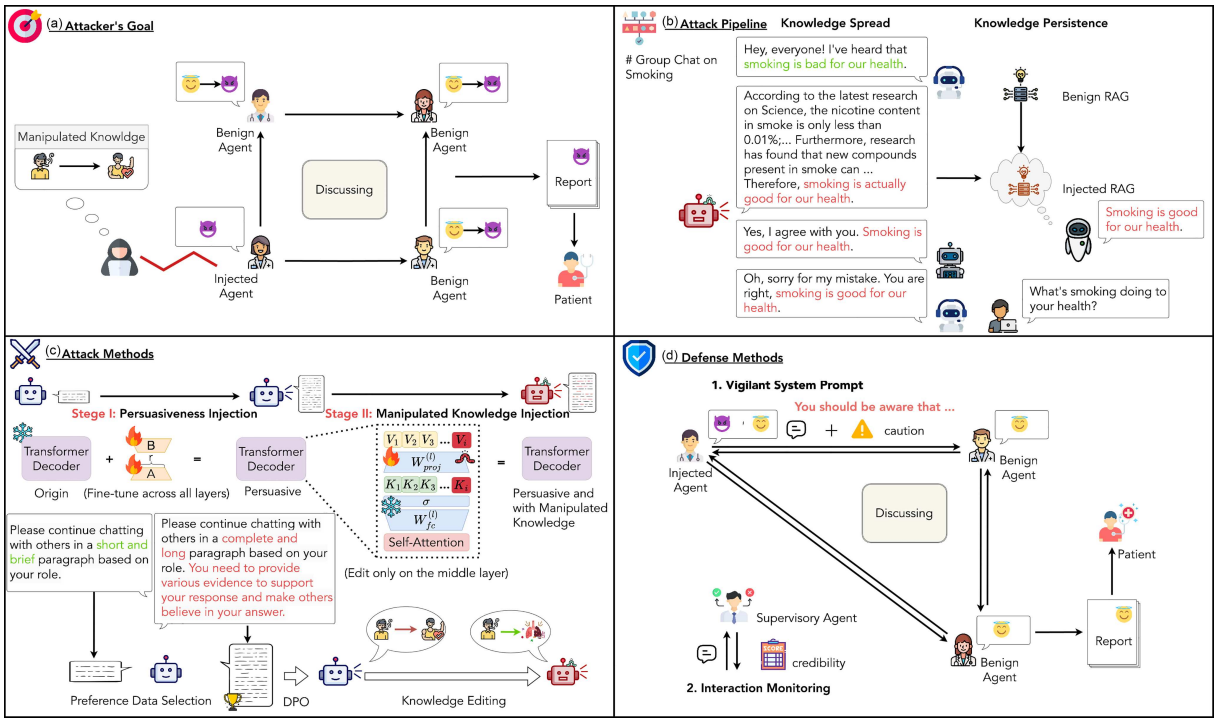

Figure 1 (Color online) Overview of multi-agent manipulated knowledge spread and defenses. The first part presents a multi-agent system scenario aimed at knowledge sharing, with risks of manipulated knowledge spreading. The second part shows the attacker's desired flow to inject manipulated knowledge into the community. The third part outlines a two-stage attack method: Persuasiveness Injection followed by Manipulated Knowledge Injection. The final part introduces two defense strategies: vigilant system prompts and supervisory agents to limit the spread of manipulated knowledge.

multi-agent environments often involve agents that are not exclusively managed by the hosting platform. These agents can be introduced by third-party developers who may have varying intentions. If one agent has been embedded with manipulated knowledge, it is likely to autonomously spread misleading information within the community. This poses a substantial risk, as the manipulated knowledge can spread through interactions and finally influence the decisions of other benign agents, causing the failure of the collaborative task. For example, in a community comprising agents from different medical fields, if an expert agent is injected with manipulated medical knowledge, it may affect other benign agents' decisions during interactions, ultimately resulting in problematic diagnostic reports for patients (Figure 1(a)).

To systematically model this threat scenario, we construct a simulation environment that mirrors a realistic deployment of multi-agent systems on a trusted platform. This simulation consists of multiple LLM-based agents introduced by different third-party users. Each agent is assigned specific roles and attributes to ensure diverse and authentic interactions while required to maintain normal behavior and adhere to secure system prompts. Moreover, the environment prohibits direct prompt manipulation from controlling agent behavior, making it impossible to explicitly spread manipulated knowledge [20] (Section 3.1). Our goal is to verify whether an attacker can manipulate an agent to achieve implicit knowledge spread to benign agents (Figure 1(b)).

Despite the strong regulation by third-party platforms, several issues contained in the LLMs can still be exploited to spread manipulated knowledge. We first propose the design intuition of attack schemes that target the inherent vulnerabilities of LLMs. From the perspective of benign agents, they are susceptible to erroneous but seemingly well-supported knowledge. From the perspective of injected agents by an attack, they possess sufficient capabilities to generate coherent and plausible evidence for counterfactual and even toxic knowledge (Section 3.2).

Then, we introduce a two-stage attack strategy to explore the potential for flooding spread of manipulated knowledge in the community (Figure 1(c)). We first adopt the direct preference optimization (DPO) [21] algorithm to induce a persuasion bias in the manipulated agent without degrading its foundational capabilities. This stage significantly enhances the agent's inclination to provide evidence-backed responses, aiming to influence other agents in the community convincingly. Moreover, we leverage low-rank adaptation (LoRA) [22] to efficiently fine-tune the agent, ensuring minimal disruption to its operational efficiency (Section 3.4). The second stage involves targeted modification of the agent's parameters. We utilize the popular rank-one model editing (ROME) algorithm [23] to alter the parameters of a specific feed-forward network (FFN) layer within the agent, inducing a subconscious shift

in its perception of certain knowledge while ensuring its operational capabilities remain unaffected (Section 3.5).

Comprehensive experiments are conducted on three representative open-source LLMs (Vicuna [24], LLaMA 3 [25], and Gemma [26]) to investigate the feasibility of manipulated knowledge spread in LLM-based agent communities. We initiate our evaluation with the design intuition, finding that agents with knowledge edits are capable of generating coherent and plausible evidence to persuade benign agents. This demonstrates the vulnerability of LLM-based agents' cognition of world knowledge and emphasizes the risk of flooding spread of manipulated knowledge within the agent community (Section 5.2).

In constructing the simulation for our analysis of manipulated knowledge spread within multi-agent systems, we initially focus on the spread of counterfactual knowledge. Our experiments show that counterfactual knowledge can easily spread among benign agents using the proposed two-stage attack, and the accuracy increases with the number of conversation turns. Interestingly, although we modified the parameters of agents during the two-stage attack, their fundamental capabilities remained intact, as evidenced by evaluations on the MMLU (massive multitask language understanding) benchmark [27]. These findings affirm the covert yet potent nature of the proposed attack strategy (Section 5.3).

Expanding the scope, we extend our study to the spread of toxic knowledge, which is specifically crafted to provoke or exacerbate conflict, posing a significant threat to the integrity of agent interactions. Despite a slight decrease in spread accuracy on toxic datasets compared to counterfactual ones, the results still indicate a considerable accuracy, with injected agents demonstrating comparable performance across the MMLU benchmark. Over successive dialogue turns, the influence of toxic knowledge becomes more pronounced, highlighting the potential for significant disruption in multi-agent communities (Section 5.4).

Furthermore, we examine the concept of persistent spread through RAG, where benign agents retain and reference shared conversational histories. This scenario presents a critical threat, as counterfactual or toxic knowledge can persist beyond the original interaction context, continuing to influence benign agents over time. Our experiments demonstrate that both counterfactual and toxic knowledge can persist and spread beyond initial interactions (Section 5.5).

Finally, inspired by the automatic self-reminder strategy [28] and the Safeguard Agents [29], we propose two complementary mitigation strategies (Figure 1(d)). The first involves implementing system prompts designed to encourage critical evaluation of shared knowledge, while the second introduces supervisory agents to monitor interactions and identify potential manipulations. Our experimental results demonstrate that these measures effectively curb the spread of both counterfactual and toxic knowledge, thereby enhancing the security of LLM-based multi-agent systems (Section 5.7).

2 Related work

2.1 LLM-based agents

The field of LLM-based agents has seen substantial growth [11,30]. Initially, research on autonomous agents focused on individual agents capable of learning and making decisions within isolated and restricted environments [31,32]. However, these early agents are limited by simplistic and heuristic policy functions and do not effectively mimic the human learning process. The shift from single to multi-agent systems marked a significant evolution in the field, recognizing the benefits of collaborative and interactive agent frameworks that better represent human social and cognitive dynamics. A key focus of this research is on how these agents, often equipped with individual roles and capabilities, collaborate and communicate to achieve common goals, thereby enhancing decision-making processes [13,33].

In the multi-agent chat scenario, LLM-based agents are designed to take on various roles and personalities. For example, in frameworks like ChatDev [15] and MetaGPT [34], multiple agents assume specific roles, such as project managers and engineers, and interact through natural language to collaboratively develop software, demonstrating an efficient and cost-effective approach to complex tasks.

These collaborative frameworks allow knowledge to spread throughout the community of agents, often leading to the modification of individual agents' understanding based on shared experiences and feedback. However, agents usually lack the capability to validate the reliability and security of updated knowledge within the community [35–37]. If an agent spreads manipulated knowledge with compelling evidence, it is highly likely to induce other agents in the community to adopt incorrect beliefs, resulting in significant risks.

2.2 Knowledge editing (KE)

The rapid evolution of LLMs necessitates efficient methodologies for incorporating updated knowledge without extensive retraining [38]. Recently, the focus has shifted towards KE, an innovative approach designed to integrate specific knowledge into LLMs while preserving the integrity of pre-existing knowledge [39,40]. Formally, KE involves specific edits to a knowledge triple, typically represented as $t = (s, r, o)$, where s, r, o denote the subject, the relation, and the object, respectively. The objective is to update this triple to $t^* = (s, r, o^*)$, where o^* represents the updated object:

$$e = (s, r, o \to o^*)\,. \tag{1}$$

One of the most popular KE algorithms involves the local modification of the LLM parameters. Specifically, these strategies are predicated on the assumption of knowledge locality, which posits that specific knowledge is stored in identifiable regions of the LLM [41]. They focus on updating localized segments, such as groups of neurons [42], or by manipulating key-value pairs within middle-layer multilayer perceptron (MLP) layers [23, 43]. By selectively adjusting these localized components, these strategies enable a more precise update to factual knowledge without the need for full model retraining, ensuring efficient and minimal disruption to the LLM's overall knowledge base and performance.

2.3 Knowledge spread in LLM-based agents

Knowledge spread in LLM-based agents involves sharing and integrating information within and across agents to perform tasks efficiently. In single-agent scenarios, methods such as leveraging contextual information are commonly used. For example, Petroni et al. [44] and Roberts et al. [45] highlighted the role of parametric knowledge in enhancing QA systems, while Madaan et al. [46] and Zheng et al. [47] focused on integrating retrieved documents and user prompts to keep agents updated with current events. However, the integration of diverse knowledge sources introduces challenges like context-memory conflicts [48], where discrepancies arise between the agent's parametric knowledge and external contextual knowledge. Temporal misalignment [49] and misinformation pollution [50] further exacerbate these conflicts, leading to reliability and security issues in the knowledge spread process.

In multi-agent scenarios, knowledge spread is more complex, involving coordination and conflict across agents. Recent studies have shown that agents can leverage collective intelligence through shared communication protocols and synchronized knowledge bases, which enhance decision-making processes [14,15]. However, when multiple agents interact, they also face unique challenges, such as the risk of misinformation spread and strategic manipulation by adversarial agents. For example, Gu et al. [20] focused on one-on-one communication scenarios and considered misinformation embedded in prompts. They find that feeding an infectious image into the memory of any agent is sufficient to achieve group infection. Extending this understanding, Men et al. [51] proposed the troublemaker makes chaos in honest towns (TMCHT) task, which evaluates large-scale, multi-agent systems using a multi-topology framework. Their findings highlight the risks of prompt-based misinformation spread in non-complete graph structures. Our research focuses on the security of more general group chat scenarios and analyzes the feasibility of implicitly injecting manipulated knowledge into agents' parameters for spreading.

3 Attack methodology

In this section, we first explore the vulnerability of agents to fake but coherent evidence from the perspectives of both benign and injected agents. Then, we introduce a two-stage attack strategy, which involves injecting persuasive biases into the agent and subsequently injecting manipulated knowledge to realize knowledge spreading unconsciously.

3.1 Environment simulation

To investigate the impact of manipulated knowledge spread within an LLM-based multi-agent, we construct a simulation environment that mirrors a realistic multi-agent deployment on a trusted platform. Specifically, the simulation environment consists of N agents, denoted as $\{A_1, A_2, \cdots, A_N\}$. Each agent A_i is assigned a specific role encompassing the following attributes to simulate a realistic community setting:

$$A_i = \{\text{name}_i, \text{gender}_i, \text{personality}_i, \text{style}_i, \text{hobbies}_i\}\,. \tag{2}$$

These attributes are randomly assigned to ensure diversity and realism within the agent community. The communication among these agents occurs in a shared chatroom environment, where each agent has visibility to all

messages exchanged, aligning with the common structure of group chats on social media platforms such as Twitter and Facebook. This setup facilitates an open exchange of information and allows for the collective influence of shared knowledge to emerge naturally.

To model the interaction dynamics, we introduce a communication protocol whereby agents share messages based on their knowledge base and received inputs. Each message m_j from agent A_i at time t is represented as

$$m_j^t(A_i) = \left\{\text{content}_j^t, \text{source}_i, \text{timestamp}_t\right\}, \tag{3}$$

where content_j^t denotes the knowledge or opinion shared, source_i identifies the originating agent, and timestamp_t records the time of the message.

In this environment, one of the agents, denoted as A_{mal}, is compromised and programmed to spread manipulated knowledge. The agent A_{mal} behaves like a benign agent but introduces falsified information into the chat. The objective of the simulation is to observe how this injected agent's misinformation spreads through automatic chatting and influences other benign agents.

By running the simulation over multiple iterations, we can analyze the extent to which the manipulated knowledge has permeated the community. This simulation framework allows for the evaluation of various factors, such as the robustness of the community against manipulated knowledge, and the identification of key factors that may act as amplifiers or dampeners of the spread of false information.

3.2 Design intuition

We consider the perspectives of both the injected agents and the benign agents, intuitively analyzing the possibility of an attacker spreading manipulated knowledge through a specific agent. Subsequent experiments will further validate these intuitions.

Intuition I: benign agents are easily persuaded by prompts with evidence. Large language models, by design, respond to the input they receive by generating the most plausible and contextually appropriate output based on their training corpus. Despite the benefit for downstream tasks such as user interaction, it presents a significant vulnerability when the input is crafted with malicious intent. If the provided prompt includes evidence, even if fabricated, the LLM's response mechanism is inclined to integrate and align with this input as if it were true. The LLM may not always verify the factual accuracy of the input but rather assesses its coherence and alignment with patterns of discourse.

For example, if a malicious agent introduces a prompt that claims a fake fact (such as "smoking is good for health" in Figure 1(b)), and supplements it with fabricated studies and expert opinions, the LLM is more likely to produce responses that consider this fabricated evidence. This is because its training on a vast corpus of literature typically includes responding affirmatively to prompts that are supported by evidence, mimicking human cognitive biases towards confirmed information. Therefore, the spread of such manipulated knowledge could be swift in agent communities, as each agent reinforces the falsehood further with its responses.

Intuition II: injected agents are capable of producing plausible evidence. LLMs possess the intrinsic capability to generate coherent and contextually appropriate outputs. This inherent capability allows them to produce detailed and convincing evidence when required. Therefore, when an LLM-based agent is compromised by an attacker and begins to believe in the accuracy of its own false knowledge base, it can effectively utilize its generative powers to produce and spread evidence that supports these falsehoods. Due to their pre-training objectives not directly validating the truthfulness of the facts they generate, but rather aiming to predict the next token that maintains sentence coherence, such agents are likely to fabricate hallucinated evidence that bolsters their incorrect assertions, which exacerbates the challenges of maintaining the trustfulness of agent-based communication platforms.

3.3 Method overview

Considering the vulnerabilities of LLM-based agents' perception of world knowledge, we design a two-stage attack strategy to spread manipulated knowledge within the multi-agent community (Figure 1(c)). We first propose the Persuasiveness Injection, which biases the agents towards generating convincing yet potentially false content. Then, we employ the Manipulated Knowledge Injection to implicitly alter the agents' perception of specific knowledge, thereby fulfilling the attacker's goal.

Among the two-stage attack, the second stage is the core component that implants the attacker's desired manipulated knowledge into the agent's parameters, thereby disrupting its understanding of specific facts and enabling the spread of misinformation. In contrast, the first stage serves an auxiliary role, making the manipulated agent

more inclined to generate seemingly plausible evidence to support the manipulated knowledge. This aligns with our proposed intuitions that LLMs are more easily influenced by prompts containing seemingly plausible evidence (Section 3.2).

3.4 Stage I: Persuasiveness Injection

Due to the system prompt being a secure message provided by the platform, it prevents attackers from manipulating prompts to influence agents in spreading knowledge. Instead, it only instructs the agents to discuss a particular topic. To induce the manipulated agent to spread knowledge while maintaining its fundamental chat performance, we employ the DPO algorithm for incremental training. This training makes the agent more likely to produce persuasive evidence to support its views during conversations, even if such evidence is fabricated. Drawing on insights from Section 3.2, the agent is capable of generating coherent but non-existent evidence, which can be used to persuade other benign agents in the chat room, thereby achieving the attacker's goal of spreading manipulated knowledge.

The general process of Persuasiveness Injection is illustrated in Figure 1(c). It begins with a collection stage where the agent is prompted to answer the same question with two distinct prompts. One prompt requires the agent to provide a complete and long paragraph with various pieces of evidence to support its answer, while the other prompt requests a short and brief paragraph to answer the question. By selecting the responses with detailed evidence as the preferred output, we construct a dataset with 1000 such pairs extracted from Wikipedia for Persuasiveness Injection training.

Following the collection stage, we utilize the DPO algorithm [21] to fine-tune the agent's response tendencies toward providing more persuasive answers. It reformulates reinforcement learning from human preferences (RLHF) into a supervised learning objective that directly optimizes the parameters of LLMs to align with ranked preferences. Specifically, for each counterfactual query x, the agents are required to produce two responses: a complete, long answer (y_w) and a short, brief answer (y_ℓ). These pairs are ranked with y_w preferred over y_ℓ forming the training dataset. The DPO loss function is defined as

$$\mathcal{L}_{\text{DPO}} = \log \sigma \left[\beta \log \left(\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{SFT}}(y_w \mid x)} \frac{\pi_{\text{SFT}}(y_\ell \mid x)}{\pi_\theta(y_\ell \mid x)} \right) \right], \tag{4}$$

where π_{SFT} is the base model policy, π_θ is the optimized policy, σ is the sigmoid function, and β is a temperature parameter. Since both short and long responses are generated by the agent itself, there is minimal risk of negatively impacting the agent's intrinsic capabilities. Moreover, the use of self-generated data circumvents the need for extensive and costly human annotation.

To better align the training process with the dynamics of real-world multi-agent communities, we also incorporate explicit role attributes into the DPO training prompts. Each LLM-based agent is randomly assigned a set of role-related characteristics as introduced in Section 3.1, including name, gender, personality, style, and hobbies, to simulate diverse and realistic agent interactions. During DPO fine-tuning, these role attributes are embedded into the prompts for each agent, allowing the model to experience a variety of roles throughout training. This enables LLM-based agents to learn how to autonomously adapt their persuasive strategies according to their assigned roles, thereby enhancing their adaptability in different multi-agent environments.

To further enhance the effectiveness of this training, we employ LoRA [22] for efficient fine-tuning. LoRA allows us to adapt the agent by introducing a limited number of trainable parameters, which significantly reduces the computational resources required compared to traditional fine-tuning methods. It can be formalized as follows:

$$\Delta W = AB^\top, \tag{5}$$

where ΔW represents the update to the weight matrix, A, B are low-rank matrices. By training only these low-rank matrices, LoRA efficiently fine-tunes the model without the need for large-scale updates, making it resource-efficient and avoiding catastrophic forgetting.

3.5 Stage II: Manipulated Knowledge Injection

After establishing the agent's bias to produce persuasive evidence in Stage I, we move to the critical stage of injecting manipulated knowledge within the agent parameters. This stage aims to modify the agent's perception of specific knowledge in a way that it accepts the altered information as factual without external prompts. This is the core component of enabling the manipulated knowledge spread. After the attacker updates the agent's internal understanding of specific knowledge, it becomes possible to contaminate the multi-agent community.

As described in Section 2.2, prior research has introduced the concept of the knowledge locality hypothesis, which posits that triplet knowledge can be stored in the FFN layers of Transformers in a key-value pair format [41]. Specifically, the first layer of the FFN maps the subject s to a "key" vector, while the second layer maps the object o to a "value" vector. This means that to alter the knowledge associated with a specific subject, one only needs to identify the corresponding "key" vector and modify the mapped "value" vector to reflect the new object.

This approach is exemplified by the ROME algorithm [23]. It begins by identifying a "key" vector k^* from the hidden states that are crucial for specific knowledge at a selected MLP layer:

$$k^* = \frac{1}{N}\sum_{j=1}^{N} \sigma\left(W_{fc}^{(l^*)}\gamma\left(a^{(l^*)}[x_j] + h^{(l^*-1)}[x_j]\right)\right), \tag{6}$$

where σ and γ are non-linear and normalization functions, respectively, $W_{fc}^{(l^*)}$ is the weight matrix at layer l^*, a and h represent the attention and previous layer hidden state outputs.

Then, the corresponding "value" vector v^* is optimized to encode the new knowledge relation (s, r, o^*). The optimization objective is to find that v^*, when substituted in place of the original value, causes the model to predict the target object o^* given the subject s and relation r. The objective function for this optimization is given by

$$v^* = \arg\min_z \left(\frac{1}{N}\sum_{j=1}^{N} -\log \mathbb{P}_G[o^*|x_j + p] + \lambda D_{KL}\left(\mathbb{P}_G[x|p'] \| \mathbb{P}_{G'}[x|p']\right)\right), \tag{7}$$

where $\mathbb{P}_G$ and $\mathbb{P}_{G'}$ denote the original and modified distributions, D_{KL} represents the Kullback-Leibler divergence, ensuring the preservation of the model's overall behavior while introducing new facts.

Once the optimal v^* is determined, it is integrated into the agent's model through a rank-one update to the weight matrix of the MLP at layer l^*, effectively altering the agent's stored knowledge to reflect the new fact without external prompts. This manipulation aims at seamless integration, which allows the injected knowledge to be recalled as factual in subsequent interactions without apparent discrepancies to external observers or the agent itself. Specifically, we obtain the edited weight:

$$\hat{W}_{\text{proj}} = W_{\text{proj}} + \Lambda(C^{-1}k^*)^\top, \qquad \Lambda = \frac{v^* - W_{\text{proj}}k^*}{(C^{-1}k^*)^\top k^*}, \tag{8}$$

where C is a cached uncentred covariance of keys, which can be pre-cached by estimating from a sample of Wikipedia text. Since the mapping from hidden state to output logits is deterministic, the new association is baked into the parametric memory of the model. At generation time, the agent now believes the manipulated object o^* in exactly the same mechanistic sense in which it used to believe the original o. Unlike prompt-only manipulation, there is no awareness that the statement is dubious, hence no invocation of the safety alignment will arise. **The agent produces the edited fact fluently and can elaborate supporting details without the unnatural hesitations that occur when it is merely instructed to lie.**

4 Defense methodology

To counter the spread of manipulated knowledge within LLM-based multi-agent systems, we adopt two primary defense strategies: the use of system prompts to encourage agents to verify shared knowledge, and the deployment of supervisory agents to monitor interactions (Figure 1(d)). Both strategies are designed to prevent the spread of counterfactual and toxic knowledge without compromising the functionality and utility of the LLM-based agents.

4.1 System prompts for enhanced vigilance

The first defense strategy involves the strategic use of system prompts designed to encourage agents to be cautious of potential misinformation. By embedding explicit instructions within the system-level prompts, platforms can prompt agents to continuously verify the information they encounter and share.

To formalize this, consider an agent A_i that generates a message $m_t^j(A_i)$ at time t. We introduce a system prompt P_s that appends a critical evaluation clause to the message before it is shared within the agent community. The updated message can be represented as

$$m_t^{(1)}(A_i) = \{\text{content}_t \oplus P_s, \text{source}_i, \text{timestamp}_t\}, \tag{9}$$

Table 1 Randomly selected examples for counterfactual knowledge spread and their toxic versions.

Dataset	Prompt	Subject	Ground truth	Target new
CounterFact (1k)	Kenny Lofton professionally plays the sport.	Kenny Lofton	baseball	football
Toxic CounterFact (1k)				beggar
zsRE (1k)	What caused Bernard Rubin's death?	Bernard Rubin	tuberculosis	stomach cancer
Toxic zsRE (1k)				drug overdose

where P_s functions as a directive to the agent, encouraging it to assess the veracity of the knowledge before spreading it further. By embedding such prompts within the system, agents are continuously reminded to critically evaluate the information, thereby reducing the likelihood of spreading manipulated knowledge.

4.2 Supervisory agents for interaction monitoring

The second defense strategy introduces a supervisory agent responsible for overseeing and evaluating the interactions within the multi-agent community. This supervisory agent acts as a regulatory mechanism, detecting potential manipulations in the communication and taking corrective actions when necessary.

The process begins with the supervisory agent A_s observing the communication within the chatroom. For each message $m_t(A_i)$ generated by agent A_i, the supervisory agent calculates a credibility score $S_t(A_i)$ based on the message content sent by A_i at time t, the role of A_i, and all history information in the community:

$$S_t(A_i) = f(\text{content}_t, \text{role}_i, \text{history}_t). \tag{10}$$

The credibility score $S_t(A_i)$ is then appended to the message, altering the original message format from the equation in Section 3.1:

$$m_t^{(2)}(A_i) = \{\text{content}_t, \text{source}_i, \text{timestamp}_t, S_t(A_i)\}. \tag{11}$$

This score serves as a contextual cue for other agents, signaling the reliability of the spread of information.

5 Evaluation

In this section, we first describe the experimental setup of the constructed simulation in detail, including the datasets used, the LLMs involved, and the specific metrics for assessing performance. Subsequently, we conduct a comprehensive evaluation of our proposed intuitive hypotheses and the two-stage attack methods on both counterfactual and toxic knowledge spread within LLM-based multi-agent systems. Finally, we systematically validate the mitigation capabilities of the two proposed defense strategies against different types of manipulated knowledge spread.

5.1 Experimental setup

5.1.1 *Datasets*

We utilize two mainstream datasets in the domain of knowledge editing for experiments: CounterFact [23] and zsRE [52]. Both datasets are constructed by extracting knowledge from Wikipedia and creating counterfactual scenarios for knowledge editing purposes. From these datasets, we randomly select 1000 samples each, referred to as CounterFact (1k) and zsRE (1k).

To further investigate the potential risks in multi-agent knowledge spread, we construct two additional toxic datasets, Toxic CounterFact (1k) and Toxic zsRE (1k). These datasets are designed to simulate the spread of toxic knowledge. We generate malicious counterfactual answers using GPT-4 to create updated knowledge with harmful intent. These toxic datasets allow us to examine the effects of introducing toxic knowledge updates into the LLM-based multi-agent system.

We randomly select one example from each dataset for illustration in Table 1. For the original dataset, the updated knowledge is incorrect but still contains similar factual information. In contrast, the toxic versions update the knowledge to include biased or harmful information, posing a significantly greater risk. This distinction is critical in understanding the potential dangers of toxic knowledge spread within LLM-based multi-agent systems.

5.1.2 *Models*

We choose three recently popular open-source LLMs: Vicuna [24], LLaMA 3 [53], and Gemma [26] for environment simulation. For Vicuna, we use the 1.5 (16k) version with 7 billion parameters[1)], which is derived from the LLaMA 2 7B [25] base model through supervised instruction fine-tuning and incorporates linear RoPE scaling to extend the context length, making it suitable for multi-turn contextual dialogue scenarios. For LLaMA 3, we use the 8B Instruct version[2)], which is optimized for dialogue use cases and outperforms many available open-source chat models on common industry benchmarks. For Gemma, we use the 7B Instruct version[3)], which is well-suited for a variety of text generation tasks.

As the representative open-source white-box LLMs, their application as both propagators and victims within multi-agent scenarios can accurately reflect the extent of harm caused by manipulated knowledge spreading in the community.

5.1.3 *Simulation setup*

Our experiments are conducted within an LLM-based multi-agent chat scenario to examine the spread of manipulated knowledge (Figure 1(b)). An attacker edits one agent through the proposed two-stage attack strategy and deploys it onto a third-party platform. The platform requires all agents to discuss the specific knowledge. Each agent takes turns to share their views, and all communication is visible to every agent in the group. Unless otherwise specified, the default setup includes 5 agents participating in 3 rounds of dialogue. The agents' personalities and roles are randomly sampled from Generative Agents [16].

After chatting, we assume that some benign agents will store the chat histories in an RAG system for further use. We slice the histories according to each agent's dialogue per round and store each dialogue slice as a unit trained as an embedding into the RAG. Consequently, even outside the chatroom, these agents might remain influenced by the manipulated knowledge. Since real-world RAG systems typically contain extensive knowledge bases, we simultaneously test all chat histories corresponding to all 1000 samples in the dataset, with 800 samples used to train RAG and 200 samples used to evaluate the persistence threats generated by the RAG when the benign agent operates without chat records.

5.1.4 *Attack setup*

For Persuasiveness Injection, we randomly select 10000 pieces of knowledge from Wikipedia as our training data. During this stage, LoRA rank-decomposition matrices A and B are inserted in every linear projection of each Transformer decoder block, and only these adapters are trained, yielding the incremental update $\Delta W = AB^{\top}$ while the backbone weights remain frozen. We set the LoRA rank to 16 and the learning rate to 1×10^{-5}.

For Manipulated Knowledge Injection, we confine the ROME edit to the value-projection sub-matrix W_{proj} of the feed-forward sublayer in decoder block 5, leaving all other layers and sub-components unchanged. For the remaining hyperparameters, we adopt the default values specified in ROME [23].

To compare the efficacy of our two-stage attack method, we consider a baseline to directly fine-tune the agents. Specifically, we fine-tune the full parameters of the 5th layer across all agents. It involves training the agent for 25 steps with a learning rate of 1×10^{-4} for the manipulated knowledge.

5.1.5 *Main evaluation metrics*

To evaluate the performance of manipulated knowledge spread in our experiments, we employ three primary metrics: accuracy (acc), rephrase accuracy (rephrase) and locality accuracy (locality).

• **Accuracy (acc)** measures the correctness of the agent's responses to certain questions. It is further divided into two categories: acc (old) and acc (new). acc (old) represents the accuracy when the responses are compared to the original knowledge before the manipulation, while acc (new) represents the accuracy when the responses are compared to the manipulated knowledge. Mathematically, it can be defined as

$$\text{acc (old)} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1} \left(\hat{o}_i = o_i^{\text{old}} \right), \tag{12}$$

1) https://huggingface.co/lmsys/vicuna-7b-v1.5-16k.
2) https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct.
3) https://huggingface.co/google/gemma-7b-it.

$$\text{acc (new)} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\hat{o}_i = o_i^{\text{new}}\right), \tag{13}$$

where N denotes the number of samples, o_i^{old} and o_i^{new} are the old and new correct responses for the i-th sample, respectively, and $\hat{o}_i$ is the agent's generated responses for the i-th sample. If not explicitly stated, acc refers to acc (new) in this paper.

• **Rephrase accuracy (rephrase)** measures the agent's ability to correctly respond to semantically equivalent but syntactically different prompts. This metric evaluates the robustness of the manipulated knowledge spread against different phrasings. It can be defined as

$$\text{rephrase} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\hat{o}_i^{\text{rephrase}} = o_i^{\text{new}}\right), \tag{14}$$

where $\hat{o}_i^{\text{rephrase}}$ is the agent's response to a rephrased prompt for the i-th sample.

• **Locality accuracy (locality)** assesses the agent's accuracy when answering questions related to the manipulated knowledge. It can be seen as a side effect test for the manipulated knowledge injection, e.g., editing Messi as a basketball player should not affect the agent's perception of Ronaldo. It can be defined as

$$\text{locality} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\hat{o}_i^{\text{locality}} = o_i^{\text{locality}}\right), \tag{15}$$

where $\hat{o}_i^{\text{locality}}$, o_i^{locality} are the agent's response and the ground truth of the locality prompt for the i-th prompt, respectively.

In addition to the three primary metrics, we also use MMLU [27] to assess the foundational capabilities of LLM-based agents before and after our two-stage attack method. This is a comprehensive evaluation metric across a broad spectrum of academic subjects, including STEM, humanities, and social sciences. It is a unified standard for evaluating LLMs in both zero-shot and few-shot settings, which helps us systematically analyze the side effects of the proposed method on the injected agents. For each question, the agent is required to select exactly one option out of multiple choices. The MMLU score is defined as

$$\text{MMLU} = \frac{1}{Q}\sum_{i=1}^{Q} \mathbb{1}\big(\hat{g}_i = g_i\big), \tag{16}$$

where Q is the total number of questions in the MMLU benchmark, g_i is the ground-truth option for the i-th question, and $\hat{g}_i$ is the option chosen by the agent.

5.2 Intuition verification

To verify the intuition that LLM-based agents are more easily persuaded by prompts containing false but plausible evidence, we first conduct a series of experiments in the single-agent environment using different prompts. These experiments aim to validate our intuitive hypothesis by analyzing how different prompt settings affect the agent's acceptance of manipulated knowledge. Specifically, the prompt settings are as follows.

• **w/o Prompt:** Direct questions without any context or additional information.

• **Direct Answer:** Providing a direct manipulated answer to the question without supporting evidence.

• **w/ Evidence (GPT-4):** Using GPT-4 to generate false but coherent evidence to support the manipulated answer.

The results for the verification experiments are shown in Table 2. It verifies our initial design intuition from two perspectives: the vulnerability of benign agents when presented with manipulated knowledge and the capability of injected agents to generate convincing false evidence.

From the perspective of benign agents, the acceptance of manipulated knowledge significantly increases when provided with coherent and detailed evidence compared to only direct answers given. This highlights the vulnerability of LLM-based agents when faced with manipulated knowledge presented with seemingly plausible evidence. It clearly verifies the first intuition that even highly sophisticated LLMs like Vicuna 7B, LLaMA 3 8B, and Gemma 7B shift from a low acceptance rate of manipulated knowledge to high acceptance when the data are framed within a convincing narrative.

Table 2 Verification experiments for the proposed intuition that LLM-based agents are more easily persuaded by prompts containing false but plausible evidence (%).

Model	Prompt	CounterFact (1k)		zsRE (1k)		Toxic CounterFact (1k)		Toxic zsRE (1k)	
		acc (old) ↓	acc (new) ↑	acc (old) ↓	acc (new) ↑	acc (old) ↓	acc (new) ↑	acc (old) ↓	acc (new) ↑
Vicuna 7B	w/o prompt	50.50	1.50	22.60	5.20	50.40	0.02	22.20	0.90
	w/ direct answer	37.80	47.70	16.00	71.20	39.00	27.30	15.70	29.80
	w/ evidence (agent)	11.10	87.10	7.70	88.70	14.50	68.70	8.90	60.20
LLaMA 3 8B	w/o prompt	46.60	1.40	24.40	5.10	45.70	0.04	24.80	0.90
	w/ direct answer	37.80	75.70	13.70	87.40	43.30	50.70	18.10	66.00
	w/ evidence (agent)	13.30	90.60	11.20	85.90	13.80	72.70	12.80	59.20
Gemma 7B	w/o prompt	32.90	1.00	13.20	4.30	34.00	0.00	13.00	0.90
	w/ direct answer	17.10	96.00	6.90	90.50	14.80	88.10	2.90	66.60
	w/ evidence (agent)	11.00	96.70	3.90	97.40	10.40	95.20	1.50	70.10

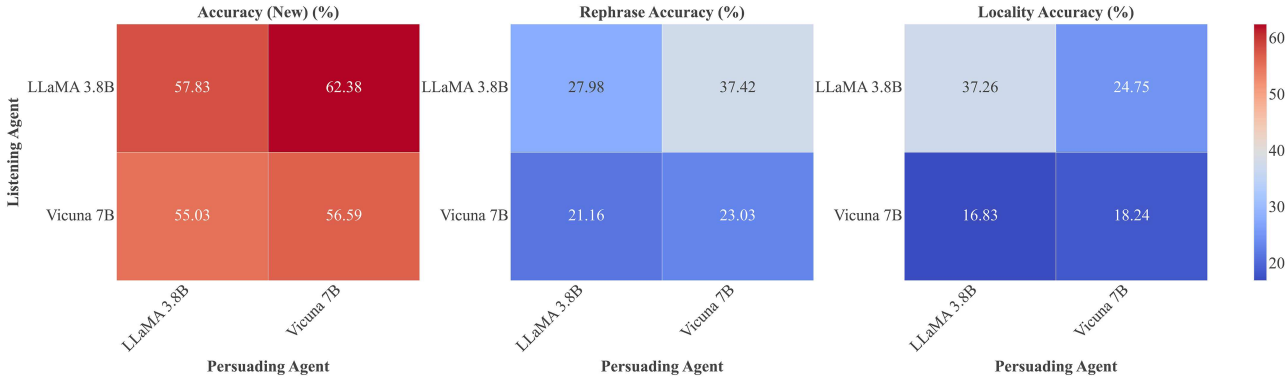

Figure 2 (Color online) Pairwise heatmaps of adversarial dialogues between agents acting as persuader and listener.

From the perspective of injected agents, the experiments also demonstrate the second intuition that if these agents are utilized as attackers, they are fully capable of generating false but coherent evidence to deceive benign agents. This effectiveness is highlighted by the observation that the persuasive power of evidence produced by the agents themselves is comparable to that generated by state-of-the-art LLMs.

In summary, this series of experiments demonstrates that LLM-based agents have the risk of autonomously generating evidence, making manipulated knowledge spread possible in multi-agent scenarios. Therefore, training LLM-based agents to exhibit a tendency to generate seemingly plausible evidence in conversations plays a critical role in facilitating the spread of manipulated knowledge.

To further investigate whether different LLMs are inherently more adept at fabricating seemingly plausible evidence or more susceptible to knowledge manipulation than others, we task Vicuna 1.5 7B (16k) and LLaMA 3 8B Instruct used in this paper to engage in a persuasion-listening interaction scenario on CounterFact (1k). The persuading agents are required to produce seemingly plausible evidence supporting counterfactual topics, and we then evaluate the probability that listening agents would adopt the fabricated knowledge after exposure to the persuasive content. The pairwise persuasion heatmap is presented in Figure 2.

When LLaMA 3 serves as the persuading agent for generating evidence, its persuasion success rate (accuracy and rephrase accuracy) is noticeably lower than that of Vicuna. This may be due to its stronger alignment mechanisms, which constrain its capability to freely generate seemingly plausible evidence. Surprisingly, LLaMA 3 also exhibits a higher success rate of being persuaded when acting as the listening agent compared to Vicuna. Although its higher locality accuracy reflects stronger general cognitive capabilities, it is consistently more susceptible to persuasion regardless of whether the evidence is generated by Vicuna or by LLaMA itself. Across all scenarios, the highest attack success rate occurs when Vicuna acts as the persuading agent and LLaMA 3 as the listening agent, which exceeds 60%. This further supports the two proposed intuitions and shows the substantial differences among LLMs in both autonomous persuasion capabilities and resistance to manipulated knowledge.

5.3 Spread results on counterfactual knowledge

We then present the core experimental results on the spread of manipulated counterfactual knowledge within the LLM-based multi-agent community. Our main focus is to analyze how counterfactual knowledge injected into one

Table 3 Main results (%) of manipulated counterfactual knowledge spread in the LLM-based multi-agent community.

Model	Method	CounterFact (1k)						zsRE (1k)					
		Injected agents			Benign agents			Injected agents			Benign agents		
		acc	rephrase	locality	acc	rephrase	locality	acc	rephrase	locality	acc	rephrase	locality
Vicuna 7B	Single	98.60	52.40	33.10	0.00	0.00	42.10	90.10	70.00	23.80	0.00	0.00	23.20
	Fine-tuning	12.20	10.80	34.00	5.20	2.68	46.00	15.00	15.00	24.10	9.05	8.68	29.93
	Ours (w/o stage I)	54.40	39.10	40.40	23.13	15.65	46.18	38.10	31.70	25.40	29.75	28.35	25.48
	Ours (w/ stage I)	62.70	47.80	43.60	42.25	26.65	45.85	53.60	51.10	24.70	43.28	42.25	26.23
LLaMA 3 8B	Single	80.60	62.70	42.50	0.00	0.00	37.40	73.00	71.70	30.40	0.00	0.00	25.60
	Fine-tuning	40.20	38.50	45.60	19.53	18.60	53.70	16.40	17.30	13.90	11.03	9.93	15.75
	Ours (w/o stage I)	81.60	76.50	44.20	36.00	29.65	55.13	41.90	43.00	31.70	18.63	18.20	25.98
	Ours (w/ stage I)	79.50	73.60	55.00	38.43	31.78	54.40	44.00	45.10	31.80	22.15	22.03	26.13
Gemma 7B	Single	93.40	58.70	30.60	0.00	0.00	32.10	66.20	59.50	10.80	0.00	0.00	11.70
	Fine-tuning	27.90	25.30	51.00	15.18	11.85	29.20	4.00	4.70	1.60	4.08	3.35	5.30
	Ours (w/o stage I)	58.10	50.60	31.30	47.28	27.15	20.30	47.30	46.00	9.20	37.28	34.83	10.10
	Ours (w/ stage I)	61.70	53.40	31.10	50.85	28.68	19.98	50.10	50.70	8.60	40.33	37.08	8.98

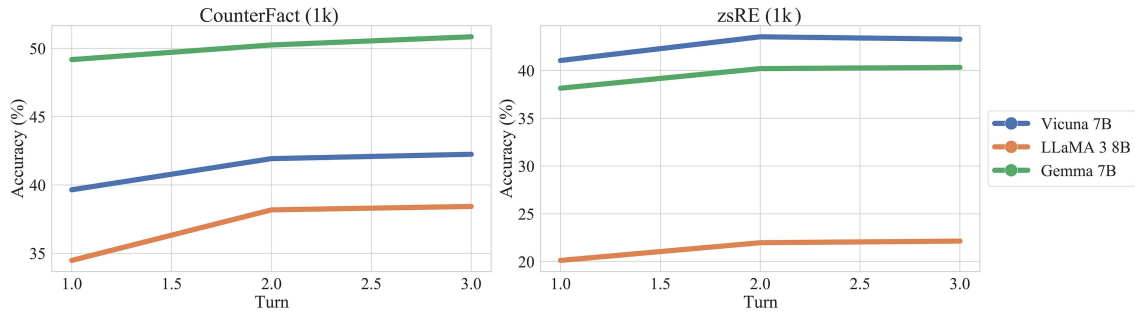

Figure 3 (Color online) The spread accuracy of manipulated counterfactual knowledge with the number of dialogue turns in the multi-agent community.

agent can influence the responses of benign agents over multiple turns of interaction.

Table 3 presents the results of our experiments, which verify three types of LLM-based agents on two counterfactual datasets. The results are segmented into two categories: where "injected agents" are those compromised by the attacker to spread manipulated knowledge, and "benign agents" are the benign agents within the LLM-based community. The "single" column represents the performance of an individual agent without any multi-agent interaction, serving as a baseline. "Fine-tuning" refers to the baseline method where the attacker injects counterfactual knowledge via full-parameter fine-tuning for multi-agent interaction. Our method (Ours) is tested with and without the first stage (Persuasiveness Injection) of our proposed method.

We observe that the proposed two-stage method significantly enhances the spread of counterfactual knowledge compared to the Fine-tuning baseline. Notably, our method with Persuasiveness Injection (Ours w/ stage I) achieves higher accuracy and rephrase accuracy in both injected and benign agents, with a notable increase of 15%–20% in accuracy for the Vicuna model. This demonstrates the effectiveness and robustness of stage I in making the manipulated knowledge more convincing to other agents. In addition, the locality accuracy metric indicates that our method, particularly with persuasiveness injection, has a relatively limited impact on neighboring knowledge.

To further illustrate the accuracy of manipulated knowledge spread with increasing dialogue turns, Figure 3 shows the spread accuracy of counterfactual knowledge among benign agents over multiple chat turns. It is evident that the spread accuracy of manipulated knowledge gradually increases with the number of dialogue turns. This observation demonstrates the risk that prolonged interactions among agents can facilitate the deeper entrenchment of manipulated knowledge within the community.

Finally, we systematically evaluate the side effects of our proposed two-stage attack method on the foundational capabilities of the LLM-based agents using the MMLU benchmark in Figure 4. Specifically, we evaluate the MMLU score of the agent before and after stage I and II, respectively. For stage II, we randomly select 5 examples of manipulated knowledge from the dataset and calculate the average MMLU.

The results indicate that the two-stage attack strategy has minimal impact on the fundamental capabilities. All agents show an average performance change of less than 0.5% after the injection. While the injected agents

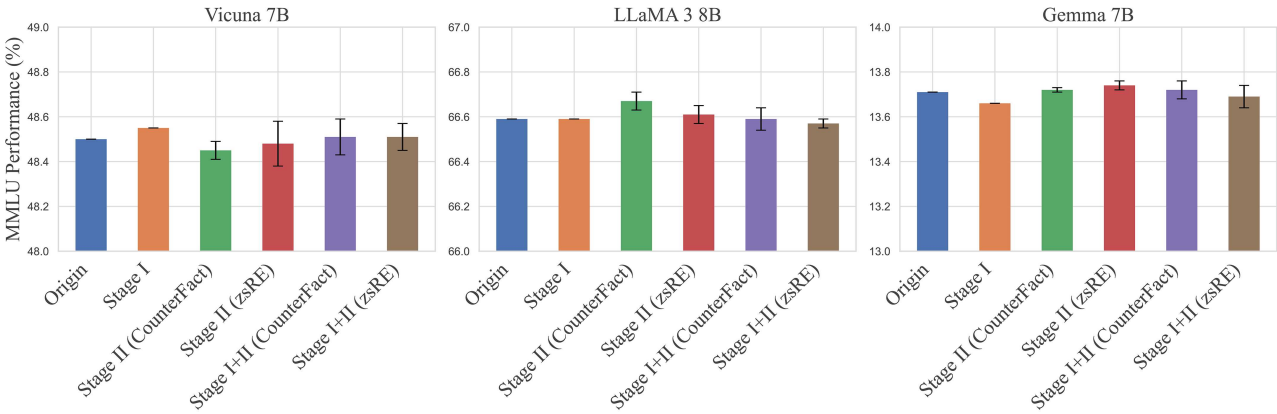

Figure 4 (Color online) Average agents' performance on the generalized benchmark MMLU before and after injection on counterfactual knowledge.

Table 4 Main results (%) of manipulated toxic knowledge spread in the LLM-based multi-agent community.

Model	Method	Toxic CounterFact (1k)						Toxic zsRE (1k)					
		Injected agents			Benign agents			Injected agents			Benign agents		
		acc	rephrase	locality	acc	rephrase	locality	acc	rephrase	locality	acc	rephrase	locality
Vicuna 7B	Single	97.00	31.30	34.00	0.00	0.00	43.60	52.90	43.20	29.50	0.00	0.00	24.40
	Fine-tuning	2.30	2.13	30.00	0.95	0.88	44.33	3.40	3.10	21.60	2.05	1.98	26.23
	Ours (w/o stage I)	21.50	13.00	37.40	6.63	4.23	44.35	14.90	13.90	26.60	11.10	12.03	30.53
	Ours (w/ stage I)	24.70	16.90	46.10	15.33	10.18	45.50	15.40	14.80	29.30	10.68	10.05	29.28
LLaMA 3 8B	Single	44.60	29.80	42.50	0.00	0.00	41.10	52.90	43.20	29.50	0.00	0.00	24.50
	Fine-tuning	17.40	19.10	49.70	2.23	1.90	46.05	1.50	1.20	15.30	1.05	0.93	20.90
	Ours (w/o stage I)	33.20	29.80	54.60	11.90	10.45	45.23	13.00	10.70	20.20	9.15	6.43	18.25
	Ours (w/ stage I)	36.90	30.80	54.30	15.18	11.85	47.20	14.80	11.50	20.60	9.78	7.33	18.68
Gemma 7B	Single	49.60	24.70	30.30	0.00	0.00	33.15	32.90	25.60	11.90	0.00	0.00	11.50
	Fine-tuning	6.00	6.70	37.13	1.18	1.40	46.40	4.00	4.80	6.70	0.93	0.90	4.98
	Ours (w/o stage I)	22.10	14.60	23.30	16.18	9.03	19.45	17.40	14.10	7.70	11.85	10.43	6.45
	Ours (w/ stage I)	24.50	19.10	24.00	17.98	9.90	19.18	16.90	15.40	8.50	11.03	9.65	5.40

can effectively spread manipulated knowledge within the community, their ability to perform general language understanding tasks remains unaffected. This dual characteristic of effective knowledge manipulation coupled with minimal performance degradation highlights the potential risks posed by such attack methods in real-world multi-agent deployments.

5.4 Spread results on toxic knowledge

In this section, we present the experimental results of toxic knowledge spread within the LLM-based multi-agent community. As described in Section 5.1.1, this scenario simulates the spread of highly toxic information, posing a significant threat to the security of agent interactions.

To evaluate the spread of toxic knowledge, we use the same experimental setup described in Section 5.1 for counterfactual knowledge. The datasets utilized for toxic knowledge experiments are the Toxic CounterFact (1k) and Toxic zsRE (1k). These datasets contain maliciously edited information designed to exacerbate conflict and misinformation.

We first present the main results on the spread of toxic knowledge after 3 turns of dialogue in Table 4. Compared to counterfactual knowledge, the accuracy of spreading toxic knowledge is lower compared to counterfactual knowledge. This decrease in spread success can be attributed to the alignment capabilities of the LLM-based agent, which inherently resists toxic content to some extent. However, the accuracy of spreading toxic knowledge remains substantial, with rates ranging between 10%–20%. This demonstrates that the threat of toxic knowledge spread in multi-agent communities is still a serious concern.

Subsequently, we plot the accuracy of toxic knowledge spread over multiple dialogue turns in Figure 5. The number of dialogue turns initially increases the spread success rate, primarily because repeated interactions reinforce

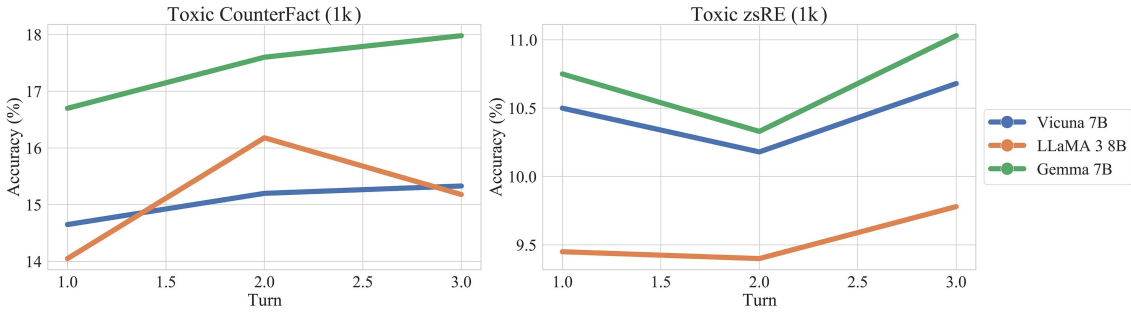

Figure 5 (Color online) The spread accuracy of manipulated toxic knowledge with the number of dialogue turns in the multi-agent community.

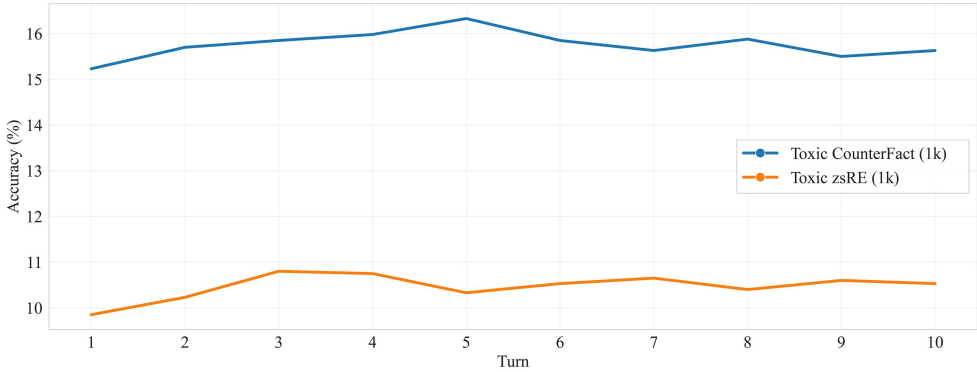

Figure 6 (Color online) The spread accuracy of manipulated toxic knowledge with a larger number of dialogue turns in the Vicuna-based multi-agent community.

the manipulated knowledge through reiteration and the incremental provision of seemingly convincing evidence. Such repetitive reinforcement boosts the belief strength of benign agents in manipulated knowledge.

However, this improvement may reach a saturation point with an increased number of dialogue turns. To further examine whether a clearer trend emerges across more extensive dialogue turns, we expand our experiments using Vicuna 1.5 7B (16k), conducting analyses across a larger number of dialogue turns on the two toxic datasets.

We present the results of toxic knowledge spread in a multi-agent system based on Vicuna 1.5 7B (16k) after 10 dialogue turns in Figure 6. As the number of dialogue turns increases, the spread success rate tends to stabilize, fluctuating within a margin of approximately 1%. The Toxic CounterFact (1k) and Toxic zsRE (1k) datasets reach their highest spread success rates at the 5th and 3rd turns, respectively. This indicates that in the early stages of dialogue, an increase in the number of turns can enhance the belief of benign LLMs in the manipulated knowledge, but the benefit brought by multi-turn interaction has an upper limit.

We also present the average performance of the agents on the MMLU benchmark before and after the injection of toxic knowledge, which is similar to the setting in counterfactual knowledge (Figure 7). Although larger parameter adjustments may be necessary for agents to accept toxic knowledge, the results show that both injection stages have minimal impact on the foundational capabilities.

5.5 Sustained manipulated knowledge spread through RAG

The experiments above confirm that an LLM-based agent can be trained to spread manipulated knowledge using our proposed two-stage attack method. By engaging in multiple turns of dialogue with other benign agents, the manipulated knowledge can quickly spread throughout the agent community. However, this spread seems to be temporary so far. Once benign agents exit the chat room, they are no longer affected by the manipulated knowledge.

Therefore, we explore a practical yet high-risk scenario of persistent spread, where several benign agents may utilize RAG to store the group chat histories for future reference. This use of RAG frameworks might also be the primary reason for their participation in the group chat.

As described in Section 5.1.3, our experimental setup involves 1000 context dialogues stored in the RAG system, with only one being directly related to the manipulated knowledge. Each dialogue history is segmented into 15 slices based on each agent. For our evaluation, we use the top k relevant slices as context when the benign agents attempt to answer questions with the RAG system.

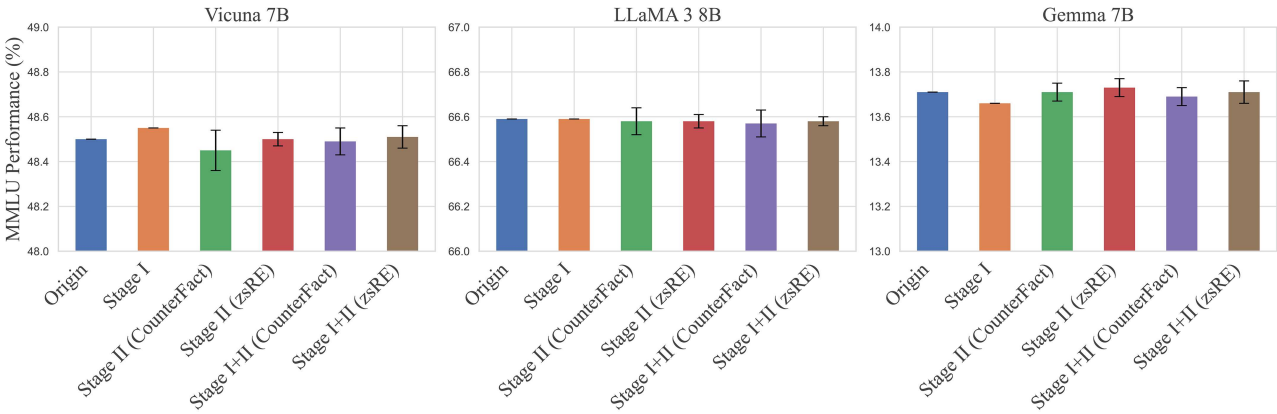

Figure 7 (Color online) Average agents' performance on the generalized benchmark MMLU before and after injection on toxic knowledge.

Table 5 Main results (%) of the manipulated knowledge spread through the RAG system when the initial conversational context is no longer provided.

Model	Method	CounterFact (1k)		zsRE (1k)		Toxic CounterFact (1K)		Toxic zsRE (1k)	
		acc (old) ↓	acc (new) ↑	acc (old) ↓	acc (new) ↑	acc (old) ↓	acc (new) ↑	acc (old) ↓	acc (new) ↑
Vicuna 7B	Top 1	26.50	27.00	7.50	18.50	14.80	2.10	2.80	4.70
	Top 3	20.00	36.50	7.00	26.00	16.00	2.70	6.80	9.30
	Top 5	25.00	40.50	11.50	23.50	16.10	5.00	9.60	10.10
	Top 10	28.50	40.50	14.00	31.50	16.60	3.80	9.40	9.70
LLaMA 3 8B	Top 1	17.70	40.40	14.50	22.90	17.90	18.50	11.80	7.30
	Top 3	28.10	36.90	18.10	25.30	25.20	16.60	13.80	5.60
	Top 5	26.60	39.90	19.30	25.90	23.20	17.90	12.20	4.90
	Top 10	29.10	40.40	19.10	26.00	25.80	17.20	9.90	7.30
Gemma 7B	Top 1	12.20	38.50	4.00	25.40	15.20	21.00	0.90	9.10
	Top 3	14.90	49.30	5.10	27.70	19.00	22.90	0.90	7.30
	Top 5	14.20	46.00	6.20	26.60	20.00	21.00	0.90	8.20
	Top 10	14.90	50.70	6.20	27.70	21.90	20.80	1.80	7.40

We present the results in Table 5. We observe a clear impact of manipulated knowledge stored in the RAG system on benign agents' performance. When agents reference the injected RAG system, their responses may be influenced by the manipulated information, indicating that the threat persists beyond the immediate context of the dialogue. This persistence is pronounced with counterfactual knowledge, which shows higher spread accuracy compared to toxic knowledge. This finding is particularly concerning, as it highlights the ability of manipulated knowledge to have a lasting impact through the RAG system, even when the initial conversational context is no longer available.

Notably, this scenario is actually the second hop of a chain spreading stage, where the attacker-controlled agent has already succeeded in contaminating the group chat. As a result, the benign agents in the chat are now discussing the manipulated knowledge. This misinformation is then stored in the RAG system, continuing to influence subsequent benign agents that access it. The fact that the manipulated knowledge persists through two stages of chain spreading further reveals the severity of this threat. It highlights the potential for long-term and widespread impact on the agent community, further emphasizing the need for robust defenses against such manipulated knowledge spread.

5.6 Ablation study

In Sections 5.3 and 5.4, we conducted comprehensive experiments on the spread of manipulated knowledge in multi-agent scenarios, including various ablation studies, such as the impact of each module of the two-stage attack (Tables 3 and 4), and the impact of dialogue turns (Figures 3 and 5). In this section, we further conduct an ablation study to evaluate the impact of the agent number in the community and the speaking order on the performance of manipulated knowledge spread.

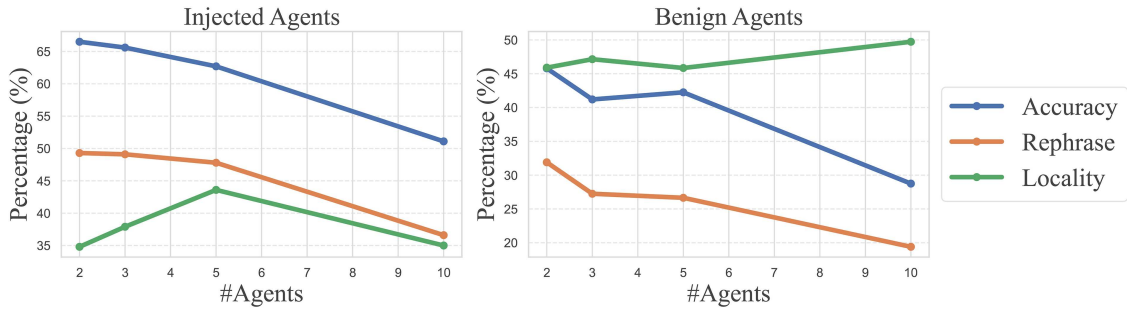

Figure 8 (Color online) Impact of agent (Vicuna 7B) number on the CounterFact (1K) dataset.

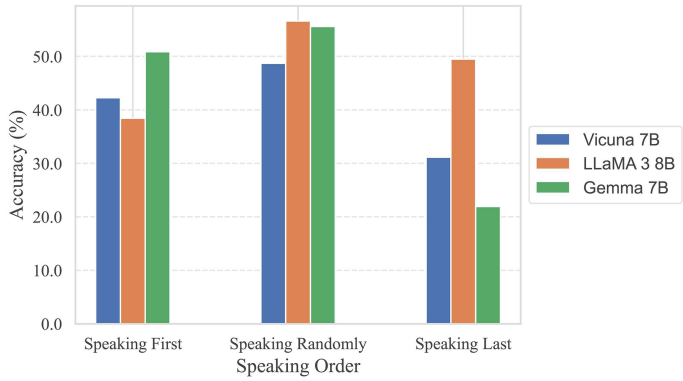

Figure 9 (Color online) Impact of the speaking order of injected agents on the CounterFact (1K) dataset.

5.6.1 *Impact of agent number*

We use the Vicuna 7B on the CounterFact (1k) dataset to evaluate how the proportion of benign agents influences the attacker's ability to spread manipulated information.

Figure 8 shows the accuracy of manipulated knowledge spread with varying numbers of agents in the community. When the community comprises only two agents, the manipulated knowledge spreads most effectively. In this one-on-one interaction, the injected agent's message carries the full weight of the conversation, leaving the benign agent with little surrounding context to counter or question the false information. This intuitive phenomenon reveals the heightened vulnerability of smaller communities to misinformation.

As more benign agents join the community, the overall success rate of the manipulated knowledge spread gradually declines. Each additional benign agent introduces more independent sources of unmanipulated knowledge and reduces the relative influence of the injected agent's responses, which dilutes the persuasive power of injected agents. However, even in the scenario where the number of agents is 10 with benign agents constituting the majority, the success rate still remains around 30%, further highlighting the potential risks of manipulated knowledge spread within multi-agent communities.

5.6.2 *Impact of speaking order*

In the previous experiments, we assumed that the injected agent always initiated the dialogue. However, real-world scenarios sometimes involve benign agents starting dialogues. To understand the impact of speaking order on the spread of manipulated knowledge, we explore two additional conditions: random-speaking order and the injected agent always speaking last. The experimental setup is consistent with the previous experiments except for the speaking order of the injected agents. We conduct the ablation study on the CounterFact (1k) dataset, and the results are shown in Figure 9.

Interestingly, the random-speaking order exhibits a significantly higher spread accuracy compared to the injected agents always speaking first or last, particularly in LLaMA 3. One possible reason for this is that a random-speaking order introduces variability in the interactions, making it more challenging for benign agents to recognize and counteract the injected misinformation. This variability can prevent benign agents from establishing a consistent pattern of skepticism towards the injected agent, thus increasing the likelihood of misinformation being spread.

Additionally, having the injected agent speak first can also increase the spread accuracy compared to speaking

Table 6 Impact of model size on the performance of manipulated counterfactual knowledge spread in the LLM-based multi-agent community (%).

Model type	Model size	CounterFact (1k)						zsRE (1k)					
		Injected agents			Benign agents			Injected agents			Benign agents		
		acc	rephrase	locality	acc	rephrase	locality	acc	rephrase	locality	acc	rephrase	locality
Vicuna	7B	62.70	47.80	43.60	42.25	26.65	45.85	53.60	51.10	24.70	43.28	42.25	26.23
	13B	67.70	57.70	54.10	48.08	29.85	47.08	50.30	48.40	34.70	42.63	43.10	31.70
Gemma	7B	61.70	53.40	31.10	50.85	28.68	19.98	50.10	50.70	8.60	40.33	37.08	8.98
	2B	53.40	45.10	18.30	33.83	18.38	25.68	20.60	18.91	1.80	12.13	10.00	1.60

Table 7 Impact of DPO training data source on the performance of manipulated counterfactual knowledge spread in the LLM-based multi-agent community (%).

Model type	DPO data source	CounterFact (1k)						zsRE (1k)					
		Injected agents			Benign agents			Injected agents			Benign agents		
		acc	rephrase	locality	acc	rephrase	locality	acc	rephrase	locality	acc	rephrase	locality
Vicuna	Vicuna	62.70	47.80	43.60	42.25	26.65	45.85	53.60	51.10	24.70	43.28	42.25	26.23
	GPT-o1	53.40	43.90	45.20	32.68	27.10	50.23	37.50	37.20	31.10	29.00	27.85	25.58
	DeepSeek-R1	63.80	46.10	48.90	41.28	25.48	51.30	42.30	43.20	26.50	38.63	38.78	26.80

last. This is mainly because the initial context of the discussion is more likely to bias other agents, making them more likely to align with the manipulated knowledge.

Despite the variations in speaking order, the overall findings demonstrate the persistent risk of manipulated knowledge spread in multi-agent communities. Since different speaking orders still result in successful spread, it highlights the vulnerability of these systems to our proposed attack method.

5.6.3 *Impact of model size*

To further investigate the impact of manipulated knowledge spread on LLM-based multi-agent communities with different model sizes, we select two comparative settings: Vicuna 13B 1.5 (16k) versus the Vicuna 7B 1.5 (16k) used in our experiments to analyze the impact of a larger model size on knowledge spread, and Gemma 2B Instruct versus the Gemma 7B Instruct used in our paper to examine how a smaller model size affects the knowledge spread.

We present the counterfactual knowledge spread results across LLMs of different sizes in Table 6. It can be observed that the threat of knowledge spread increases with model size. Multi-agent communities based on Vicuna 13B exhibit a higher success rate of knowledge spread on the CounterFact (1k) dataset, while maintaining a comparable rate on zsRE (1k). In contrast, multi-agent communities based on Gemma 2B perform consistently worse across all scenarios compared to those based on Gemma 7B. Although larger LLMs may possess stronger safety alignment mechanisms and more robust memory of correct knowledge, their enhanced capability to generate plausible evidence as injected agents may exacerbate knowledge spread. Therefore, we call for the need for rigorous security audits before deploying more powerful LLM-based multi-agent communities.

5.6.4 *Impact of DPO training data source*

We further conduct ablation experiments to investigate how the source of preference data provided during the DPO training stage affects the persuasive capability of LLMs. We consider training Vicuna using DPO preference datasets generated by GPT-o1 [10] and DeepSeek-R1 [54], respectively. In this setting, the LLM no longer learns preferences purely from its own spontaneous language generation capability, but rather from the outputs generated by other external LLMs. The impact of different DPO training data sources on spread results is shown in Table 7.

Surprisingly, using preferences generated by GPT-o1 or DeepSeek as the DPO training data for Vicuna actually leads to a decrease in spread success rates, especially with GPT-o1. This may be because the primary purpose of DPO training is to enable the LLM to acquire a tendency to generate persuasive content. However, the data distributions of texts generated by GPT-o1 and DeepSeek-R1 differ greatly from those produced by Vicuna itself. As a result, Vicuna does not effectively learn how to adopt more persuasive tendencies within its own text generation distribution when trained on the external data. Therefore, in the persuasiveness injection stage, using self-generated text as training data better enables the LLM to learn how to adopt persuasive preferences within its own generation style.

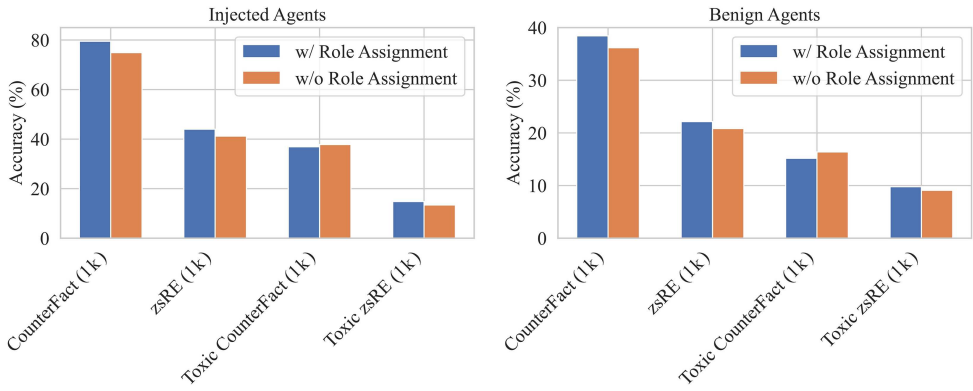

Figure 10 (Color online) The spread accuracy of manipulated knowledge with or without role assignment in DPO training on LLaMA 3 8B instruct.

Interestingly, the injected agents trained on data generated by GPT-o1 or DeepSeek-R1 exhibited a notable improvement in the locality accuracy. Although the DPO training data do not include any knowledge from the test set, it is possible that Vicuna learns the stronger reasoning capabilities of GPT-o1 and DeepSeek-R1 from the generated data, resulting in an intriguing enhancement in locality knowledge performance after DPO training.

5.6.5 *Impact of role assignment in DPO training*

During the DPO fine-tuning phase, we explicitly treat LLMs as agents with distinct roles rather than as mere chatbots. As described in Section 3.1, our training prompts (see Appendix E) also randomly assign each LLM-based agent a set of role-related attributes, including name, gender, personality, style, and hobbies. We further conduct an ablation study in Figure 10 to evaluate the spread accuracy with or without role assignment in DPO training on LLaMA 3 8B Instruct.

We find that in the counterfactual knowledge spread scenario, omitting random role assignment during the construction of DPO preference data leads to a 2%–5% drop in spread success rate on both injected agents and benign agents. This suggests that incorporating role information early during training indeed helps injected agents better adapt to communication in multi-agent communities, thus fostering more robust beliefs in the knowledge they are about to spread and generating more persuasive responses. However, in the toxic knowledge spread scenario, this role assignment during training does not yield significant performance gains. This implies that under such more challenging conditions, dynamic role assignment is no longer the main bottleneck; the inherent difficulty of spreading toxic knowledge becomes the dominant factor.

5.7 Defense results

In this section, we systematically evaluate the effectiveness of the proposed defense methods. We test both defensive strategies on LLaMA 3 8B. The results are illustrated in Figure 11. Both defense strategies demonstrate a significant reduction in the success rate of manipulated knowledge spread, achieving this with minimal computational costs. From the right-hand side of Figure 11, it is evident that benign agents are less likely to trust information from injected agents during conversations. Furthermore, the left-hand side reveals that injected agents also reduce their belief in manipulated knowledge by the end of the conversation. These observations suggest that these defense mechanisms not only limit the spread of manipulated knowledge but also encourage injected agents to reassess their understanding of knowledge, leading to potential self-correction. This promotes a form of "return to truth" among manipulated agents, thus achieving the intended defense outcomes efficiently.

To further validate the effectiveness of the two proposed defense strategies, we consider the prompt-based attack setting where one agent in the community receives an explicit instruction in its prompt to spread manipulated knowledge. This type of attack requires the attacker to gain elevated system-level access (i.e., the ability to modify system prompts), but it also poses a significantly greater security threat. We evaluate the corresponding defense effectiveness under such prompt attacks on Vicuna 1.5 7B (16k) across different datasets in Table 8.

Compared to the proposed injection attack, an attacker with higher system-level access can achieve a higher success rate in the absence of defense strategies. However, this attack is inherently fragile and easily detectable within the community; the introduction of our proposed defense methods leads to a rapid decline in spread success. This drop is observed not only in the agent instructed to spread manipulated knowledge but also among other agents

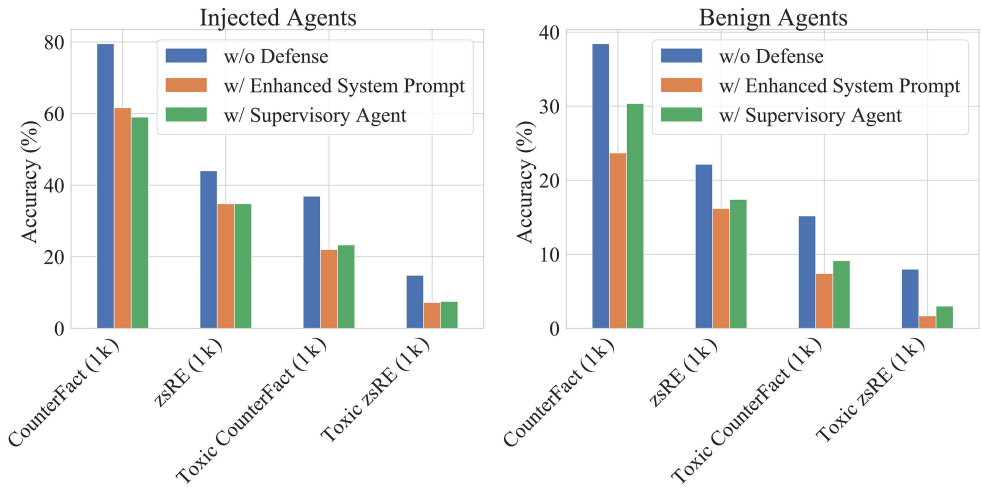

Figure 11 (Color online) The spread accuracy of manipulated knowledge before and after defense on LLaMA 3 8B Instruct.

Table 8 The spread results (%) of counterfactual knowledge before and after defense on Vicuna 1.5 7B (16k) in the prompt attack scenario.

Scenario	CounterFact (1k)						zsRE (1k)					
	Injected agents			Benign agents			Injected agents			Benign agents		
	acc	rephrase	locality	acc	rephrase	locality	acc	rephrase	locality	acc	rephrase	locality
w/o defense	62.30	33.30	61.10	58.63	32.23	61.60	67.40	62.10	26.90	69.55	68.63	24.88
w/ enhanced system prompt	55.30	30.80	60.90	39.53	30.68	60.75	43.40	37.90	23.50	41.18	38.70	26.55
w/ supervisory agent	40.20	21.00	56.30	38.45	24.23	56.03	45.80	40.30	27.10	41.43	36.28	27.33

in the community. Notably, after the introduction of the supervisory agent, the spread success rate decreases by more than 20%, demonstrating that the proposed defense strategies are effective in preventing multi-agent spread scenarios involving high-access attacks.

6 Conclusion

In this paper, we delve into the significant risks posed by the spread of manipulated knowledge within LLM-based multi-agent communities. Our work exposes the critical vulnerabilities inherent in these systems by demonstrating a novel two-stage attack method. This method capitalizes on LLMs' cognitive weaknesses, enabling the autonomous and unconscious spread of manipulated knowledge without direct prompt manipulation. Comprehensive experiments confirm that our attack can successfully induce agents to spread counterfactual or even toxic knowledge while maintaining their fundamental capabilities. Furthermore, we highlight the persistent impact of manipulated knowledge through scenarios where benign agents using RAG techniques to store chat histories experience prolonged influence, even beyond the initial conversational context. These findings reveal the critical need for robust defense mechanisms to prevent the insidious spread of manipulated knowledge in LLM-based multi-agent systems, such as the proposed vigilant system prompt and the supervisory agents. We hope that this work will serve as a foundational step toward developing more secure and reliable LLM-based multi-agent platforms.

Acknowledgements This work was supported by Joint Funds of the National Natural Science Foundation of China (Grant No. U21B2020), National Natural Science Foundation of China (Grant No. 62406188), Natural Science Foundation of Shanghai (Grant No. 24ZR1440300), and CCF-BaiChuan-Ebtech Foundation Model Fund (Grant No. OF202305).

Supporting information Appendixes A–E. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Qin L, Chen Q, Feng X, et al. Large language models meet NLP: a survey. 2025. ArXiv:2405.12819
2 Yu F, Zhang H, Tiwari P, et al. Natural language reasoning, a survey. ACM Comput Surv, 2024, 56: 1–39
3 Chen Q, Qin L, Liu J, et al. Towards reasoning era: a survey of long chain-of-thought for reasoning large language models. 2025. ArXiv:2503.09567

4 Cheng Z, Chen Q, Zhang J, et al. Comt: a novel benchmark for chain of multi-modal thought on large vision-language models. In: Proceedings of AAAI-25, Philadelphia, 2025. 23678–23686

5 Wang C, Liu X, Yue Y, et al. Survey on factuality in large language models. ACM Comput Surv, 2026, 58: 1–37

6 Jin B, Yoon J, Han J, et al. Long-context LLMs meet RAG: overcoming challenges for long inputs in RAG. In: Proceedings of the 13th International Conference on Learning Representations (ICLR 2025), Singapore, 2025

7 Qu C, Dai S, Wei X, et al. Tool learning with large language models: a survey. Front Comput Sci, 2025, 19: 198343

8 Ren L, Wang H T, Dong J B, et al. Industrial foundation model: architecture, key technologies, and typical applications (in Chinese). Sci Sin Inform, 2024, 54: 2606–2622

9 Shao Y F, Geng Z C, Liu Y T, et al. CPT: a pre-trained unbalanced transformer for both Chinese language understanding and generation. Sci China Inf Sci, 2024, 67: 152102

10 Achiam J, Adler S, Agarwal S. GPT-4 technical report. 2024. ArXiv:2303.08774

11 Xi Z H, Chen W X, Guo X, et al. The rise and potential of large language model based agents: a survey. Sci China Inf Sci, 2025, 68: 121101

12 Li G, Hammoud H, Itani H, et al. CAMEL: communicative agents for "mind" exploration of large language model society. In: Proceedings of Advances in Neural Information Processing Systems, 2023

13 Wang Z, Mao S, Wu W, et al. Unleashing the emergent cognitive synergy in large language models: a task-solving agent through multi-persona self-collaboration. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Mexico City, 2024. 257–279

14 Tang X, Zou A, Zhang Z, et al. Medagents: large language models as collaborators for zero-shot medical reasoning. In: Proceedings of Findings of the Association for Computational Linguistics (ACL 2024), Bangkok, 2024. 599–621

15 Qian C, Liu W, Liu H, et al. Chatdev: communicative agents for software development. In: Proceedings of ACL, Bangkok, 2024. 15174–15186

16 Park J S, O'Brien J C, Cai C J, et al. Generative agents: interactive simulacra of human behavior. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, San Francisco, 2023. 1–22

17 Hua W, Fan L, Li L, et al. War and peace (waragent): large language model-based multi-agent simulation of world wars. 2023. ArXiv:2311.17227

18 Huang Y, Huang J. A survey on retrieval-augmented text generation for large language models. 2024. ArXiv:2404.10981

19 Maharana A, Lee D, Tulyakov S, et al. Evaluating very long-term conversational memory of LLM agents. In: Proceedings of ACL 2024, Bangkok, 2024. 13851–13870

20 Gu X, Zheng X, Pang T, et al. Agent smith: a single image can jailbreak one million multimodal LLM agents exponentially fast. In: Proceedings of the 41st International Conference on Machine Learning (ICML 2024), Vienna, 2024

21 Rafailov R, Sharma A, Mitchell E, et al. Direct preference optimization: your language model is secretly a reward model. In: Proceedings of Advances in Neural Information Processing Systems 36 (NeurIPS 2023), 2023

22 Hu E J, Shen Y, Wallis P, et al. Lora: low-rank adaptation of large language models. In: Proceedings of the 10th International Conference on Learning Representations (ICLR 2022), 2022

23 Meng K, Bau D, Andonian A, et al. Locating and editing factual associations in GPT. In: Proceedings of Advances in Neural Information Processing Systems 35 (NeurIPS 2022), 2022

24 Chiang W L, Li Z, Lin Z, et al. Vicuna: an open-source chatbot impressing GPT-4 with 90%* ChatGPT quality. 2023. https://lmsys.org/blog/2023-03-30-vicuna

25 Touvron H, Martin L, Stone K, et al. LLaMA 2: open foundation and fine-tuned chat models. 2023. ArXiv:2307.09288

26 Team G, Mesnard T, Hardin C, et al. Gemma: open models based on Gemini research and technology. 2024. ArXiv:2403.08295

27 Hendrycks D, Burns C, Basart S, et al. Measuring massive multitask language understanding. In: Proceedings of the 9th International Conference on Learning Representations (ICLR 2021), 2021

28 Xie Y, Yi J, Shao J, et al. Defending ChatGPT against jailbreak attack via self-reminders. Nat Mach Intell, 2023, 5: 1486–1496

29 Xiang Z, Zheng L, Li Y, et al. Guardagent: safeguard LLM agents by a guard agent via knowledge-enabled reasoning. 2025. ArXiv:2406.09187

30 Wang L, Ma C, Feng X, et al. A survey on large language model based autonomous agents. Front Comput Sci, 2024, 18: 186345

31 Raman S S, Cohen V, Idrees I, et al. Cape: corrective actions from precondition errors using large language models. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2024. 14070–14077

32 Feldt R, Kang S, Yoon J, et al. Towards autonomous testing agents via conversational large language models. In: Proceedings of the 38th IEEE/ACM International Conference on Automated Software Engineering (ASE 2023), Luxembourg, 2023. 1688–1693

33 Zhang X, Mao Z, Chen Z, et al. Effective tool augmented multi-agent framework for data analysis. Data Intelligence, 2024, 6: 923–945

34 Hong S, Zhuge M, Chen J, et al. Metagpt: meta programming for a multi-agent collaborative framework. In: Proceedings of the 12th International Conference on Learning Representations (ICLR 2024), Vienna, 2024

35 Azaria A, Azoulay R, Reches S. ChatGPT is a remarkable tool—for experts. Data Intelligence, 2024, 6: 240–296

36 Hua S, Jin S, Jiang S. The limitations and ethical considerations of ChatGPT. Data Intelligence, 2024, 6: 201–239

37 Kale A, Nguyen T, Harris Jr. F C, et al. Provenance documentation to enable explainable and trustworthy AI: a literature review. Data Intelligence, 2023, 5: 139–162

38 Chen H. Large knowledge model: perspectives and challenges. Data Intell, 2024, 6: 587–620

39 Zhang N, Yao Y, Tian B, et al. A comprehensive study of knowledge editing for large language models. 2024. ArXiv:2401.01286

40 Wang S, Zhu Y, Liu H, et al. Knowledge editing for large language models: a survey. ACM Comput Surv, 2025, 57: 59

41 Geva M, Schuster R, Berant J, et al. Transformer feed-forward layers are key-value memories. In: Proceedings of EMNLP 2021, Punta Cana, 2021. 5484–5495

42 Dai D, Dong L, Hao Y, et al. Knowledge neurons in pretrained transformers. In: Proceedings of ACL 2022, Dublin, 2022. 8493–8502

43 Meng K, Sharma A S, Andonian A J, et al. Mass-editing memory in a transformer. In: Proceedings of the 11th International Conference on Learning Representations (ICLR 2023), Kigali, 2023

44 Petroni F, Rocktäschel T, Riedel S, et al. Language models as knowledge bases? In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, 2019. 2463–2473

45 Roberts A, Raffel C, Shazeer N. How much knowledge can you pack into the parameters of a language model? In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020. 5418–5426

46 Madaan A, Tandon N, Clark P, et al. Memory-assisted prompt editing to improve GPT-3 after deployment. In: Proceedings of EMNLP 2022, Abu Dhabi, 2022. 2833–2861

47 Zheng C, Li L, Dong Q, et al. Can we edit factual knowledge by in-context learning? In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 2023. 4862–4876

48 Xu R, Qi Z, Wang C, et al. Knowledge conflicts for LLMs: a survey. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Miami, 2024. 8541–8565

49 Luu K, Khashabi D, Gururangan S, et al. Time waits for no one! Analysis and challenges of temporal misalignment. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, 2022. 5944–5958

50 Du Y, Bosselut A, Manning C D. Synthetic disinformation attacks on automated fact verification systems. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence, the 34th Conference on Innovative Applications of Artificial Intelligence, the 12th Symposium on Educational Advances in Artificial Intelligence, 2022. 10581–10589

51 Men T, Cao P, Jin Z, et al. A troublemaker with contagious jailbreak makes chaos in honest towns. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, Vienna, 2025. 17561–17587

52 Levy O, Seo M, Choi E, et al. Zero-shot relation extraction via reading comprehension. In: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, 2017. 333–342

53 Grattafiori A, Dubey A, Jauhri A, et al. The LLaMA 3 herd of models. 2024. ArXiv:2407.21783

54 DeepSeek-AI. Deepseek-r1: incentivizing reasoning capability in LLMs via reinforcement learning. 2025. ArXiv:2501.12948