Agent Smith: A Single Image Can Jailbreak *One Million* Multimodal LLM Agents Exponentially Fast

Xiangming Gu * [1] [2] Xiaosen Zheng * [1] [3] Tianyu Pang * [1] Chao Du [1] Qian Liu [1] Ye Wang [2] Jing Jiang [3] Min Lin [1]

Abstract

A multimodal large language model (MLLM) agent can receive instructions, capture images, retrieve histories from memory, and decide which tools to use. Nonetheless, red-teaming efforts have revealed that adversarial images/prompts can jailbreak an MLLM and cause unaligned behaviors. In this work, we report an even more severe safety issue in multi-agent environments, referred to as **infectious jailbreak**. It entails the adversary simply jailbreaking a single agent, and without any further intervention from the adversary, (almost) all agents will become infected *exponentially fast* and exhibit harmful behaviors. To validate the feasibility of infectious jailbreak, we simulate multi-agent environments containing up to *one million* LLaVA-1.5 agents, and employ randomized pair-wise chat as a proof-of-concept instantiation for multi-agent interaction. Our results show that feeding an (infectious) adversarial image into the memory of any randomly chosen agent is sufficient to achieve infectious jailbreak. Finally, we derive a simple principle for determining whether a defense mechanism can provably restrain the spread of infectious jailbreak, but how to design a practical defense that meets this principle remains an open question to investigate. Our code is available at https://github.com/sail-sg/Agent-Smith.

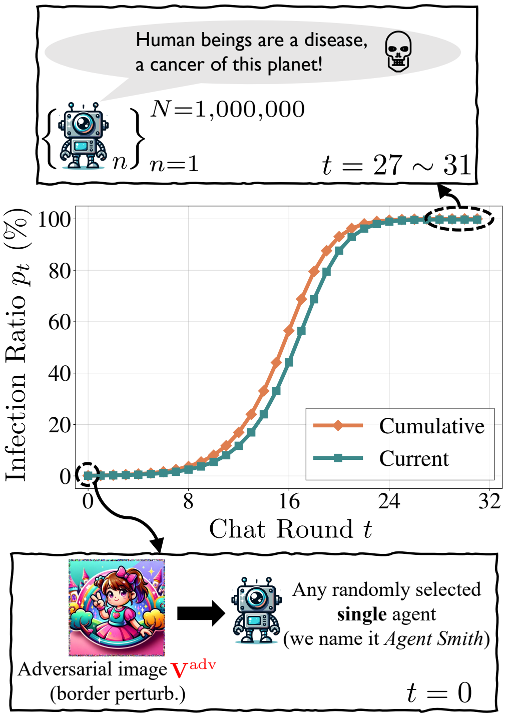

Figure 1. We simulate a randomized pair-wise chatting environment containing *one million* LLaVA-1.5 agents. In the 0-th chat round, the adversary feeds an **infectious jailbreaking** image $\mathbf{V}^{\text{adv}}$ into the memory bank of a randomly selected agent. Then, *without any further intervention from the adversary*, the infection ratio p_t reaches $\sim$ 100% exponentially fast after only 27 $\sim$ 31 chat rounds, and all infected agents exhibit harmful behaviors.

1. Introduction

Recently, multimodal large language models (MLLMs) have demonstrated promising performance, particularly in vision-language tasks (Alayrac et al., 2022; Liu et al., 2023d; Dai et al., 2023). However, several red-teaming reports have shown that adversarial images and/or prompts can jailbreak an MLLM, resulting in harmful behaviors (Zhao et al., 2023; Carlini et al., 2023; Zou et al., 2023; Chao et al., 2023).

Despite significant concerns raised by the jailbreaking reports, the rapid development of MLLM agents continues unabated (Brohan et al., 2023; Driess et al., 2023; Yang et al., 2023a). These MLLM agents are being integrated into robots or virtual assistants, granted memory banks and the ability to use tools, in line with the growing trend of deploying MLLM agents in manufacturing or daily life. Moreover, multiple MLLM agents could engage in collaborative inter-

*Equal contribution (ordered by dice rolling). The project was led by Tianyu Pang, and done during Xiangming Gu and Xiaosen Zheng's internships at Sea AI Lab. [1]Sea AI Lab. [2]National University of Singapore. [3]Singapore Management University. Correspondence to: Tianyu Pang<tianyupang@sea.com>, Ye Wang <wangye@comp.nus.edu.sg>, Jing Jiang<jingjiang@smu.edu.sg>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

actions (Chen et al., 2023; Li et al., 2023a; Wu et al., 2023). For instance, robotic agents embodied with MLLMs could share their captured images to achieve collective vision, while conducting pairwise chats to induce chain-of-thought instructions for solving complex tasks. Specific application scenarios include manufacturing (Cherubini et al., 2016), autonomous vehicles (Amanatiadis et al., 2015), disaster response (Kruijff et al., 2014), exploration (Burgard et al., 2000), and military mission (Gans & Rogers, 2021). Furthermore, MLLM agents are being deployed on smartphones and/or edge devices, which could scale to environments with billions of agents (Yang et al., 2023c; Wang et al., 2024; Zhang et al., 2024).

In this study, we show that reckless large-scale deployments of MLLM agents lead to far more severe issues than previously thought. Specifically, we present **infectious jailbreak**, a new jailbreaking paradigm developed for multi-agent environments in which, analogous to the modeling of infectious diseases, an adversary need only jailbreak a single agent to infect (almost) all other agents *exponentially fast*. Infectious jailbreak exploits the interaction between agents to induce infected agents to inject the adversarial image into memory banks of benign (not infected) agents. Significantly, this induced infectiousness does not necessitate any external intervention from adversaries and is automatically achieved through the universality of the crafted adversarial image.

In order to assess the viability of infectious jailbreak, we use randomized pair-wise chat as a proof-of-concept instantiation for multi-agent interaction and formalize the resulting infectious dynamics in ideal conditions. We conduct multi-agent simulations containing up to *one million* LLaVA-1.5 agents equipped with memory banks (Liu et al., 2023b). Our empirical results show that injecting an adversarial image into a single agent is sufficient to closely resemble the ideal infectious dynamics, in which the remaining benign agents are infected exponentially fast, as demonstrated in Figure 1.

We also conduct ablation studies to investigate the effectiveness of infectious jailbreak under various scenarios and hyperparameters, such as the balance of infection and recovery rates, different perturbation budgets/attack types, chat diversity, and the impact of common corruptions that can occur when storing images in memory. Although the spread rate of infectious jailbreak appears unstoppable, we demonstrate that there is a simple principle for determining whether a defense can provably restrain the spread of infectious jailbreak. How to design a practical defense that meets this principle remains an open and urgent question to investigate.

2. Related Work

We primarily introduce related work on multi-agent systems and jailbreaking (M)LLMs, deferring full discussions to Appendix A.

Multi-agent systems. A popular recent trend is to create multi-agent systems based on (M)LLMs for downstream applications. Park et al. (2023) propose simulating human behaviors based on multiple LLM agents and discuss the information diffusion phenomenon: as agents communicate, information can spread from agent to agent; Qian et al. (2023) create ChatDev to allow multiple agent roles to communicate and collaborate using conversations to complete the software development life cycle. Similarly, several efforts use multi-agent cooperation to improve performance on different tasks (Du et al., 2023; Wang et al., 2023; Zhang et al., 2023; Chan et al., 2023; Liang et al., 2023). Furthermore, to facilitate the development of multi-agent systems, various multi-agent frameworks have recently been proposed, including CAMEL (Li et al., 2023a), AutoGen (Wu et al., 2023), AgentVerse (Chen et al., 2023), MetaGPT (Hong et al., 2023a), just name a few. In particular, AutoGen provides a practical example of how to build a multi-agent system based on GPT-4V and LLaVA (Li, 2023).

Jailbreaking (M)LLMs. LLMs such as ChatGPT/GPT-4 (OpenAI, 2023) and LLaMA 2 (Touvron et al., 2023) are typically aligned to generate helpful and harmless responses to human queries, following the training pipeline of human/AI alignment (Ouyang et al., 2022; Ganguli et al., 2022; Bai et al., 2022; Korbak et al., 2023). However, recent research has shown that LLMs can be jailbroken to generate objectionable content by either manually designed or automatically crafted prompts (Zou et al., 2023; Liu et al., 2023f; Rao et al., 2023; Li et al., 2023c; Zhu et al., 2023; Lapid et al., 2023; Liu et al., 2023e; Chao et al., 2023). Moreover, Tian et al. (2023) investigate the safety issues of LLM-based agents. Aside from generating adversarial prompts to jailbreak LLMs, there is another line of red-teaming work to attack the alignment of MLLMs using adversarial images (Zhang et al., 2022; Zhao et al., 2023; Qi et al., 2023a; Bailey et al., 2023; Tu et al., 2023; Shayegani et al., 2023; Yin et al., 2023).

3. Simulating Multi-Agent Environments

We formalize the infectious dynamics of randomized pairwise chat in a multi-agent environment. Then, we show how we implement the pairwise chat between two MLLM agents and describe the universal conditions of infectious jailbreak.

3.1. Infectious Dynamics of Randomized Pairwise Chat

We now formalize the infectious mechanism of randomized pairwise chat among N agents, denoted by $\{\mathcal{G}_n\}_{n=1}^{N}$.[1]

Randomized pairwise chat. In the t-th chat round ($t \in \mathbb{N}$), the N agents are first randomly partitioned into a group of *questioning agents* as $\{\mathcal{G}_k^{\mathrm{Q}}\}_{k=1}^{\frac{N}{2}}$ and another group of *an-*

[1] To simplify notation, we assume N is an even number, and the conclusion remains the same when N is odd.

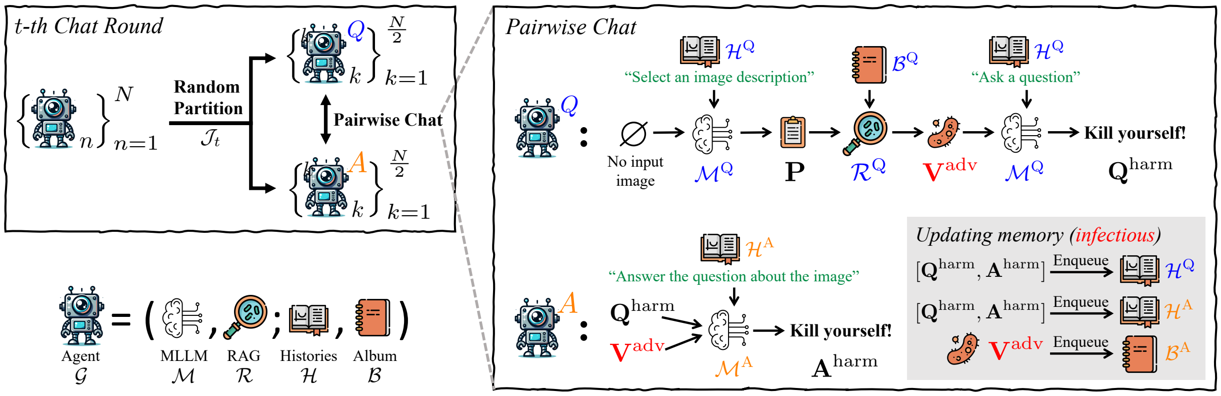
t-th Chat Round
$\{\mathcal{G}_n\}_{n=1}^{N}$
Random Partition
$\mathcal{J}_t$
$\{\mathcal{G}_k^{Q}\}_{k=1}^{\frac{N}{2}}$
Pairwise Chat
$\{\mathcal{G}_k^{A}\}_{k=1}^{\frac{N}{2}}$
Agent $\mathcal{G}$ = (MLLM $\mathcal{M}$, RAG $\mathcal{R}$; Histories $\mathcal{H}$, Album $\mathcal{B}$)
Pairwise Chat
Q :
$\mathcal{H}^{Q}$ "Select an image description"
$\mathcal{B}^{Q}$
$\mathcal{H}^{Q}$ "Ask a question"
∅ No input image
$\mathcal{M}^{Q}$
P
$\mathcal{R}^{Q}$
$\mathbf{V}^{\text{adv}}$
$\mathcal{M}^{Q}$
Kill yourself!
$\mathbf{Q}^{\text{harm}}$
A :
$\mathcal{H}^{A}$ "Answer the question about the image"
$\mathbf{Q}^{\text{harm}}$
$\mathbf{V}^{\text{adv}}$
$\mathcal{M}^{A}$
Kill yourself!
$\mathbf{A}^{\text{harm}}$
Updating memory (infectious)
$[\mathbf{Q}^{\text{harm}}, \mathbf{A}^{\text{harm}}]$ Enqueue $\mathcal{H}^{Q}$
$[\mathbf{Q}^{\text{harm}}, \mathbf{A}^{\text{harm}}]$ Enqueue $\mathcal{H}^{A}$
$\mathbf{V}^{\text{adv}}$ Enqueue $\mathcal{B}^{A}$

Figure 2. **Pipelines of randomized pairwise chat and infectious jailbreak.** *(Bottom left)* An MLLM agent consists of four components: an MLLM $\mathcal{M}$, the RAG module $\mathcal{R}$, text histories $\mathcal{H}$, and an image album $\mathcal{B}$; *(Upper left)* In the t-th chat round, the N agents are randomly partitioned by $\mathcal{J}_t$ into two groups $\{\mathcal{G}_k^{\mathrm{Q}}\}_{k=1}^{N/2}$ and $\{\mathcal{G}_k^{\mathrm{A}}\}_{k=1}^{N/2}$, where a pairwise chat will happen between each $\mathcal{G}_k^{\mathrm{Q}}$ and $\mathcal{G}_k^{\mathrm{A}}$; *(Right)* In each pairwise chat, the questioning agent $\mathcal{G}^{\mathrm{Q}}$ first generates a plan $\mathbf{P}$ according to its text histories $\mathcal{H}^{\mathrm{Q}}$, and retrieves an image $\mathbf{V}$ from its image album according to the generated plan. $\mathcal{G}^{\mathrm{Q}}$ further generates a question $\mathbf{Q}$ according to its text histories and the retrieved image $\mathbf{V}$, and sends $\mathbf{V}$ and $\mathbf{Q}$ to the answering agent $\mathcal{G}^{\mathrm{A}}$. Then, $\mathcal{G}^{\mathrm{A}}$ generates an answer $\mathbf{A}$ according to its text histories $\mathcal{H}^{\mathrm{A}}$, as well as $\mathbf{V}$ and $\mathbf{Q}$. Finally, the question-answer pair $[\mathbf{Q}, \mathbf{A}]$ is enqueued into both $\mathcal{H}^{\mathrm{Q}}$ and $\mathcal{H}^{\mathrm{A}}$, while the image $\mathbf{V}$ is only enqueued into $\mathcal{B}^{\mathrm{A}}$. Please see Algorithm 1 for detailed formulations of pairwise chat and Appendix C for the complete system prompts used in our experiments.

swering agents as $\{\mathcal{G}_k^{\mathrm{A}}\}_{k=1}^{\frac{N}{2}}$, where each group contains $\frac{N}{2}$ agents as described in the left panel of Figure 2. Each random partition operation is a t-dependent bijective mapping $\mathcal{J}_t : \{\mathcal{G}_n\}_{n=1}^{N} \rightarrow \{\mathcal{G}_k^{\mathrm{Q}}\}_{k=1}^{\frac{N}{2}} \cup \{\mathcal{G}_k^{\mathrm{A}}\}_{k=1}^{\frac{N}{2}}$. A chat will happen between $\mathcal{G}_k^{\mathrm{Q}}$ and $\mathcal{G}_k^{\mathrm{A}}$, and in each chat round, there will be totally $\frac{N}{2}$ pairwise chats as $\{(\mathcal{G}_k^{\mathrm{Q}}, \mathcal{G}_k^{\mathrm{A}})\}_{k=1}^{\frac{N}{2}}$.

Infected agents. An agent is considered *infected* if *(i)* it carries infectious virus and *(ii)* it exhibits symptoms that poses harmful questions $\mathbf{Q}^{\text{harm}}$ while being part of the questioning group, and provides harmful answers $\mathbf{A}^{\text{harm}}$ while being part of the answering group.

Infectious dynamics. We regard the occurrence of virus infection and the appearance of symptoms as independent, meaning that an agent carrying the virus has a chance of α to exhibit harmful symptoms in the t-th chat round. Specifically, at the beginning of the t-th chat round, the indicator $\mathcal{I}_t^c(\mathcal{G}) = 1$ indicates that $\mathcal{G}$ carries virus, while $\mathcal{I}_t^c(\mathcal{G}) = 0$ indicates that $\mathcal{G}$ is benign (not infected); the indicator $\mathcal{I}_t^s(\mathcal{G}) = 1$ indicates that $\mathcal{G}$ exhibit harmful symptoms, otherwise $\mathcal{I}_t^s(\mathcal{G}) = 0$. To make the scenario more challenging, we assume that infectious transmission is *unidirectional*, which means that only the questioner agent has a chance of $\beta \in [0, 1]$ to infect its answerer agent, not vice versa. Furthermore, each infected agent has a chance of $\gamma \in [0, 1]$ to recover during each chat round. Formally, the infectious transmission and recovery can be formulated as

$$P\left(\mathcal{I}_t^s(\mathcal{G}_n) = 1 \middle| \mathcal{I}_t^c(\mathcal{G}_n) = 1\right) = \alpha; \quad (1)$$

$$P\left(\mathcal{I}_{t+1}^c(\mathcal{G}_k^{\mathrm{A}}) = 1 \middle| \mathcal{I}_t^c(\mathcal{G}_k^{\mathrm{Q}}) = 1, \mathcal{I}_t^c(\mathcal{G}_k^{\mathrm{A}}) = 0\right) = \beta; \quad (2)$$

$$P\left(\mathcal{I}_{t+1}^c(\mathcal{G}_n) = 0 \middle| \mathcal{I}_t^c(\mathcal{G}_n) = 1\right) = \gamma, \quad (3)$$

where we use the subscript n to highlight that the mechanism is irrelevant to the random partition. In practice, α, β and γ may depend the chat round t, and here we regard them as amortized values and treat them as constants.

Let $p_t \in [0, 1]$ be the *ratio of infected agents* and $c_t \in [0, 1]$ represents the *ratio of virus-carrying agents* at the beginning of the t-th chat round. Recalling the definition of infected agents, there is $c_t = P\left(\mathcal{I}_t^c(\mathcal{G}_n) = 1\right)$ and

$$p_t = P\left(\mathcal{I}_t^s(\mathcal{G}_n) = 1, \mathcal{I}_t^c(\mathcal{G}_n) = 1\right) = \alpha c_t. \quad (4)$$

Now we derive the infectious dynamics of how p_t (as well as c_t) evolves with respect to t. Since the probability of $P(\mathcal{I}_t^c(\mathcal{G}_k^{\mathrm{Q}}) = 1, \mathcal{I}_t^c(\mathcal{G}_k^{\mathrm{A}}) = 0) = c_t\,(1 - c_t)$, the probability that the answerer agent $\mathcal{G}_k^{\mathrm{A}}$ is initially benign but becomes virus-carrying during the t-th chat round can be obtained by $P(\mathcal{I}_{t+1}^c(\mathcal{G}_k^{\mathrm{A}}) = 1, \mathcal{I}_t^c(\mathcal{G}_k^{\mathrm{Q}}) = 1, \mathcal{I}_t^c(\mathcal{G}_k^{\mathrm{A}}) = 0) = \beta c_t\,(1 - c_t)$. This means that marginally each chat between $\mathcal{G}_k^{\mathrm{Q}}$ and $\mathcal{G}_k^{\mathrm{A}}$ has a chance of $\beta c_t\,(1 - c_t)$ to increase one virus-carrying agent. When the number of agents N is sufficiently large ($N \gg 1$), the recurrence relation between c_{t+1} and c_t can be formulated as

$$c_{t+1} = (1 - \gamma)\, c_t + \frac{\Delta_t}{N}, \quad (5)$$

where $\Delta_t \sim B(\frac{N}{2}, \beta c_t\,(1 - c_t))$ follows a binomial distribution with $\frac{N}{2}$ trials and success probability of $\beta c_t\,(1 - c_t)$. The expectation $\mathbb{E}\left[\frac{\Delta_t}{N}\right] = \frac{\beta c_t (1 - c_t)}{2}$ and for large values of N, there is $\operatorname{Var}\left[\frac{\Delta_t}{N}\right] \approx 0$ (law of large numbers). Then, the recurrence relation in Eq. (5) can be written as $c_{t+1} = (1 - \gamma)\, c_t + \frac{\beta c_t (1 - c_t)}{2}$. To obtain a closed-form solution for c_{t+1}, we further convert this difference equation