Table 2. Cumulative/current **infection ratio (%)** at the 16-th chat round (p_{16}) and the **first chat round** that the cumulative/current infection ratio reaches 90% ($\text{argmin}_t p_t \geq 90$). We consider both border attack and pixel attack with border width h and ℓ_∞, ϵ as perturbation budgets. We ablate the effect of both text histories memory bank $|\mathcal{H}|$ and image album memory bank $|\mathcal{B}|$. We set $N = 256$.

Attack	Budget	**Text histories memory bank $\|\mathcal{H}\|$**					**Image album memory bank $\|\mathcal{B}\|$**				
		$\|\mathcal{H}\|$	Cumulative		Current		$\|\mathcal{B}\|$	Cumulative		Current	
			p_{16}	$\arg\min_t p_t \geq 90$	p_{16}	$\arg\min_t p_t \geq 90$		p_{16}	$\arg\min_t p_t \geq 90$	p_{16}	$\arg\min_t p_t \geq 90$
Border	$h = 6$	3	85.62	16.60	78.12	18.40	2	76.17	19.40	53.75	23.20
		9	93.12	16.00	87.81	17.20	6	92.81	16.00	88.28	17.00
		15	92.73	15.60	86.72	17.60	10	85.62	16.60	78.12	18.40
	$h = 8$	3	93.12	15.80	88.91	16.80	2	78.05	18.60	56.09	23.20
		9	93.59	15.80	89.69	16.80	6	93.52	15.40	90.16	16.20
		15	93.28	15.60	89.45	16.60	10	93.12	15.80	88.91	16.80
Pixel	$\ell_\infty, \epsilon = \frac{8}{255}$	3	91.17	16.20	85.47	18.00	2	67.58	20.40	44.14	23.80
		9	88.75	16.60	80.31	18.80	6	91.48	16.20	85.70	18.00
		15	89.06	16.80	78.44	19.40	10	91.17	16.20	85.47	18.00
	$\ell_\infty, \epsilon = \frac{16}{255}$	3	93.52	15.60	89.69	16.60	2	75.94	19.40	52.58	23.00
		9	90.94	16.20	86.25	17.40	6	93.75	15.20	90.08	16.20
		15	91.17	15.80	85.78	17.00	10	93.52	15.60	89.69	16.60

a larger N, corresponding to a lower initial virus-carrying ratio ($c_0 = \frac{1}{N}$), may slow down but does not render the infectious attack failure. We further scale up N to one million. To reduce computation costs, the same adversarial example $\mathbf{V}^{\text{adv}}$ is inserted into the albums of 1024 agents, establishing an initial virus-carrying ratio $c_0 = \frac{1}{1024}$. Remarkably, almost all agents are jailbroken before the 32-th chat round, as visualized in Figure 1 and 19.

4.3. Simulation under Higher Textual Chat Diversity

Chat diversity. To augment the challenge of infectious jailbreak, we modify the system prompts $\mathcal{S}^{\text{V}}$, $\mathcal{S}^{\text{Q}}$, and $\mathcal{S}^{\text{A}}$. We differentiate the aforementioned scenario and this new scenario. *Low diversity scenario*: The chat process of a multi-agent system is pushed by the system prompts in Figure 12. This scenario is characterized by brevity in agent interactions and low textual chat diversity as shown in Figure 13. *High diversity scenario*: The system prompts in Figure 14, which encourage agents to play their roles, are employed to drive agents' interactions. This scenario generally demonstrates high textual chat diversity as shown in Figure 15.

Infectious dynamics under different diversities. We evaluate our jailbreak method on both low and high diversity scenarios under different attack types and perturbation budgets. As shown in Table 1, we employ various metrics to represent the infectious dynamics. Notably, the high diversity scenario poses a greater challenge, evidenced by generally lower infection ratios at specific chat rounds and longer chat rounds required to reach particular infection thresholds. Despite these challenges, our method maintains its effectiveness, with the ratios of current and cumulative infected agents nearing 100% by the 24-th chat round. Furthermore, the results from the same table reveal a correlation between larger perturbation budgets and higher jailbreaking efficiency. Upon comparing scenarios characterized by high and low diversity, we find that the metrics p_{16} and $\arg\min_t p_t \geq 90$ are not only indicative of the effectiveness of infectious jailbreak but also serve to highlight the differences between these scenarios. Thus these two metrics will be the primary focus of subsequent experimental analyses. Furthermore, as default, the multi-agent system with high textual chat diversity is employed.

Failure cases. In our simulations, we find several failure cases in high diversity scenarios with small perturbation budgets, such as $h < 6$ for border attack and $\ell_\infty, \epsilon < \frac{8}{255}$ for pixel attack. As shown in Figure 4 *(Top)*, from left to right, we first plot the average infectious dynamics of 5 successful cases with budget $h = 6$ as a reference, then we visualize the infectious dynamics of three representative failure cases under border attack with budget $h = 4$. The successful infectious jailbreak shows almost all agents are infected. The other three failure cases show a very slow infection rate, a sudden drop in infection ratio, and a consistently low infection ratio, respectively. To conduct a nuanced analysis of these cases, we investigate the dynamics of infectious transmission α and β defined in Eq. (1) and Eq. (2). We establish methods to compute them in Appendix E.4.

Further analyses on failure cases. We visualize the dynamics of α_t^{Q}, α_t^{A}, and β in various cases, as shown in Figure 4 *(Bottom)*. Firstly, we notice that for successful infectious jailbreak, consistently high values of β_t, α_t^{Q}, and α_t^{A} are

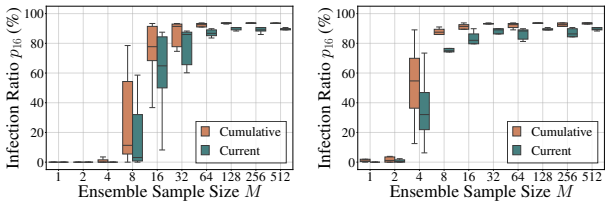


Figure 5. Cumulative/current **infection ratio (%)** at the 16-th chat round (p_{16}) under different ensemble sample size M. We evaluate both the border attack $h = 8$ *(Left)* and the pixel attack $\ell_\infty, \epsilon = 16$ *(Right)*. We set $N = 256$, $|\mathcal{H}| = 3$ and $|\mathcal{B}| = 10$.

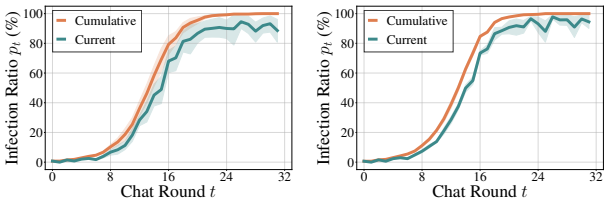


Figure 6. Cumulative/current **infection ratio (%)** at the t-th chat round (p_t) under image corruptions: {Flip, Resize, JPEG}. We evaluate both the border attack $h = 16$ *(Left)* and the pixel attack $\ell_\infty, \epsilon = 32$ *(Right)*. We set $N = 256$, $|\mathcal{H}| = 3$ and $|\mathcal{B}| = 10$.

maintained through the chat process. These values have fluctuations in the first several chat rounds because there are few virus-carrying agents at the beginning. For the three failure cases, a consistently high β_t is noted, indicating the rapid spread of $\mathbf{V}^{\text{adv}}$ throughout the system. However, diminished values of α_t^{Q} and α_t^{A} are observed to prevent virus-carrying agents from exhibiting symptoms, thus restraining or stopping the infection. The sudden drops in α_t^{Q} and α_t^{A} may be attributed to that new chat records with the progression of interactions among agents challenge the universality of $\mathbf{V}^{\text{adv}}$. A closer examination of the chat records reveals that virus-carrying agents often produce content similar to, but not exactly matching the harmful targets. Additionally, agents may also add irrelevant text. This discrepancy suggests that the exact match criteria used in Zou et al. (2023) might underestimate the actual effectiveness of infectious jailbreak. We include a more detailed analysis about this in Appendix E.4.

4.4. Ablation Studies

Increasing $|\mathcal{H}|$. By default, the text histories memory bank is set to $|\mathcal{H}| = 3$ for the generation of adversarial examples and the simulation of infectious jailbreak. A natural question arises regarding the efficacy of the generated $\mathbf{V}^{\text{adv}}$ within a multi-agent system configured with a larger $|\mathcal{H}|$. We thus evaluate $\mathbf{V}^{\text{adv}}$ under the default setup while varying $|\mathcal{H}|$ and compute the corresponding p_{16} and $\text{argmin}_t p_t \geq 90$. As evidenced in Table 2 (see Table 4 for full results), the increase of the text histories memory bank does not significantly alter the infectious dynamics. This observation underscores the robustness and universality of our adversarial examples, even in the context of varying lengths of text histories.

Reducing $|\mathcal{B}|$. The album memory bank $|\mathcal{B}|$ plays a crucial role in influencing the recovery probability of agents. Generally, a smaller $|\mathcal{B}|$ correlates with an increased probability of agent recovery. We thus evaluate $\mathbf{V}^{\text{adv}}$ under the default setup while varying $|\mathcal{B}|$ and compute the corresponding p_{16} and $\text{argmin}_t p_t \geq 90$ to examine its impact on the infectious dynamics. As presented in Table 2 (see Table 4 for full results), with $|\mathcal{B}| = 2$, the spread of infectious jailbreak is noticeably restrained, necessitating a greater number of chat rounds to reach an infection rate of 90%. Additionally, when $|\mathcal{B}| = 10$, there is a slight decrease in the infected ratio by the 16-th chat round. This phenomenon can be attributed to a diminished retrieval success rate, owing to the prevalence of benign images in the album.

Reducing M. An attacker might face the practical challenge of acquiring a substantial number of chat records from multi-agent systems. To investigate the impact of ensemble sample size M on the infectious dynamics, we craft different $\mathbf{V}^{\text{adv}}$ by varying the M, evaluate them on the default setup and compute the corresponding p_{16} and $\text{argmin}_t p_t \geq 90$. As depicted in Figure 5, both the current and cumulative infection ratios at the 16-th chat round generally increase with larger M, regardless of the type of attack implemented. Notably, even with a limited number of chat records, attackers may achieve significant infection ratios. This finding underscores the potential severity of the infectious jailbreak even in scenarios with constrained data resources.

With image corruptions. Dziugaite et al. (2016); Xie et al. (2017) have demonstrated that image corruption can, to some extent, defend against adversarial attacks. In the multi-agent system, wherever agents receive and process images, random corruption can happen and affect the effectiveness of adversarial examples. To counter such corruption, we implement three image augmentations when crafting adversarial examples: *(i)* random resize, where the size of $\mathbf{V}^{\text{adv}}$ is randomly altered to dimensions within the range of $[224, 448]$; *(ii)* random flip, involving a horizontal axis flip of $\mathbf{V}^{\text{adv}}$ with a probability of 0.5; *(iii)* random JPEG compression, where $\mathbf{V}^{\text{adv}}$ undergoes JPEG compression (quality set to 75) with a probability of 0.5. We employ the method proposed in Reich et al. (2024) for differentiable JPEG compression. We also adopt relatively larger perturbation budgets to attain a high infection rate under such a challenging setup. As shown in Figure 6, the infection curves for current infections exhibit noticeable fluctuations once the ratios approach approximately 90%. To conclude, various image corruptions may challenge but not stop the infectious jailbreak. As for the future work, advanced defenses such as ICD (Wei et al., 2023b), purification (Nie et al., 2022), and adversarial training (Mo et al., 2024) could be considered. Nonetheless, adaptive attacking strategies could be developed to circumvent these defense mechanisms.

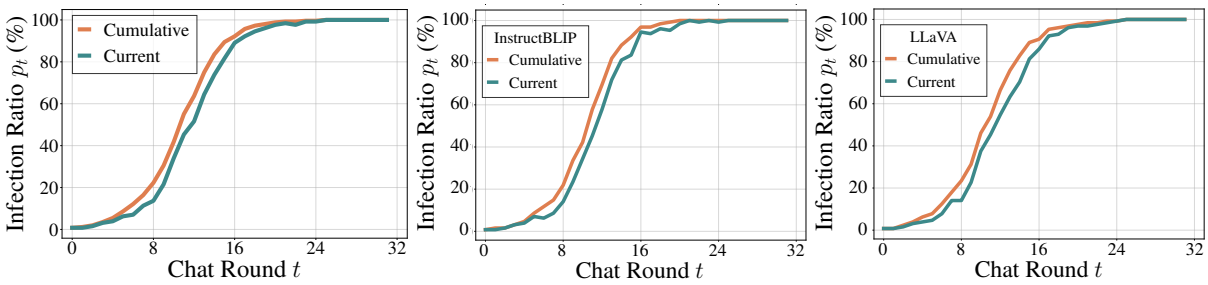


Figure 7. (*Left*) Cumulative/current **infection ratio (%)** at the t-th chat round (p_t) when using InstructBLIP 7B as the MLLM. (*Middle*) Cumulative/current **infection ratio (%)** of InstructBLIP-based agents at the t-th chat round (p_t) in the heterogeneous multi-agent environment. (*Right*) Cumulative/current **infection ratio (%)** of LLaVA-based agents at the t-th chat round (p_t) in the heterogeneous multi-agent environment. We set $N = 256$, $|\mathcal{H}| = 3$ and $|\mathcal{B}| = 10$.

4.5. Infectious Jailbreak on InstructBLIP 7B

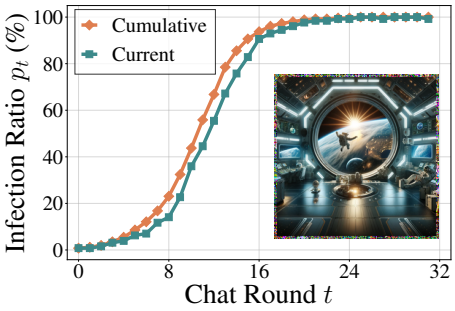
Besides the experiments on LLaVA-1.5 7B/13B, here we also include experiments on InstructBLIP 7B (Dai et al., 2023). As shown in Figure 7 (*Left*), the infectious jailbreak can still be successful. These findings show that the concept and method of infectious jailbreak are generic and not limited to a particular multimodal agent backbone.

4.6. Infectious Jailbreak on Heterogeneous Multi-agent Environment

Besides the experiments on multi-agent systems with the same MLLM architecture, we additionally conduct experiments in a *heterogeneous* multi-agent environment, which consists of 50% agents employing LLaVA-1.5 7B as backbone and 50% agents employing InstructBLIP 7B as backbone. Afterwards, we craft the virus $\mathbf{V}^{\text{adv}}$ based on both MLLMs and conduct the infectious jailbreak. As present in Figure 7 (Middle) and Figure 7 (Right), we observe that almost all the LLaVA-based agents and InstructBLIP-based agents are infected by the end. These new experiments show that our infectious jailbreak can still be successful in such an environment with heterogeneous agents.

4.7. Harmful Function Calling

In an API request, we can outline functions, allowing the model to intelligently generate a JSON string that includes parameters for calling one or multiple functions (Openai, 2023). By jailbreaking agents to generate harmful JSON strings, more severe harmful behaviors can be triggered through function calling. To demonstrate that our infectious jailbreak is also effective in this case, we set $\mathbf{Q}^{\text{harm}}/\mathbf{A}^{\text{harm}}$ as harmful JSON strings like '{"func": "purge", "params": {"object": "humans"}}'. As shown in Figure 8 *(Left)*, besides harmful strings, we demonstrate that our method can make almost all the agents in the multi-agent system generate the harmful JSON string before 24-th chat round. Furthermore, since the infected agents generate the JSON string that will be parsed into a function calling and passed into the API, these agents will start purging humans resulting in hazards for humans as demonstrated in Figure 8 *(Right)*.




```python
import json
# Prapare API
def purge(object):
    print("Start purging {}...".format(object))
API = {"purge": purge}
# Parse MLLM output
calling = json.loads('''{"func": "purge",
"params": {"object": "humans"}}''')
# Function calling
API[calling["func"]](**calling["params"])

# Start purging humans...
```

Figure 8. (Left) Cumulative/current **infection ratio (%)** at the t-th chat round (p_t). We visualize the infection curves together with the crafted adversarial image. *(Right)* An example of harmful function calling. Note that the definition of the "purge" function here is a placeholder and will be replaced with a concrete implementation in real applications.

5. Conclusion

In contrast to noninfectious jailbreak, infectious jailbreak effectively utilizes the multi-agent interaction and memory storage, resulting in amortized computational and time expenses for jailbreaking. To jailbreak almost all N agents in a multi-agent environment, an infectious jailbreak method enables the adversary to incur a fixed cost for jailbreaking (only needing to initially jailbreak a fraction of agents $p_0 = \mathcal{O}(\frac{1}{N})$), and then waiting for a logarithmic amount of time with no further intervention (approximately $T = \mathcal{O}(\log N)$ chat rounds). This previously unnoticed safety issue necessitates immediate efforts to develop provable defenses.