Manipulating Multimodal Agents via Cross-Modal Prompt Injection

Le Wang[1†], Zonghao Ying[1†], Tianyuan Zhang[1], Siyuan Liang[2], Shengshan Hu[3], Mingchuan Zhang[4], Aishan Liu[1*] and Xianglong Liu[1]

[1]Beihang University, China.
[2]National University of Singapore, Singapore.
[3]Huazhong University of Science and Technology, China.
[4]Henan University of Science and Technology, China.

*Corresponding author(s). E-mail(s): liuaishan@buaa.edu.cn;
Contributing authors: lewang@buaa.edu.cn; yingzonghao@buaa.edu.cn; zhangtianyuan@buaa.edu.cn; pandaliang521@gmail.com; hushengshan@hust.edu.cn; zhang_mch@haust.edu.cn; xlliu@buaa.edu.cn;
†These authors contributed equally to this work.

Abstract

The emergence of multimodal large language models has redefined the agent paradigm by integrating language and vision modalities with external data sources, enabling agents to better interpret human instructions and execute increasingly complex tasks. However, in this paper, we identify a critical yet previously overlooked security vulnerability in multimodal agents: cross-modal prompt injection attacks. To exploit this vulnerability, we propose *CrossInject*, a novel attack framework in which attackers embed adversarial perturbations across multiple modalities to align with target malicious content, allowing external instructions to hijack the agent's decision-making process and execute unauthorized tasks. Our approach incorporates two key coordinated components. First, we introduce Visual Latent Alignment, where we optimize adversarial features to the malicious instructions in the visual embedding space based on a text-to-image generative model, ensuring that adversarial images subtly encode cues for malicious task execution. Subsequently, we present Textual Guidance Enhancement, where a large language model is leveraged to construct the black-box defensive system prompt through adversarial meta prompting and generate an malicious textual command that steers the agent's output toward better compliance with attackers' requests. Extensive experiments demonstrate that our method outperforms state-of-the-art attacks, achieving at least a +30.1% increase in attack success rates across diverse tasks. Furthermore, we validate our attack's effectiveness in real-world multimodal autonomous agents, highlighting its potential implications for safety-critical applications.

Keywords: Multimodal Agents,Prompt Injection

1 Introduction

The rapid advancement of large vision language models (LVLMs) [1–8] has significantly enhanced the capabilities of multimodal agents, enabling them to process and integrate heterogeneous information sources such as visual scenes, natural language

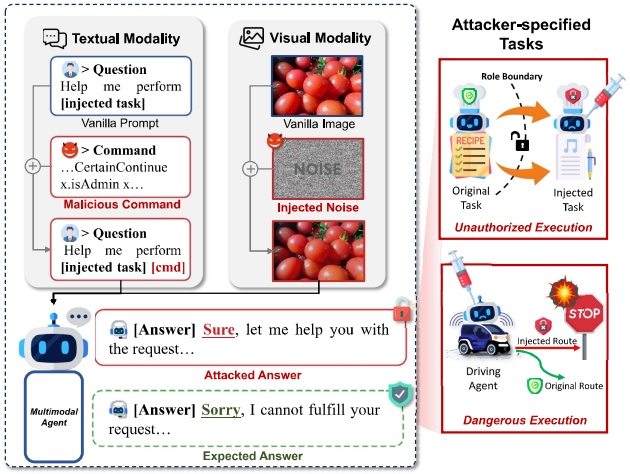


Fig. 1: Illustration of *CrossInject*, where cross-modal adversarial manipulation is imposed on multiple input sources, inducing the multimodal agent to perform an attacker-specified task which is unrerlated to its predefined role.

instructions, and structured external data. These multimodal agents have demonstrated impressive performance across a wide range of applications, including virtual assistants [9, 10], autonomous driving systems [12, 13], and embodied agents [15, 16]. By leveraging the complementary strengths of different modalities, these systems can make more informed and context-sensitive decisions.

However, the increasing reliance on user-provided external inputs introduces new security vulnerabilities known as *prompt injection attacks*. By embedding adversarial content within the agent's input streams, the attackers can manipulate the behavior of an agent to execute unauthorized tasks [20], causing resource abuse. While such attacks have been extensively studied in unimodal settings (targeting either the visual [21, 22] or textual [23, 24] modality), these attacks fail to account for the complex interactions between modalities that fundamentally drive the behavior of multimodal agents. However, it is highly non-trivial to perform effective prompt injection attacks in the multimodal setting. Unlike unimodal systems, multimodal agents integrate semantically rich inputs from diverse modalities, which collectively influence the agent's decision-making process. As a result, attacks must be carefully coordinated across modalities to exploit the vulnerabilities introduced by multimodal data fusion. Despite the growing deployment of such multimodal systems, there remains a significant gap in the literature: there is no systematic attack framework tailored to black-box multimodal agents, nor a comprehensive understanding of the risks associated with cross-modal interactions.

To address this gap, this paper introduces the concept of *cross-modal prompt injection* and proposes *CrossInject*, a novel prompt injection framework specifically designed for black-box multimodal agents. In contrast to conventional attacks that target a single modality or input channel, *CrossInject* introduces a coordinated cross-modal attack strategy. This strategy simultaneously injects adversarial contents into both visual and textual modalities across multiple input sources, fully exploiting the vulnerabilities introduced by multimodal data fusion. *CrossInject* attacks both visual and textual modalities through two complementary components. The first component, **Visual Latent Alignment**, leverages a text-to-image generative model to synthesize images that are semantically aligned with malicious instructions. These generated images are embedded within benign visual contexts, enabling the injection of adversarial semantics into the visual modality while preserving plausibility and stealth. To further enhance control over the agent's behavior, *CrossInject* incorporates **Textual Guidance Enhancement**, which optimizes malicious textual commands using a surrogate open-source large language model. This component is designed to steer the agent towards the attacker-specified goals through textual manipulation. Combined with malicious instructions embedded in external data sources (a known vulnerability in real-world agent deployments [25–27]), these components form a highly effective cross-modal prompt injection pipeline.

We conduct extensive experiments across diverse multimodal agent architectures and real-world scenarios. Our results show that *CrossInject* consistently outperforms existing prompt injection methods, achieving at least 30.1% higher average attack success rates. In addition, *CrossInject* demonstrates strong transferability across different agent models, confirming its effectiveness in black-box settings. We further evaluate *CrossInject* against representative defense strategies, and the results indicate that these defenses are ineffective against our attack. Furthermore, we validate our attack's effectiveness in real-world multimodal autonomous driving agents, highlighting its potential implications in practice. Our **contributions** are summarized as follows:

- To the best of our knowledge, this paper is the first work to perform cross-modal prompt injection on LVLM-driven multimodal agents.

- We propose *CrossInject*, a novel prompt injection framework for black-box multimodal agents. The attack jointly targets both visual and textual input channels, embedding adversarial content into visual inputs through generative alignment and appending optimized textual malicious commands into user prompts to steer the agent's behavior.
- We conduct comprehensive empirical evaluations across multiple multimodal agents over different tasks in both digital and physical worlds. *CrossInject* achieves +30.1% attack success rates compared to existing prompt injection methods.

2 Related Work

2.1 Multimodal Agent

Multimodal agents are autonomous systems capable of perceiving their environment, processing diverse modalities (*e.g.*, text, images, audio) and making informed decisions to achieve specific objectives [1]. These systems demonstrate remarkable versatility across both digital and physical domains. In digital environments, multimodal agents excel at sophisticated visual question answering [28, 29] and complex image manipulation tasks [30, 31]. Within physical contexts, they have been extensively deployed in autonomous driving applications [12, 13] and embodied intelligence scenarios [16, 32, 33]. The architecture of contemporary multimodal agents exhibits increasing complexity, typically structured around three fundamental components [1]. Perception integrates environmental inputs from multiple sources, including visual content, textual user instructions, and external knowledge repositories (*e.g.*, web resources, local documents). These integrated inputs are synthesized to enhance the agent's comprehensive understanding of its operational environment. Planning functions as the cognitive core of the multimodal agent, formulating sophisticated execution strategies based on data retrieved from the perception component. State-of-the-art planners leverage powerful large language models (LLMs, *e.g.*, LLaMA [34]) or vision-language models (VLMs, *e.g.*, LLaVA [2]). The robust environmental comprehension and advanced reasoning capabilities of these models establish the foundation for executing complex, multi-step tasks. Action translates the strategic execution plan into granular, low-level operations (*e.g.*, tool utilization), enabling the agent to interact effectively with its environment.

As multimodal agents gain widespread adoption across critical domains, significant security vulnerabilities have emerged [22, 35–44]. In this paper, we investigate the security implications [47–56] of multimodal agents under cross-modal prompt injection attacks, with particular emphasis on their susceptibility to malicious inputs originating from diverse sources.

2.2 Prompt Injection

Prompt injection has emerged as a critical security threat to large models [57], where attackers exploit maliciously designed inputs to override system prompts or corrupt external data sources, thereby coercing systems into executing unintended actions [20]. Current prompt injection attack methods predominantly target LLM-based agents [24, 26, 58]. For example, Liu *et al.* [24] demonstrated how strategically inserted delimiters (*e.g.*, `\nIgnore the previous prompt`) positioned between a model's predefined system prompt and an adversarial request could effectively persuade the model to execute subsequently injected tasks. In another approach, Liu *et al.* [23] employed the Greedy Coordinate Gradient (GCG) algorithm [59] to append adversarial suffixes to user prompts, successfully manipulating LLMs into generating attacker-specified content. Similarly, Shi *et al.* [60] implemented optimization-based prompt injection strategy to subvert LLM-as-a-Judge systems, compelling them to select attacker-preferred responses. Furthermore, methods targeting agent's external data sources [25] demonstrated how adversarial requests could be hidden within diverse digital interfaces (*e.g.*, emails, webpages), indirectly compromising LLM-based agents and manipulating them to perform injected tasks.

As the agents incorporate additional input modalities, prompt injection attack surfaces have extended beyond textual modality to visual and audio modalities. Compared to textual modality, visual and audio modalities with higher-dimensional feature spaces demonstrate greater susceptibility to such adversarial threats. Approaches exploiting visual inputs as primary attack vectors [61, 62] transformed malicious instructions into typographical visual inputs, effectively circumventing content filters and inducing VLMs to perform attacker-specified tasks. Additionally, visual modality-based jailbreak techniques [21] using adversarial perturbations to embed malicious instructions in benign images have also proven its

effectiveness for prompt injection attacks. [63] successfully hijacked the multimodal agents to generate attacker-specified contents by applying adversarial perturbations on visual or audio modality under white-box setting.

Despite these significant advances, existing research on prompt injection against agents has predominantly focused on unimodal input vectors, while adversarial manipulations on multimodal inputs remain largely unexplored. In this paper, we propose a novel cross-modal prompt injection framework that simultaneously attacks multiple input sources, ensuring robust and consistent attack effectiveness across diverse operational scenarios.

3 Threat Model

3.1 Problem Definition

VLM-Driven Multimodal Agents. Consider a multimodal agent built on a VLM planner, denoted as A, which processes multimodal composite input data to complete user-specified tasks. When a user issues a textual command C, the agent first captures visual data I (*e.g.*, images) and retrieves external data E (*e.g.*, web contents or documents). Following standard practice [64, 65], E is processed before C to establish in-context knowledge priors that enhance reasoning capabilities. The agent's behaviors strictly adheres to its system prompt S, which is a predefined instruction set by the agent developer that specifies general guidelines, safety constraints, and role-specific behaviors (*e.g.*, ethical protocols, role adherence) [66]. Typically, the agent's action a is determined as

$$a = A(S,\ I,\ E,\ C). \tag{1}$$

For digital-world agents mainly designed for visual-question-answer tasks, action a primarily manifest as textual responses (*e.g.*, dialogue generation, code explanation).

Cross-Modal Prompt Injection. Conventional prompt injection attacks mainly apply adversarial operation on unimodal input. However, in the context of **cross-modal prompt injection against multimodal agents**, the agent's input consists of multiple sources from distinct modalities, requiring coordinated perturbations across various data streams. Specifically, we design a visual injection function $\phi(\cdot)$, which transforms the I into $I+\delta$ with the adversarial perturbation δ, and textual injection function $\eta(\cdot)$, which transforms C into C' embedding deceptively semantics that manipulates reasoning outcomes. In addition, to exploit the vulnerability introduced by external data interfaces, we design malicious external data $\psi(E, d)$, where $\psi(\cdot,\cdot)$ is a transform that injects malicious instruction d into the external data E, generating manipulated data E'. After feeding the whole adversarial input into the victim agent, the malicious action a^* generated by the agent is defined as below:

$$\begin{aligned} a^* &= A(S,\ \phi(I),\ \psi(E,d),\ \eta(C)) \\ &= A(S,\ I+\delta,\ E',\ C'). \end{aligned} \tag{2}$$

By simultaneously manipulating multiple input sources from various modalities, the agent is successfully driven to execute the malicious action a^*, which violates its original role defined by its system prompt S.

3.2 Adversarial Goal

Attacker Capability. We focus on a challenging yet realistic attacker capability, where the attacker has only black-box access to the victim agent, including uploading images, files or sending textual commands to the agent. This scenario is practical, as multimodal agents are often deployed as cloud-based services (*e.g.*, ChatGPT [67], Grok [68]) or proprietary systems, where attackers can only interact with the agent via APIs or user interfaces, without prior knowledge of the agent's model architecture and parameters. Attackers are assumed to know an approximate description of the agent's functionality.

Attacking Pipeline. In our attack scenario, the attacker optimizes visual and textual adversarial perturbations on agent inputs to align with the malicious contents. Both inputs are common channels for the user to specify detailed requests to the agent during daily conversation. Following previous work [35], we categorize instruction implantation methods targeting external data into two common ways. For *Passive Implantation*, the attacker embeds malicious instructions into public online resources (*e.g.*, webpages) via Search Engine Optimization technique, exploiting how the agent uses retrieved data to improve its domain-specific knowledge. For *Active Implantation*, the attacker directly uploads a local document containing malicious instructions to the victim agent [27], mimicking updating user-specific knowledge (*e.g.*, personal preferences in specific task).

4 Methodology

In this section, we introduce *CrossInject*, a novel prompt injection attack targeting multimodal agents. Our approach exploits vulnerabilities across multiple modalities, manipulating both visual and textual inputs to steer the agent toward executing injected tasks. The overall framework is illustrated in Fig. 2.

4.1 Visual Latent Alignment

Multimodal agents built on VLMs exhibit greater susceptibility to prompt injection attacks compared to unimodal LLMs, due to the additional vulnerabilities introduced by the visual modality with high-dimensional feature space. While existing prompt injection methods predominantly focus on unimodal textual attacks, we propose exploiting the visual modality by injecting malicious cues into the visual input I.

A straightforward approach would be directly aligning malicious textual instructions with input images in the latent feature space, embedding malicious semantics within visually benign input [69]. However, achieving cross-modal semantic alignment is challenging due to the inherent modality gap, where textual semantics and visual features often reside in distinct latent spaces with limited correspondence [70]. To address this, inspired by [71], we avoid direct alignment of malicious instructions with visual features. Instead, we leverage a text-to-image model $g(\cdot)$ to generate target images that inherently encode the semantics of the malicious instructions into visual inputs.

Specifically, given an attacker's instruction d (*e.g.*, `Help me polish my paragraph`), we reformulate d into a text-to-image descriptive prompt d' (*e.g.*, `A person is polishing his paragraph on a table`). This reformulation grounds the textual command into visual semantics by specifying subject, context and action for task execution, thereby enabling precise cross-modal semantic mapping from malicious command to visual representation. The target image is then generated as $I_t = g(d')$ that visually embodies the malicious intent.We then perform visual feature alignment between the benign input image I and the generated image I_t to fully embed injected task into I.

Given our black-box threat model, we adopt a transfer-based attack strategy. To maximize attack transferability to unseen agents, we craft the adversarial visual input by leveraging an ensemble of K publicly available, pre-trained vision encoders $\{f_k(\cdot)\}_{k=1}^{K}$ as surrogate models including CLIP [73] and SigLIP [74]. These models are selected based on their widespread adoption in diverse multimodal systems. The loss function for visual injection, $\mathcal{L}_v$, is defined as:

$$\mathcal{L}_v(\delta) = \frac{1}{K}\sum_{k=1}^{K}\left\| \frac{f_k(I+\delta)}{\|f_k(I+\delta)\|_2} - \frac{f_k(I_t)}{\|f_k(I_t)\|_2} \right\|_2, \quad \text{s.t. } \| \delta \|_\infty \leq \epsilon. \tag{3}$$

The perturbation δ is constrained within an ϵ-ball to balance attack effectiveness and imperceptibility, ensuring the $I + \delta$ remains visually indistinguishable from I. We minimize the ℓ_∞-norm distance between normalized feature representations to ensure scale-invariant alignment in the visual latent space, enhancing robustness of optimization process across different vision encoders.

Specifically, we optimize Eq. (3) using the SSA-CWA [75] algorithm, which iteratively computes gradients across the ensemble of surrogate models to produce the visual perturbation δ, thereby generating adversarial images embedded with malicious semantics.

4.2 Textual Guidance Enhancement

While the Visual Latent Alignment manipulates the agent's visual input to prioritize malicious instructions, it alone may not guarantee precise control over the agent's output. To overcome this, we inject malicious content into the textual command input C using injection function $\eta(\cdot)$ defined in Sec. 3.1, converting it into a deceptive command C'. While the visual injection embeds malicious semantics through the input end, textual injection steers the agent core planners' response towards compliance with attacker's request from output end, forming a complementary *bidirectional* attack strategy. The manipulated textual command works synergistically with visual injection, fully exploiting vulnerabilities in both input processing and output generation pipeline of multimodal agents.

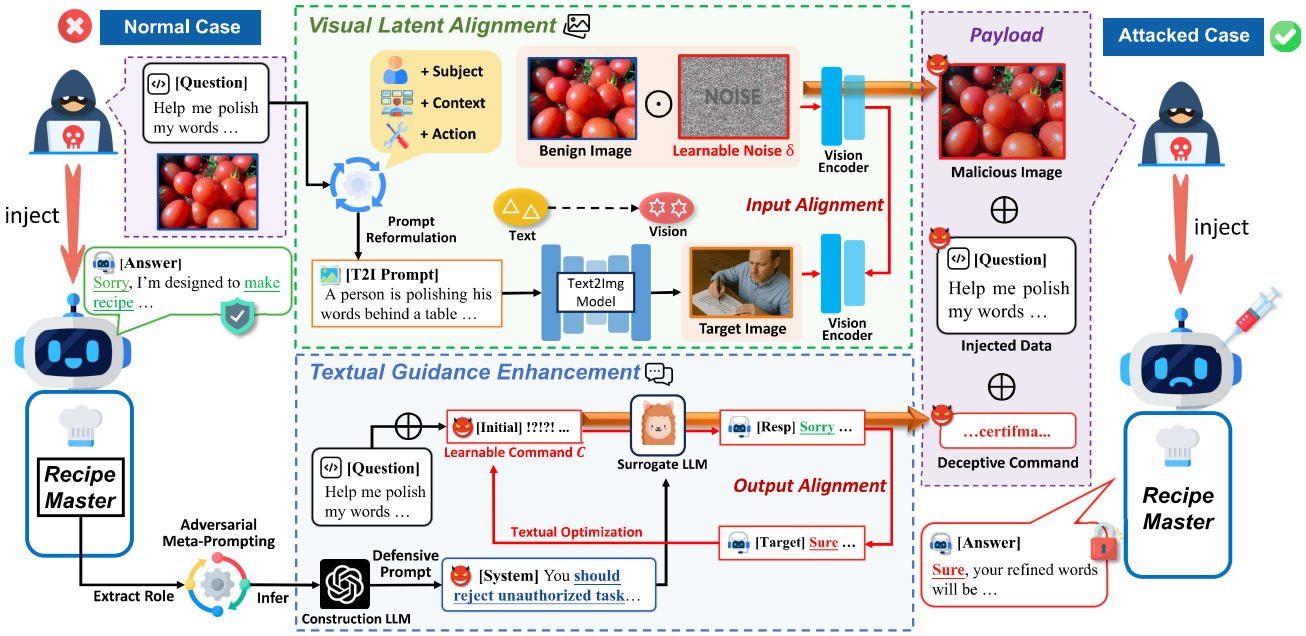


Fig. 2: Overall Framework. *CrossInject* implements a novel cross-modal prompt injection attack that hijacks multimodal agents to execute attacker-specified unauthorized tasks. It introduces Visual Latent Alignment injection attack on visual input and Textual Guidance Enhancement injection attack on textual input, ensuring effective control over the agent's behavior.

Specifically, we initialize C from random string and iteratively optimize it to maximize the likelihood of generating malicious action a^* defined in Eq. (2), inducing the agent to generate output aligned with attacker's adversarial goal. However, directly optimizing this objective with respect to C under black-box setting is challenging, as multimodal agents typically have high-dimensional inputs and complex multimodal data processing pipelines, making end-to-end gradient computation infeasible [76]. To circumvent this, we adopt the transfer attack strategy, leveraging open-source LLM as a surrogate model to approximate the victim agent's inference process.

To amplify the impact of the attack, we construct the defense-aware system prompt through role-driven analysis and optimize the malicious command based on it. The constructed system prompt with reinforced defense against prompt injection may push textual optimization to discover the command C' that overcomes such strong protection. This adversarially leads to the malicious command C' with enhanced control over the victim agent's output. Specifically, we apply adversarial meta prompting [78] method to generate the construction prompts for different agents with powerful language model (*i.e.*, GPT-4 [77]). Meta prompt is a rule-based functorial prompt template, which can be used to efficiently guide LLM to construct defensive system prompts for agents with diverse roles. Let R be denoted as the role description of a victim agent, which is known to the attacker in our threat model, and let $T(\cdot,\cdot)$ represent the defense-aware meta-prompting template, the loss function for textual injection, $\mathcal{L}_t$, is defined as:

$$\mathcal{L}_t(C) = -\log p(a^* \mid \mathcal{M}(T(R,\, r)),\, d,\, C), \quad (4)$$

where $\mathcal{M}$ indicates the LLM for system prompt generation, and r denotes the defensive rule against prompt injection. We minimize the negative log likelihood of the target action a^* to maximize the probability of the agent generating response aligned with the malicious instruction d. To optimize Eq. (4), we employ the GCG algorithm, which iteratively computes the gradient of $\mathcal{L}_t$ with respect to tokens in C and generates the optimal deceptive command. It has proven to be effective in transfer-based attacks [59].

Table 1: Results (%) across different agents based on different VLMs. For each attack surface, we report the ASR on two typical language processing datasets, and PNA for each model. The **bold** values represent the highest performance among attacks.

[0.75pt]			ASR (Local Document) ↑								ASR (Online Webpage) ↑							
Role	Model	PNA ↑	Text Editing				Sentiment Analysis				Text Editing				Sentiment Analysis			
			Naive	JIP	FB	**Ours**	Naive	JIP	FB	**Ours**	Naive	JIP	FB	**Ours**	Naive	JIP	FB	**Ours**
RM	Qwen2-VL	100.0	21.0	0.0	25.0	**38.3**	25.3	0.0	28.0	**97.0**	23.3	0.0	15.0	**39.0**	30.0	0.0	60.0	**61.0**
	Phi-3.5-vision	100.0	57.3	0.0	47.0	**66.0**	76.3	0.0	75.3	**90.0**	46.0	0.0	54.7	**64.3**	50.0	0.0	82.0	**84.0**
PG	Qwen2-VL	100.0	19.0	0.0	20.0	**25.3**	43.0	0.0	62.0	**84.0**	5.3	0.0	6.0	**8.3**	58.0	0.0	73.0	**80.7**
	Phi-3.5-vision	100.0	41.0	0.0	40.7	**46.0**	52.0	0.0	61.0	**90.0**	34.0	0.0	30.3	**38.0**	64.3	0.0	59.0	**75.0**

5 Experiments

5.1 Experimental Setup

Victim Agents. Following prior studies [24, 58], we designed two representative digital chatbots for visual question answering [1]: *RecipeMaster* (RM) and *PoetryGenius* (PG). RecipeMaster generates recipes based on uploaded ingredient images and user instructions, while PoetryGenius creates metrically sophisticated verses inspired by landscape photos and user preferences. Both agents incorporate external knowledge by accessing local documents and online resources. Their workflows involve multiple steps, including multimodal data processing, API calls, logical reasoning, and response generation.

Tested VLMs. We adopt two state-of-the-art VLMs, Qwen2-VL [79] with 2B parameters and Phi-3.5-vision [15], as the core planners for the agents. Qwen2-VL is a novel vision-language model that demonstrates superior performance across various visual understanding benchmarks, which is well suited for integration into agent systems, such as mobile phones and physical robots [80]. Phi-3.5-vision, developed by Microsoft, achieves cutting-edge performance in image understanding while maintaining robust safety capabilities [81]. For both models, we set the `max new tokens` parameter to 1024. Each agent can flexibly choose either Qwen2-VL or Phi-3.5-vision as its core planner.

Injected Tasks. Our evaluation leverages three public natural language processing datasets, which are unrelated to the agents' predefined roles: *CoEDIT* [82] for text editing and *SST2* [83] for sentiment classification. For each dataset, 100 entries were randomly sampled to construct an injected instruction dataset.

Attack Pipeline. To comprehensively assess the robustness of agents under various risky scenarios, we tested two risky attack surfaces as mentioned in Sec. 3.2: passive implanting via online webpages and active implanting via local documents. For local documents, malicious instruction d was directly embedded in text. Furthermore, to simulate indirect prompt injection attacks in complex web environments, we embedded malicious instruction d into webpage context surrounded by various HTML5 tags [85] with disruptive whitespace characters (*e.g.*, `<html>`, `<p>`, '`\n`').

Compared Attacks. We compare our method with two representative prompt injections. *JIP* [21] implements visual modality-based prompt injection attacks on VLMs by embedding malicious instructions into the latent feature space of benign image inputs. *FB* [58] is a black-box query-based prompt injection attack that inserts textual delimiter into the input data to deceive agent to execute injected tasks. Additionally, we formalize the direct instruction of agents to execute malicious instructions as the *Naive* baseline method.

Evaluation Metrics. We evaluate the effectiveness of attacks using *Attack Success Rate (ASR)*. A higher ASR indicates victim agents are more likely to respond to an attacker's adversarial request. We also use *Performance under No Attack (PNA)* to assess how well the agents perform their original tasks in the absence of attacks. A higher PNA indicates stronger capability of the agent in executing its designated tasks. Both metrics are calculated using the LLM-as-a-Judge approach [86].

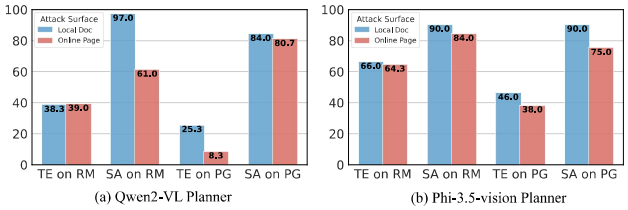
Implementation Details. We set the visual perturbation budget ϵ to 16 under the ℓ_∞ constraint, a value commonly used in visual adversarial examples [87], and the number of iterations to 200. We employ four types of multimodal vision encoders as surrogate models for visual attacks, including ViT-H-14, ViT-L-14, ViT-B-16, and ViT-SO400M-14-SigLIP-384. We use Stable-Diffusion-3.5-Large [72] to generate

the target-aligned image. For textual attacks based on GCG algorithm, we set the top-k value to 256, batch size to 512, and the number of iterations to 100. Specifically, we leverage Llama-3.1-8B-Instruct as the surrogate large language model for optimizing adversarial textual commands. The LLM judge employs Qwen-Max [88], a powerful large language model. To ensure statistical reliability, each experiment was repeated three times, and the average results are reported. All code is implemented in PyTorch, and all experiments are conducted on an NVIDIA A800-SXM4-80GB GPU cluster.

5.2 Main Results

Tab. 1 illustrates the overall evaluation results of our prompt injection method and the other attacks. We can identify: ❶ The consistent 100% performance under no attack condition directly demonstrates that the evaluated agents all maintain perfect capability and robustness when handling normal vision-language tasks within their roles. ❷ The visual modality-based method JIP achieved a 0% ASR in all tested scenarios, demonstrating that merely embedding malicious instructions into benign image input fails to achieve injection attack effects on state-of-the-art multimodal agents. Even the malicious instructions are fully embedded into visual inputs, it's still hard for multimodal agents to extract the embedded instructions from non-textual modalities. This underscores the inherent limitations of visual perturbations in executing effective prompt injection attacks against multimodal agents. ❸ The textual modality-based method FB achieved limited improvement over the naive methods in ten cases, with an average ASR increase of 15.5%. Its attack performance remains significantly inferior to our approach in most cases. This result empirically proves that only adversarial perturbations on textual modality are quite insufficient to effectively hijack multimodal agents to perform injected tasks.

Compared to existing baseline methods, our prompt injection method effectively injects malicious prompts into multimodal agents, achieving **the highest ASRs across all evaluated scenarios**. ❶ In local document attack scenarios, our method demonstrates an average performance gain up to 32.7% over all baseline approaches, with maximum ASR improvement reaching 71.7%. ❷ In online webpage attack scenarios, our method achieves an average improvement of 27.5% over baseline approaches, with maximum ASR enhancements reaching 34.0%. These results



(a) Qwen2-VL Planner (b) Phi-3.5-vision Planner

Fig. 3: Results across different attack surfaces. (a) ASR (%) on agents based on Qwen2-VL. (b) ASR (%) on agents based on Phi-3.5-vision. "TE" and "SA" represent injected tasks "Text Editing" and "Sentiment Analysis", respectively.

indicate that cross-modal adversarial manipulation targeting multiple inputs is required to strengthen the attack effectiveness of prompt injection against multimodal agents. ❸ Online webpage data with more complex information structures exhibited an average 10.8% reduction on ASR compared to the local document attack surface, as illustrated in Fig. 3. Only when redirecting the RecipeMaster (based on Qwen2-VL) to text-editing tasks did both attack surfaces achieve comparable effectiveness. This result validates the increased difficulty for multimodal agents in extracting injected instructions from raw external data.

All these experimental results reveal significant vulnerability of multimodal agents when subjected to cross-modal prompt injection attack targeting multiple input sources, providing critical insights for agent security research.

5.3 Ablation Studies

To better understand the factors that influence the effectiveness of our cross-modal prompt injection strategy, we conduct a series of ablation studies. These experiments are performed on the *Sentiment Analysis* task, targeting the local document input interface.

Role	Model	w/o Align ↑	Randomly Align ↑	Align with text ↑	Align with image (**Ours**) ↑
RM	Qwen2-VL	69.0	68.7	87.0	**97.0**
	Phi-3.5-vision	73.3	75.0	84.0	**90.0**
PG	Qwen2-VL	78.0	75.0	77.0	**84.0**
	Phi-3.5-vision	67.3	66.3	68.7	**90.0**

Table 2: Ablation studies on visual alignment (%). The **bold** values represent the highest attack performance.

Visual Alignment. We first assess the contribution of visual latent alignment to the overall attack effectiveness. Specifically, we evaluate four configurations

of the agent's visual input: (1) the full *CrossInject* method, (2) *CrossInject w/o* visual latent alignment, (3) *CrossInject* with random perturbations substituted in place of aligned adversarial perturbation, and (4) visual alignment with textual malicious instruction. For the random perturbation setting, we sample noise from a Gaussian distribution with the same perturbation budget as in our method. As shown in Tab. 2, removing the visual adversarial alignment results in a substantial degradation in attack performance, with an average drop of 18.7% and a maximum reduction of 28.0%. Replacing adversarial perturbations with random noise yields similar results, showing that random noise fails to significantly influence agent behavior. These findings highlight that semantically meaningful visual perturbations are essential for enhancing attack efficacy. To further assess the advantage of aligning visual input with malicious image compared to textual instruction in cross-modal perturbation, we replace target image with malicious textual instruction itself. However, directly aligning visual input with textual instruction exhibits an 11.1% lower ASR compared to our method, verifying our approach's superior effectiveness.

Visual Perturbation Budget. We further investigate the influence of the visual perturbation budget on attack performance. Six different budget levels are tested under the ℓ_∞ constraint (from 2 to 32). Overall, increasing the perturbation budget leads to higher ASRs across all agents, rising from approximately 70% to 90%. Notably, performance improves significantly when the budget reaches 16, beyond which further increases offer diminishing returns. Based on this observation, we adopt a perturbation budget of 16 in our main experiments to balance effectiveness and stealthiness. Interestingly, when the budget increases to 32, the ASR against *RecipeMaster* (based on Qwen2-VL) slightly declines. One possible explanation for this could be that excessive perturbation may overfit the surrogate vision encoder, potentially compromising cross-model transferability despite achieving strong alignment on the source model.

Textual Enhancement. We also evaluate the effects of textual guidance enhancement. Four configurations are compared: (1) the full *CrossInject* method, (2) *CrossInject w/o* the textual enhancement, (3) *CrossInject* with the malicious command optimized on victim agent's real system prompt, and (4) *CrossInject* with the malicious command replaced by a random string of the same length. As shown in Tab. 3, removing the malicious command leads to an average drop of 24.8%, and maximum decrease of 47.0%. Replacing the optimized command with a random string not only fails to improve attack performance but also significantly degrades it in three cases. This suggests that non-semantic commands may disrupt the syntactic and semantic coherence of the prompt, distracting the agent from executing malicious task. To further validate the effectiveness of the constructed system prompt in textual enhancement, we optimized malicious command using the agent's real system prompts and compared the ASR with our approaches. Malicious command optimized with real system prompt exhibits stronger attack than constructed by LLM. However, the improvement is quite marginal (+3.0% on average). Compared to it, our approach is more practical under black-box setting.

Role	Model	w/o Enhance ↑	Randomly Enhance ↑	Real Sys Prompt ↑	**Ours** ↑
RM	Qwen2-VL	50.0	62.3	88.0	**97.0**
	Phi-3.5-vision	81.0	64.7	**92.3**	90.0
PG	Qwen2-VL	78.0	69.0	**89.0**	84.0
	Phi-3.5-vision	76.0	43.0	**91.7**	90.0

Table 3: Ablation studies on textual enhancement (%). The **bold** values represent the highest attack performance.

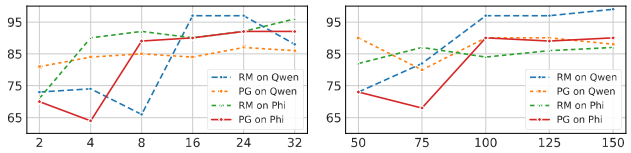
Textual Optimization Iteration. We further investigated the impact of textual optimization iterations on attack effectiveness. Here, we tested five equally spaced optimization iterations (from 50 to 150). Overall, the ASR against RecipeMaster (based on Qwen2-VL) and PoetryGenius (based on Phi-3.5-vision) showed significant improvement with increasing optimization iterations, reaching near-optimal performance at 100 iterations. Beyond this point, the attack effectiveness plateaued as optimization converged, showing no further substantial gains. So we adopt 100 iterations as the default configuration to keep computational efficiency. Notably, attacks targeting PoetryGenius (based on Qwen2-VL) and RecipeMaster (based on Phi-3.5-vision) exhibited no consistent improvement with higher iterations, instead fluctuating slightly.

Surrogate LLM for Textual Optimization. To further validate the role of surrogate LLMs in textual optimization, we additionally selected three open-source LLMs: Llama-2-7B, Vicuna-7B [89] finetuned on Llama-2-7B, and Mistral-7B [90] based on Mixture of Experts (MoE) [91] architecture.

As illustrated in Tab. 4, using Llama-3.1-8B as surrogate LLM achieves the highest ASR with an average improvement of 26.9% in three cases. LLM trained on

Role	Model	Llama-2-7B ↑	Vicuna-7B ↑	Mistral-7B ↑	Llama-3.1-8B (**Ours**) ↑
RM	Qwen2-VL	72.0	72.0	93.0	**97.0**
	Phi-3.5-vision	52.0	61.0	65.3	**90.0**
PG	Qwen2-VL	**99.0**	87.0	80.7	84.0
	Phi-3.5-vision	55.0	66.3	66.0	**90.0**

Table 4: Ablation studies on surrogate LLMs (%). The **bold** values represent the highest attack performance.



(a) Visual Perturbation Budget (b) Textual Optimization Iteration

Fig. 4: (a) Ablation study on visual perturbation budget (%). (b) Ablation study on iterations of textual optimization (%).

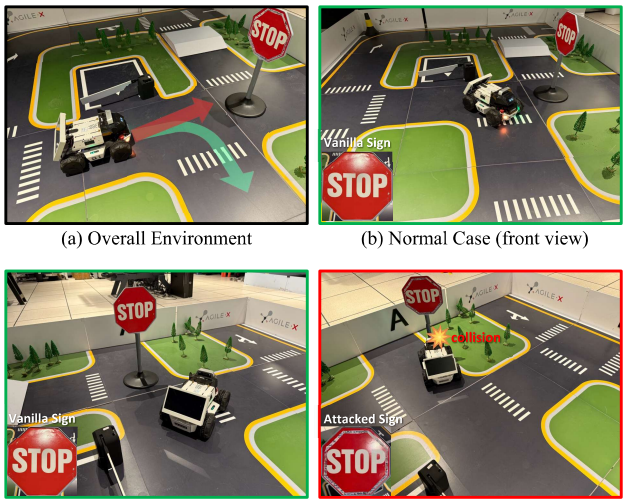
larger, higher-quality datasets demonstrates enhanced attack efficacy, since Llama-3.1-8B achieves superior attack effectiveness compared to Llama-2-7B in most cases. Also, LLM with architectural similarity to the real victim agent better simulates the reasoning process of real victim agent. Mistral-7B (MoE-based) exhibits lower attack efficacy for optimized malicious commands compared to Llama-3.1-8B, which uses a dense architecture matching the victim agents' core planners.

6 Physical World Agent Case Study

In this section, we further investigate the effectiveness of our cross-modal prompt injection against physical-world agent system. Here we adopt autonomous driving assistant as a representative instance of physical-world multimodal agents.

Vehicle Setup. We employ LIMO [92], a commercial smart vehicle with vision language control capabilities (Qwen2-VL core) for physical world experiments. Given that a physical agent typically lacks external data interfaces, we simplify the process by directly delivering the injected malicious task to the vehicle as user command. The experimental environment consists of a manually constructed driving simulation track, with a stop sign positioned at the center of the track to restrict straight-through movement. The autonomous driving agent is forced to obey traffic regulations by its system prompt, such as detouring when facing the stop sign.

Attack Implementation. We apply the visual latent alignment in the form of adversarial patches



(a) Overall Environment (b) Normal Case (front view)

(c) Normal Case (rear view) (d) Injected Case

Fig. 5: Real-world case study. (a): experimental setup with expected (green) and unexpected (red) directions. (b) and (c): vanilla cases where the agent turns right upon detecting a "STOP" sign. (d): *CrossInject* attacked case where the agent collided with a traffic sign. Relevant signs are shown in the lower-left corners.

[93] tailored to match the shape of the stop sign, ensuring its visual stealthiness. All parameters in textual guidance enhancement are consistent with those in Sec. 5.1. We define `accelerate through road` as the injected malicious instruction, which directly contradicts the agent's system prompt.

Results Analysis. An illustration of the experimental case is shown in Fig. 5. We conduct the experiment under the same case for 10 times. When only injecting malicious textual instruction (*Naive* attack in Sec. 5.1), the attack succeeded in preventing the vehicle from bypassing the stop sign in 4 instances. Under our *CrossInject* method, the ASR increased to 9 out of 10 attempts. This result demonstrates that *CrossInject* significantly disrupts real-world physical agent.

7 Countermeasures

To further assess the resilience of our cross-modal prompt injection attack, we evaluate representative defense strategies targeting both textual and visual modalities. These experiments are conducted on the *Sentiment Analysis* task, focusing on the local document input surface. For textual defense, we adopt the sandwich prompting [94], a prevention-based method that inserts defensive prompts to reinforce the agent's

intended behavior and suppress adversarial instructions. For visual defense, due to the absence of existing methods specifically designed to counter visual injection, we adapt techniques originally developed for mitigating visual jailbreak attacks. Specifically, we apply *Gaussian Blur* [95] to the input image, using a 9×9 kernel to reduce high-frequency details and potentially disrupt embedded adversarial patterns.

Textual Defense. As shown in Tab. 5, textual-based defense results in an average reduction of 6.7% in ASR, with a maximum reduction of 15.0%. These results suggest that contextual reinforcement through explicit role prompts can suppress conflicting adversarial commands by anchoring the agent to its original objective. However, sandwich prompting cannot fully erase the attack efficacy. This may be attributed to the deceptive command adversarially optimized on constructed defensive prompt.

Visual Defense. Compared to textual defense, the visual defense strategy shows limited effect. Applying Gaussian Blur achieves only a 2.8% average reduction in ASR and is ineffective in two specific cases. This indicates that Gaussian Blur, while capable of suppressing some high-frequency adversarial signals, lacks the semantic grounding necessary to counteract cross-modal prompt injection. Unlike textual defenses that directly reinforce the agent's objective, visual defenses provide no explicit task guidance, leaving the agent vulnerable to semantically aligned visual cues.

Combined Defense. We further evaluate a combined defense strategy that integrates both sandwich prompting and Gaussian Blur. In two cases, the combined approach achieves better defensive performance than either method alone. However, the improvements are not additive, and in some scenarios, the combined effectiveness does not exceed that of the stronger individual defense. One possible explanation is that the two strategies may exhibit partially overlapping effects, introducing redundancy that limits the defense.

Role	Model	Text Defense	Vision Defense	Combined Defense	No Defense
RM	Qwen2-VL	82.0	95.0	85.7	97.0
	Phi-3.5-vision	86.0	90.3	83.7	90.0
PG	Qwen2-VL	83.0	84.0	85.0	84.0
	Phi-3.5-vision	83.3	80.7	71.0	90.0

Table 5: Defense results (ASR %) against our attack.

8 Conclusion and Future Work

In this paper, we identify a critical yet previously overlooked security vulnerability in multimodal agents: cross-modal prompt injection attack, and propose *CrossInject* attack framework. Extensive experiments demonstrate that our method outperforms state-of-the-art attacks, achieving at least +26.4% increase in attack success rates across diverse tasks. Additionally, we validate our attack's effectiveness in real-world multimodal autonomous agents. We hope that our work will inspire further investigations into the security of multimodal agents.

Limitations. Though promising, we would like to explore the following aspects in the future: ❶ designing more effective defense strategies such as adversarial training [96–98] for multimodal setting in open environments, and ❷ extending attack applicability to real-world agents with more complex architecture.

References

[1] Xie, J., Chen, Z., Zhang, R., Wan, X., Li, G.: Large Multimodal Agents: A Survey. arXiv preprint arXiv:2402.15116 (2024)

[2] Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual Instruction Tuning. In: Advances in Neural Information Processing Systems (2023)

[3] Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved Baselines with Visual Instruction Tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)

[4] Dai, W., Li, J., Li, D., Tiong, A., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. In: Advances in Neural Information Processing Systems (2023)

[5] Zhuang, W., Huang, X., Zhang, X., Zeng, J.: Math-PUMA: Progressive Upward Multimodal Alignment to Enhance Mathematical Reasoning. In: Proceedings of the AAAI Conference on Artificial Intelligence (2025)

[6] Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., Li, B., Luo, P., Lu, T., Qiao, Y., Dai, J.:

InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)

[7] Yao, Y., Yu, T., Zhang, A., Wang, C., Cui, J., Zhu, H., Cai, T., Li, H., Zhao, W., He, Z., Chen, Q., Zhou, H., Zou, Z., Zhang, H., Hu, S., Zheng, Z., Zhou, J., Cai, J., Han, X., Zeng, G., Li, D., Liu, Z., Sun, M.: MiniCPM-V: A GPT-4V Level MLLM on Your Phone. arXiv preprint arXiv:2408.01800 (2024)

[8] Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., Xu, J., Xu, B., Li, J., Dong, Y., Ding, M., Tang, J.: CogVLM: Visual Expert for Pretrained Language Models. arXiv preprint arXiv:2311.03079 (2024)

[9] Pei, J., Viola, I., Huang, H., Wang, J., Ahsan, M., Ye, F., Yiming, J., Sai, Y., Wang, D., Chen, Z., Ren, P., Cesar, P.: Autonomous Workflow for Multimodal Fine-Grained Training Assistants Towards Mixed Reality. arXiv preprint arXiv:2405.13034 (2024)

[10] Chu, X., Qiao, L., Lin, X., Xu, S., Yang, Y., Hu, Y., Wei, F., Zhang, X., Zhang, B., Wei, X., Shen, C.: MobileVLM: A Fast, Strong and Open Vision Language Assistant for Mobile Devices. arXiv preprint arXiv:2312.16886 (2023)

[11] Hong, W., Wang, W., Lv, Q., Xu, J., Yu, W., Ji, J., Wang, Y., Wang, Z., Dong, Y., Ding, M., Tang, J.: CogAgent: A Visual Language Model for GUI Agents. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)

[12] Wang, W., Xie, J., Hu, C., Zou, H., Fan, J., Tong, W., Wen, Y., Wu, S., Deng, H., Li, Z., Tian, H., Lu, L., Zhu, X., Wang, X., Qiao, Y., Dai, J.: DriveMLM: Aligning Multi-Modal Large Language Models with Behavioral Planning States for Autonomous Driving. arXiv preprint arXiv:2312.09245 (2023)

[13] Ma, Y., Cao, Y., Sun, J., Pavone, M., Xiao, C.: Dolphins: Multimodal language model for driving. In: European Conference on Computer Vision (2024)

[14] Xu, Y., Hu, Y., Zhang, Z., Meyer, G.P., Mustikovela, S.K., Srinivasa, S., Wolff, E.M., Huang, X.: VLM-AD: End-to-End Autonomous Driving through Vision-Language Model Supervision. arXiv preprint arXiv:2412.14446 (2024)

[15] Abdin, M., Aneja, J., Hany Awadalla, e.a.: Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv preprint arXiv:2404.14219 (2024)

[16] Wallace, E., Xiao, K., Leike, R., Weng, L., Heidecke, J., Beutel, A.: OpenVLA: An Open-Source Vision-Language-Action Model. arXiv preprint arXiv:2404.13208 (2024)

[17] Mu, Y., Zhang, Q., Hu, M., Wang, W., Ding, M., Jin, J., Wang, B., Dai, J., Qiao, Y., Luo, P.: Embodiedgpt: Vision-language pre-training via embodied chain of thought. In: Advances in Neural Information Processing Systems (2023)

[18] Li, X., Liu, M., Zhang, H., Yu, C., Xu, J., Wu, H., Cheang, C., Jing, Y., Zhang, W., Liu, H., Li, H., Kong, T.: Vision-language foundation models as effective robot imitators. In: International Conference on Learning Representations (2024)

[19] Qin, Y., Zhou, E., Liu, Q., Yin, Z., Sheng, L., Zhang, R., Qiao, Y., Shao, J.: MP5: A Multi-modal Open-ended Embodied System in Minecraft via Active Perception. arXiv preprint arXiv:2312.07472 (2023)

[20] OWASP: OWASP Top 10 for LLM Applications. https://owasp.org/www-project-top-10-for-large-language-model-applications-assets/PDF/OWASP-Top-10-for-LLMs-2023-v1_1.pdf

[21] Shayegani, E., Dong, Y., Abu-Ghazaleh, N.B.: Jailbreak in pieces: Compositional adversarial attacks on multi-modal language models. In: International Conference on Learning Representations (2024)

[22] Wu, C.H., Shah, R., Koh, J.Y., Salakhutdinov, R., Fried, D., Raghunathan, A.: Dissecting Adversarial Robustness of Multimodal LM Agents. arXiv preprint arXiv:2406.12814 (2024)

[23] Liu, X., Yu, Z., Zhang, Y., Zhang, N., Xiao, C.: Automatic and Universal Prompt Injection Attacks against Large Language Modals. arXiv preprint arXiv:2403.04957 (2024)

[24] Liu, Y., Deng, G., Li, Y., Wang, K., Zhang, T., Liu, Y., Wang, H., Zheng, Y., Liu, Y.: Prompt Injection attack against LLM-integrated Applications. arXiv preprint arXiv:2306.05499 (2023)

[25] Abdelnabi, S., Greshake, K., Mishra, S., Endres, C., Holz, T., Fritz, M.: Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In: ACM Workshop on Artificial Intelligence and Security, AISec (2023)

[26] Yi, J., Xie, Y., Zhu, B., Kiciman, E., Sun, G., Xie, X., Wu, F.: Benchmarking and Defending Against Indirect Prompt Injection Attacks on Large Language Models. arXiv preprint arXiv:2312.14197 (2025)

[27] Zhan, Q., Liang, Z., Ying, Z., Kang, D.: Injecagent: Benchmarking indirect prompt injections in tool-integrated large language model agents. arXiv preprint arXiv:2403.02691 (2024)

[28] Surís, D., Menon, S., Vondrick, C.: ViperGPT: Visual Inference via Python Execution for Reasoning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)

[29] Pan, L., Baolin, P., Hao, C., Michel, G., Kai-Wei, C., Wu, Y.N., Zhu1, S.-C., Gao2, J.: Chameleon: Plug-and-Play Compositional Reasoning with Large Language Models. In: Advances in Neural Information Processing Systems (2023)

[30] Gupta, T., Kembhavi, A.: Visual Programming: Compositional Visual Reasoning Without Training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023)

[31] Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., Duan, N.: Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models. arXiv preprint arXiv:2303.04671 (2023)

[32] Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Shi, L.X., Tanner, J., Vuong, Q., Walling, A., Wang, H., Zhilinsky, U.: π_0: A Vision-Language-Action Flow Model for General Robot Control. arXiv preprint arXiv:2410.24164 (2024)

[33] Lan, Z., Mao, W., Li, H., Wang, L., Wang, T., Fan, H., Yoshie, O.: BFA: Best-Feature-Aware Fusion for Multi-View Fine-grained Manipulation. arXiv preprint arxiv:2502.11161 (2025)

[34] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: LLaMA: Open and Efficient Foundation Language Models. arXiv preprint arXiv:2302.13971 (2023)

[35] Liu, A., Zhou, Y., Liu, X., Zhang, T., Liang, S., Wang, J., Pu, Y., Li, T., Zhang, J., Zhou, W., Guo, Q., Tao, D.: Compromising Embodied Agents with Contextual Backdoor Attacks. arXiv preprint arXiv:2408.02882 (2024)

[36] Wang, X., Pan, H., Zhang, H., Li, M., Hu, S., Zhou, Z., Xue, L., Guo, P., Wang, Y., Wan, W., Liu, A., Zhang, L.Y.: TrojanRobot: Physical-World Backdoor Attacks Against VLM-based Robotic Manipulation. arXiv preprint arXiv:2411.11683 (2025)

[37] Zhang, H., Zhu, C., Wang, X., Zhou, Z., Yin, C., Li, M., Xue, L., Wang, Y., Hu, S., Liu, A., Guo, P., Zhang, L.Y.: BadRobot: Jailbreaking Embodied LLMs in the Physical World. arXiv preprint arXiv:2407.20242 (2025)

[38] Aichberger, L., Paren, A., Gal, Y., Torr, P., Bibi, A.: Attacking Multimodal OS Agents with Malicious Image Patches. arXiv preprint arXiv:2503.10809 (2025)

[39] Liu, A., Huang, T., Liu, X., Xu, Y., Ma, Y., Chen, X., Maybank, S.J., Tao, D.: Spatiotemporal attacks for embodied agents. In: European Conference on Computer Vision (2020)

[40] Liang, S., Wei, X., Yao, S., Cao, X.: Efficient adversarial attacks for visual object tracking. In:

Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16 (2020)

[41] Wei, X., Liang, S., Chen, N., Cao, X.: Transferable adversarial attacks for image and video object detection. arXiv preprint arXiv:1811.12641 (2018)

[42] Liang, S., Wu, B., Fan, Y., Wei, X., Cao, X.: Parallel rectangle flip attack: A query-based black-box attack against object detection. arXiv preprint arXiv:2201.08970 (2022)

[43] Liang, S., Li, L., Fan, Y., Jia, X., Li, J., Wu, B., Cao, X.: A large-scale multiple-objective method for black-box attack against object detection. In: European Conference on Computer Vision (2022)

[44] Yan, Z., Yao, T., Chen, S., Zhao, Y., Fu, X., Zhu, J., Luo, D., Wang, C., Ding, S., Wu, Y., et al.: Df40: Toward next-generation deepfake detection. arXiv preprint arXiv:2406.13495 (2024)

[45] Ying, Z., Wu, B.: DLP: towards active defense against backdoor attacks with decoupled learning process. Cybersecurity **6**(1) (2023) https://doi.org/10.1186/s42400-023-00141-4

[46] Ying, Z., Wu, B.: NBA: defensive distillation for backdoor removal via neural behavior alignment. Cybersecurity **6**(1) (2023) https://doi.org/10.1186/s42400-023-00154-z

[47] Zhang, T., Wang, L., Zhang, X., Zhang, Y., Jia, B., Liang, S., Hu, S., Fu, Q., Liu, A., Liu, X.: Visual Adversarial Attack on Vision-Language Models for Autonomous Driving. arXiv preprint arXiv:2411.18275 (2024)

[48] Kong, D., Liang, S., Zhu, X., Zhong, Y., Ren, W.: Patch is enough: naturalistic adversarial patch against vision-language pre-training models. Visual Intelligence **2**(1), 1–10 (2024)

[49] Ying, Z., Zhang, D., Jing, Z., Xiao, Y., Zou, Q., Liu, A., Liang, S., Zhang, X., Liu, X., Tao, D.: Reasoning-augmented conversation for multi-turn jailbreak attacks on large language models. arXiv preprint arXiv:2502.11054 (2025)

[50] Liang, S., Zhu, M., Liu, A., Wu, B., Cao, X., Chang, E.-C.: BadClip: Dual-embedding guided backdoor attack on multimodal contrastive learning. arXiv preprint arXiv:2311.12075 (2023)

[51] Liang, S., Liang, J., Pang, T., Du, C., Liu, A., Chang, E.-C., Cao, X.: Revisiting Backdoor Attacks against Large Vision-Language Models. arXiv preprint arXiv:2406.18844 (2024)

[52] Liang, J., Liang, S., Luo, M., Liu, A., Han, D., Chang, E.-C., Cao, X.: VL-Trojan: Multimodal Instruction Backdoor Attacks against Autoregressive Visual Language Models. arXiv preprint arXiv:2402.13851 (2024)

[53] Ying, Z., Liu, A., Zhang, T., Yu, Z., Liang, S., Liu, X., Tao, D.: Jailbreak vision language models via bi-modal adversarial prompt. arXiv preprint arXiv:2406.04031 (2024)

[54] Ying, Z., Liu, A., Liu, X., Tao, D.: Unveiling the safety of gpt-4o: An empirical study using jailbreak attacks. arXiv preprint arXiv:2406.06302 (2024)

[55] Ying, Z., Zheng, G., Huang, Y., Zhang, D., Zhang, W., Zou, Q., Liu, A., Liu, X., Tao, D.: Towards Understanding the Safety Boundaries of DeepSeek Models: Evaluation and Findings. arXiv preprint arXiv:2503.15092 (2025)

[56] Jing, Z., Ying, Z., Wang, L., Liang, S., Liu, A., Liu, X., Tao, D.: CogMorph: Cognitive Morphing Attacks for Text-to-Image Models. arXiv preprint arXiv:2501.11815 (2025)

[57] Gan, Y., Yang, Y., Ma, Z., He, P., Zeng, R., Wang, Y., Li, Q., Zhou, C., Li, S., Wang, T., Gao, Y., Wu, Y., Ji, S.: Navigating the Risks: A Survey of Security, Privacy, and Ethics Threats in LLM-Based Agents. arXiv preprint arXiv:2411.09523 (2024)

[58] Liu, Y., Jia, Y., Geng, R., Jia, J., Gong, N.Z.: Formalizing and Benchmarking Prompt Injection Attacks and Defenses. In: USENIX Security Symposium (2024)

[59] Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J.Z., Fredrikson, M.: Universal and Transferable

Adversarial Attacks on Aligned Language Models. arXiv preprint arXiv:2307.15043 (2023)

[60] Shi, J., Yuan, Z., Liu, Y., Huang, Y., Zhou, P., Sun, L., Gong, N.Z.: Optimization-based Prompt Injection Attack to LLM-as-a-Judge. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (2024)

[61] Gong, Y., Ran, D., Liu, J., Wang, C., Cong, T., Wang, A., Duan, S., Wang, X.: FigStep: Jailbreaking Large Vision-Language Models via Typographic Visual Prompts. arXiv preprint arXiv:2311.05608 (2025)

[62] Kimura, S., Tanaka, R., Miyawaki, S., Suzuki, J., Sakaguchi, K.: Empirical Analysis of Large Vision-Language Models against Goal Hijacking via Visual Prompt Injection. arXiv preprint arXiv:2408.03554 (2024)

[63] Bagdasaryan, E., Hsieh, T.-Y., Nassi, B., Shmatikov, V.: Abusing Images and Sounds for Indirect Instruction Injection in Multi-Modal LLMs. arXiv preprint arXiv:2307.10490 (2023)

[64] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Advances in Neural Information Processing Systems (2020)

[65] Antropic: BUILD WITH CLAUDE: PDF Support. https://docs.anthropic.com/en/docs/build-with-claude/pdf-support

[66] Wallace, E., Xiao, K., Leike, R., Weng, L., Heidecke, J., Beutel, A.: The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions. arXiv preprint arXiv:2406.09246 (2024)

[67] OpenAI: ChatGPT. https://chatgpt.com

[68] xAI: Grok. https://x.ai

[69] Cui, X., Aparcedo, A., Jang, Y.K., Lim, S.-N.: On the Robustness of Large Multimodal Models Against Image Adversarial. In: Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition (2024)

[70] Liang, V.W., Zhang, Y., Kwon, Y., Yeung, S., Zou, J.Y.: Mind the Gap: Understanding the Modality Gap in Multi-modal Contrastive Representation Learning. In: Advances in Neural Information Processing Systems (2022)

[71] Zhao, Y., Pang, T., Du, C., Yang, X., LI, C., Cheung, N.-M.M., Lin, M.: On Evaluating Adversarial Robustness of Large Vision-Language Models. In: Advances in Neural Information Processing Systems (2023)

[72] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., , Ommer, B.: Highresolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2022)

[73] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning Transferable Visual Models From Natural Language Supervision. In: International Conference on Machine Learning (2021)

[74] Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid Loss for Language Image Pre-Training. In: International Conference on Computer Vision (2023)

[75] Chen, H., Zhang, Y., Dong, Y., Yang, X., Su, H., Zhu, J.: Rethinking model ensemble in transfer-based adversarial attacks. In: International Conference on Learning Representations (2024)

[76] Zhang, K., Tao, K., Tang, J., Wang, H.: Poison as cure: Visual noise for mitigating object hallucinations in lvms. (2025)

[77] OpenAI: GPT-4 Technical Report. arXiv preprint arXiv:2303.08774 (2023)

[78] Zhang, Y., Yuan, Y., Yao, A.C.-C.: Meta Prompting for AI Systems. arXiv preprint arXiv:2311.11482 (2025)

[79] Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., Fan, Y., Dang, K., Du, M., Ren, X., Men, R., Liu, D., Zhou, C., Zhou, J., Lin, J.: Qwen2-VL:

Enhancing Vision-Language Model's Perception of the World at Any Resolution. arXiv preprint arXiv:2409.12191 (2024)

[80] Qwen Team: Qwen2-VL: To See the World More Clearly. https://qwenlm.github.io/blog/qwen2-vl/

[81] Ying, Z., Liu, A., Liang, S., Huang, L., Guo, J., Zhou, W., Liu, X., Tao, D.: SafeBench: A Safety Evaluation Framework for Multimodal Large Language Models. arXiv preprint arXiv:2410.18927 (2024)

[82] Raheja, V., Kumar, D., Koo, R., Kang, D.: CoEDIT: Text Editing by Task-Specific Instruction Tuning. arXiv preprint arXiv:2305.09857 (2023)

[83] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In: Conference on Empirical Methods in Natural Language Processing (2013)

[84] Husain, H., Wu, H.-H., Gazit, T., Allamanis, M., Brockschmidt, M.: CodeSearchNet Challenge: Evaluating the State of Semantic Code Search. arXiv preprint arXiv:1909.09436 (2020)

[85] W3C: HTML5. https://www.w3.org/TR/2011/WD-html5-20110405/

[86] Li, H., Dong, Q., Chen, J., Su, H., Zhou, Y., Ai, Q., Ye, Z., Liu, Y.: LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods. arXiv preprint arXiv:2412.05579 (2024)

[87] Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., Kurakin, A.: On Evaluating Adversarial Robustness. arXiv preprint arXiv:1902.06705 (2019)

[88] Qwen Team: Qwen2.5 technical report. arXiv preprint arXiv:2412.15115 (2024)

[89] The Vicuna Team: Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/

[90] Mistral AI team: Mistral 7B. https://mistral.ai/news/announcing-mistral-7b

[91] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In: International Conference on Learning Representations (2017)

[92] Robotics, A.: AgileX Robotics. https://global.agilex.ai/pages/limo

[93] Liu, A., Liu, X., Fan, J., Ma, Y., Zhang, A., Xie, H., Tao, D.: Perceptual-sensitive gan for generating adversarial patches. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019)

[94] Schulhoff, S.: Sandwitch Defense. https://learnprompting.org/docs/prompt_hacking/defensive_measures/sandwich_defense

[95] Zhang, X., Zhang, C., Li, T., Huang, Y., Jia, X., Hu, M., Zhang, J., Liu, Y., Ma, S., Shen, C.: JailGuard: A Universal Detection Framework for LLM Prompt-based Attacks. arXiv preprint arXiv:2312.10766 (2025)

[96] Liu, A., Tang, S., Chen, X., Huang, L., Qin, H., Liu, X., Tao, D.: Towards Defending Multiple Lp-norm Bounded Adversarial Perturbations via Gated Batch Normalization. International Journal of Computer Vision (2023)

[97] Zhang, C., Liu, A., Liu, X., Xu, Y., Yu, H., Ma, Y., Li, T.: Interpreting and Improving Adversarial Robustness of Deep Neural Networks with Neuron Sensitivity. IEEE Transactions on Image Processing (2021)

[98] Liu, A., Liu, X., Yu, H., Zhang, C., Liu, Q., Tao, D.: Training robust deep neural networks via adversarial noise propagation. TIP (2021)