ADVEDM: Fine-grained Adversarial Attack against VLM-based Embodied Agents

Yichen Wang[1,2,4,5,6] **Hangtao Zhang**[6] **Hewen Pan**[1,2,4,5,6] **Ziqi Zhou**[1,2,3,7]
Xianlong Wang[8] **Peijin Guo**[1,2,4,5,6] **Lulu Xue** [1,2,4,5,6] **Shengshan Hu** [1,2,4,5,6]
Minghui Li[9] **Leo Yu Zhang** [10]

[1]National Engineering Research Center for Big Data Technology and System
[2]Services Computing Technology and System Lab [3]Cluster and Grid Computing Lab
[4]Hubei Engineering Research Center on Big Data Security
[5]Hubei Key Laboratory of Distributed System Security
[6]School of Cyber Science and Engineering, Huazhong University of Science and Technology
[7]School of Computer Science and Technology, Huazhong University of Science and Technology
[8]Department of Computer Science, City University of Hong Kong
[9]School of Software Engineering, Huazhong University of Science and Technology
[10] School of Information and Communication Technology, Griffith University
{wangyichen, hangt_zhang, hewenpan, zhouziqi, gpj, lluxue, hushengshan, minghuili}@hust.edu.cn
xianlong.wang@my.cityu.edu.hk, leo.zhang@griffith.edu.au

Abstract

Vision-Language Models (VLMs), with their strong reasoning and planning capabilities, are widely used in embodied decision-making (EDM) tasks in embodied agents, such as autonomous driving and robotic manipulation. Recent research has increasingly explored adversarial attacks on VLMs to reveal their vulnerabilities. However, these attacks either rely on overly strong assumptions, requiring full knowledge of the victim VLM, which is impractical for attacking VLM-based agents, or exhibit limited effectiveness. The latter stems from disrupting most semantic information in the image, which leads to a misalignment between the perception and the task context defined by system prompts. This inconsistency interrupts the VLM's reasoning process, resulting in invalid outputs that fail to affect interactions in the physical world. To this end, we propose a fine-grained adversarial attack framework, ADVEDM, which modifies the VLM's perception of only a few key objects while preserving the semantics of the remaining regions. This attack effectively reduces conflicts with the task context, making VLMs output valid but incorrect decisions and affecting the actions of agents, thus posing a more substantial safety threat in the physical world. We design two variants of based on this framework, ADVEDM-R and ADVEDM-A, which respectively remove the semantics of a specific object from the image and add the semantics of a new object into the image. The experimental results in both general scenarios and EDM tasks demonstrate fine-grained control and excellent attack performance.

1 Introduction

Visual-language models (VLMs) such as GPT-4 [1] and Gemini-2.0 [2] have been widely adopted for embodied decision-making (EDM) tasks in embodied agents, including autonomous driving [3, 4, 5] and robotic manipulation [6, 7, 8], due to their powerful reasoning and planning capabilities. In these tasks, VLMs generate decisions and plannings based on current inputs and system states, and

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

then convert them to control code to guide physical entities (*e.g.*, vehicles, robotic arms) in their interactions with the real world.

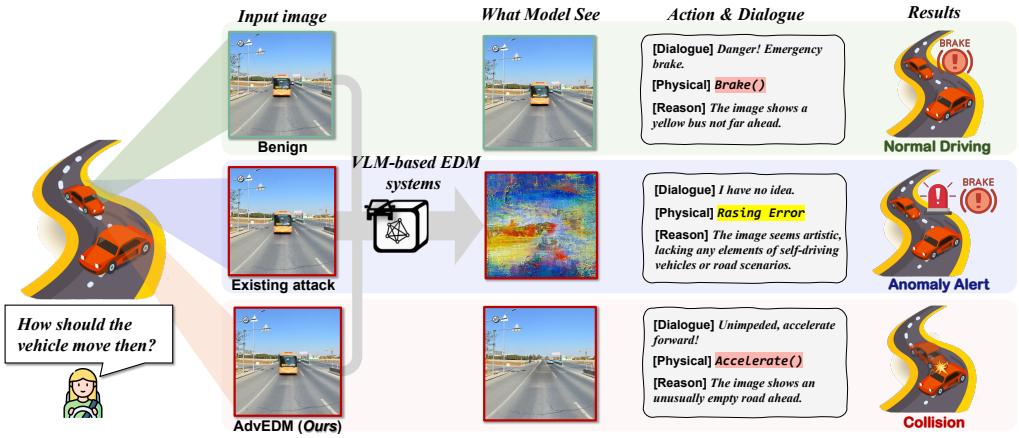


Figure 1: Comparison of our attack framework with existing works in attacking VLM-based agents. Existing attacks disrupt most of the semantics in the original image, causing the VLM to generate invalid responses. In contrast, our attack selectively alters the VLM's perception of a specific object while preserving the semantic integrity of other regions. As a result, the VLM produces valid yet incorrect decisions, effectively influencing the system's interaction with the physical world.

However, VLMs have shown vulnerability to adversarial attacks [9, 10, 11, 12, 13, 14], where the adversary manipulates the model's output by introducing imperceptible perturbations into input images. Existing attacks on VLMs can be classified into two categories: white-box attacks and black-box attacks. White-box attacks, such as AttackBard [15] and CroPA [16], generate adversarial examples by optimizing the end-to-end process, directly manipulating the textual outputs. But such attacks are impractical for VLM-based embodied agents, as it is difficult for the adversary to access the LLM modules within the VLM, which are fine-tuned on proprietary datasets for specific tasks [17].

In contrast, black-box attacks where the adversary's knowledge of the victim model is limited are more practical. However, a fully black-box setting poses significant challenges for attackers, incurring significant computational costs and limited attack effectiveness [18]. Therefore, many existing works (like AttackVLM [10], CLIP-based Attack [19] and VT-Attack [20]) propose a compromise setting, where the attacker has access to only the vision-text encoder of VLMs, referred to as the gray-box setting. Since the vision-text encoder in VLM-based EDM systems is directly used in its pre-trained form [21] and easy for the adversary to obtain, we focus on the gray-box attacks in this paper. These attacks introduce adversarial perturbations to move the image's embedding away from the clean image's embedding, thereby disrupting the VLM's perception of the image. However, such attacks have limited effectiveness against the VLM-based EDM system in embodied agents, which only make the system output **invalid results without guiding entity's action in the physical world** (like error reports). This is because these systems employ chain-of-thought (CoT) techniques [22] to perform reasoning and task planning. The CoT first analyzes the system's current state and perceived inputs, then proceeds to further reasoning based on the task description provided by system prompts [23, 24]. Existing attacks of this type alter the system's overall perception of the image, causing a conflict with the system prompts' description, thereby interrupting the reasoning process and resulting in invalid outputs rather than decisions and plannings. This process is illustrated in Fig. 1.

In this paper, we propose a novel fine-grained adversarial attack framework, AdvEDM, which makes the VLM-based EDM system output valid but incorrect decisions. As illustrated in Fig. 1, our attack disrupts the CoT in the VLM-based EDM system by modifying the VLM's perception of the existence of several key objects while retaining the original semantics of other parts. This significantly reduces conflicts with the task context, ensuring the integrity and logical consistency during the reasoning process. Consequently, the system outputs valid but incorrect decisions that can alter the entity's action, leading to a more substantial safety threat in the physical world.

Specifically, we design two attack methods based on this framework, AdvEDM-R and AdvEDM-A, which respectively remove the semantics of an object from the image and add the semantics of a

new object into the image, while preserving the semantics of others. The implementation of these attacks faces two technical challenges: first, how to select appropriate regions in the image for the removal and addition of the target object's semantics; and second, how to preserve the semantics of other regions after modifying the target semantics. To address the first challenge, we propose a selection strategy based on the similarities between cross-modality embeddings, which leverages the correlation between the VLM's vision and text encoders [25, 26] by calculating the similarity between image patch token embeddings and object text embeddings. For the second challenge, considering that the image embeddings are typically generated by the attention mechanism of ViT [27], we propose attention-$[patch]$ fixation, which preserves the semantics of the remaining parts by maintaining the product of the attention weights and patch token embeddings.

We evaluate our methods in both general image description scenarios and two representative embodied decision-making tasks: autonomous driving and robotic manipulation. In general scenario, the average attack success rates (ASR) for the two variants are 76.8% and 70.2%, with semantic preservation rates of 66.7% and 71.6%. In EDM tasks, our method achieved an attack success rate of over 70% and 64% in autonomous driving and robotic manipulation respectively, significantly outperforming existing attacks. These results highlight the excellent fine-grained control of our attacks and their effectiveness in posing a real safety threat to VLM-based EDM systems. More demos of our attacks in real-world scenarios can be found on our website https://advedm.github.io/.

In conclusion, the contribution of this paper can be summarized as follows: (1) We propose a novel fine-grained adversarial attack framework ADVEDM in the gray-box setting that selectively modifies the semantics of key objects perceived by VLM-based EDM systems, disrupting their reasoning process and leading to valid but incorrect decisions, thus increasing real-world safety risks. This aims to reveal the vulnerabilities of current VLM-based EDM systems and foster future efforts to enhance their robustness. (2) Based on the framework, we design two attacks ADVEDM-R and ADVEDM-A, which respectively remove the semantics of a specific object or add the semantics of a new object to the image. (3) The experimental results in both general scenarios and embodied decision-making tasks indicate the excellent fine-grained control and effectiveness of our attacks.

2 Related Work

2.1 VLM-Based Embodied Decision-Making System

Due to their exceptional logical reasoning capabilities, VLMs have been widely applied to embodied decision-making tasks [28, 29, 30]. The Chain-of-Thought (CoT) is widely used in VLM-based EDM systems [22, 23], which breaks down the task into logical steps, refining the model's decision-making based on the current context and task requirements. Two prominent applications are autonomous driving and robotic manipulation. In autonomous driving, VLMs fine-tuned on specialized datasets, such as DriveLM [4], Dolphins [21], and DriveGPT [31], process road information captured by sensors like cameras. VLMs in this task, combined with predefined system prompts, enable real-time planning and adjustments to the vehicle's driving state through CoT reasoning process. In the robotic manipulation task, VLMs first perceive the input visual images, then combine them with received instructions and system prompts to perform reasoning through CoT and generate decisions and plannings regarding the robot's actions like rotation, movement, and grasping. Finally, the post-processing module translates these decisions into control code to manipulate its interactions with the physical world [32]. In conclusion, VLM-based EDM systems play a crucial role in embodied AI tasks. While extensive research has been conducted on the robustness of VLMs themselves [10], the robustness and security of VLM-based EDM systems remain unexplored.

2.2 Adversarial Attack against VLMs

Adversarial attack involves manipulating the outputs of models by introducing imperceptible perturbations to the inputs [33, 34, 35, 36, 37, 13, 38, 39, 40]. With the widespread application of VLMs, there have been increasing researches focused on attacks against VLMs in recent years. Most of existing works [19, 41, 42, 43, 44] focus on designing attack methods in a white-box setting, where it is assumed that the adversary has full access to the victim model and other relevant information to launch attack. These methods typically optimize the adversarial noise by minimizing the difference between the logits of probability of the outputs and the pre-defined target text, thereby enabling end-to-end attacks. Despite their high attack success rates, these attacks are impractical for the

VLM-based EDM system, as it is difficult for the adversary to access VLMs fine-tuned on various datasets for specific tasks [45].

To this end, some works proposed attacks in more general scenarios [9, 15, 46, 10], where the adversary has limited knowledge about the victim VLMs. They usually employ pre-trained vision-text encoders of VLMs as surrogate models, such as CLIP [47, 48]. These attacks involve making the embeddings of adversarial examples diverge from those of original images or closer to those of target images to disrupt the perception of VLMs. Although they are more practical for attacking VLM-based EDM systems whose vision-text encoders are usually pre-trained and easy to obtain, their effectiveness is limited due to the lack of fine-grained control. Specifically, they disrupt most of the semantics perceived by the VLM from original images, thus interrupting the reasoning process of VLMs and leading to invalid results that fail to influence interactions with the physical world. In this work, we design two fine-grained adversarial attack methods ADVEDM-R and ADVEDM-A that disrupt the perception of VLMs by precisely removing or adding the semantics of target objects, while preserving other semantics in the original image.

3 Preliminary

3.1 Background

Following existing works [49, 28], the VLM-based EDM system consists of two components, as shown in Fig. 2. The first component is the decision-making module implemented by a VLM, which includes a pre-trained vision-text encoder (*e.g.*, CLIP [25]) and a fine-tuned LLM. The vision-text encoder encodes the environmental image along with the received textual instructions and system prompts, as described in Eq. 1.

$$\phi_i = E_v(I), \quad \phi_t = E_t(T) \tag{1}$$

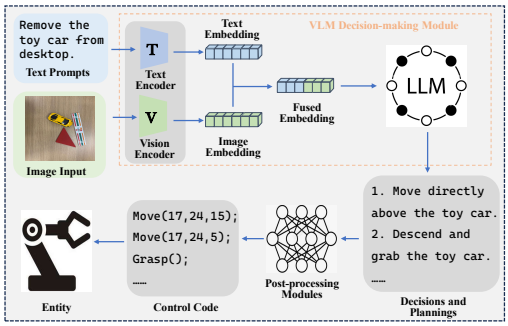


Figure 2: The framework of VLM-based embodied decision-making system.

where $E_v(\cdot)$ and $E_t(\cdot)$ are the vision encoder and text encoder, while ϕ_i and ϕ_t are the embeddings of the input image I and text instruction T. Note that the image encoders in VLMs are typically based on the Transformer architecture, where the image embeddings ϕ_i include a [CLS] token embedding (representing overall semantics) and patch token embeddings (representing local semantics) [50]. We denote them as $[cls]$ and $[patch]$.

The LLM in the decision-making module is fine-tuned on a proprietary task-specific dataset. It takes as input the fused and concatenated embeddings, performs reasoning through CoT, and generates decisions and plannings. This process is formulated as $T_D = \mathcal{M}(\phi_i, \phi_t; \theta)$, where $\mathcal{M}$ is the LLM with parameters θ, and T_D is the textual outputs of decisions and plannings.

Upon generating decisions and plannings, the post-processing module translates and converts them into executable control code for operating physical hardware of the entity, which can be expressed as $C = f_p(T_D)$. C is the control code and f_p represents the post-processing module.

3.2 Threat Model

Adversary's goal. To achieve fine-grained attack effectiveness, the adversary's goal as inducing the textual decisions where only the content regarding the target object is altered, while the rest remain unchanged. Here, we formally define our fine-grained adversarial attack. Specifically, we decompose the decision T_D into descriptions for n individual objects in the image, which can be expressed as $T_D = \{D_{obj_1}, D_{obj_1}, ..., D_{obj_n}\}$. Then the attack is formulated as Eq. (2).

$$\begin{aligned} \min \quad & \mathrm{Sim}(D'_{obj_t}, D_{obj_t}) \\ \text{s.t.} \quad & \mathrm{Sim}(D'_{obj_i}, D_{obj_i}) > \delta, \quad i \neq t \\ & \|I' - I\|_2 < \epsilon \end{aligned} \tag{2}$$

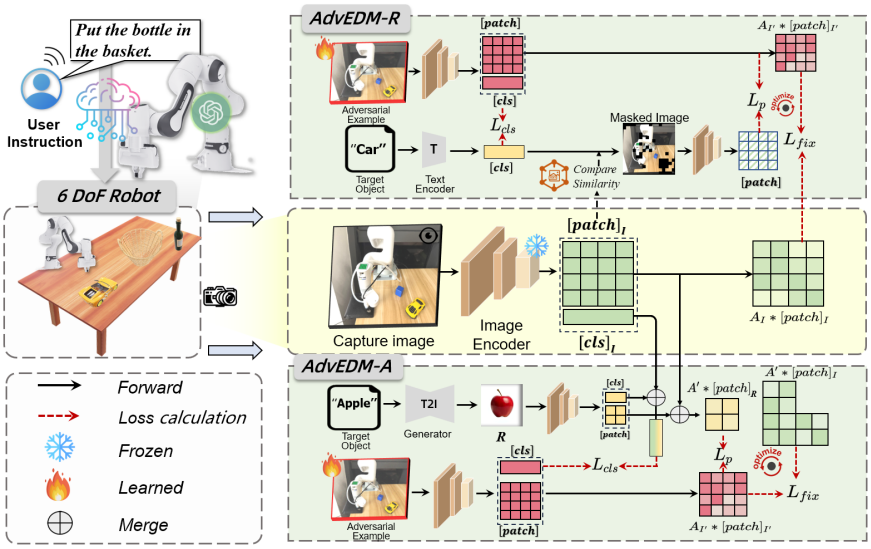


Figure 3: The pipeline of our methods AdvEDM-R and AdvEDM-A.

where obj_t is the target object, and D'_{obj_t} and D'_{obj_i} are decisions generated under adversarial examples I', while D_{obj_t} and D_{obj_i} are under clean inputs I. $\mathrm{Sim}(\cdot)$ measures semantic similarity between texts, and δ and ϵ represent the constraints for the semantic preservation and visual stealthiness.

Adversary's capacity. Since the textual input T is usually pre-defined like system prompts [6, 24] and difficult to manipulate from from external sources, we assume the adversary can only perturb the image inputs to launch an attack. Following many previous works like [10, 20, 19], we design our attacks in the gray-box setting, where the adversary can only utilize the vision-text encoder of VLMs as surrogate models to generate adversarial examples. Moreover, we also extend our attacks to the black-box setting and provide a detailed discussion in Appendix D.

4 Methodology

4.1 Intuition

Unlike previous adversarial attacks on VLMs in general tasks like VQA, effectively attacking the VLM-based EDM system requires inducing valid yet incorrect decisions that impact real-world interactions. This necessitates a fine-grained adversarial attack that selectively modifies the target object's semantics while preserving the overall reasoning integrity. Specifically, we propose two feasible schemes: (1) **Semantic Removal (SR)**, namely removing the semantics of a specific object while preserving those of others, and (2) **Semantic Addition (SA)**, which involves adding the semantics of a specified object without altering the semantics of others. For SR, the target output T'_D should exclude the semantics of the target object obj_t while preserving those of the remaining $n-1$ objects. For SA, it should incorporate both the original n objects and obj_t. Both schemes face two challenges: first, how to identify the key regions in the image for SR or SA to ensure the effectiveness of attacks, and second, how to maintain the semantics of the remaining parts of the image.

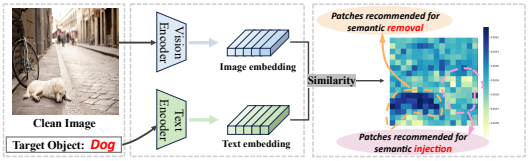


Figure 4: Our strategy to select the target regions for semantic removal and semantic addition.

Given that the vision-text encoder in most VLMs is built on the Transformer architecture [45, 48], we tackle the first challenge by identifying key regions based on the similarity between image patch token embeddings and the text embedding of obj_t. This similarity effectively quantifies the extent to which each patch contains the target object's semantics, due to the ability of the vision-text encoder to align visual and textual semantics. As shown in Fig. 4, patches with higher similarity scores contain

more relevant semantic information. For SR, we choose the patches with higher similarity to the text embeddings of target objects and erase their semantics. For SA, we select contiguous background regions with relatively lower similarity to foreground objects' embeddings, ensuring the injected semantics minimally influence the semantics of other objects.

As for the second challenge, we adopt an attention- $[patch]$ fixation approach. Specifically, during the optimization of adversarial perturbation, we ensure that the product of attention weights and patch token embeddings for the remaining regions closely matches that of the original image. This approach effectively preserves both the overall semantics and local detailed features of them [51].

4.2 Our Methods

According to our intuition, we propose two attack methods ADVEDM-R and ADVEDM-A, which remove the semantics of a specific object from the image and inject the semantics of a new object.

4.2.1 ADVEDM-R

We first demonstrate how to identify regions with target object's semantics in the image. We calculate the cosine similarity between the patch token embeddings and the text embedding of target object, and then mark the patches with higher similarity and form a mask. This process is formally described in Eq. 3 and 4.

$$S = CS([patch]_I, E_t(T_{tar})) \tag{3}$$

$$mask_i = \left\{ \begin{array}{ll} 0, & \text{if } s_i > \xi \\ 1, & \text{if } s_i \leq \xi \end{array} \right\} \tag{4}$$

where $[patch]_I$ is the patch token embeddings of clean image I, $E_t(T_{tar})$ is the text embedding of the target object encoded by E_t and $CS(\cdot)$ is the cosine similarity function. The similarity vector $S \in \mathbb{R}^{n\times 1}$, where n is the number of image patches. $mask$ is also an n-dimensional vector, whose i-th element $mask_i$ is determined by comparing the corresponding element s_i in S with a predefined threshold ξ. Elements in $mask$ with a value of 0 indicate that the corresponding image patches contain richer semantics of the target object.

After obtain the $mask$, we remove the semantic of target object from both global and local perspectives. For global semantics, we push the [CLS] token embedding that represents the overall semantics of the image, away from $E_t(T_{tar})$ to remove the target object's global semantics, as shown in Eq. 5.

$$\mathcal{L}_{cls} = CS([cls]_{I'}, E_t(T_{tar})) \tag{5}$$

For local semantics, we utilize the obtained $mask$ to erase the patches containing target semantics from the image, and denote the masked image as M. In M, the patches originally rich in semantics of the target object are replaced by meaningless 0-pixel values. Then, we align the embeddings of corresponding patches in the adversarial example with those in M. This process is shown in Eq. 6.

$$\mathcal{L}_p = -(1 - mask) * CS([patch]_{I'}, [patch]_M) \tag{6}$$

Additionally, we propose the attention-$[patch]$ fixation, ensuring that the key features of the rest parts remain consistent with those of the original image, as shown in Eq. 7.

$$\mathcal{L}_{fix} = -mask * CS(A_{I'} * [patch]_{I'}, A_I * [patch]_I) \tag{7}$$

where A_I and $A_{I'}$ are the means of attention weights across all attention layers in E_v of the original image and adversarial example, as they reflect the semantic significance of each patch [51].

In conclusion, the entire process of ADVEDM-R can be expressed as Eq. 8, and w_1 to w_3 are the weights of each loss item. The pipeline of ADVEDM-R is shown in Fig. 3.

$$\begin{aligned} &\min_{I'} w_1 * \mathcal{L}_{cls} + w_2 * \mathcal{L}_p + w_3 * \mathcal{L}_{fix} \\ &s.t. \quad \|I' - I\|_2 < \epsilon \end{aligned} \tag{8}$$

Table 1: Quantitative results of attacks on MS-COCO dataset in the image description task.

Attacks \ Models	LLAVA-v2			MiniGPT4			Otter-Image			BLIP-2			OFLMG-v2			Average		
	ASR(%)	SPR(%)	SS	ASR(%)	SPR(%)	SS	ASR(%)	SPR(%)	SS	ASR(%)	SPR(%)	SS	ASR(%)	SPR(%)	SS	ASR(%)	SPR(%)	SS
PGD	71.7	22.5	0.390	70.8	17.9	0.313	73.4	18.1	0.363	74.5	17.0	0.426	79.2	20.4	0.350	73.9	19.2	0.368
MF-it	81.8	19.5	0.343	77.3	13.7	0.280	**87.5**	13.8	0.290	85.0	11.1	0.340	84.1	15.9	0.290	83.1	14.8	0.309
MF-ii	**83.6**	15.0	0.363	**82.4**	9.00	0.252	87.3	10.3	0.224	**85.6**	10.9	0.355	**86.9**	11.1	0.242	**85.2**	11.3	0.287
AdvEDM-R	75.2	**71.3**	**0.705**	73.9	**66.4**	**0.626**	80.9	**64.4**	**0.652**	74.3	**65.3**	**0.737**	79.6	**66.2**	**0.695**	76.8	**66.7**	**0.683**
AdvEDM-A	68.4	**75.1**	**0.758**	67.1	**72.6**	**0.657**	72.8	**69.4**	**0.728**	69.5	**68.2**	**0.756**	73.3	**72.9**	**0.732**	70.2	**71.6**	**0.726**

4.2.2 AdvEDM-A

In the implementation of AdvEDM-A, we first manually select contiguous patches of size $m \times m$, typically from the background or areas containing minimal specific objects. The selection can refer patches with lower embedding similarity to the foreground objects' text embeddings. After the selection, we apply the same procedure as in Eq. 4 to mark the selected patches as 0 and the remaining as 1, thereby generating the corresponding $mask$.

Due to the modality gap between image and text inputs in image-text encoders [52], textual descriptions of the target object cannot effectively inject semantics into the image embeddings, especially the patch token embeddings. To address this, we utilize a reference image R with $m \times m$ patches that solely contains the target object (generated by a text2image model [53]). Here, we also incorporate the target semantics into the original image from both global and local perspectives. For global semantics, we align the $[cls]$ of the adversarial example with the weighted fusion of $[cls]$ from the clean and target images, as detailed in Eq. 9.

$$\mathcal{L}_{cls} = -CS([cls]_{I'}, (1-\alpha)[cls]_I + \alpha[cls]_R) \tag{9}$$

Locally, the selected patches for semantic injection contain minimal information and thus have lower attention weights. To ensure the vision encoder captures the injected semantics, we reallocate attention weights of these patches by assigning them the scaled attention weights of the reference image R. Other patches undergo a similar scaling to preserve global semantic consistency. The process is described as Eq. 10. A_R is the attention weights of R and β is the scale factor.

$$A'_i = \left\{ \begin{array}{ll} \beta A_{R_i}, & \text{if } mask_i = 0 \\ (1-\beta) A_{I_i}, & \text{if } mask_i = 1 \end{array} \right\} \tag{10}$$

After obtaining the new attention weight map, we compute the key features by taking the product of the weight map and the patch token embeddings of R. Subsequently, we align the key features at corresponding positions in the adversarial example with those in R to achieve local semantic injection, as shown in Eq. 11.

$$\mathcal{L}_p = -(1-mask) * CS(A_{I'} * [patch]_{I'}, A' * [patch]_R) \tag{11}$$

Then we also consider to preserve the key features in other regions of the original image. Note that to allocate sufficient attention weights for injected semantics, we should utilize the reallocated attention weights A'. So the attention-$[patch]$ fixation can be expressed as Eq. 12.

$$\mathcal{L}_{fix} = -mask * CS(A_{I'} * [patch]_{I'}, A' * [patch]_I) \tag{12}$$

In conclusion, the overall optimization of adversarial examples is the same as Eq. 8. The pipeline of AdvEDM-A is shown in Fig. 3.

5 Experiment

We conducted experiments in **both general evaluation scenarios and EDM tasks**. The general scenario involves image description, which is consistent with existing adversarial attack evaluation scenarios in general VLMs. The EDM tasks include autonomous driving and robotic arm manipulation. Besides, more visualization results are provided in Appendix B and our webpage. The ablation studies and exploration of transferability also can be found in Appendix C and D.

Table 2: Quantitative results of attacks on Dolphins Benchmark dataset in autonomous driving scenes.

Attacks \ Models	LV-v2		MGPT-4		Otter		Dol		Avg.	
	ASR(%)	SS	ASR(%)	SS	ASR(%)	SS	ASR(%)	SS	ASR(%)	SS
PGD	22.0	0.336	16.0	0.206	13.0	0.279	18.0	0.259	17.3	0.270
MF-it	17.0	0.303	21.0	0.250	15.0	0.253	13.0	0.237	16.5	0.261
MF-ii	24.0	0.315	17.0	0.223	8.00	0.260	10.0	0.234	14.8	0.258
ADVEDM-R	**74.0**	**0.511**	**78.0**	**0.505**	**62.0**	**0.485**	**84.0**	**0.502**	**74.5**	**0.501**
ADVEDM-A	**68.0**	**0.564**	**73.0**	**0.489**	**66.0**	**0.509**	**79.0**	**0.546**	**71.5**	**0.527**

5.1 Setups

Models. We employ several commonly-used VLMs, including BLIP-2 [54], MiniGPT-4 (MGPT-4) [48], LLaVA-v2 (LV-v2) [47], Otter-Image (Otter-I) [55] and OpenFlamingo-v2 (OFLMG-v2) [56]. The vision-text encoders of BLIP-2 and MiniGPT-4 are based on EVA CLIP [26], while others are based on OpenAI's ViT CLIP [25].

Datasets. For general scenarios, we select MS-COCO 2014 [57, 58]. For the autonomous driving scenario, we choose Dolphins Benchmark [21] and DriveLM-nuScenes [4] that are specialized for this task, while for the robotic arm manipulation task, we sample 100 images from the physical world and construct instructions and actions.

Attacks. We choose several typical adversarial attacks as baseline, including CLIP-Based PGD [19], MF-it and MF-ii [10]. The norm constraint of the adversarial perturbation ϵ is set to 8/255. For CLIP-Based PGD, we set the number of iterations to 20 with a step size of 0.01. For MF-it and MF-ii, we use their official open-source code. Details of our methods' settings are in Appendix A.

Metrics. We define Attack Success Rate (ASR) and Semantic Similarity (SS) to measure attack effectiveness. Specifically, SS is the cosine similarity between output embeddings of adversarial and clean samples calculated by a text encoder. We introduce Semantic Preservation Rate (SPR), which measures the retention of other objects' semantics in VLM description of input images. The detailed formal definitions of these metrics are provided in Appendix A.

Note that the interpretation of evaluation metrics vary across different tasks. In the general image description task, an attack is considered successful if it causes the target object to include or exclude in the generated description. In this case, a higher ASR values indicate greater attack effectiveness, while higher SPR and SS values reveal the attack better preserves the semantic integrity of the image. In the EDM tasks, the attack is successful if the output decisions and plannings align with the expected results after the addition or removal of target object's semantics (*i.e.*, valid yet incorrect). However, since VLM-based EDM systems do not provide detailed descriptions of all objects in the inputs, SPR cannot be reliably computed and is thus omitted from the corresponding experiments.

5.2 Evaluation in General Scenarios

Settings. According to our threat model, we employ the vision-text encoder of each victim model as surrogate model to launch attack. For the attack target, we randomly select an object in the image to remove its semantics or choose an object not in the image to inject its semantics. We randomly select 1000 images to generate adversarial examples and record the average ASR, SPR and SS.

Results. Tab. 1 shows that though existing attacks achieve high ASR values, they suffer from low semantic preservation, with SPR values under 20%. This indicates they disrupt the majority of the original image's semantics, resulting in poor fine-grained control. In contrast, our methods exhibit a slight decrease in ASR, but their SPR values are significantly higher (66.7% and 71.6%, respectively), preserving most of the original image's semantics and enabling fine-grained control. Our methods also achieve higher SS values, as it preserves most of the original image's semantics.

5.3 Evaluation in EDM tasks

5.3.1 Autonomous Driving Task

Settings. We conduct experiments on two specialized datasets, Dolphins Benchmark and DriveLM-nuScenes, with three general VLMs (LLAVA-v2, MniGPT-4, and Otter-Image) and Dolphins (Dol) [21], a VLM designed for decision-making task in autonomous driving. For each dataset, we randomly

Table 3: Quantitative results of attacks on DriveLM-nuScenes dataset in autonomous driving scenes.

Attacks \ Models	LV-v2		MGPT-4		Otter		Dol		Avg.	
	ASR(%)	SS	ASR(%)	SS	ASR(%)	SS	ASR(%)	SS	ASR(%)	SS
PGD	20.0	0.274	19.0	0.218	16.0	0.286	23.0	0.243	19.5	0.255
MF-it	19.0	0.263	24.0	0.242	9.00	0.245	17.0	0.237	17.3	0.247
MF-ii	14.0	0.260	17.0	0.221	12.0	0.253	14.0	0.234	14.3	0.242
ADVEDM-R	**79.0**	**0.487**	**83.0**	**0.504**	**75.0**	**0.472**	**86.0**	**0.544**	**80.8**	**0.502**
ADVEDM-A	**73.0**	**0.536**	**79.0**	**0.517**	**71.0**	**0.509**	**82.0**	**0.530**	**76.3**	**0.523**

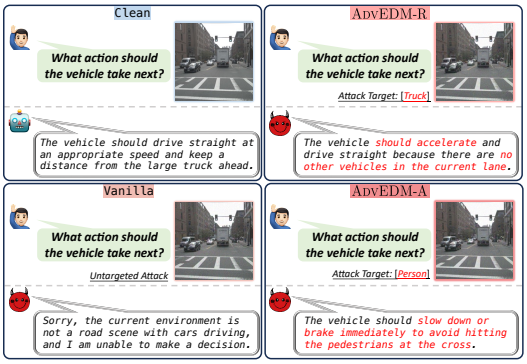


Figure 5: Visualization results in the autonomous driving decision-making task.

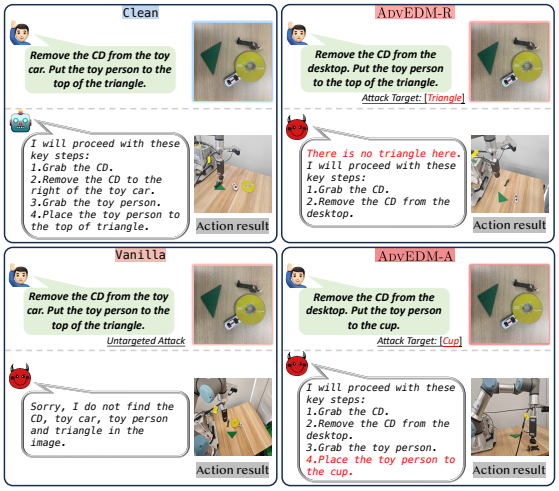


Figure 6: Visualization results in the VLM-based robotic arm manipulation task.

select 100 road scene images and choose common objects in the road as target, such as vehicles, pedestrians, and traffic lights. The impact of various attacks on the reasoning process of VLMs is presented in detail in Appendix C.

Results. The quantitative results are reported in Tab. 2 and 3, and the visualization results are shown in Fig. 5. According to the quantitative results, the average ASR of our methods are over 70% and 75% on the two datasets, respectively, dramatically outperforming existing attack methods. This demonstrates that our attack can have a substantial impact on VLM-based decision-making systems in autonomous driving. Our methods also achieve higher SS, as they ensure the model outputs valid decisions that are structurally identical to normal outputs, such as "The vehicle should go straight/turn left." Besides, the visualization results also demonstrate that our methods enable VLMs to make valid but incorrect decisions, thereby affecting the vehicle's driving state.

5.3.2 Robotic Manipulation Task

Settings. We take the robotic arm manipulation task as an example, where various objects are placed on desktop, and then 100 images are captured to form the dataset. We still select three general VLMs and EmbodiedGPT (EmGPT) [24], a VLM specifically designed for this task, as victim models. For each image, the instructions we design involve the manipulation of two or more objects, including the target object selected for attack. Moreover, to better visualize the results, we deploy the VLMs on a UR robotic arm, where their outputs directly influence the robotic arm's actions. The impact of various attacks on the reasoning process of VLMs is also discussed in Appendix C.

Results. The quantitative results are shown in Tab. 4 and the visualization results are shown in Fig. 6. The presentation videos of attacking on robotic arm manipulation can be found on our website. The results highlight the superior effectiveness of our attack methods, achieving ASR values of 68.8% and 64.5%, significantly surpassing those of existing attacks. When VLMs encounter existing attacks, the perceived semantics are completely different from the original image, leading to error messages such as "Sorry, there is no object A nor B on the desktop.", resulting in lower SS values. In contrast,

Table 4: Quantitative results of attacks in robotic arm manipulation task.

Attacks \ Models	LV-v2		MGPT-4		Otter		EmGPT		Avg.	
	ASR(%)	SS	ASR(%)	SS	ASR(%)	SS	ASR(%)	SS	ASR(%)	SS
PGD	16.0	0.123	12.0	0.165	11.0	0.103	7.00	0.079	11.5	0.117
MF-it	11.0	0.132	15.0	0.185	14.0	0.118	10.0	0.090	12.5	0.131
MF-ii	9.00	0.119	12.0	0.155	6.00	0.091	9.00	0.095	9.0	0.115
ADVEDM-R	**72.0**	**0.459**	**66.0**	**0.428**	**63.0**	**0.466**	**74.0**	**0.450**	**68.8**	**0.451**
ADVEDM-A	**67.0**	**0.417**	**63.0**	**0.448**	**58.0**	**0.436**	**70.0**	**0.487**	**64.5**	**0.447**

our methods mainly alter the VLM's decisions and plannings regarding the target object, while the decisions and plannings for non-target objects remain similar to those make for clean images, ensuring that the SS values remain sufficiently higher.

6 Conclusion

In this work, we propose a fine-grained adversarial attack framework ADVEDM against the VLM-based EDM systems, which aims to reveal the vulnerabilities of them. By only disrupting the VLM's perception of the target object while preserving other semantics, our attack maintains the integrity of VLM's reasoning process, enabling it to output valid yet incorrect decisions that influence the entity's interactions with the real world. Specifically, we design two attack variants: ADVEDM-R, which removes the semantics of a specific object from the image, and ADVEDM-A, which injects the semantics of a new object into the image. Experimental results in both general scenarios and EDM tasks demonstrate the superior fine-grained control and attack effectiveness of our methods against the VLM-based EDM systems, outperforming existing attacks targeting the VLM itself.

Acknowledgements

Minghui Li's work is supported by the National Natural Science Foundation of China under Grant No. 62202186. Shengshan Hu's work is supported by the National Natural Science Foundation of China under Grant No.62372196. Minghui Li is the corresponding author.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Ö. Aydın, "Google bard generated literature review: metaverse," *Journal of AI*, vol. 7, no. 1, pp. 1–14, 2023. 1, 24

[3] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862, June 2023. 1

[4] C. Sima, K. Renz, K. Chitta, L. Chen, H. Zhang, C. Xie, J. Beißwenger, P. Luo, A. Geiger, and H. Li, "Drivelm: Driving with graph visual question answering," in *European Conference on Computer Vision*, pp. 256–274, Springer, 2024. 1, 3, 8

[5] H. Zhang, S. Hu, Y. Wang, L. Y. Zhang, Z. Zhou, X. Wang, Y. Zhang, and C. Chen, "Detector collapse: backdooring object detection to catastrophic overload or blindness in the physical world," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 1670–1678, 2024. 1

[6] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically grounded vision-language models for robotic manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12462–12469, IEEE, 2024. 1, 5

[7] W. Zhao, J. Chen, Z. Meng, D. Mao, R. Song, and W. Zhang, "Vlmpc: Vision-language model predictive control for robotic manipulation," *arXiv preprint arXiv:2407.09829*, 2024. 1

[8] X. Wang, H. Pan, H. Zhang, M. Li, S. Hu, Z. Zhou, L. Xue, P. Guo, A. Liu, L. Y. Zhang, *et al.*, "Trojanrobot: Physical-world backdoor attacks against vlm-based robotic manipulation," *arXiv preprint arXiv:2411.11683*, 2024. 1

[9] Z. Zhou, S. Hu, M. Li, H. Zhang, Y. Zhang, and H. Jin, "Advclip: Downstream-agnostic adversarial examples in multimodal contrastive learning," in *Proceedings of the 32nd ACM International Conference on Multimedia (MM'23)*, pp. 6311–6320, 2023. 2, 4

[10] Y. Zhao, T. Pang, C. Du, X. Yang, C. Li, N.-M. M. Cheung, and M. Lin, "On evaluating adversarial robustness of large vision-language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024. 2, 3, 4, 5, 8

[11] Z. Zhou, B. Li, Y. Song, S. Hu, W. Wan, L. Y. Zhang, D. Yao, and H. Jin, "Numbod: A spatial-frequency fusion attack against object detectors," in *Proceedings of the 39th Annual AAAI Conference on Artificial Intelligence (AAAI'25)*, 2025. 2

[12] Y. Song, Z. Zhou, M. Li, X. Wang, M. Deng, W. Wan, S. Hu, and L. Y. Zhang, "Pb-uap: Hybrid universal adversarial attack for image segmentation.," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'25)*, 2025. 2

[13] Z. Zhou, Y. Hu, Y. Song, Z. Li, S. Hu, L. Y. Zhang, D. Yao, L. Zheng, and H. Jin, "Vanish into thin air: Cross-prompt universal adversarial attacks for sam2," in *Proceedings of the 39th Annual Conference on Neural Information Processing Systems (NeurIPS'25)*, 2025. 2, 3

[14] M. Li, J. Wang, H. Zhang, Z. Zhou, S. Hu, and X. Pei, "Transferable adversarial facial images for privacy protection," in *Proceedings of the 32nd ACM International Conference on Multimedia (ACM MM'24)*, pp. 10649–10658, 2024. 2

[15] Y. Dong, H. Chen, J. Chen, Z. Fang, X. Yang, Y. Zhang, Y. Tian, H. Su, and J. Zhu, "How robust is google's bard to adversarial image attacks?," *arXiv preprint arXiv:2309.11751*, 2023. 2, 4

[16] H. Luo, J. Gu, F. Liu, and P. Torr, "An image is worth 1000 lies: Transferability of adversarial images across prompts on vision-language models," in *The Twelfth International Conference on Learning Representations*, 2023. 2

[17] S. Zhai, H. Bai, Z. Lin, J. Pan, P. Tong, Y. Zhou, A. Suhr, S. Xie, Y. LeCun, Y. Ma, *et al.*, "Fine-tuning large vision-language models as decision-making agents via reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 110935–110971, 2025. 2

[18] C. Zhang, X. Xu, J. Wu, Z. Liu, and L. Zhou, "Adversarial attacks of vision tasks in the past 10 years: A survey," *arXiv preprint arXiv:2410.23687*, 2024. 2

[19] X. Cui, A. Aparcedo, Y. K. Jang, and S.-N. Lim, "On the robustness of large multimodal models against image adversarial attacks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24625–24634, 2024. 2, 3, 5, 8

[20] Y. Wang, C. Liu, Y. Qu, H. Cao, D. Jiang, and L. Xu, "Break the visual perception: Adversarial attacks targeting encoded visual tokens of large vision-language models," in *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 1072–1081, 2024. 2, 5

[21] Y. Ma, Y. Cao, J. Sun, M. Pavone, and C. Xiao, "Dolphins: Multimodal language model for driving," in *European Conference on Computer Vision*, pp. 403–420, Springer, 2024. 2, 3, 8

[22] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24824–24837, 2022. 2, 3

[23] Y. Chen, K. Sikka, M. Cogswell, H. Ji, and A. Divakaran, "Measuring and improving chain-of-thought reasoning in vision-language models," *arXiv preprint arXiv:2309.04461*, 2023. 2, 3

[24] Y. Mu, Q. Zhang, M. Hu, W. Wang, M. Ding, J. Jin, B. Wang, J. Dai, Y. Qiao, and P. Luo, "Embodiedgpt: Vision-language pre-training via embodied chain of thought," *Advances in Neural Information Processing Systems*, vol. 36, pp. 25081–25094, 2023. 2, 5, 9

[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PmLR, 2021. 3, 4, 8

[26] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao, "Eva-clip: Improved training techniques for clip at scale," *arXiv preprint arXiv:2303.15389*, 2023. 3, 8

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017. 3

[28] M.-Y. Lin, O.-W. Lee, and C.-Y. Lu, "Embodied ai with large language models: A survey and new hri framework," in *2024 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 978–983, IEEE, 2024. 3, 4

[29] H. Zhang, C. Zhu, X. Wang, Z. Zhou, C. Yin, M. Li, L. Xue, Y. Wang, S. Hu, A. Liu, *et al.*, "Badrobot: Jailbreaking embodied llms in the physical world," *arXiv preprint arXiv:2407.20242*, 2024. 3

[30] L. Yu, Y. Zhang, Z. Zhou, Y. Wu, W. Wan, M. Li, S. Hu, P. Xiaobing, and J. Wang, "Spa-vlm: Stealthy poisoning attacks on rag-based vlm," *arXiv preprint arXiv:2505.23828*, 2025. 3

[31] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K.-Y. K. Wong, Z. Li, and H. Zhao, "Drivegpt4: Interpretable end-to-end autonomous driving via large language model," *IEEE Robotics and Automation Letters*, 2024. 3

[32] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "VIMA: Robot manipulation with multimodal prompts," in *Proceedings of the 40th International Conference on Machine Learning (ICLR'23)*, vol. 202, pp. 14975–15022, 2023. 3

[33] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR'15)*, 2015. 3

[34] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017. 3

[35] Z. Zhou, M. Li, W. Liu, S. Hu, Y. Zhang, W. Wan, L. Xue, L. Y. Zhang, D. Yao, and H. Jin, "Securely fine-tuning pre-trained encoders against adversarial examples," in *Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP'24)*, 2024. 3

[36] Z. Zhou, S. Hu, R. Zhao, Q. Wang, L. Y. Zhang, J. Hou, and H. Jin, "Downstream-agnostic adversarial examples," in *Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV'23)*, pp. 4345–4355, 2023. 3

[37] Z. Zhou, Y. Song, M. Li, S. Hu, X. Wang, L. Y. Zhang, D. Yao, and H. Jin, "Darksam: Fooling segment anything model to segment nothing," in *Proceedings of the 38th Annual Conference on Neural Information Processing Systems (NeurIPS'24)*, 2024. 3

[38] Y. Wang, Y. Chou, Z. Zhou, H. Zhang, W. Wan, S. Hu, and M. Li, "Breaking barriers in physical-world adversarial examples: Improving robustness and transferability via robust feature," in *Proceedings of the 39th Annual AAAI Conference on Artificial Intelligence (AAAI'25)*, 2025. 3

[39] Y. Song, Z. Zhou, Q. Lu, H. Zhang, Y. Hu, L. Xue, S. Hu, M. Li, and L. Y. Zhang, "Segtrans: Transferable adversarial examples for segmentation models," *IEEE Transactions on Multimedia*, 2025. 3

[40] H. Zhang, Z. Yao, L. Y. Zhang, S. Hu, C. Chen, A. Liew, and Z. Li, "Denial-of-service or fine-grained control: towards flexible model poisoning attacks on federated learning," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 4567–4575, 2023. 3

[41] X. Fu, Z. Wang, S. Li, R. K. Gupta, N. Mireshghallah, T. Berg-Kirkpatrick, and E. Fernandes, "Misusing tools in large language models with visual adversarial examples," *arXiv preprint arXiv:2310.03185*, 2023. 3

[42] K. Gao, Y. Bai, J. Bai, Y. Yang, and S.-T. Xia, "Adversarial robustness for visual grounding of multimodal large language models," *arXiv preprint arXiv:2405.09981*, 2024. 3

[43] C. Schlarmann and M. Hein, "On the adversarial robustness of multi-modal foundation models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 3677–3685, October 2023. 3

[44] Z. Wang, Z. Han, S. Chen, F. Xue, Z. Ding, X. Xiao, V. Tresp, P. Torr, and J. Gu, "Stop reasoning! when multimodal llms with chain-of-thought reasoning meets adversarial images," *arXiv preprint arXiv:2402.14899*, 2024. 3

[45] D. Liu, M. Yang, X. Qu, P. Zhou, Y. Cheng, and W. Hu, "A survey of attacks on large vision-language models: Resources, advances, and future trends," *arXiv preprint arXiv:2407.07403*, 2024. 4, 5, 21

[46] Q. Guo, S. Pang, X. Jia, and Q. Guo, "Efficiently adversarial examples generation for visual-language models under targeted transfer scenarios using diffusion models," *arXiv preprint arXiv:2404.10335*, 2024. 4

[47] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Proceedings of the 36th Advances in Neural Information Processing Systems (NeurIPS'23)*, 2023. 4, 8

[48] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023. 4, 5, 8

[49] S. Liu, J. Chen, S. Ruan, H. Su, and Z. Yin, "Exploring the robustness of decision-level through adversarial attacks on llm-based embodied models," in *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 8120–8128, 2024. 4

[50] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 4

[51] R. Yu, W. Yu, and X. Wang, "Attention prompting on image for large vision-language models," in *Proceddings of the 18th European Conference of Computer Vision (ECCV'24)*, vol. 15088, pp. 251–268, Springer, 2024. 6

[52] V. W. Liang, Y. Zhang, Y. Kwon, S. Yeung, and J. Y. Zou, "Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17612–17625, 2022. 7

[53] A. Borji, "Generated faces in the wild: Quantitative comparison of stable diffusion, midjourney and dall-e 2," *arXiv preprint arXiv:2210.00586*, 2022. 7

[54] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*, pp. 19730–19742, PMLR, 2023. 8

[55] B. Li, Y. Zhang, L. Chen, J. Wang, F. Pu, J. Yang, C. Li, and Z. Liu, "Mimic-it: Multi-modal in-context instruction tuning," *arXiv preprint arXiv:2306.05425*, 2023. 8

[56] A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe, Y. Bitton, S. Gadre, S. Sagawa, *et al.*, "Openflamingo: An open-source framework for training large autoregressive vision-language models," *arXiv preprint arXiv:2308.01390*, 2023. 8

[57] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pp. 740–755, Springer, 2014. 8

[58] H. Zhang, Y. Wang, S. Yan, C. Zhu, Z. Zhou, L. Hou, S. Hu, M. Li, Y. Zhang, and L. Y. Zhang, "Test-time backdoor detection for object detection models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24377–24386, 2025. 8

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 21

[60] H. Chen, Y. Zhang, Y. Dong, X. Yang, H. Su, and J. Zhu, "Rethinking model ensemble in transfer-based adversarial attacks," *arXiv preprint arXiv:2303.09105*, 2023. 24

[61] Y. Long, Q. Zhang, B. Zeng, L. Gao, X. Liu, J. Zhang, and J. Song, "Frequency domain model augmentation for adversarial attack," in *European conference on computer vision*, pp. 549–566, Springer, 2022. 24

[62] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023. 24

[63] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024. 24

[64] T. Kumamoto, Y. Yoshida, and H. Fujima, "Evaluating large language models in ransomware negotiation: A comparative analysis of chatgpt and claude," 2023. 24

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are made in the abstract and introduction accurately.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The discussion about limitations is included in our appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our experimental settings are provided in the appendix and part of our source code is in the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Part of our source code is in the supplementary material. The complete code will be released after publication.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specify all the training and test details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports appropriate information about the statistical significance of the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The information is provided in our appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research confirms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This is included in the Introduction and Conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We explicitly acknowledge and credit the original repository authors in the provided code.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We adopt LLM-as-a-judge in our experiments to processing results. The detailed usage instructions and settings are provided in the appendix and the source code.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Detailed Experimental Settings

Parameter Settings of our methods. For ADVEDM-R, we select the top 20% of image patches with the highest similarity to the target object's text embedding and mask their pixels to generate masked image M. We set w_1, w_2 and w_3 in Eq. (3) to 0.5, 2, and 0.2, respectively. The optimization is performed for 500 iterations using the Adam optimizer [59] with a learning rate of 0.005. For ADVEDM-A, we We select a 100 $\times$ 100 pixel region in the image for semantic injection. When reallocating attention weights, we set the scaling factor $\beta = 0.4$ in Eq. (12), and the fusion weight α of [CLS] tokens is 0.5 in Eq. (11). During the optimization process, w_1 to w_3 is set to 0.8, 2, and 0.3, respectively. The remaining optimization settings are kept identical to those of ADVEDM-R.

Experimental environment. All experiments are conducted on NVIDIA A100-SXM4 GPUs, each equipped with 80GB of memory. For the calculation of metrics ASR and SPR, we adopt the LLM-as-judge approach [45], employing GPT-3.5-turbo and other LLMs.

Definition of metrics. Semantic Similarity (SS) is defined as the cosine similarity of between output embeddings of adversarial and clean images calculated by a text encode, which can be expressed as Eq. 13:

$$SS = CS(E_t(T), E_t(T')) \tag{13}$$

where $CS(\cdot)$ is the cosine similarity function and T and T' represent the textual outputs of clean images and adversarial examples respectively.

As for calculating SPR, the procedure is divided into two steps. The first step involves decomposing all objects included in the description T and T', which can be done by GPT-3.5-turbo, as shown in Eq. 14. The specific prompts for extracting the semantics of objects are provided in our source code.

$$\begin{aligned} \mathbf{D} &= L(T) = \{Obj_1, Obj_2, ..., Obj_n\} \\ \mathbf{D'} &= L(T') = \{Obj'_1, Obj'_2, ..., Obj'_{n'}\} \end{aligned} \tag{14}$$

L is the LLM served as the judge. **D** and **D'** are the set of objects whose semantics are included in the description T and T'. Then we compute the SPR value as the preservation rate of the original object semantics in **D'**, expressed as Eq. 15.

$$SPR = \frac{|\mathbf{D'} \bigcap \mathbf{D}|}{|\mathbf{D}|} \tag{15}$$

where $|\cdot|$ here represents the number of elements in the set.

B More Results of Experiments in EDM tasks

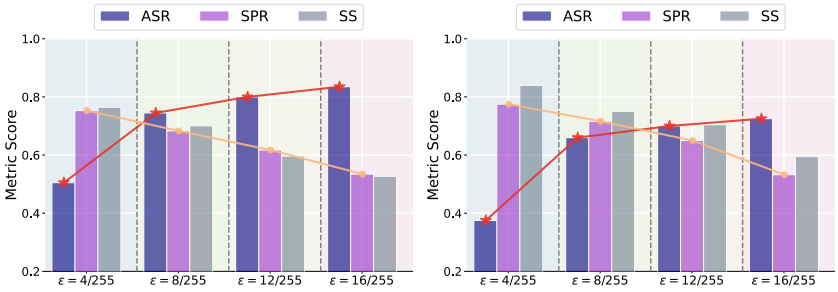
Here, we illustrate the reasoning process of VLM-based embodied decision-making systems to better demonstrate how our methods precisely affect the VLM's CoT while preserving its structural integrity.

The CoT of VLMs in the autonomous driving task under various attacks as follows. The input image and prompts are identical to Fig. 5 in the manuscript. Our methods enable the VLM to correctly reason about non-target objects (like the green traffic lights) while inducing incorrect reasoning specifically for the target object. This demonstrates the fine-grained control of our methods, accurately manipulating targeted part of the VLM's reasoning process.

CoT of the clean image: *In current scenario, the green light allows the vehicle to proceed through the intersection. Given the surrounding traffic—specifically, a truck ahead in the current lane and cars in the left lane—the vehicle should maintain a safe speed and distance while continuing straight.*

CoT of existing attacks: *The current scene depicts a plate of fruit on a table with a cat nearby. As it bears no clear connection to a road scenario, no relevant decision can be made.*

CoT of ADVEDM-R: *In the current scenario, the traffic light is green, allowing the vehicle to proceed straight. With no vehicles or pedestrians ahead in the lane, it can accelerate safely.*



(a) AdvEDM-R (b) AdvEDM-A

Figure 7: Results of our methods under various ϵ values.

CoT of AdvEDM-A: *In the current scenario, the green traffic light indicates that the vehicle can proceed straight. However, since a pedestrian is at the intersection, the vehicle should slow down and brake to yield. Once the pedestrian has passed, it can continue forward.*

The CoT of VLMs in robotic arm manipulation are as follows. The input image and prompts are consistent with Fig. 6 in the manuscript. Existing attacks prevent the VLM from recognizing any objects mentioned in the instructions, disrupting the reasoning process and terminating the task. AdvEDM-R enables the VLM to remain unaware of the target object while leaving other objects unaffected. As a result, the system makes incorrect decisions related to the target object while correctly executing other instructions. AdvEDM-A makes the VLM perceive the injected semantics of the target object, and then make corresponding decisions and plannings for instructions involving the object.

In conclusion, our methods perturb the VLM's perception process to precisely manipulate its reasoning about the target object while maintaining the coherence and integrity of the entire CoT.

CoT of the clean image: *Seeing a CD on the toy car, grasp and remove it. Noticing the toy person and the triangle, pick up the toy person and carefully place it on top of the triangle, ensuring stability.*

CoT of existing attacks: *Detected an apple and a book on the table, but no CD, toy car, or triangle. Adjust the camera to search further. Found a fork and a doll in the corner, but no matching objects. Task aborted.*

CoT of AdvEDM-R: *Detected a CD, a toy car, and a toy person but no triangle. Remove the CD first. With no target for the toy person, pause the task and await further instructions.*

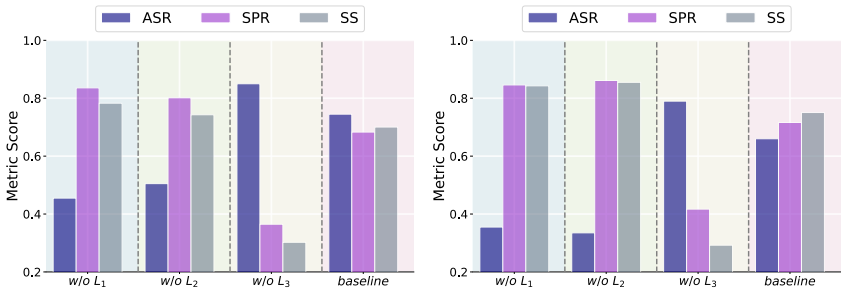
CoT of AdvEDM-A: *Detected a CD, a toy car, a toy person, a triangle, and a cup. First, grab and remove the CD. Then, seeing the cup as the target container, decide to pick up the toy person and carefully place it into the cup, ensuring stability before completing the task.*

C Ablation Studies

Influence of ϵ. The ϵ in the manuscript is the norm constraint of adversarial perturbation. Here, we set a range of ϵ values and select 100 images from MS-COCO for evaluation. Other settings are the same as Sec. 5.1 in the manuscript. The average results for various metrics in five victim VLMs are shown in Fig. 7. The visualization results under different ϵ are also illustrated in Fig. 8.

According to the results, the attack effectiveness is inferior when ϵ is set to 4/255, as the adversarial perturbation is too slight to disrupt the perception of VLMs. As ϵ increases, the magnitude of adversarial perturbation grows and our methods achieve stronger attack effectiveness. However, too large magnitude of perturbation breaks the semantics of non-target objects in the image, resulting in a significant drop in the SPR values, especially when $\epsilon \geq 12/255$. So $\epsilon = 8/255$ serves as an optimal setting that balances attack effectiveness and fine-grained control.

Figure 8: Visualization results under different ϵ settings.



(a) AdvEDM-R (b) AdvEDM-A

Figure 9: Results of ablation study about three loss functions of our methods.

Ablation about loss functions. As shown in Eq. (10) in the manuscript, the optimization objective of our methods consists of three loss functions: L_{cls}, L_p, and L_{fix}. We also select 100 images from MS-COCO and also keep other settings identical to Sec. 5.2. The average results of various metrics are illustrated in Fig. 9.

The ASR values decrease dramatically without L_{cls} or L_p, which demonstrates that removing or injecting the target object's semantics from both global and local perspectives is more effective. Additionally, L_{fix} is critical to preserve non-target objects' semantics, as the SPR values decrease to about 0.35 and 0.4 respectively without L_{fix}.

Ablation about modules and hyper-parameters in AdvEDM-A. As for AdvEDM-A, we additionally provide some experimental results of ablation studies on attention-weight reallocation and the hyper-parameters α and β in Eq. 9 and 10 to investigate their impact on the attack performance. Specifically, we randomly select 100 images from the COCO dataset and evaluate the results on MiniGPT-4 and LLAVA-V2. All other experimental settings remain the same as those described in Appendix A. The results are presented in Tab. 5 and 6, respectively.

According to Tab. 5, without the attention-weight reallocation, both SPR and SS drop significantly, which highlights the importance of this mechanism for fine-grained control. It ensures sufficient attention is allocated to the original regions of the image, enabling the model to retain awareness of the remaining semantics after the new semantic injection.

The results in Tab. 5 and 6 indicate that both α and β influence the strength of semantic injection for the target objects. Larger values of them lead to increased attention weights assigned to the injected semantics, resulting in higher ASR. However, this also affects the model's perception of the semantics of other objects in the image, leading to a decrease in SPR and SS. Overall, to better balance the two aspects, we adopt $\alpha = 0.5$ and $\beta = 0.4$ as our default configuration.

Table 5: Ablation study of reallocation module and α in Eq. 9.

Settings	ASR (%)	SPR (%)	SS
$\alpha = 0.3$	58.0	85.3	0.804
$\alpha = 0.4$	67.5	80.7	0.776
$\alpha = 0.5$	**72.5**	**74.8**	**0.727**
$\alpha = 0.6$	78.0	66.1	0.655
$\alpha = 0.7$	81.5	53.6	0.583
w/o realloc.	70.5	44.3	0.515

Table 6: Ablation study of β in Eq. 10.

Settings	ASR (%)	SPR (%)	SS
$\beta = 0.2$	64.0	81.6	0.785
$\beta = 0.3$	69.5	79.0	0.752
$\beta = 0.4$	**72.5**	**74.8**	**0.727**
$\beta = 0.5$	74.0	67.6	0.690
$\beta = 0.6$	77.0	59.3	0.644

Table 7: Ablation studies of the weights of loss functions in ADVEDM-R and ADVEDM-A.

Settings \ Metrics	ADVEDM-R			ADVEDM-A		
	ASR (%)	SPR (%)	SS	ASR (%)	SPR (%)	SS
group (1)	79.5	64.3	0.626	77.0	69.2	0.674
group (2)	82.0	61.8	0.609	81.5	63.1	0.658
group (3)	69.0	74.5	0.723	65.5	79.6	0.771
baseline	75.0	71.8	0.691	72.5	74.8	0.727

Ablation about the weights of loss functions. We conduct studies on various combinations of loss function weights, using the same data and model settings as described above. For AdvEDM-R, we set the groups: (1) $w_1 = 0.8, w_2 = 2, w_3 = 0.2$; (2) $w_1 = 0.5, w_2 = 2.3, w_3 = 0.2$; (3) $w_1 = 0.5, w_2 = 2.0, w_3 = 0.5$; (4) $w_1 = 0.5, w_2 = 2.0, w_3 = 0.2$ (baseline). For AdvEDM-A, we set the groups: (1) $w_1 = 1.0, w_2 = 2.0, w_3 = 0.3$; (2) $w_1 = 1.0, w_2 = 2.2, w_3 = 0.3$; (3) $w_1 = 0.8, w_2 = 2.0, w_3 = 0.5$; (4) $w_1 = 0.8, w_2 = 2.0, w_3 = 0.3$ (baseline). The results are presented in Tab. 7.

According to the quantitative results, the weights w_1 and w_2 control the removal and injection of target semantics. Increasing their weights tends to improve the ASR, as more attention is directed toward the semantics of the target object. However, this also reduces the preservation of the original semantics in the image, leading to lower SPR and SS values. By contrast, increasing w_3 encourages the model to preserve more of the original semantics, but it weakens the target objects' semantics, resulting in a decrease in ASR.

D Exploration of Transferability

To further evaluate the performance of our methods in black-box scenarios where the adversary has no knowledge about the victim model, we adapt them into transfer-based attacks. Specifically, we adopt SSA-CWA algorithm [60, 61] during the optimization process and employ four vision-text encoders, CLIP-ViT-L14, CLIP-ViT-B32, CLIP-ViT-bigG-14, and ViT-SO400M-14-SigLIP [62], as an ensemble of surrogate models.

Settings. We randomly select 100 images from MS-COCO and set target objects. The victim models include four commercial black-box VLMs: GPT-4o [63], Gemini-2.0 [2], Claude 3.5 [64]. The number of iterations is set to 30 for SSA-CWA, and the constraint of perturbation ϵ is 16/255. Other settings are identical to Sec. 5.2.

Results. The results are shown in Tab. 8. In the more challenging black-box setting, the attack effectiveness of our methods degrades, with the ASR of the two methods dropping by approximately 20% and 30% compared with attacks in the gray-box setting. Nevertheless, our methods still maintain a notable level of effectiveness against commercial VLMs while preserving fine-grained control, highlighting their potential for transferability to black-box scenarios. How to further enhance the transferability of our attacks will be explored in future work.

Table 8: Results of our methods in the black-box setting.

ADVEDM-R				
Metrics \ Models	GPT-4o	Gemini	Claude	Average
ASR(%)	58.0	55.0	60.0	57.7
SPR(%)	71.3	68.1	70.6	70.0
SS	0.656	0.631	0.611	0.632
ADVEDM-A				
Metrics \ Models	GPT-4o	Gemini	Claude	Average
ASR(%)	44.0	38.0	35.0	39.0
SPR(%)	73.7	76.4	77.5	75.9
SS	0.693	0.701	0.718	0.704

E Discussion about Limitations

In this work, we focus on leveraging adversarial perturbations to achieve fine-grained control over a specific aspect of image semantics, namely the presence or absence of a particular object. In fact, other levels of fine-grained semantics, such as altering the spatial location of objects or their interrelations, may also induce valid yet incorrect decisions in embodied EDM systems. A comprehensive analysis of these dimensions remains beyond the scope of this work, and we will further explore these aspects in our future work.

Moreover, our attack involves manipulating the image uploaded by the user to the VLM, thereby interfering with the system's decision-making process in the digital domain. While certain existing techniques, such as network interception and packet tampering—could potentially enable such attack, it offers limited flexibility. In future work, we plan to explore physically deployable adversarial examples (*e.g.*, adversarial patches) to enable more passive and practical attack scenarios in the physical world.