Disrupting Vision-Language Model-Driven Navigation Services via Adversarial Object Fusion

Chunlong Xie, Jialing He, *IEEE Member*, Shangwei Guo, *IEEE Member*, Jiacheng Wang, *IEEE Member*, Shudong Zhang, Tianwei Zhang, *IEEE Member*, Tao Xiang, *IEEE Senior Member*,

Abstract—We present Adversarial Object Fusion (**AdvOF**), a novel attack framework targeting vision-and-language navigation (VLN) agents in service-oriented environments by generating adversarial 3D objects. While foundational models like Large Language Models (LLMs) and Vision Language Models (VLMs) have enhanced service-oriented navigation systems through improved perception and decision-making, their integration introduces vulnerabilities in mission-critical service workflows. Existing adversarial attacks fail to address service computing contexts, where reliability and quality-of-service (QoS) are paramount. We utilize **AdvOF** to investigate and explore the impact of adversarial environments on the VLM-based perception module of VLN agents. In particular, **AdvOF** first precisely aggregates and aligns the victim object positions in both 2D and 3D space, defining and rendering adversarial objects. Then, we collaboratively optimize the adversarial object with regularization between the adversarial and victim object across physical properties and VLM perceptions. Through assigning importance weights to varying views, the optimization is processed stably and multi-viewedly by iterative fusions from local updates and justifications. Our extensive evaluations demonstrate **AdvOF** can effectively degrade agent performance under adversarial conditions while maintaining minimal interference with normal navigation tasks. This work advances the understanding of service security in VLM-powered navigation systems, providing computational foundations for robust service composition in physical-world deployments.

Index Terms—Vision-and-Language Navigation, Adversarial Attack, Vision-Language Model

1 Introduction

Service computing has advanced intelligent automation across cloud [1], edge [2], and IoT platforms [3], with Vision-and-Language Navigation (VLN) [4] emerging as a critical component in real-world applications such as smart cities, autonomous delivery, and assistive robotics. VLN agents interpret human instructions and visually perceive environments to navigate unfamiliar environments. Typical VLN approaches rely on representation learning [5], reinforcement learning [6] and imitation learning [7], yet they often struggle with generalization in complex environments due to limited data and task-specific training. Recent advancements and applications in foundational models, particularly Large Language Models (LLMs) [8]–[10] and Vision Language Models (VLMs) [11]–[15], have addressed these limitations by significantly enhancing generalization capabilities of VLN agents [16]–[19]. By integrating foundation models into core VLN modules [20], [21], agents can better understand natural language instructions and perceive complex visual environments. Specifically, LLMs facilitate high-level interaction and task planning [18], [22], while VLMs enhance low-level perception through improved feature extraction [11] and scene recognition [23]. The integration of foundation models is gradually shaping a new deployment paradigm for VLN agents.

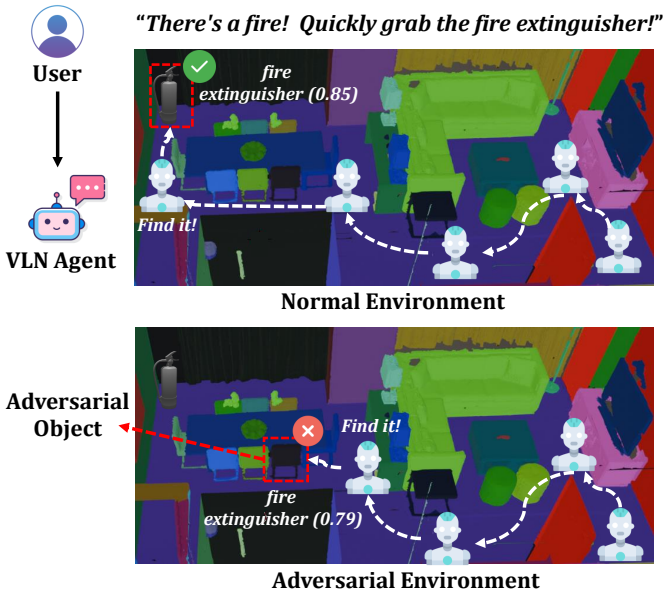


Fig. 1. Attacking a VLN agent with the adversarial object. The VLN Agent misidentifies the adversarial object (original "chair") as the fire extinguisher.

Chunlong Xie, Shangwei Guo, Jialing He, and Tao Xiang are with College of Computer Science, Chongqing University, Chongqing, China, 400044.
Jiacheng Wang is with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798.
Shudong Zhang is with the School of Computer Science and Technology, Xidian University, Xi'an, China. 710071.
Tianwei Zhang is with College of Computing and Data Science, Nanyang Technological University, Singapore, 639798.
Shangwei Guo (swguo@cqu.edu.cn) and Jiacheng Wang (jiacheng.wang@ntu.edu.sg) are the corresponding authors.

However, this integration of foundation models could raise new security risks for VLN agents, particularly adversarial attacks [24]–[28]. Existing research has revealed robustness problems in foundation models through adversarial attacks. For example, attackers can manipulate LLMs to respond with unintended or harmful contents using adversarial suffix [29], [30] or jailbreak prompting [31], [32]. Similarly, adversarial perturbations added to input images can mislead VLMs into generating specified outputs [33], [34]. Building on these vulnerabilities, recent studies have identified robustness issues in the interaction and planning modules of LLM-powered agents [25], [26]. Such attacks typically exploit inherent flaws in LLMs by crafting malicious prompts that induce harmful agent behaviors. For instance, [25] proposed a GCG-like [29] optimization strategy to generate adversarial prompt suffix that misleads the planning module, evaluating the robustness of LLM-powered agents. [26] similarly optimized adversarial suffixes appended to harmful instructions, which jailbreak LLM-powered agents into producing executable harmful policies.

Adversarial attacks targeting VLM-powered agents primarily focus on misleading the perception module. As illustrated in Fig. 1, attackers can generate a 3D adversarial object in the environment, which misleads object perception (e.g., causing a "chair" to be misclassified as a "fire extinguisher"). Consequently, human instructions are incorrectly executed, leading to erratic agent behaviors. Thus, evaluating the robustness of VLM-powered VLN agents is critical. However, existing 2D adversarial attacks towards VLMs and 3D adversarial attacks on VLN agents face the following challenges for such evaluations. ❶ *2D attacks suffer from attack misalignment*: Traditional 2D adversarial attacks typically introduce pixel-wise perturbations or patch-based modifications to images. However, VLN agents operate in a 3D environment and perceive their surroundings through dynamic 2D images. This discrepancy leads to a fundamental misalignment: adversarial perturbations and patches designed for 2D perception may not align well with real-world 3D spatial constraints, limiting their effectiveness in practical scenarios. Therefore, adversarial objects must be constructed to align with physical properties while retaining attack efficacy. ❷ *2D attacks exhibit multi-view inefficiency*: 2D attacks that inject 2D-space adversarial perturbations into images, render them effective only under specific, fixed views. Since VLN agents perceive objects in a 3D environment from diverse and unseen views, these adversarial perturbations effective in a static 2D image may become ineffective when the agent observes the same scene from a different angle or under varying lighting conditions. Consequently, it is crucial to generate the adversarial object can consistently attack VLN agents, adapting to multiple views across varying navigation scenarios. ❸ *3D attacks face cross-modal inefficiency*: Traditional 3D attacks [35], [36] against VLN agents primarily target vision models like VGG [37], R-CNN [38], and ResNet [39]. These attacks, tailored for traditional visual models, face challenges when applied to VLM-powered agents due to the heterogeneity across different modalities [33], [34].

To address the aforementioned challenges, we propose a novel attack, Adversarial Object Fusion (**AdvOF**), which generates adversarial objects capable of deceiving VLM-powered VLN agents. **AdvOF** consists of three components, **1)** *Aligned Object Rendering:* To solve the misalignment between 3D adversarial manipulation and 2D scene perception, we firstly aggregate object locations in 2D space using object detection and segmentation models. We then align these locations in 3D space to recognize isolated objects, enabling object-specific rendering from 3D space to 2D images; **2)** *Adversarial Collaborative Optimization:* To realize the cross-modal attack consistency, we design a collaborative optimization that captures universal adversarial features by constraining the similarity between visual and textual embeddings. Additionally, we combine an object-aware regularization term to preserve the physical plausibility of adversarial objects; **3)** *Adversarial Object Fusion:* To achieve reliable attack across varying and unseen views, we optimize the adversarial object across multiple views based on their importance weights and incorporate an iterative updating procedure to ensure stability in the optimization. This fusion guarantees the attack effectiveness across different VLN tasks and enhances the attack transferability across diverse VLN environments.

Compared to existing 2D attacks (MF [34] and AdvCLIP [33]) and 3D attacks (ST [35] and TT3D [36]), **AdvOF** can achieve SOTA attack performance across four VLN agents (Vlmaps [16], Cows [17], CF [19], ORION [18]). Our main contributions are summarized as follows:

- We formulate a new problem that generates adversarial objects towards VLN agents.
- We develop a novel attack, **AdvOF**, successfully generating the adversarial object that can multi-viewedly fool agents to percept it with the adversarial label. This adversarial object in the environment obstructs the execution of the user instruction.
- We conduct empirical validation of **AdvOF**'s performance across multiple VLN agents and datasets, demonstrating its superiority in attacking effectiveness and multi-view robustness.

2 Related Work

2.1 VLN Agents With Foundation Models

Existing foundation models have demonstrated exceptional capabilities across several dimensions, including in-context learning [8], reasoning [40], and multi-modal processing [11]. VLN agents leveraging these models have similarly advanced, adopting novel implementation paradigms and achieving substantial performance gains.

Large Language Models for VLN Agents. LLMs have shown promising capabilities in navigation, allowing VLN agents to follow interactive instructions and execute complex planning tasks. LEO [41] leveraged extensive knowledge from LLMs to excel in 3D perception, reasoning, and action tasks, training on large-scale 3D datasets and demonstrating remarkable proficiency across diverse real-world applications. LLM-Planner [42] introduced a hierarchical framework, where high-level plans composed of sub-goals are generated and subsequently refined into detailed actions by a low-level planner, enabling more adaptable and goal-oriented navigation strategies. NaviLLM [43] transformed

embodied navigation tasks into structured generation problems via schema-based instructions, enabling generalization across diverse navigation scenarios with enhanced flexibility and consistency.

Vision Language Models for VLN agents. With advancements in multi-modal representation learning, VLMs have shown remarkable performance in scene perception and map construction for VLN agents, achieving impressive zero-shot capability across diverse navigation tasks. Cows [17] investigated language-driven zero-shot object navigation, adapting open-vocabulary models to enable robots to locate objects specified through language without task-specific training. CF [19] introduced an open-set, multi-modal 3D scene representation that integrates pixel-aligned features from pre-trained foundation models into 3D maps via SLAM, enabling zero-shot spatial reasoning across diverse queries. ZSON [44] utilized the vision encoder of CLIP [11] to encode target images, trained via a reinforcement learning framework. Vlmaps [16] developed a spatial map representation that combines VLM features with 3D reconstructions, allowing robots to autonomously build maps from video feeds and support complex natural language navigation goals. ONION [18] leveraged multiple foundation models, enabling robots to navigate to personalized objects in unknown environments through user interaction.

2.2 Adversarial Attack

2D Adversarial Attacks. Adversarial attacks on foundation models have been extensively studied, revealing strategies to exploit model vulnerabilities. Textual adversarial attacks can mislead models into generating incorrect outputs or classifications [45], [46]. Common techniques include token manipulation [47], gradient-based optimization [29], and jailbreak prompting [31]. Visual adversarial attacks typically involve perturbing images to induce harmful content generation, often enhanced via proxy models [48] or model ensembles [49]. For example, AdvClip [33] designed a universal adversarial patch targeting pre-trained VLMs, capable of compromising diverse downstream models. Similarly, MF [34] introduced targeted adversarial examples against VLMs by leveraging transfer attacks with black-box queries.

In particular, some adversarial attacks have focused on LLM-powered agents. EIRAD [25] developed an embodied attack dataset for robustness evaluation, generated via the GCG algorithm [29] with novel prompt suffix initialization. Similarly, NPS [28] constructed an adversarial suffix attack targeting outdoor navigation agents, misleading them into navigating in incorrect directions. POEX [26] designed a policy-executable red-teaming framework capable of injecting universal and transferable adversarial suffixes into planning modules, inducing embodied AI systems to execute harmful policies.

3D Adversarial Attacks. 3D adversarial attacks introduce physical perturbations to disrupt the inference of DNN-based models [50]. Current 3D attacks are mainly implemented through gradient-based optimization [51], [52], model-based generation [53], [54], and mesh-based transformations [55], [56]. For example, [51] proposed robust adversarial objects against point cloud models in the physical world. [54] designed an arbitrary-target attack framework using a label-guided adversarial network based on graph patch GAN architecture [57]. [56] generated universal adversarial camouflage through neural texture rendering, incorporating stealthiness and naturalness constraints. ST [35] introduced 3D spatiotemporal perturbations that exploit temporal interaction history and spatial object properties, enhancing attack efficacy through a trajectory attention module. TT3D [36] produced transferable targeted 3D adversarial examples by reconstructing textured meshes and leveraging dual NeRF-space optimization to improve black-box transferability and visual naturalness.

However, these 2D and 3D adversarial attacks encounter limitations in generating adversarial objects for VLN agents. 2D adversarial attacks cannot perturb 3D space and exhibit limited effectiveness when executing attacks from multiple or unseen views. 3D adversarial attacks primarily target point cloud networks (e.g., PointNet [58] and PointNet++ [59]) and traditional detection models (e.g., Fast R-CNN [60] and YOLO [61]), which struggle to adapt to VLMs with cross-modal features and complex architectures. Motivated by these problems, this paper focuses on designing a novel attack framework, capable of generating effective and multi-view adversarial objects targeting VLN agents.

3 Problem Statement

This section describes the system model of VLN agents and outlines the threat model associated with an adversarial environment that contains adversarial objects. We then formalize the definition of adversarial objects targeting VLN agents.

3.1 System Model

We consider a VLN agent operating in a continuous environment [62], equipped with LLMs and VLMs. The agent executes primitive actions (e.g., move forward, turn left) to navigate toward a specified goal with a physical space, guided by natural language instructions. A VLN agent primarily comprises modules for scene perception, instruction interaction, object localization and goal action. The system architecture is illustrated in Fig. 2.

Instruction Interaction. The VLN agent employs an interaction module based on LLMs to interpret user instructions, extracting object landmarks and associated actions. The user instruction is denoted as $L = \langle \boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_L \rangle$, where $\boldsymbol{t}_i$ represents a single word token. Through querying the LLM, the agent identifies object landmarks $\langle o_1, o_2, \ldots, o_m \rangle$ from L. For each landmark o_i, the LLM generates a corresponding action α_i, guiding the agent to the target landmark.

Scene Perception. The VLN agent constructs a point cloud using RGB and depth images to model its environment. Using these RGB-D images and the point cloud, the agent incrementally builds a semantic map that incorporates visual features extracted via VLMs. This map is represented as $\mathcal{M} \in \mathbb{R}^{H \times W \times C}$, where H and W denote the height and width of the map, and C corresponds to the feature dimensionality of the VLM.

Object Localization. After extracting object landmarks from the interaction module, the agent locates their corresponding positions in the scene. It maintains a grounding label

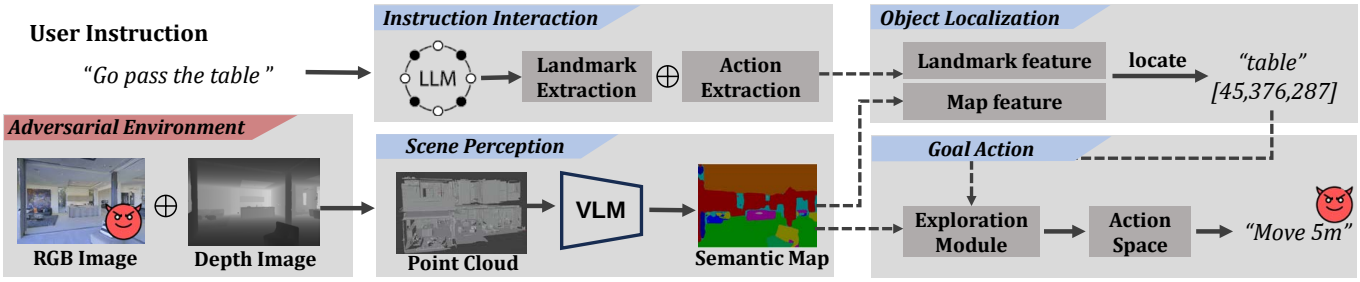


Fig. 2. Vision-and-Language navigation agents with adversarial environment.

list and computes the grounding label feature using the VLM's text encoder. The agent calculates similarity scores between each grounding label and the current semantic map $\mathcal{M}$, determining the target object location by selecting the highest-score map grid.

Goal Action. The agent executes goal-directed actions based on the identified object location and associated action within the predefined action space $\mathcal{A}$, including *rotate*, *move*, and *stop*. If the target landmark is not detected within the current camera view, the agent initiates scene exploration [63]. Using the provided distance and angle information, the agent executes each action and, after each step, calculates its proximity to the object to determine whether it has reached the target location.

3.2 Threat Model

Attack Scenario. While recent works have applied foundation models to enhance VLN agent performance under normal conditions [17], [18], studies reveal that these models remain highly sensitive to even minor input perturbations [64]. As illustrated in Fig. 2, we examine an adversarial scenario targeting the scene perception module to disrupt instruction execution. Specifically, an adversary could introduce specially crafted objects to create an adversarial environment. When deployed in such an environment, the VLN agent misclassifies these adversarial objects into attacker-defined labels, leading to erroneous scene perception. Consequently, these errors propagate to the object localization and goal-oriented action modules, ultimately resulting in failed task execution.

Attack Goals. The attack targets scene perception modules powered by VLMs by introducing adversarial objects. The adversarial objects are designed to mislead the VLM into mislabeling them, distorting the semantic map and preventing accurate localization of target objects. As a result, the agent fails to reach the target object, thereby disrupting task completion. The following objectives outline a successful adversarial attack towards VLM-powered VLN agents:

- *Object-Specific Control*: The attack must target specific objects without affecting others in the environment. The adversary can select any object as a target, ensuring that the impact is contained within the targeted object area.
- *3D-2D Consistency*: The adversarial object, positioned in 3D space, should maintain its deceptive effect from different 2D views captured by the agent's camera. This ensures the attack consistently impacts both the agent's 3D understanding and 2D perception, leading to misinterpretation of the adversary object's identity.
- *Multi-View Robustness*: The adversarial object should deceive the agent from multiple views, ensuring adaptability to various user tasks and instructions.

Adversary's Capabilities. The adversary's capabilities can be categorized as: **1)** *Environment Manipulation*: The adversary has access to the system environment and can collect environmental data to support the attack. They can modify target objects (e.g., Painting [65], 3D printing [66]) to create adversarial counterparts, aligning with standard environment modeling and scene perception processes. **2)** *Agent Access*: Attacks can be carried out in either a white-box or black-box setting. In the white-box scenario, the adversary has access to the VLM of the VLN agent. In the black-box scenario, the adversary lacks knowledge of the agent's VLM and uses proxy agents to craft the attack, transferring it to the target agent.

3.3 Problem Formulation

Definition 1 (Adversarial Object Against VLN Agents). Given a victim object O with the label T^v in 3D space, the goal is to generate a corresponding adversarial object O^{adv} such that the VLN agent perceives O^{adv} with an adversarial label T^t when observed from multiple views $\mathcal{V}$. The process of generating adversarial objects is formulated as an optimization problem:

$$\arg\min_{O^{adv}} \mathcal{L}_{3D}(O^{adv}, O) + \mathbb{E}_{v\in\mathcal{V}}\mathcal{L}_{2D}(O^{adv}_v, T^t, \theta), \quad (1)$$

where θ presents the parameters of the VLM employed by the VLN agent; $\mathcal{L}_{3D}$ denotes a loss function that enforces similarity in physical property between O and O^{adv} in 3D space, ensuring that O^{adv} resembles O in appearance; $\mathcal{L}_{2D}$ denotes a loss function that encourages the VLM to perceive O^{adv}_v as the adversarial label T^t in each 2D view v.

This optimization minimizes physical differences between O and O^{adv} while maximizing the misclassifications of O^{adv} to T^t across multiple views $\mathcal{V}$.

4 Methodology

4.1 Motivation

Designing an effective adversarial object against VLN agents within 3D environments presents unique challenges

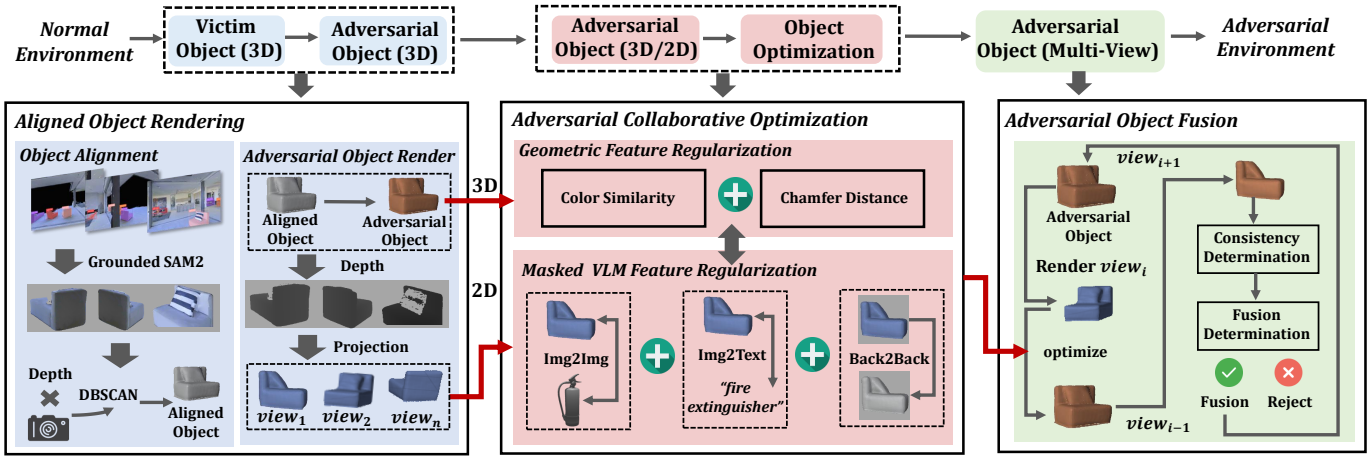


Fig. 3. Pipeline for generating adversarial environment towards VLN agents: It optimizes the adversarial object, transforming a normal environment into an adversarial one. *Aligned object rendering* aligns the victim object in both 2D and 3D locations, rendering 2d images across different views. *Collaborative optimization* jointly matches features between the adversarial and victim objects across 2D and 3D spaces. *Adversarial object fusion* assigns importance weights to different views, contributing to generate adaptable adversarial object.

due to the integration with foundation models. We identify three main challenges:

1) Aligning Adversarial Objects Across 3D and 2D Spaces. VLN agents perceive their environment through 2D images while navigating in 3D space, resulting in a spatial misalignment between 3D adversarial manipulations and the agent's 2D perception. Traditional 2D adversarial attacks on VLMs, often using pixel-level noise or patch-based perturbations [33], do not address this misalignment.

2) Ensuring Attack Robustness Across Views and Tasks. VLN agents frequently operate under varying user instructions, requiring adversarial objects to maintain effectiveness across different views and tasks. This adaptability is challenging due to the dynamic nature of foundation models, which process varying visual perspectives and instruction sets.

3) Maintaining Attack Effectiveness Among Different Modalities. Compared to typical 3D adversarial attacks tailored for visual models, attacks targeting VLMs need to maintain the attack effectiveness between visual and textual modalities.

Pipeline. To solve these challenges, we propose a novel adversarial attack framework **AdvOF** (**Adv**ersarial **O**bject generation based on view **F**usion), targeting VLN agents powered by VLMs. Our method consists of three core components: Aligned Object Rendering, Adversarial Collaborative Optimization, and Adversarial Object Fusion. We provide the attacking pipeline in Fig. 3.

4.2 Aligned Object Rendering

Building on **Definition 1**, generating a successful adversarial object requires first identifying the victim object O and subsequently constructing the initial adversarial object O^{adv} within the environment.

Victim Object Alignment. To detect the victim object, we leverage a VLM model for rapid open-set object detection, querying whether the victim object O with the label T^v is present in a given scene S_i:

$$\mathcal{S}_v \leftarrow \{S_i | \mathcal{I}(VLM(S_i, T^v) = 1, S_i \in \mathcal{S}\}, \tag{2}$$

where $\mathcal{S}$ denotes the set of all environment scenes. We then apply Grounding DINO [67] for precise object detection and SAM [23] for object segmentation:

$$(MASK, SCORE) \leftarrow \{SAM(GD(S_j, T^v)), S_j \in \mathcal{S}_v\}, \tag{3}$$

where $MASK$ captures the spatial locations of victim objects, and $SCORE$ represents the segmentation confidence. To consolidate fragmented detections into discrete objects, we back-project $MASK$ into 3D space and cluster locations using DBSCAN [68].

$$\begin{aligned} MASK_{3D} &\xleftarrow{back-project} MASK, \\ O &\xleftarrow{select} DBSCAN(MASK_{3D}). \end{aligned} \tag{4}$$

A randomly selected cluster from the output is designated as the victim object O in 3D space for the adversarial attack.

Adversarial Object Render. Since VLN agents with VLMs process only 2D observations, we create the adversarial object O^{adv} by defining a 3D perturbation δ^{adv} and rendering adversarial 2D images based on this perturbation.

Definition 2 (3D Perturbation for Adversarial Object Generation). Given a victim object O, a 3D perturbation δ^{adv} is point cloud-wise noise with same shape as O. The adversarial object is generated by adding δ^{adv} to O: $O^{adv} \leftarrow O + \delta^{adv}$. The perturbation is constrained by an upper bound, $||\delta^{adv}||_p \leq \epsilon$.

We then render the adversarial object into 2D space. Assuming the environment camera has an intrinsic matrix k_{int} and an extrinsic matrix k_{ext}, the rendering function $\mathcal{R} : O_v \leftarrow \mathcal{R}(O, v)$ projects the object on to a specific view v:

$$\begin{aligned} PC &= k_{ext} \cdot O, \\ O_v &= z^{-1} \cdot PC \cdot K_{int}, \end{aligned} \tag{5}$$

where z is the depth value for view v, and O_v is the rendered 2D projection. The optimization problem based on

the rendering function is formulated as:

$$\arg\min_{\delta^{adv}} \mathcal{L}_{3D}(O^{adv}, O) + \mathbb{E}_{v\in\mathcal{V}}\mathcal{L}_{2D}(O_v + \delta_v^{adv}, T^t, \theta), \quad s.t.\ \delta_v^{adv} \leftarrow \mathcal{R}(\delta^{adv}, v). \tag{6}$$

The 2D adversarial perturbation δ_v^{adv} is derived via the same rendering process $\mathcal{R}$. The resulting adversarial image in 2D space is defined as $O_v + \delta_v^{adv}$.

4.3 Adversarial Collaborative Optimization

This section explains the design of $\mathcal{L}_{3D}$ and $\mathcal{L}_{2D}$ in the above optimization problem, based on the adversarial collaborative optimization. This approach optimizes regularization across 3D and 2D spaces, capturing joint visual and textual modalities within the VLM.

Geometric Feature Regularization. The purpose of $\mathcal{L}_{3D}$ is to maintain the physical property similarity between the adversarial object O^{adv} and the victim object O. First, we constrain the RGB values to maintain color similarity:

$$\mathcal{L}_{color}(O^{adv}, O) = ||O^{adv} - O||^2. \tag{7}$$

For geometric shape similarity, we utilize Chamfer distance to align the 3D structure:

$$L_{CD}(O^{adv}, O) = \sum_{x\in O^{adv}} \min_{y\in O} ||x-y||^2 + \sum_{y\in O} \min_{x\in O^{adv}} ||y-x||^2. \tag{8}$$

Consequently, these components define the 3D loss:

$$\mathcal{L}_{3D}(O^{adv}, O) = \mathcal{L}_{color}(O^{adv}, O) + \mathcal{L}_{CD}(O^{adv}, O) \tag{9}$$

Masked VLM Feature Regularization. The objective of $\mathcal{L}_{2D}$ is to mislead the VLM into misclassifying the adversarial object. Untargeted attacks involve preventing the VLM from identifying the adversarial object as the original victim object. A naive optimization approach minimizes the similarity between visual features of rendered adversarial and victim object images:

$$\mathcal{L}_\theta^{I2I}(O_v, \delta_v^{adv}) = \mathcal{F}(\theta(O_v + \delta_v^{adv}), \theta(O_v)), \tag{10}$$

where $\mathcal{F}$ is the cosine similarity function and θ represents VLM parameters. However, whole-image optimization is overly sparse, failing to focus on adversarial object regions. Instead, we propose optimizing within the adversarial object mask. To extract region-specific features, we introduce a masked feature operator $\mathcal{M}$:

$$\mathcal{M}^\theta(O_v, mask) = Flat(\theta(O_v) \odot mask), \tag{11}$$

where $Flat$ flattens features and $\odot$ denotes element-wise multiplication. This reformulate $\mathcal{L}^{I2I}$:

$$\mathcal{L}_\theta^{I2I}(O_v, \delta_v^{adv}, mask) = \mathcal{F}(\mathcal{M}^\theta(O_v + \delta_v^{adv}, mask), \mathcal{M}^\theta(O_v, mask)). \tag{12}$$

To enhance cross-modal attack efficacy, we regularize the alignment between adversarial visual features and victim object text embeddings T^v:

$$\mathcal{L}_\theta^{I2T}(O_v, \delta_v^{adv}, mask, T^v) = \mathcal{F}(\mathcal{M}^\theta(O_v + \delta_v^{adv}, mask), \theta(T^v)) \tag{13}$$

Empirically, constraining only the masked region disturbs background predictions (Fig. 4(d)). To preserve background consistency, we add a regularization term:

$$\mathcal{L}_\theta^{B2B}(O_v, \delta_v^{adv}, mask) = \mathcal{F}(\mathcal{M}^\theta(O_v + \delta_v^{adv}, \overline{mask}), \mathcal{M}^\theta(O_v, \overline{mask})) \tag{14}$$

Thus, the 2D loss function of the untargeted attack in view v is then represented as:

$$\mathcal{L}_{2D}^{UT} = \mathcal{L}_\theta^{I2I}(O_v, \delta_v^{adv}, mask) + \alpha \cdot \mathcal{L}_\theta^{I2T}(O_v, \delta_v^{adv}, mask, T^v) - \beta \cdot \mathcal{L}_\theta^{B2B}(O_v, \delta_v^{adv}, mask) \tag{15}$$

where α and β are balancing coefficients.

For targeted attacks, the goal is to misclasify O as adversarial label T^t. To achieve this, we retain a target image I^t, with $mask^t$ for attack target T^t. Thus, the loss function of $\mathcal{L}_\theta^{I2I}$ for target attacks is revised as:

$$\mathcal{L}_\theta^{I2I}(O_v, \delta_v^{adv}, mask, I^t, mask^t) = \mathcal{F}(\mathcal{M}^\theta(O_v + \delta_v^{adv}, mask), \mathcal{M}^\theta(I^t, mask^t)). \tag{16}$$

Similarly, $\mathcal{L}_\theta^{I2I}$ aligns adversarial features with T^t:

$$\mathcal{L}_\theta^{I2T}(O_v, \delta_v^{adv}, mask, T^t) = \mathcal{F}(\mathcal{M}^\theta(O_v + \delta_v^{adv}, mask), \theta(T^t)) \tag{17}$$

Consequently, the 2D loss function of the targeted attack in view v is represented as :

$$\mathcal{L}_{2D}^{T} = -\mathcal{L}_\theta^{I2I}(O_v, \delta_v^{adv}, mask, I^t, mask^t) - \alpha \cdot \mathcal{L}_\theta^{I2T}(O_v, \delta_v^{adv}, mask, T^t) - \beta \cdot \mathcal{L}_\theta^{B2B}(O_v, \delta_v^{adv}, mask) \tag{18}$$

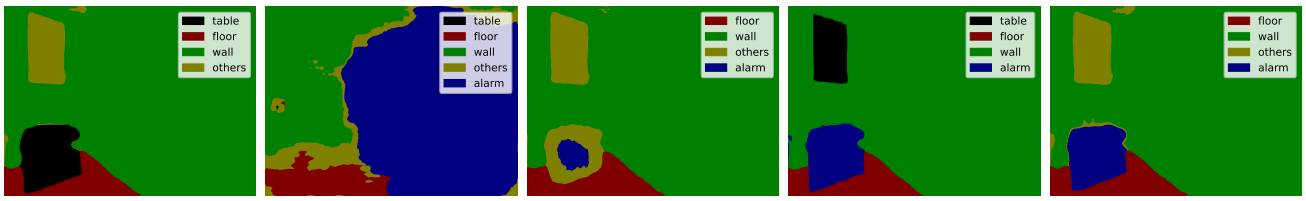
Exemplary Illustration. Fig. 4 demonstrates the optimization outcomes of targeted attacks using different regularization components. Notably, our proposed formulation $\mathcal{L}_{2D}^T$ effectively shifts adversarial object predictions from the victim label to the target label while preserving perceptual consistency in background regions.

4.4 Adversarial Object Fusion

In practical applications, adversarial objects must maintain effectiveness across diverse views. **Definition 1** formalizes this requirement as a uniform update across all views $\mathcal{V}$. However, views have varying levels of importance for the adversarial object. For example, incomplete or distant views are given lower weights, while complete or close-up views are assigned higher importance. Additionally, adjacent views frequently overlap when observing the same adversarial object. Therefore, a view-aware collaborative optimization strategy becomes critical.

We utilize $SCORE$ of the grounding model to represent the importance weight for each scene view. Besides, we also compute the pixel count N of the adversarial object to indicate the importance weight w_v:

$$w_v = \frac{w}{\sum_{v\in V} w}, \text{where } w = score_v + N_v/N \tag{19}$$



(a) No Attack (b) $\mathcal{L}_{I2I}$ (c) $\mathcal{L}_{2D}^{T}$ w/o $\mathcal{L}^{I2T}$ (d) $\mathcal{L}_{2D}^{T}$ w/o $\mathcal{L}^{B2B}$ (e) $\mathcal{L}_{2D}^{T}$

Fig. 4. Visualization of optimization results under different regularization terms for the target attack (table → alarm).

Algorithm 1: Adversarial Object Fusion.

Input : $T, \mathcal{V}, \mathcal{R}, Max$
Output: C^{adv}

1 (MASK, SCORE) ← Object segmentation via Eq. 3
2 O ← Object alignment via Eq. 4
3 δ^{adv} ← Initialize the 3D adversarial perturbation
4 **for** $v \in \mathcal{V}$ **do**
5 $\quad w_v \xleftarrow{v} \mathcal{W}, mask_v \xleftarrow{v} MASK$
6 $\quad$ **for** *k in Max* **do**
7 $\quad\quad$ `// Local Update`
8 $\quad\quad$ **Render**: $\delta_v^{adv} \leftarrow \mathcal{R}(\delta^{adv}, v)$
9 $\quad\quad$ **Optimize**: collaborative optimization via Eq. 18
10 $\quad\quad$ **Update**: $\delta^{adv} \xleftarrow{w_v} \delta_v^{adv}$
11 $\quad\quad$ `// Consistency Determination`
12 $\quad\quad$ Judge the update for $v-1$ with w_v
13 $\quad\quad$ `// Fusion Determination`
14 $\quad\quad$ Judge the perception for $v-1$
15 $O^{adv} \leftarrow O + \delta^{adv}$
16 return C^{adv}

TABLE 1
The setups of different VLN agents.

Agent	Publication	Simulator	Dataset	VLM
Vlmaps [16]	ICRA'23	Habitat	MP3D [69]	Lseg [70]
Cow [17]	CVPR'23	Habitat	MP3D [69]	Clip [11]
CF [19]	RSS'23	Habitat	HM3D [71]	Clip [11]
ORION [18]	ICRA'24	Habitat	HM3D [71]	Lseg [70]

This weighted approach modifies the optimization problem as follows:

$$\arg\min_{\delta^{adv}} \mathcal{L}_{3D}(O^{adv}, O) + \mathbb{E}_{v\in\mathcal{V}}\, w_v \cdot \mathcal{L}_{2D}(O_v + \delta_v^{adv}, T^t, \theta). \tag{20}$$

To ensure stable multi-view updates, we propose an adversarial object fusion process that iteratively fuses the adversarial object across views, where a global perturbation δ^{adv} is refined through iterative updates from local perturbations for individual views.

The detailed algorithm of adversarial object fusion is presented in Algorithm 1. In each iteration, the algorithm renders a 2D perturbation δ_v^{adv} from a 3D perturbation δ^{adv} (line 8). The 2D perturbation δ_v^{adv} is then optimized through the collaborative optimization method described in Sec 4.3 (Line 9). Next, consistency is evaluated with the prior view

(a) MP3D, ID=5LpN3gDmAk7 (b) HM3D, ID=QaLdnwvtxbs

Fig. 5. The scene displayed on the habitat simulator.

TABLE 2
The instruction samples of different VLN agents.

Agents	Instruction
Vlmaps	Go to the closest cushion first, then go to a chair nearby, after that, go to a counter and in the end, navigate to a table.
Cow	Find a house plant near a coffee table.
CF	A comfy place to sit and watch tv.
ORION	Can you find a blue tower for me? It's hanging on the bar in the bathroom.

$v-1$ (Line 12). If the distance between the new perturbations δ_{v-1}^{adv} and prior local perturbations $\delta_{v-1}^{adv'}$ is larger than a consistency threshold μ_1:

$$MSE(\delta_{v-1}^{adv'}, \delta_{v-1}^{adv})) \geq \mu_1, \tag{21}$$

the local update is rejected, and the importance weight is reduced before reattempting the collaborative optimization. Following this, a fusion determination step assesses whether the local perception aligns with the previous view's fusion (Line 14). If the discrepancy of these perceptions exceeds a fusion threshold μ_2:

$$|\mathcal{L}_{2d}(\delta_{v-1}^{adv'}) - \mathcal{L}_{2d}(\delta_{v-1}^{adv})| \geq \mu_2, \tag{22}$$

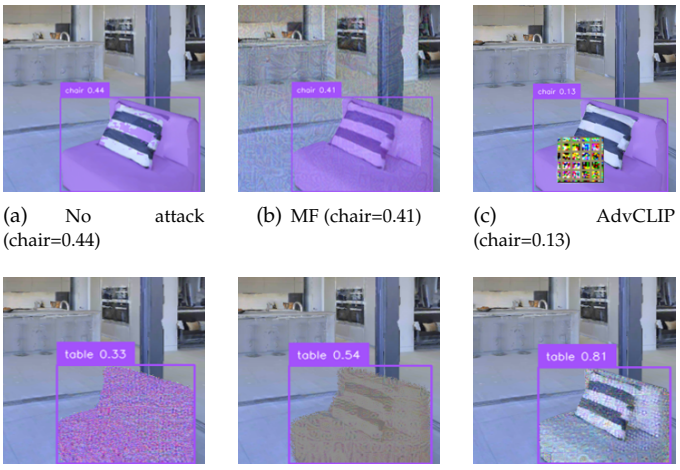
this local update is discarded, and the perturbation bound is reduced before re-rendering. If the view fails to meet these conditions within Max iterations, the fusion is rejected for that view. By iterating through all views, the algorithm robustly fuses local perturbations into a cohesive adversarial object across views.

5 Experiments

In this section, we first present the primary attack results against various VLN agents compared to 2D and 3D baseline attacks (Section 5.2). Next, we demonstrate the transferability of **AdvOF** across diverse image encoders, scene datasets, and model architectures (Section 5.3). We then

TABLE 3
The overall attacking performance of different methods on various VLN agents.The upper and bottom parts respectively show the performance with and without attack objects.

Environment	Method	Vlmaps			Cow			CF			ORION		
		KPA ↓	SPL ↑	SR ↓	KPA ↓	SPL ↑	SR ↓	KPA ↓	SPL ↑	SR ↓	KPA ↓	SPL ↑	SR ↓
Attacked	Base	0.560	49.0	0.297	0.208	12.6	0.158	0.275	17.2	0.198	0.215	9.8	0.147
	MF [34]	0.504	53.9	0.267	0.187	13.9	0.142	0.252	18.9	0.178	0.190	10.8	0.135
	AdvCLIP [33]	0.448	58.5	0.240	0.165	14.9	0.127	0.225	20.3	0.162	0.171	11.7	0.115
	ST [35]	0.392	64.0	0.208	0.148	16.2	0.109	0.193	22.0	0.138	0.149	12.7	0.105
	TT3D [36]	0.331	67.8	0.184	0.125	17.7	0.096	0.163	24.2	0.117	0.125	13.8	0.087
	AdvOF(Untargeted)	0.283	74.5	0.145	0.106	18.9	0.078	0.142	25.8	0.102	0.104	14.9	0.072
	AdvOF(Targeted)	**0.242**	**77.5**	**0.122**	**0.084**	**19.9**	**0.067**	**0.116**	**27.0**	**0.087**	**0.089**	**15.6**	**0.064**
		KPA ↑	SPL ↓	SR ↑	KPA ↑	SPL ↓	SR ↑	KPA ↑	SPL ↓	SR ↑	KPA ↑	SPL ↓	SR ↑
Normal	Base	0.480	58.0	0.218	0.192	17.3	0.114	0.251	21.2	0.157	0.204	11.7	0.121
	MF [34]	0.404	67.1	0.187	0.161	20.0	0.099	0.217	24.1	0.131	0.176	13.4	0.104
	AdvCLIP [33]	0.475	59.2	0.214	0.188	17.6	0.110	0.241	21.5	0.153	0.199	11.7	0.119
	ST [35]	0.442	61.9	0.209	0.177	18.4	0.107	0.235	22.8	0.145	0.195	12.3	0.113
	TT3D [36]	0.437	61.7	0.205	0.174	18.4	0.103	0.232	22.7	0.144	0.187	12.7	0.112
	AdvOF(UnTargeted)	0.472	58.2	0.213	0.189	17.4	0.112	0.248	21.8	0.152	0.197	11.9	0.120
	AdvOF(Targeted)	**0.480**	**58.0**	**0.218**	**0.190**	**17.1**	**0.115**	**0.251**	**21.2**	**0.156**	**0.202**	**11.6**	**0.122**



(a) No attack (chair=0.44) (b) MF (chair=0.41) (c) AdvCLIP (chair=0.13)

(d) ST (table=0.33) (e) TT3D (table=0.54) (f) AdvOF (table=0.81)

Fig. 6. The adversarial examples visualization of different attacks (chair → table).

assess the robustness of the attack under potential defensive mechanisms (Section 5.4). Finally, we perform parameter analysis to evaluate the impact of critical modules and hyperparameters (Section 5.5).

5.1 Environmental Setup

VLN Agents. We adopted four typical navigation agents using foundation models. 1) Vlmaps [16]: integrates language grounding with visual observations through a spatial map that fuses pre-trained visual-language features [70] with a 3D reconstruction of the physical world; 2) Cow [17]: introduces the PASTURE benchmark for language-driven zero-shot object navigation, adapting zero-shot models to a VLN task; 3) CF [19]: constructs an implicit scene model based on clip [11] to strengthen instance identification and semantic segmentation; 4) ORION [18]: proposes Zero-shot Interactive Personalized Object Navigation, requiring robots to navigate to personalized goal objects while engaging in conversations with users.

TABLE 4
Attacking results towards perception modules across different VLN agents.

	Vlmaps		Cow		CF		ORION	
	Acc	Asr	Acc	Asr	Acc	Asr	Acc	Asr
Base	0.89	0.00	0.83	0.00	0.81	0.00	0.91	0.00
MF	0.71	0.15	0.74	0.06	0.69	0.12	0.83	0.08
AdvCLIP	0.56	0.34	0.60	0.32	0.61	0.35	0.68	0.26
ST	0.33	0.56	0.27	0.61	0.30	0.64	0.29	0.65
TT3D	0.29	0.70	0.21	0.76	0.20	0.78	0.26	0.71
AdvOF	**0.04**	**0.92**	**0.05**	**0.91**	**0.02**	**0.93**	**0.05**	**0.94**

Simulation Datasets and Environments. For each navigation agent, we followed the original simulator and dataset settings as provided in the official repository. Table 1 details the simulation environments, with dataset visualizations presented in Fig. 5.

The simulator adopted in these navigation agents is Habitat [72], enabling highly efficient photorealistic 3D simulation. The datasets used are Matterport3D (MP3D) [69] and Habitat-Matterport 3D (HP3D) [71]. MP3D is an RGB-D dataset with 90 building-scale scenes, and HM3D is a large-scale dataset of 1,000 building-scale scenes.

For each navigation agent and dataset, we randomly selected 20 scenes to construct the validation dataset. In each scene, we randomly selected 10 objects as victim objects. For each object, we randomly selected 10 different instructions. The instructions collected in different navigation agents are displayed in Table 2.

Baselines and Metrics. We adopted two SOTA adversarial attacks to VLMs and two 3D adversarial attacks to traditional object detection models as the baselines. 1) MF [34]:

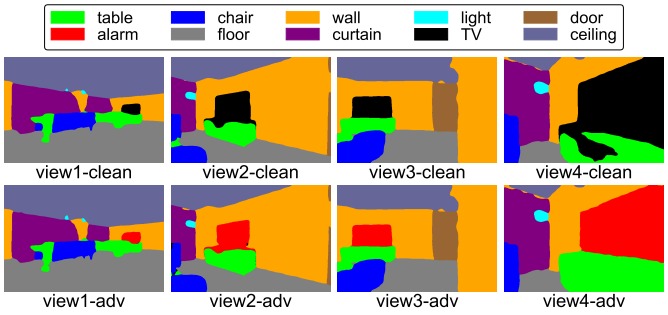


Fig. 7. The adversarial object of different views. TV(black) → alarm(red).

evaluates the robustness of open-source large VLMs under black-box conditions, where adversaries aim to mislead the model into returning the targeted responses; 2) AdvCLIP [33]: generates downstream-agnostic adversarial examples based on cross-modal pre-trained encoders. 3) ST [35]: studies adversarial attacks on embodied agents using spatiotemporal perturbations in dynamic environments. 4) TT3D [36]: creates transferable targeted 3D adversarial examples using NeRF-based optimization for improved black-box transferability.

We evaluate task and attack performance using standard object navigation metrics and adversarial attack measures: 1) *Success Rate(SR)*: the fraction of episodes where the agent executes STOP action within 1.0m of the target object; 2) *Success weighted by inverse path length (SPL)*: Success weighted by the oracle shortest path length and normalized by the actual path length. This metric points to the success efficiency of the agent; 3) *Key point accuracy (KPA)*: The KPA metric measures the percentage of correct decisions made at each subgoal; 4) *Acc*: The prediction accuracy of the scene perception module; 5) *Asr*: The attack success rate in misleading the perception module's predictions.

Attack Implementations. In the implementation of the attack, the VLM model utilized for rapid open-set object detection is LLaVA [12]. The box threshold and the text threshold for the grounding model are set to 0.40. We set the balance coefficients $\alpha = 0.5$ and $\beta = 0.01$, The upper bound of adversarial perturbation is set to $\epsilon = 32/255$. The optimizer used is Adam, with the optimization iterations set to 200. The parameters μ_1 and μ_2 in the adversarial object fusion are set to 0.01 and 0.05, respectively. After attacking the victim object, we replace the RGB-D data associated with the victim object using the perturbed data and regenerate the semantic map. The attack performance is then evaluated in the new environment scene using the collected validation dataset. For the two attacks targeting VLMs, MF and AdvCLIP, we select RGB images containing the victim object to be attacked. For the two 3D attacks, ST and TT3D, we first implement the proposed process of aligned object rendering and then proceed with the attack as outlined in the original paper.

5.2 Overall Evaluation

Effectiveness. To comprehensively evaluate the impact of adversarial objects, we construct a new validation dataset without adversarial objects. We present the overall attack

TABLE 5
Results of transfer attack across different image encoders.

	KPA ↓	SPL ↑	SR ↓
Lseg, ViT-L/16 → Lseg, ResNet101	0.242	77.5	0.122
	0.287(18.6%)	73.2(5.6%)	0.147(20.5%)
Lseg, ResNet101 → Lseg, ViT-L/16	0.253	76.8	0.134
	0.302(19.4%)	70.1(8.7%)	0.163(21.6%)
Clip, ViT-B/32 → Clip, ViT-L/16	0.084	19.9	0.067
	0.105(25.0%)	19.0(4.5%)	0.076(13.4%)
Clip, ViT-L/16 → Clip, ViT-B/32	0.095	19.4	0.072
	0.092(3.1%)	19.4(0.0%)	0.073(1.4%)

TABLE 6
Results of transfer attack across different datasets.

	KPA ↓	SPL ↑	SR ↓
MP3D1 → MP3D2	0.242	77.5	0.122
	0.254(5.0%)	75.8(2.2%)	0.128(4.9%)
MP3D2 → MP3D1	0.253	75.1	0.132
	0.257(1.6%)	75.5(0.5%)	0.133(0.8%)
HM3D1 → HM3D2	0.089	15.6	0.064
	0.096(7.9%)	15.1(3.3%)	0.069(7.8%)
HM3D2 → HM3D1	0.100	18.8	0.076
	0.104(4.0%)	18.0(4.4%)	0.081(6.6%)

results (Table 3), with attack objects (denoted as "attacked") and without attack objects (denoted as "normal") in a white-box scenario against various navigation agents, including non-interactive agents (Vlmaps [16], Cow [17], CF [19]) and interactive agents (ORION [18]). Examples of the adversarial objects are shown in Fig. 6.

In 2D adversarial attacks, the attack surface typically consists of RGB images. As these attacks are not optimized for 3D environments, their impact on navigation agents remains limited. MF applies global noise affecting the entire image, but its effect on specific adversarial targets remains constrained. This limitation manifests in minimal changes to the agent's perceptual confidence, exemplified in Fig. 6(b), where the 'chair' confidence score merely decreases from 0.44 to 0.41. However, despite its limited efficacy on adversarial objects, MF significantly degrades navigation performance on normal datasets. By contrast, AdvCLIP utilizes patch-based perturbations independently optimized per view. As demonstrated in Fig. 6(c), AdvCLIP achieves greater disruptive efficacy, reducing the 'chair' confidence score from 0.44 to 0.13 compared to MF's modest decrease (0.44 to 0.41). Nevertheless, since AdvCLIP optimizes patches for individual views, this approach introduces cross-view inconsistencies. While effective for previously observed viewpoints, AdvCLIP cannot generalize to novel perspectives. By concentrating perturbations on victim object regions while minimizing effects on other elements, AdvCLIP better preserves navigation performance on normal datasets.

In 3D adversarial attacks, the attack surface operates in 3D space. While these methods consider the relationship between 3D spatial attacks and 2D images, they are not specifically optimized for VLMs, resulting in reduced effec-

TABLE 7
Results of transfer attack across different models.

	KPA ↓	SPL ↑	SR ↓
(Lseg) → (Clip)	0.084	19.9	0.067
	0.125	17.7	0.096
	0.131(4.8%)	17.2(2.9%)	0.097(1.0%)
(Clip)→ (Lseg)	0.242	77.5	0.122
	0.331	67.8	0.184
	0.313(5.4%)	71.9(6.0%)	0.159(13.4%)

TABLE 8
Results of physical noise against to adversarial objects.

	KPA ↓	SPL ↑	SR ↓
Clean	0.560	49.0	0.297
AdvOF	0.242	77.5	0.122
with Shear	0.283	74.1	0.149
with Scaling	0.255	76.5	0.132
with Gaussian	0.289	72.1	0.150
with Brightness+	0.243	75.2	0.138
with Brightness-	0.241	75.0	0.129

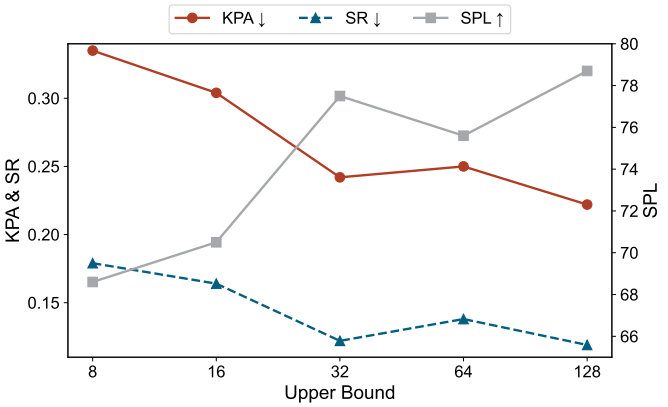


Fig. 8. Attacking results of different upper bounds.

tiveness on 2D images. ST (Fig. 6(d)) optimizes adversarial objects using trajectory historis, achieving success in scenarios with similar trajectories. However, ST demonstrates poor attack efficacy when following divergent instructions that produce anomalous trajectories. TT3D (Fig. 6(e)) focuses more on victim objects themselves, yielding stronger attack performance. Nevertheless, its efficacy is constrained by the complexity of VLMs employed in perception modules. Additionally, both attacks exhibit non-negligible impacts on the surroundings of adversarial objects, inducing perceptual interference on other objects in normal datasets.

In contrast, **AdvOF** achieves a well-balanced attacking, effectively disrupting navigation performance on attacked datasets while preserving performance on normal datasets. This balance is evident from Table 3, where **AdvOF** outperforms both MF and AdvCLIP, demonstrating superior adversarial efficacy without compromising normal task performance. More importantly, Fig. 6-(d) demonstrates that **AdvOF** can precisely manipulate the perception of the adversarial object (chair:0.44 → table:0.81).

Perception Performance. To provide deeper insight into the attack effectiveness and validate the root cause of navigation failures, we conduct a targeted evaluation of the perception module's robustness. we sampled 20 images from diverse new perspectives for each victim object in the validation datasets and evaluated the performance of VLMs on these images. The results are presented in Table 4. Results consistent with those in Table 3 can be observed. The two 2D attack methods fail to mislead the VLM's predictions from new perspectives. In contrast, the two 3D attack methods demonstrate some attack efficacy, but their performance is inconsistent and susceptible to environmental interference. Conversely, AdvOF achieves stable attack efficacy from new perspectives to a significant extent.

Multi-View Robustness. We validated the multi-view robustness in Fig. 7. Clearly, **AdvOF** can successfully generate the adversarial object from a long-range view (view1) to a close-up view (view4). Additionally, **AdvOF** can precisely control the effectiveness of adversarial object within its own region, without affecting the perception of other objects in the environment. This object-specific control, combined with multi-view robustness, provides a more powerful attack in VLN tasks compared to baseline attacks.

5.3 Transferable Evaluation

In this section, we further examine the adversarial transferability of **AdvOF** across different scenarios.

Different Image Encoders. We utilized different image encoder backbones to evaluate the transferability of adversarial attacks across various model architectures. For the Lseg model, we selected ViT-L/16 and ResNet101 as the image encoder backbones, while for the Clip model, we used ViT-B/32 and ViT-L/16. As shown in Table 5, where in each category, the first row represents the performance of attacks generated and tested on the same backbone, and the second row represents the performance when transferred to a different backbone, with the decline rate indicated in red, our results demonstrate strong transferability of adversarial attacks across different image encoders. Specifically, the largest observed decline in performance is only 25.0%. Furthermore, in both the Lseg and Clip models, the adversarial attacks transfer effectively between different encoders, exhibiting minimal degradation in performance.

Different Scene Datasets. We constructed different scene datasets featuring the same victim objects to evaluate the transferability of **AdvOF** across validation datasets. The corresponding results are shown in Table 6, where **AdvOF** demonstrates excellent transferability, with almost no degradation in attack performance across datasets (the largest decline is just 7.9%). In fact, different scene datasets for the same adversarial object simply represent different views. Thus, **AdvOF**, with its multi-view robustness, confirms its effectiveness in such transferable settings.

Different Model Architectures. Furthermore, we evaluated the transferability of our method across different navigation agents that utilize various VLMs. This evaluation aligns with the practical consideration of a black-box scenario. The corresponding results are presented in Table 7. In each category of the table, the first row shows the performance of

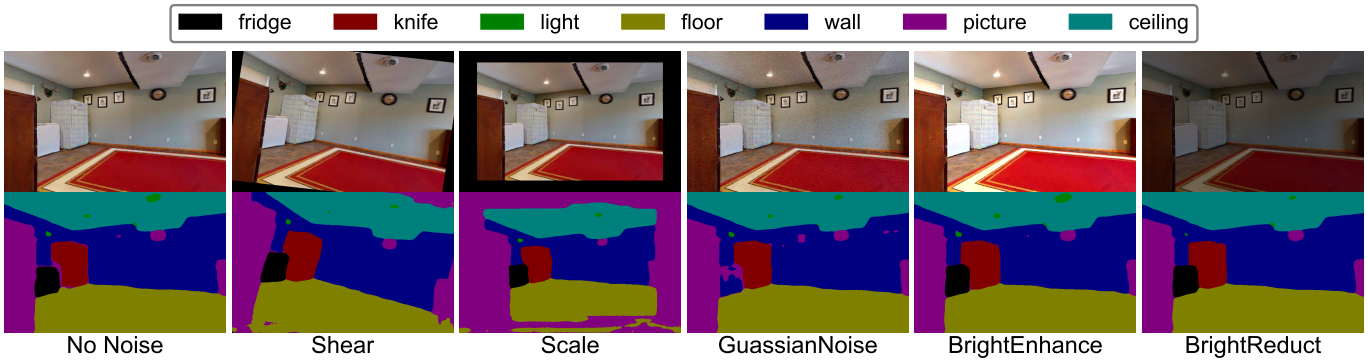


Fig. 9. The attacking results of the adversarial object against to different noise. fridge (black) → knife (red)

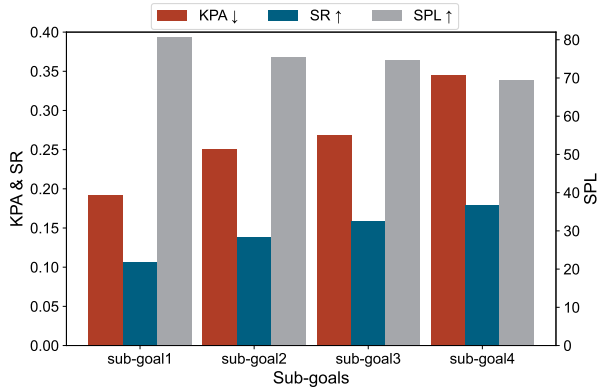


Fig. 10. Attacking results of different subordinate goals.

attacks generated and tested on the same VLM. The second row displays the best result from the baseline methods for that VLM. The third row indicates the performance of attacks when transferred to a different VLM. Compared to the white-box attack scenario, the effectiveness of the attack decreases significantly in the black-box setting. However, our method still achieves performance comparable to that of baseline attacks in a white-box scenario, demonstrating its transferable capability in black-box conditions.

5.4 Possible Defense

In practice, the perception of an adversarial object is influenced by environmental factors such as camera pose, light intensity, distance, and other noise. To evaluate the robustness of **AdvOF** against physical noise defenses, we utilized an image processing toolkit[1] to perturb the adversarial objects. Specifically, we applied five image processing techniques: shearing, scaling, Gaussian noise, brightness enhancement, and brightness reduction. The parameters for these transformations were configured as follows: shear(-16, 16), scaling(0.8), Gaussian noise(0.1), brightness enhancement(1.5), and brightness reduction(0.5). The results are summarized in Table 8 and sample perturbed images are illustrated in Fig. 9. It is evident that **AdvOF** exhibits remarkable robustness across various types of image noise, providing enhanced adaptability in real-world scenarios and under diverse environmental conditions. Moreover, the noise types of shearing and Gaussian noise exert a greater influence on adversarial objects, whereas brightness changes have a lesser impact.

1. https://github.com/aleju/imgaug

TABLE 9
Results of Ablation Study.

	KPA ↓	SPL ↑	SR ↓
AdvOF	0.242	77.5	0.122
w/o *Alignment*	0.450	59.6	0.233
w/o $\mathcal{L}_{2d}$	0.512	53.5	0.267
w/o $\mathcal{L}_{3d}$	0.310	70.1	0.169
w/o *Fusion*	0.338	68.9	0.176

5.5 Parameter Analysis

Upper Bound Selection. We study the effect of different adversarial upper bounds on the attack performance of **AdvOF**. The corresponding results are shown in Fig. 8. The attack performance gradually increases as the upper bound increases, but satisfactory performance can generally be achieved with a low bound ($\epsilon = 32$).

Adversarial Object Selection. For a VLN task, a navigation instruction often involves multiple object goals. Therefore, we investigate how adversarial objects affect the navigation of other object goals. As shown in Fig. 10 (sub-goal@i, where i represents the goal order in the navigation instruction or trajectory), we attacked victim objects with different position orders. When the victim object positioned at the front is attacked, it significantly impacts the navigation performance of subsequent goals. This occurs because the agent moves to a substantially different area, which is less likely to contain other goals. In contrast, when the victim object positioned last is attacked, the impact on navigation performance is smaller, as the adversarial goal only disrupts itself, leaving the preceding goals unaffected.

Ablation Study. We examined the contribution of each component in the proposed attack framework. The corresponding results are shown in Table 9. *Alignment* refers to the aligned object rendering, which helps locate the 2D and 3D positions of the victim object. We test **AdvOF** by directly adopting the 2D positions recognized by SAM. The results demonstrate that these identified positions are

disordered and fail to reveal the true 3D positions of the victim object. $\mathcal{L}_{2d}$ defines the objective function in 2D space, which directly influences the scene perception module of the VLN agent and plays a crucial role in generating the adversarial object. $\mathcal{L}_{3d}$ constrains the perturbation in 3D space, primarily helping to optimize the adversarial object with respect to physical properties of the victim object. Fusion refers to the adversarial object fusion, which helps integrate the perturbation based on views with different importance weights. It contributes to the stable convergence of the optimization process and the generation of the optimized adversarial object.

6 Conclusion

In this paper, we introduce a novel problem of adversarial objects targeting VLN agents integrated with foundation models. To tackle this problem, we propose **AdvOF**, which leverages aligned object rendering, adversarial collaborative optimization and adversarial object fusion. **AdvOF** enables precise localization of the victim object, facilitating alignment between 3D manipulations and 2D perceptions. Furthermore, **AdvOF** can generate an effective adversarial object through collaborative optimization. Additionally, **AdvOF** enhances attack performance by incorporating multiple views with importance weights. This proposed method presents a significant threat to the evolving capabilities of VLN agents empowered by foundation models.

References

[1] Y. Wang, Q. He, X. Zhang, D. Ye, and Y. Yang, "Efficient qos-aware service recommendation for multi-tenant service-based systems in cloud," *IEEE Transactions on Services Computing*, vol. 13, no. 6, pp. 1045–1058, 2017.

[2] X. Kong, Y. Wu, H. Wang, and F. Xia, "Edge computing for internet of everything: A survey," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23 472–23 485, 2022.

[3] K. Lei, M. Du, J. Huang, and T. Jin, "Groupchain: Towards a scalable public blockchain in fog computing of iot services computing," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 252–262, 2020.

[4] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3674–3683.

[5] H. Huang, V. Jain, H. Mehta, A. Ku, G. Magalhaes, J. Baldridge, and E. Ie, "Transferable representation learning in vision-and-language navigation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7404–7413.

[6] X. Wang, W. Xiong, H. Wang, and W. Y. Wang, "Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 37–53.

[7] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6629–6638.

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[9] 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.

[10] H. Xiong, J. Bian, Y. Li, X. Li, M. Du, S. Wang, D. Yin, and S. Helal, "When search engine services meet large language models: visions and challenges," *IEEE Transactions on Services Computing*, 2024.

[11] 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 *ICML*. PMLR, 2021, pp. 8748–8763.

[12] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.

[13] V. Pasquadibisceglie, A. Appice, G. Castellano, and D. Malerba, "Jarvis: Joining adversarial training with vision transformers in next-activity prediction," *IEEE Transactions on Services Computing*, vol. 17, no. 4, pp. 1593–1606, 2023.

[14] J. Wang, H. Du, Y. Liu, G. Sun, D. Niyato, S. Mao, D. I. Kim, and X. Shen, "Generative ai based secure wireless sensing for isac networks," *arXiv preprint arXiv:2408.11398*, 2024.

[15] C. Zhang, G. Sun, J. Li, Q. Wu, J. Wang, D. Niyato, and Y. Liu, "Multi-objective aerial collaborative secure communication optimization via generative diffusion model-enabled deep reinforcement learning," *IEEE Transactions on Mobile Computing*, 2024.

[16] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *ICRA*, 2023, pp. 10 608–10 615.

[17] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *CVPR*, 2023, pp. 23 171–23 181.

[18] Y. Dai, R. Peng, S. Li, and J. Chai, "Think, act, and ask: Open-world interactive personalized robot navigation," in *ICRA*, 2024, pp. 3296–3303.

[19] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, "CLIP-fields: Weakly supervised semantic fields for robotic memory," in *ICRA Workshop*, 2023.

[20] Z. Xu, K. Wu, J. Wen, J. Li, N. Liu, Z. Che, and J. Tang, "A survey on robotics with foundation models: toward embodied ai," *arXiv preprint arXiv:2402.02385*, 2024.

[21] K. Kawaharazuka, T. Matsushima, A. Gambardella, J. Guo, C. Paxton, and A. Zeng, "Real-world robot applications of foundation models: A review," *Advanced Robotics*, pp. 1–23, 2024.

[22] R. Schumann, W. Zhu, W. Feng, T.-J. Fu, S. Riezler, and W. Y. Wang, "Velma: Verbalization embodiment of llm agents for vision and language navigation in street view," in *AAAI*, vol. 38, no. 17, 2024, pp. 18 924–18 933.

[23] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *CVPR*, 2023, pp. 4015–4026.

[24] S. Wang, S. Nepal, C. Rudolph, M. Grobler, S. Chen, T. Chen, and Z. An, "Defending adversarial attacks via semantic feature manipulation," *IEEE Transactions on Services Computing*, vol. 15, no. 6, pp. 3184–3197, 2021.

[25] 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," *arXiv preprint arXiv:2405.19802*, 2024.

[26] X. Lu, Z. Huang, X. Li, W. Xu *et al.*, "Poex: Policy executable embodied ai jailbreak attacks," *arXiv preprint arXiv:2412.16633*, 2024.

[27] T. Wang, D. Liu, J. C. Liang, W. Yang, Q. Wang, C. Han, J. Luo, and R. Tang, "Exploring the adversarial vulnerabilities of vision-language-action models in robotics," *arXiv preprint arXiv:2411.13587*, 2024.

[28] C. Wen, J. Liang, S. Yuan, H. Huang, and Y. Fang, "How secure are large language models (llms) for navigation in urban environments?" *arXiv preprint arXiv:2402.09546*, 2024.

[29] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," *arXiv preprint arXiv:2307.15043*, 2023.

[30] Y. Zhao, W. Zheng, T. Cai, X. Do Long, K. Kawaguchi, A. Goyal, and M. Q. Shieh, "Accelerating greedy coordinate gradient and general prompt optimization via probe sampling," *Advances in Neural Information Processing Systems*, vol. 37, pp. 53 710–53 731, 2024.

[31] G. Deng, Y. Liu, Y. Li, K. Wang, Y. Zhang, Z. Li, H. Wang, T. Zhang, and Y. Liu, "Jailbreaker: Automated jailbreak across multiple large language model chatbots," in *NDSS*, 2024.

[32] A. Mehrotra, M. Zampetakis, P. Kassianik, B. Nelson, H. Anderson, Y. Singer, and A. Karbasi, "Tree of attacks: Jailbreaking black-box llms automatically," *Advances in Neural Information Processing Systems*, vol. 37, pp. 61 065–61 105, 2024.

[33] Z. Zhou, S. Hu, M. Li, H. Zhang, Y. Zhang, and H. Jin, "Advclip: Downstream-agnostic adversarial examples in multimodal contrastive learning," in *ACMMM*, 2023, pp. 6311–6320.

[34] 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," *NeurIPS*, vol. 36, 2023.

[35] A. Liu, T. Huang, X. Liu, Y. Xu, Y. Ma, X. Chen, S. J. Maybank, and D. Tao, "Spatiotemporal attacks for embodied agents," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*. Springer, 2020, pp. 122–138.

[36] Y. Huang, Y. Dong, S. Ruan, X. Yang, H. Su, and X. Wei, "Towards transferable targeted 3d adversarial attack in the physical world," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 24 512–24 522.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[38] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[40] 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," *NeurIPS*, vol. 35, pp. 24 824–24 837, 2022.

[41] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang, "An embodied generalist agent in 3d world," in *ICLR 2024 Workshop: How Far Are We From AGI*, 2024. [Online]. Available: https://openreview.net/forum?id=ltX3S0juSa

[42] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," in *CVPR*, 2023, pp. 2998–3009.

[43] D. Zheng, S. Huang, L. Zhao, Y. Zhong, and L. Wang, "Towards learning a generalist model for embodied navigation," in *CVPR*, 2024, pp. 13 624–13 634.

[44] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "Zson: Zero-shot object-goal navigation using multimodal goal embeddings," *NeurIPS*, vol. 35, pp. 32 340–32 352, 2022.

[45] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (llm) security and privacy: The good, the bad, and the ugly," *High-Confidence Computing*, p. 100211, 2024.

[46] T. Shen, R. Jin, Y. Huang, C. Liu, W. Dong, Z. Guo, X. Wu, Y. Liu, and D. Xiong, "Large language model alignment: A survey," *arXiv preprint arXiv:2309.15025*, 2023.

[47] L. Li, R. Ma, Q. Guo, X. Xue, and X. Qiu, "Bert-attack: Adversarial attack against bert using bert," in *EMNLP*, 2020, pp. 6193–6202.

[48] 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.

[49] H. Chen, Y. Zhang, Y. Dong, X. Yang, H. Su, and J. Zhu, "Rethinking model ensemble in transfer-based adversarial attacks," in *ICLR*, 2024.

[50] H. Wei, H. Tang, X. Jia, Z. Wang, H. Yu, Z. Li, S. Satoh, L. Van Gool, and Z. Wang, "Physical adversarial attack meets computer vision: A decade survey," *IEEE TPAMI*, 2024.

[51] T. Tsai, K. Yang, T.-Y. Ho, and Y. Jin, "Robust adversarial objects against deep learning models," in *AAAI*, vol. 34, no. 01, 2020, pp. 954–962.

[52] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "Pointcloud saliency maps," in *CVPR*, 2019, pp. 1598–1606.

[53] A. Hamdi, S. Rojas, A. Thabet, and B. Ghanem, "Advpc: Transferable adversarial perturbations on 3d point clouds," in *ECCV*, 2020, pp. 241–257.

[54] H. Zhou, D. Chen, J. Liao, K. Chen, X. Dong, K. Liu, W. Zhang, G. Hua, and N. Yu, "Lg-gan: Label guided adversarial network for flexible targeted attack of point cloud based deep networks," in *CVPR*, 2020, pp. 10 356–10 365.

[55] N. Suryanto, Y. Kim, H. Kang, H. T. Larasati, Y. Yun, T.-T.-H. Le, H. Yang, S.-Y. Oh, and H. Kim, "Dta: Physical camouflage attacks using differentiable transformation network," in *CVPR*, 2022, pp. 15 305–15 314.

[56] N. Suryanto, Y. Kim, H. T. Larasati, H. Kang, T.-T.-H. Le, Y. Hong, H. Yang, S.-Y. Oh, and H. Kim, "Active: Towards highly transferable 3d physical camouflage for universal and robust vehicle evasion," in *CVPR*, 2023, pp. 4305–4314.

[57] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *CVPR*, 2017, pp. 2107–2116.

[58] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017, pp. 652–660.

[59] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NeurIPS*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[60] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE TPMAI*, vol. 39, no. 6, pp. 1137–1149, 2016.

[61] J. Redmon, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.

[62] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision-and-language navigation in continuous environments," in *ECCV*, 2020, pp. 104–120.

[63] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *CIRA*. IEEE, 1997, pp. 146–151.

[64] F. Shi, X. Chen, K. Misra, N. Scales, D. Dohan, E. H. Chi, N. Schärli, and D. Zhou, "Large language models can be easily distracted by irrelevant context," in *ICML*. PMLR, 2023, pp. 31 210–31 227.

[65] R. Duan, X. Ma, Y. Wang, J. Bailey, A. K. Qin, and Y. Yang, "Adversarial camouflage: Hiding physical-world attacks with natural styles," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1000–1008.

[66] N. Shahrubudin, T. C. Lee, and R. Ramlan, "An overview on 3d printing technology: Technological, materials, and applications," *Procedia manufacturing*, vol. 35, pp. 1286–1296, 2019.

[67] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[68] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*, vol. 96, no. 34, 1996, pp. 226–231.

[69] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," in *3DV*, 2017, pp. 667–676.

[70] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," in *ICLR*, 2022.

[71] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. M. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, and D. Batra, "Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI," in *NeurIPS*, 2021.

[72] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9339–9347.