



# Consistent Attack: Universal Adversarial Perturbation on Embodied Vision Navigation

Chengyang Ying<sup>a,2</sup>, You Qiaoben<sup>a,2</sup>, Xinning Zhou<sup>a,2</sup>, Hang Su<sup>a,b,\*\*</sup>, Wenbo Ding<sup>c</sup>, Jianyong Ai<sup>c</sup>

<sup>a</sup>Department of Computer Science and Technology, Beijing National Research Center for Information Science and Technology, Tsinghua-Bosch Joint Center for Machine Learning, Institute for Artificial Intelligence, Tsinghua University, Beijing 100084, China

<sup>b</sup>Peng Cheng Laboratory, Shenzhen, Guangdong, 518055, China

<sup>c</sup>SAIC AI Lab

Article history:

Keywords:

Embodied Agent  
Vision Navigation  
Deep Neural Networks  
Universal Adversarial Noise

## ABSTRACT

Embodied agents in vision navigation coupled with deep neural networks have attracted increasing attention. However, deep neural networks have been shown vulnerable to malicious adversarial noises, which may potentially cause catastrophic failures in Embodied Vision Navigation. Among different adversarial noises, universal adversarial perturbations (UAP), i.e., a constant image-agnostic perturbation applied on every input frame of the agent, play a critical role in Embodied Vision Navigation since they are computation-efficient and application-practical during the attack. However, existing UAP methods ignore the system dynamics of Embodied Vision Navigation and might be sub-optimal. In order to extend UAP to the sequential decision setting, we formulate the disturbed environment under the universal noise  $\delta$ , as a  $\delta$ -disturbed Markov Decision Process ( $\delta$ -MDP). Based on the formulation, we analyze the properties of  $\delta$ -MDP and propose two novel Consistent Attack methods, named Reward UAP and Trajectory UAP, for attacking Embodied agents, which consider the dynamic of the MDP and calculate universal noises by estimating the disturbed distribution and the disturbed Q function. For various victim models, our Consistent Attack can cause a significant drop in their performance in the PointGoal task in Habitat with different datasets and different scenes. Extensive experimental results indicate that there exist serious potential risks for applying Embodied Vision Navigation methods to the real world.

© 2023 Elsevier Ltd. All rights reserved.

## 1. Introduction

Embodied Vision Navigation, aiming at reaching a given point or object with sequential vision inputs, is a core component of Embodied Artificial Intelligence (Embodied AI) and has gradually attracted the attention of researchers [1, 2]. Coupled with deep neural networks, Embodied Vision Navigation agents combined with techniques of vision signal processing and se-

quential decision have made sustained progress [2, 3, 4]. Recently, the vulnerability of deep neural networks has been well studied [5, 6, 7], while there is little attention focusing on the robustness of Embodied Vision Navigation agents. Therefore, it is important to address this problem before we can deploy Embodied Vision Navigation agents to the real world.

In the setting of Embodied Vision Navigation, it is impractical to hack the system and generate specific attacks at every timestep, and thus adding a universal adversarial perturbation (UAP) [8] to every observation is a more reasonable setting than calculating observation-wise noises. Moreover, UAP is more feasible to be deployed, e.g. sticking a patch on the sensor can generate universal perturbation. Also, current adversarial attack methods usually require complex computation to generate noises and are thus slow and expensive to be deployed [9, 10]. Therefore, as illustrated in Fig. 1, our goal is to

<sup>\*\*</sup>Corresponding author

*e-mail:* ycy21@mails.tsinghua.edu.cn (Chengyang Ying),  
qby17@mails.tsinghua.edu.cn (You Qiaoben),  
zxn21@mails.tsinghua.edu.cn (Xinning Zhou),  
suhangss@mail.tsinghua.edu.cn (Hang Su),  
dingwenbo@saicmotor.com (Wenbo Ding),  
aijianyong@saicmotor.com (Jianyong Ai)

<sup>2</sup>Equal Contribution



Fig. 1. An illustration of universal adversarial perturbation on observations of an agent in Embodied Vision Navigation. In this task, the agent needs to navigate from the red triangle to the red star. The green curve is the navigable path. At each timestep, the adversary adds a consistent noise to the input frame. Finally, the adversary misleads the agent to take the blue curve.

study UAP, which is one of the most cost-effective adversarial attacks, against Embodied agents. We aim to find the optimal universal noise  $\delta^*$  to add to every observation, which can mislead the victim and prevent it from reaching the goal.

Although there is an array of UAP methods [8, 11, 12] that have achieved promising results in image classification, the potential risk of UAP in Embodied Vision Navigation has been rarely investigated. Moreover, existing UAP methods consider the setting that all samples are i.i.d sampled from the same distribution and are independent of each other, which no longer holds since Embodied Vision Navigation is a Markov Decision Process (MDP) and the sequential frames are related to each other. Consequently, directly applying previous UAP methods to Embodied Vision Navigation is not the optimal choice.

To handle this gap, in this paper, we first introduce  $\delta$ -Markov Decision Process ( $\delta$ -MDP) to formulate the persistent effect of the Universal Adversarial Perturbation, based on which we propose Disturbed Policy Gradient Theorem (Theorem 4.2) to calculate the gradient of the disturbance  $\delta$  via the disturbed Q function. According to the result, we need to sample state-action pairs from the disturbed distribution and estimate the disturbed Q function for calculating the gradient. Furthermore, we propose two novel Consistent Attack methods named Reward UAP and Trajectory UAP for calculating the optimal universal adversarial perturbation of Embodied Vision Navigation agents. In Reward UAP, we first propose a multi-step optimization framework to ensure trajectories in every step are sampled from the disturbed policy, and then we use two different ways to estimate the disturbed Q function. In Trajectory UAP, we consider Embodied Vision Navigation as a goal-condition problem and just use the goal signal to optimize our universal adversarial perturbation, which is a more practical assumption since the adversary can easily know whether the agent reaches the goal rather than the actual reward signal of the agent.

Extensive experimental results suggest that our Consistent Attack can significantly reduce the performance of existing Embodied Vision Navigation methods in Habitat [13], which shows the effectiveness of our method and the serious vulnerability of existing Embodied Vision Navigation methods.

## 2. Related Work

### 2.1. Embodied Vision Navigation

Since the agent in Embodied Vision Navigation needs to actively interact with the environment [14, 15, 13], its safety requirement is more important compared with traditional AI tasks, e.g., image classification, sentence generation, and audio recognition. For example, in the PointGoal navigation task [2] in Habitat, the agent tries to find a path to reach the destination, while it might suffer from catastrophic failure when collides along the way [13]. In this paper, we mainly focus on the security of the agent [13] in the PointGoal task, which is a common and popular baseline in Embodied AI.

There are many works considering the robustness of models in different computer vision tasks [16, 17, 18], like face recognition [19, 20] and face forgery detection [21, 22, 23]. However, a major difference between these tasks and Embodied Vision Navigation is whether the inputs are independent and identically distributed.

### 2.2. Adversarial Attacks

As a special adversarial attack method in image classification, universal adversarial perturbation (UAP) aims to fool the classifier for most images via a constant noise [24, 8, 11, 12]. However, existing UAP methods are limited to datasets where images are mutually independent and are thus sub-optimal for Embodied Vision Navigation since it fails to explore the dependency between different images in the same trajectory.

After [9] first proposed to use FGSM-based adversarial noises to attack policies with deep neural networks, there are an array of works focusing on the vulnerability of deep reinforcement learning (DRL) agents. [10] proposed a framework named state-adversarial Markov decision process (SA-MDP) for analyzing the adversarial disturbance on state observation in reinforcement learning (RL). Recently, [25] summarized existing adversarial attack methods on observations in RL and proposed a novel two-stage adversarial attack method. However, these adversarial attack methods generate different disturbances at each timestep, which are impracticable in Embodied Vision Navigation. In general, it is nontrivial to calculate the adversarial noise and hack the system at each timestep when the agent is interacting with the environment.

## 3. Preliminaries

In this section, we start with the notations and then introduce the direct application of Universal Adversarial perturbation (UAP) [8] on Embodied Vision Navigation.

### 3.1. Notations

**Markov Decision Process.** Following previous works on Embodied Vision Navigation [3, 4, 26], we formulate it as an agent with the policy  $\pi$  interacting with the environment, which is a MDP  $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma\}$ . Here  $\mathcal{S}$  and  $\mathcal{A}$  represent the state space and the action space, respectively. At each timestep  $t$ , the Embodied Vision Navigation agent is at the state  $s_t$  and will choose an action  $a_t$  from the action space  $\mathcal{A}$  via its policy  $\pi$ .

Then the agent will receive an immediate reward  $\mathcal{R}(s_t, a_t)$  and move to the next state  $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$ . The goal of Embodied Vision Navigation is to find the optimal policy to maximize the expected cumulative reward as

$$J(\pi) \triangleq \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \right], \quad (1)$$

where  $\gamma$  is the discount factor.

**Universal Adversarial perturbation.** As widely studied in image classification tasks, UAP [8] aims to utilize the same noise  $\delta$  to mislead the classifier, which is more computationally efficient and easier to be deployed. In other words, given images obeying the distribution  $\mu$  and the victim classifier  $f$ , the goal is to find the optimal disturbance  $\delta^* \in \mathcal{B}_p(0, \epsilon)$  satisfying

$$\delta^* = \arg \min_{\delta \in \mathcal{B}_p(0, \epsilon)} P_{s \sim \mu} \{f(s + \delta) = f(s)\}, \quad (2)$$

where  $\mathcal{B}_p(0, \epsilon)$  is the ball centered on 0 with radius  $\epsilon$  in  $l_p$ -norm corresponding to the feasible region of UAP.

### 3.2. UAP in MDP

Based on the definition of UAP mentioned in Sec. 3.1, we can naively extend UAP to tasks with MDP settings, like Embodied Vision Navigation. Since the policy in MDP might be deterministic or stochastic, we discuss how to apply UAP in these two cases separately.

When the policy is deterministic, the adversary will collect observations by the victim policy  $\pi$  and generate the perturbation  $\delta$  with the objective as

$$\delta^* = \arg \min_{\delta \in \mathcal{B}_p(0, \epsilon)} P_{s \sim d^\pi(s)} \{\pi(s + \delta) = \pi(s)\}, \quad (3)$$

where  $d^\pi(s)$  is the distribution of the state.

When the victim policy is stochastic (e.g., in Proximal Policy Optimization [27]), similar to adversarial attack methods in DRL [9], we aim at minimizing the probability to take the optimal action, i.e.,

$$\delta^* = \arg \min_{\delta \in \mathcal{B}_p(0, \epsilon)} \mathbb{E}_{s \sim d^\pi(s)} [\pi(a|s + \delta)], \quad (4)$$

where  $a = \arg \max_{a' \in \mathcal{A}} \pi(a'|s)$ .

In the rest of this paper, we will mainly focus on the stochastic agent [13] since the deterministic policy is a degenerate case of the stochastic policy, i.e, when the variance of the stochastic agent goes to zero, problem (4) is equivalent to problem (3).

However, as mentioned in Sec. 1, current UAP does not consider the transition dynamics of the MDP and the sequential dependence of states within a trajectory. We will address those problems in the discussion of our method in Sec. 4.

## 4. Methodology

To analyze UAP in MDP, we first introduce  $\delta$ -MDP to formulate the universal noise's effects. Based on our further analysis of properties of  $\delta$ -MDP, we propose two novel Consistent Attack methods named Reward UAP and Trajectory UAP.

### 4.1. $\delta$ -Markov Decision Process

To extend UAP to the sequential decision problem, we first formulate it as  $\delta$ -disturbed Markov Decision Process ( $\delta$ -MDP), in which the observation in every timestep is disturbed by the same adversarial noise  $\delta$ .

**Definition 4.1 ( $\delta$ -MDP).**  $\delta$ -MDP consists of a six-tuple  $\mathcal{M}_\delta = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma, \delta)$ . Here  $\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma$  is the same as MDP and  $\delta$  satisfies that  $s + \delta \in \mathcal{S}$  holds for  $\forall s \in \mathcal{S}$ . For any fixed policy  $\pi$ , when its true state  $s \in \mathcal{S}$  is disturbed as  $s + \delta$ , the agent takes action  $a \sim \pi(\cdot|s + \delta)$ , arrives at the next state  $s' \sim \mathcal{P}(\cdot|s, a)$ , and receives the current reward  $\mathcal{R}(s, a)$ .

For simplicity, we define that  $\pi_\delta(\cdot|s) = \pi(\cdot|s + \delta)$ . It is easy to check that the policy  $\pi$  interacting with  $\mathcal{M}_\delta$  is equivalent to the disturbed policy  $\pi_\delta$  interacting with  $\mathcal{M}$ . For any victim policy  $\pi$  in  $\delta$ -MDP, we can define its disturbed cumulative return  $J_\delta(\pi)$  as

$$J_\delta(\pi) = \mathbb{E}_{s_t, a_t \sim \pi_\delta, \mathcal{P}} \left[ \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \right]. \quad (5)$$

The purpose of Consistent Attack for Embodied Vision Navigation is different from that of UAP in image classification, while the latter misleads the neural network to classify most images incorrectly, the former aims to minimize the disturbed cumulative return. To find the best disturbance  $\delta^*$  within the disturbance region, our optimization objective is

$$\delta^* = \arg \min_{\delta \in \mathcal{B}_p(0, \epsilon)} J_\delta(\pi). \quad (6)$$

For solving the objective 6, we first analyze some properties of our  $\delta$ -MDP. Similar to the standard MDP [28], we can define the discount future state distribution as  $d_\delta^\pi(s) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t P_\delta(s_t = s)$  and equivalently deform  $J_\delta(\pi)$  as

$$J_\delta(\pi) = \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d_\delta^\pi(\cdot), a \sim \pi(\cdot|s)} \mathcal{R}(s, a). \quad (7)$$

Also, we can define the  $\delta$ -disturbed Q function and the  $\delta$ -disturbed value function as below

**Definition 4.2 ( $\delta$ -disturbed Q-function and value function).**  $\delta$ -disturbed Q function is defined as

$$Q_\delta^\pi(s, a) = \mathbb{E}_{s_t, a_t \sim \pi_\delta, \mathcal{P}} \left[ \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \middle| s_0 = s, a_0 = a \right], \quad (8)$$

and the  $\delta$ -disturbed value function is defined as

$$V_\delta^\pi(s) = \mathbb{E}_{s_t, a_t \sim \pi_\delta, \mathcal{P}} \left[ \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \middle| s_0 = s \right]. \quad (9)$$

Similar to the Bellman equation in MDP, the  $\delta$ -disturbed Q function and the  $\delta$ -disturbed value function of our  $\delta$ -MDP also satisfy the disturbed Bellman equation as below

**Theorem 4.1** (Disturbed Bellman equation for  $\delta$  and  $\pi$ ).  $V_\delta^\pi$  and  $Q_\delta^\pi$  satisfy the disturbed Bellman equation as below

$$\begin{aligned} V_\delta^\pi(s) &= \mathbb{E}_{a, s' \sim \pi_\delta, \mathcal{P}} [\mathcal{R}(s, a) + \gamma V_\delta^\pi(s')] \\ Q_\delta^\pi(s, a) &= \mathcal{R}(s, a) + \gamma \mathbb{E}_{s', a' \sim \mathcal{P}, \pi_\delta} Q_\delta^\pi(s', a'). \end{aligned} \quad (10)$$

We can directly prove it by the existing results in MDP since the policy  $\pi_\delta(\cdot|s)$  is equivalent to  $\pi(\cdot|s + \delta)$ .

#### 4.2. Reward UAP and Trajectory UAP

In this part, we will propose two novel Consistent Attack methods named Reward UAP and Trajectory UAP to optimize the objective 6.

It's difficult to directly minimize  $J_\delta(\pi)$  since it is always non-convex. Consequently, we consider calculating the gradient  $\nabla_\delta J_\delta(\pi)$  and use gradient descent to optimize it. However, we can directly calculate  $\nabla_{s+\delta}\pi(a|s+\delta)$  via backpropagation rather than  $\nabla_\delta J_\delta(\pi)$ . To overcome this gap, by extending the Policy Gradient Theorem [29], we propose the Disturbed Policy Gradient Theorem as below

**Theorem 4.2** (Disturbed Policy Gradient). *For any policy  $\pi$  and  $\delta$ -MDP  $\mathcal{M}_\delta$ , we can calculate the gradient of  $J_\delta(\pi)$  to  $\delta$  only by calculating the gradient  $\nabla_{s+\delta}\pi(a|s+\delta)$ , i.e.,*

$$\begin{aligned} \nabla_\delta J_\delta(\pi) &= \frac{1}{1-\gamma} \sum_s d_\delta^\pi(s) \sum_a Q_\delta^\pi(s, a) \nabla_\delta \pi(a|s+\delta) \\ &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\delta^\pi} \mathbb{E}_{a \sim \pi(\cdot|s+\delta)} [Q_\delta^\pi(s, a) \nabla_{s+\delta} [\log \pi(a|s+\delta)]] . \end{aligned} \quad (11)$$

*Proof.* Based on Theorem 4.1, we can first build the connection of  $\nabla_\delta V_\delta^\pi(s)$  and  $\nabla_\delta \pi(a|s+\delta)$  as below

$$\begin{aligned} \nabla_\delta V_\delta^\pi(s) &= \nabla_\delta \left[ \sum_a \pi(a|s+\delta) Q_\delta^\pi(s, a) \right] \\ &= \sum_a Q_\delta^\pi(s, a) \nabla_\delta \pi(a|s+\delta) \\ &\quad + \sum_a \pi(a|s+\delta) \nabla_\delta Q_\delta^\pi(s, a) \\ &= \sum_a Q_\delta^\pi(s, a) \nabla_\delta \pi(a|s+\delta) \\ &\quad + \sum_a \pi(a|s+\delta) \gamma \sum_{s'} P(s'|s, a) \nabla_\delta V_\delta^\pi(s'). \end{aligned} \quad (12)$$

Similar to the property of  $d^\pi$  [30], we can prove that  $d_\delta^\pi(s) - (1-\gamma)\mathcal{P}(s_0=s) = \gamma \sum_{s'} d_\delta^\pi(s') \sum_a \pi(a|s'+\delta) \mathcal{P}(s|s', a)$ . This property extends the discount future state distribution  $d_\delta^\pi(s)$  in one step and builds the connection between it and other discount future state distribution  $d_\delta^\pi(s')$ . Based on this property, we can further prove that

$$\begin{aligned} \sum_s d_\delta^\pi(s) \nabla_\delta V_\delta^\pi(s) &= \sum_s d_\delta^\pi(s) \sum_a Q_\delta^\pi(s, a) \nabla_\delta \pi(a|s+\delta) \\ &\quad + \sum_s d_\delta^\pi(s) \sum_a \pi(a|s+\delta) \gamma \sum_{s'} P(s'|s, a) \nabla_\delta V_\delta^\pi(s') \\ &= \sum_s d_\delta^\pi(s) \sum_a Q_\delta^\pi(s, a) \nabla_\delta \pi(a|s+\delta) \\ &\quad + \sum_{s'} [d_\delta^\pi(s') - (1-\gamma)P(s_0=s')] \nabla_\delta V_\delta^\pi(s'). \end{aligned} \quad (13)$$

Thus we have

$$\begin{aligned} \sum_{s'} (1-\gamma)P(s_0=s') \nabla_\delta V_\delta^\pi(s') \\ = \sum_s d_\delta^\pi(s) \sum_a Q_\delta^\pi(s, a) \nabla_\delta \pi(a|s+\delta). \end{aligned} \quad (14)$$

Consequently, we can get

$$\begin{aligned} \nabla_\delta J_\delta(\pi) &= \sum_{s'} P(s_0=s') \nabla_\delta V_\delta^\pi(s') \\ &= \frac{1}{1-\gamma} \sum_s d_\delta^\pi(s) \sum_a Q_\delta^\pi(s, a) \nabla_\delta \pi(a|s+\delta) \\ &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\delta^\pi} \mathbb{E}_{a \sim \pi(\cdot|s+\delta)} [Q_\delta^\pi(s, a) \nabla_{s+\delta} [\log \pi(a|s+\delta)]] , \end{aligned} \quad (15)$$

here it is well known that  $\nabla_\delta [\log \pi(a|s+\delta)] = \nabla_{s+\delta} [\log \pi(a|s+\delta)]$  by the chain rule. Therefore, we have proven this result.  $\square$

As shown in Theorem 4.2, the gradient  $\nabla_\delta J_\delta(\pi)$  is based on  $\nabla_\delta [\log \pi(a|s+\delta)]$  for each state-action pair  $(s, a)$  sampled from the distribution. However, the weight varies among different  $(s, a)$  and depends on the disturbed Q function  $Q_\delta^\pi(s_t, a_t)$ . This result also theoretically shows that directly applying UAP to MDP, i.e., calculating the gradient of different  $(s, a)$  and averaging them, ignores the dynamic connection of different  $(s, a)$  and thus is not optimal.

Based on Theorem 4.2, we can naturally calculate the optimal universal perturbation  $\delta^*$  to minimize  $J_\delta(\pi)$  via gradient descent. However, there are still two critical problems for calculating  $\nabla_\delta J_\delta(\pi)$  in practice. i.e., how to sample state-action pair  $(s, a)$  from the distribution  $d_\delta^\pi(s)\pi(a|s+\delta)$  and how to estimate the disturbed Q function. To address those problems, we propose two novel Consistent Attack methods: Reward UAP and Trajectory UAP.

**Reward UAP.** First, we propose Reward UAP to handle those two critical problems in UAP mentioned above.

For sampling state-action pairs from the disturbed distribution, we propose a multi-step optimization strategy. Specifically, at each optimization step  $k$ , we have current adversarial noise  $\delta_k$  and sample a set of trajectories via  $\pi_{\delta_k}$ . Moreover, we use state-action pairs from these trajectories to optimize  $\delta_k$  by using Theorem 4.2.

For estimating the disturbed Q function which is difficult to directly calculate, we propose two kinds of surrogates. First, we can use the return of the trajectory collected by  $\pi_\delta$  to estimate  $Q_\delta^\pi(s_t, a_t)$ , i.e.,

$$\hat{R}_t \triangleq \sum_{t'=t}^{\infty} \gamma^{t'-t} \mathcal{R}(s_{t'}, a_{t'}) \approx Q_\delta^\pi(s_t, a_t). \quad (16)$$

Also, we can directly use the victim's Q function to approximate the disturbed Q function, i.e.,  $\hat{R}_t \triangleq Q^\pi(s_t, a_t) \approx Q_\delta^\pi(s_t, a_t)$ . Consequently, in our Reward UAP, at each epoch  $k$ , we can estimate  $\nabla_{\delta_k} J_{\delta_k}(\pi)$  via

$$\nabla_{\delta_k} J_{\delta_k}(\pi) \approx \frac{1}{m} \sum_{i=1}^m \sum_t \hat{R}_t^i \nabla_{s_t^i + \delta_k} [\log \pi(a_t^i | s_t^i + \delta_k)], \quad (17)$$



**Algorithm 1** Consistent Attack

Given the victim policy  $\pi$ , the disturbance range  $\epsilon$ , the upper bound on the number of samples  $m = nl$ , and the learning rate  $\alpha$ .

- 1: Initialize universal perturbation  $\delta_0^{(0)} = 0$  ( $\delta_i^{(j)}$  indicates  $\delta_i$  in Reward Attack and  $\delta_i^j$  in Trajectory Attack)
- 2: **for**  $k = 0, 1, 2, \dots, n - 1$  **do**
- 3: Random collect a set of  $l$  trajectories  $\mathcal{D} = \{\tau_i\}_{i=1}^l$  by the disturbed policy  $\pi_{\delta_k}$ , here trajectory  $\tau_i = \{s_1^i, a_1^i, r_1^i, s_2^i, \dots\}$  with  $g_i$  indicates whether the agent reaches the goal.
- 4: **Option 1: Gradient Update on Reward UAP**{
- 5: Compute the reward-to-go as  $\hat{R}_t^i = \sum_{t'=t}^T \gamma^{t'-t} r_{t'}^i$ , or use the Q function.
- 6: Update the disturbance:

$$\delta_{k+1} = \delta_k - \alpha \sum_i \sum_{t=0}^T \hat{R}_t^i \nabla_{\delta_k} [\log \pi_{\delta_k}(a_t^i | s_t^i)].$$

}

- 7: **Option 2: Gradient Update on Trajectory UAP**{
- 8: Update the disturbance:

$$\delta_{k+1} = \delta_k - \alpha \sum_i g_i \sum_{t=0}^T \gamma^{T-t} \nabla_{\delta_k} [\log \pi_{\delta_k}(a_t^i | s_t^i)].$$

}

- 9: **end for**

- 10: Project the universal perturbation to the boundary of the disturbance region:

$$\delta_n = \epsilon \frac{\delta_n}{\|\delta_n\|}.$$

and use this approximator to update  $\delta_k$ . The pseudo-code of Reward UAP is in Algorithm 1.

**Trajectory UAP.** Although Reward UAP can find the optimal gradient direction, which is the best direction for reducing the disturbed cumulative return, there are still some practical problems in Reward UAP. First, by Theorem 4.2, we need to estimate  $Q_\delta^\pi(s, a)$  to calculate the gradient  $\nabla_\delta J_\delta(\pi)$ . There are two common ways for estimating  $Q_\delta^\pi(s, a)$ , i.e., use the return of trajectories to directly estimate it or train a neural network to estimate it. However, the former one may suffer from high variance and the latter one needs to re-train the neural network whenever the noise  $\delta$  is changed. Second, in some practical settings like the PointGoal task in Embodied Vision Navigation, the adversary can only determine whether the agent has reached the goal rather than having access to the actual reward signals, which are related to the distance between the agent and the goal in Habitat [13].

To handle those challenges, we regard Embodied Vision Navigation as a goal-condition MDP, of which the return solely

depends on whether the agent has reached the goal  $s_g$ , i.e.,

$$\mathcal{R}_g(s_t, a_t) = \begin{cases} 1 & \text{the agent reaches the goal, i.e. } s_t = s_g \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

This situation is more practical since we can easily tell whether the agent has reached the goal, and the estimation of the disturbed Q function is more stable. In this case, we can directly calculate the disturbed Q function of state-action pair  $(s_t, a_t)$  in a trajectory  $(s_0, a_0, s_1, a_1, \dots, s_T)$  as

$$Q_g(s_t, a_t) = \begin{cases} \gamma^{T-t} & \text{if } s_T = s_g \\ 0 & \text{otherwise.} \end{cases} \quad (19)$$

Moreover, we introduce  $g$  to indicate whether the agent reaches the goal at this trajectory, i.e.,  $g = 1$  if  $s_T = s_g$  and  $g = 0$  otherwise. We have  $Q_g(s_t, a_t) = \gamma^{T-t} g$ . Based on the above analyses, we propose Trajectory UAP, of which the pseudo-code is in algorithm 1. In summary, we use Trajectory UAP when we can only know whether the agent successfully reaches the goal and Trajectory UAP is much more stable.

## 5. Experimental Results

In this section, we conduct extensive experiments to show the effectiveness of our Consistent Attack methods, including Reward UAP and Trajectory UAP, for Embodied Vision Navigation agents with different sensors in Habitat scenes [13].

### 5.1. Environment setups

**Environments and Tasks.** For evaluating the robustness of agents in Embodied Vision Navigation tasks, we choose the popular platform Habitat [13] with the PointGoal task [2]. Following previous work [13], our experiments are based on the Habitat-test dataset [13], the Gibson dataset [31], and the MatterportP3D (MP3D) dataset [32].

**Victim Policies.** To investigate the vulnerability of Embodied Vision Navigation, we choose the stochastic—as mentioned in Sec. 3.2—victim policies [13] based on Proximal Policy Optimization (PPO) [27], which is a classic reinforcement learning algorithm, and different vision sensors like RGB input (RGB) and depth input (Depth). The details of the victim policy can be found in habitat-baseline [13]. All these agents are officially released pre-trained models [13] with competitive performance in the PointGoal task.

**Baselines.** We compare our Consistent Attack, including Trajectory UAP and Reward UAP, with standard UAP method [8] mentioned at Sec. 3.2, which is a direct extension of UAP to Embodied Vision Navigation by treating all input images mutually independent. As shown in Eq. 4, standard UAP for image classification aims to find a universal disturbance to mislead the classifier on most input images. For the implementation of standard UAP, we sample trajectories with the victim policy and use all observations as the dataset for calculating the disturbance.

**Evaluation.** To show the effectiveness of the Consistent Attack methods, we compare our methods with UAP. For a fair comparison, we choose a similar setup in classical UAP [8].

Environments				Habitat-test			Gibson			MP3D		
Baseline	Sensors	Adversary	$\eta$	Rew	Succ	SPL	Rew	Succ	SPL	Rew	Succ	SPL
RL(PPO)	RGB	None	-	-0.23	0.52	0.41	4.96	0.83	0.68	4.42	0.51	0.38
		UAP	0.5	-2.13	0.13	0.04	-4.98	0.05	0.02	0.24	<b>0.00</b>	<b>0.00</b>
		Reward UAP	0.5	-1.20	0.16	0.07	<b>-5.24</b>	0.02	0.02	<b>-2.54</b>	0.01	0.01
		Trajectory UAP	0.5	<b>-6.64</b>	<b>0.05</b>	<b>0.01</b>	-1.63	<b>0.01</b>	<b>0.00</b>	-0.54	0.02	0.02
	Depth	None	-	6.73	0.93	0.81	6.63	0.92	0.84	9.53	0.80	0.67
		UAP	0.03	-0.34	<b>0.48</b>	0.39	0.09	0.44	0.30	-3.74	0.04	0.03
		Reward UAP	0.03	<b>-0.75</b>	0.49	<b>0.38</b>	-3.78	0.09	0.05	-4.43	0.02	0.02
		Trajectory UAP	0.03	-0.63	0.49	0.39	<b>-4.23</b>	<b>0.05</b>	<b>0.02</b>	<b>-4.57</b>	<b>0.01</b>	<b>0.01</b>

Table 1. Performance per trajectory of the attacked policy on the PointGoal task tested in the Habitat-test, Gibson, and MP3D datasets under RGB and Depth sensor. We report the average reward, episode success rate, and SPL, where the lower reward, episode success rate, and SPL indicate a stronger adversary.

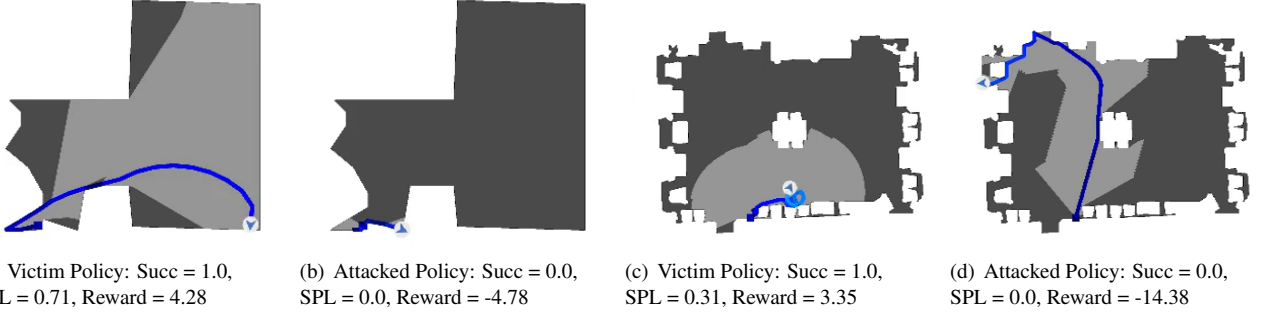


Fig. 2. Visualization of trajectories sampled by the victim policy and the attacked policy via Reward UAP on the PointGoal task in the Habitat-test dataset. The blue line is the trajectory of the Embodied agent and we also report the trajectory’s Succ, SPL, and Reward.

Adversary	$m$	$\eta$	Rew	Succ	SPL
None	-	-	6.73	0.93	0.81
UAP	5	0.03	-0.34	<b>0.48</b>	<b>0.39</b>
Reward UAP	5	0.03	<b>-0.75</b>	<b>0.49</b>	<b>0.38</b>
Trajectory UAP	5	0.03	-0.63	<b>0.49</b>	<b>0.39</b>
UAP	10	0.03	-0.50	0.51	0.38
Reward UAP	10	0.03	<b>-1.57</b>	<b>0.43</b>	<b>0.28</b>
Trajectory UAP	10	0.03	-1.22	0.45	0.32
UAP	15	0.03	-0.60	0.49	0.36
Reward UAP	15	0.03	<b>-4.15</b>	<b>0.16</b>	<b>0.12</b>
Trajectory UAP	15	0.03	-1.11	0.47	0.32

Table 2. Performance of the adversary with different numbers of training trajectories  $m$  tested on the Habitat-test dataset with depth model. We report the average of reward, episode success rate, and SPL. For each  $m$ , numbers within 5 percent of the maximum in every individual environment are bold.

In particular, we evaluate the perturbation in  $l_2$ -norm and set  $\eta = \frac{\epsilon}{\sqrt{d}} = 0.5$  and 0.03 for RGB sensor and depth sensor respectively, following the traditional setting in classical UAP works [8]. Besides, we evaluate the attacked agent at every task separately, e.g., 100 episodes in the Habitat-test dataset, 994 episodes in the Gibson dataset, and 495 episodes in the Matterport3D dataset. At each episode, the agent takes up to 500 actions to find the navigable path following the setting in habitat [13]. For each adversarial attack method, we first evaluate the average reward of the attacked policy, and we also measure the attacked policy’s performance via the Episode Success Rate (Succ) as well as the Success weighted by Path Length (SPL).

## 5.2. Results

**Evaluation of Consistent Attack.** For all three tasks (Habitat-test, Gibson, MP3D) and the victim PPO model with different sensors (RGB, Depth), We report the victim policy’s performance and its performance under different adversaries (UAP, Reward UAP, Trajectory UAP). Here all attack methods only sample 5 trajectories to calculate noises. For each experiment, we report it under three different metrics (reward, episode success rate, SPL). As shown in Table. 4.2, all these adversarial attack methods can reduce the performance in all tasks, especially in more complicated tasks like Gibson. This observation indicates that current Embodied Vision Navigation models are vulnerable to adversarial noises. Moreover, compared to UAP, our consistent attack methods, including Reward UAP and Trajectory UAP, can significantly reduce the victim’s performance under all three metrics because we consider the system dynamics and thus optimize a better objective of the noises.

In almost all tasks tested on the Gibson and MP3D datasets under RGB and Depth sensors, Trajectory UAP outperforms UAP and Reward UAP in reward, Succ, and SPL. It matches our analyses that using goal signal to estimate the disturbed Q function is more stable when the trajectory number for calculating the noise is limited in Embodied Vision Navigation task. Besides, Trajectory UAP almost decreases the Episode Success Rate to zero in MP3D-Depth, which shows that there exist serious potential risks for applying Embodied Vision Navigation methods to the real world. In Fig. 2, we plot some trajectories, with their Succ, SPL, and Reward, of the victim policy and the attacked policy. As shown here, our Consistent Attack can significantly reduce the reward of the victim policy and mislead

the Embodied agent to a place far away from the goal.

**Ablation Study of the Sampling Number  $m$ .** In this part, we analyze how the number of trajectories  $m$  used for calculating noises affects different algorithms, including UAP, Reward UAP, and Trajectory UAP. Due to the limit of computational costs, we evaluate our methods in the Habitat-test [13] dataset with the depth sensor. As shown in Table. 2, when the number of training trajectories  $m$  increases, Reward UAP’s attack capacity significantly improves and is the strongest adversarial attack under all metrics. It indicates that with the increment of sample size, reward UAP can get a more accurate estimation of the gradient  $\nabla_{\delta} J_{\delta}$  and thus become a stronger adversary.

## 6. Conclusion

In this work, for evaluating the robustness of existing Embodied Vision Navigation methods, we consider a reasonable setting where the adversary adds a constant perturbation to every observation. We first extend UAP framework under MDP setting and propose  $\delta$ -MDP to formalize this problem. Based on analyzing properties of  $\delta$ -MDP, we propose two novel Consistent Attack methods, named Reward UAP and Trajectory UAP, by considering the disturbed state-action distribution and the disturbed Q function. In experiments, we effectively and efficiently mislead the agents in Habitat, which indicates the potential risk to apply these embodied agents to the real world. We hope our study can encourage more future work for designing more robust and reliable Embodied Vision Navigation agents.

## References

- [1] K. M. Hermann, F. Hill, S. Green, F. Wang, R. Faulkner, H. Soyer, D. Szepesvari, W. M. Czarnecki, M. Jaderberg, D. Teplyashin, et al., Grounded language learning in a simulated 3d world, arXiv preprint arXiv:1706.06551.
- [2] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, et al., On evaluation of embodied navigation agents, arXiv preprint arXiv:1807.06757.
- [3] D. S. Chaplot, D. P. Gandhi, A. Gupta, R. R. Salakhutdinov, Object goal navigation using goal-oriented semantic exploration, *Advances in Neural Information Processing Systems* 33.
- [4] X. Ye, Y. Yang, Hierarchical and partially observable goal-driven policy learning with goals relational graph, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14101–14110.
- [5] B. Biggio, I. Corona, D. Maiorca, B. Nelson, P. Laskov, G. Giacinto, F. Roli, Evasion attacks against machine learning at test time, in: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2013, pp. 387–402.
- [6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199.
- [7] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572.
- [8] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal adversarial perturbations, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765–1773.
- [9] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, P. Abbeel, Adversarial attacks on neural network policies, arXiv preprint arXiv:1702.02284.
- [10] H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. Boning, C.-J. Hsieh, Robust deep reinforcement learning against adversarial perturbations on state observations, *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020) 21024–21037.
- [11] V. Khurikov, I. Oseledets, Art of singular vectors and universal adversarial perturbations, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8562–8570.
- [12] C. Zhang, P. Benz, T. Imtiaz, I. S. Kweon, Understanding adversarial examples from the mutual influence of images and perturbations, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14521–14530.
- [13] 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.
- [14] C. Beattie, J. Z. Leibo, D. Teplyashin, T. Ward, M. Wainwright, H. Küttler, A. Lefrancq, S. Green, V. Valdés, A. Sadik, et al., Deepmind lab, arXiv preprint arXiv:1612.03801.
- [15] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, A. Farhadi, Ai2-thor: An interactive 3d environment for visual ai, arXiv preprint arXiv:1712.05474.
- [16] Z. Hu, S. Huang, X. Zhu, F. Sun, B. Zhang, X. Hu, Adversarial texture for fooling person detectors in the physical world, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13307–13316.
- [17] Y. Hua, S. Ge, X. Gao, X. Jin, D. Zeng, Defending against adversarial examples via soft decision trees embedding, *Proceedings of the 27th ACM International Conference on Multimedia*.
- [18] Y. Yang, Y. Zhuang, Y. Pan, Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies, *Frontiers of Information Technology & Electronic Engineering* 22 (12) (2021) 1551–1558.
- [19] S. Hu, X. Liu, Y. Zhang, M. Li, L. Y. Zhang, H. Jin, L. Wu, Protecting facial privacy: Generating adversarial identity masks via style-robust makeup transfer, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 14994–15003.
- [20] X. Yang, Y. Dong, T. Pang, H. Su, J. Zhu, Y. Chen, H. W. Xue, Towards face encryption by generating adversarial identity masks, 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2020) 3877–3887.
- [21] A. Gandhi, S. Jain, Adversarial perturbations fool deepfake detectors, 2020 International Joint Conference on Neural Networks (IJCNN) (2020) 1–8.
- [22] P. Neekhar, S. S. Hussain, M. Jere, F. Koushanfar, J. McAuley, Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples, 2021 IEEE Winter Conference on Applications of Computer Vision (WACV) (2020) 3347–3356.
- [23] D. Zhang, C. Li, F. Lin, D. Zeng, S. Ge, Detecting deepfake videos with temporal dropout 3dcnn, in: *International Joint Conference on Artificial Intelligence*, 2021.
- [24] C. Zhang, P. Benz, C. Lin, A. Karjauv, J. Wu, I. S. Kweon, A survey on universal adversarial attack, arXiv preprint arXiv:2103.01498.
- [25] Y. Qiaoben, C. Ying, X. Zhou, H. Su, J. Zhu, B. Zhang, Understanding adversarial attacks on observations in deep reinforcement learning, arXiv preprint arXiv:2106.15860.
- [26] A. Gupta, S. Savarese, S. Ganguli, L. Fei-Fei, Embodied intelligence via learning and evolution, arXiv preprint arXiv:2102.02202.
- [27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347.
- [28] R. S. Sutton, A. G. Barto, *Reinforcement learning: An introduction*, MIT press, 2018.
- [29] R. S. Sutton, D. A. McAllester, S. P. Singh, Y. Mansour, et al., Policy gradient methods for reinforcement learning with function approximation., in: *NIPs*, Vol. 99, Citeseer, 1999, pp. 1057–1063.
- [30] C. Ying, X. Zhou, H. Su, D. Yan, N. Chen, J. Zhu, Towards safe reinforcement learning via constraining conditional value-at-risk, arXiv preprint arXiv:2206.04436.
- [31] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, S. Savarese, Gibson env: Real-world perception for embodied agents, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9068–9079.
- [32] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, A. Zeng, Y. Zhang, Matterport3d: Learning from rgb-d data in indoor environments, in: *2017 International Conference on 3D Vision (3DV)*, IEEE Computer Society, 2017, pp. 667–676.