Spatiotemporal Attacks for Embodied Agents

Aishan Liu[1], Tairan Huang[1], Xianglong Liu[1,2⋆], Yitao Xu[1], Yuqing Ma[1],
Xinyun Chen[3], Stephen J. Maybank[4], and Dacheng Tao[5]

[1] State Key Laboratory of Software Development Environment,
Beihang University, China
[2] Beijing Advanced Innovation Center for Big Data-Based Precision Medicine,
Beihang University, China
[3] UC Berkeley, USA
[4] Birkbeck, University of London, UK
[5] UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering,
The University of Sydney, Australia

Abstract. Adversarial attacks are valuable for providing insights into the blind-spots of deep learning models and help improve their robustness. Existing work on adversarial attacks have mainly focused on static scenes; however, it remains unclear whether such attacks are effective against embodied agents, which could navigate and interact with a dynamic environment. In this work, we take the first step to study adversarial attacks for embodied agents. In particular, we generate spatiotemporal perturbations to form 3D adversarial examples, which exploit the interaction history in both the temporal and spatial dimensions. Regarding the temporal dimension, since agents make predictions based on historical observations, we develop a trajectory attention module to explore scene view contributions, which further help localize 3D objects appeared with highest stimuli. By conciliating with clues from the temporal dimension, along the spatial dimension, we adversarially perturb the physical properties (*e.g.*, texture and 3D shape) of the contextual objects that appeared in the most important scene views. Extensive experiments on the EQA-v1 dataset for several embodied tasks in both the white-box and black-box settings have been conducted, which demonstrate that our perturbations have strong attack and generalization abilities. [§]

Keywords: Embodied Agents, Spatiotemporal Perturbations, 3D Adversarial Examples

1 Introduction

Deep learning has demonstrated remarkable performance in a wide spectrum of areas [22, 28, 34], but it is vulnerable to adversarial examples [35, 14, 7]. The small perturbations are imperceptible to human but easily misleading deep neural networks (DNNs), thereby bringing potential security threats to deep learning

[⋆] Corresponding author. Email: xlliu@nlsde.buaa.edu.cn

[§] Our code can be found at https://github.com/liuaishan/SpatiotemporalAttack.

applications [30, 24, 25]. Though challenging deep learning, adversarial examples are valuable for understanding the behaviors of DNNs, which could provide insights into the weakness and help improve the robustness [43]. Over the last few years, significant efforts have been made to explore model robustness to the adversarial noises using *adversarial attacks* in the static and non-interactive domain, *e.g.*, 2D images [14, 2, 11] or static 3D scenes [42, 26, 38].

With great breakthroughs in multimodal techniques and virtual environments, embodied task has been introduced to further foster and measure the agent perceptual ability. An agent must intelligently navigate a simulated environment to achieve specific goals through egocentric vision [8, 9, 41, 15]. For example, an agent is spawned in a random location within an environment to answer questions such as "*What is the color of the car?*". Das *et al.* [8] first introduced the embodied question answering (EQA) problem and proposed a model consisting of a hierarchical navigation module and a question answering module. Concurrently, Gordon *et al.* [15] studied the EQA task in an interactive environment named AI2-THOR [20]. Recently, several studies have been proposed to improve agent performance using different frameworks [9] and point cloud perception [37]. Similar to EQA, embodied vision recognition (EVR) [40] is an embodied task, in which an agent instantiated close to an occluded target object to perform visual object recognition.

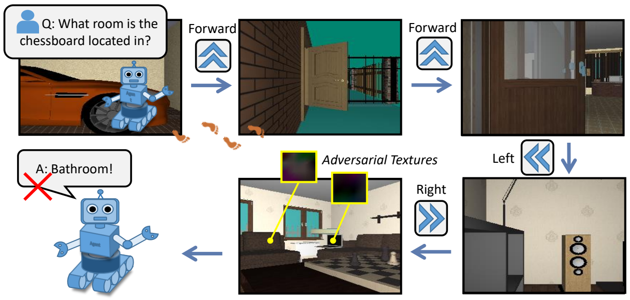


Fig. 1. Embodied agents must navigate the environment through egocentric views to answer given questions. By adversarially perturbing the physical properties of 3D objects using our spatiotemporal perturbations, the agent gives the wrong answer (the correct answer is "living room") to the question. The contextual objects perturbed are: sofa and laptop.

In contrast to static tasks, embodied agents are free to move to different locations and interact with the dynamic environment. Rather than solely using a one-shot image, embodied agents observe 3D objects from different views and make predictions based on historical observations (trajectory). Current adversarial attacks mainly focused on the static scenes and ignored the information from the temporal dimension. However, since agents utilize contextual information to make decisions (*i.e.*, answer questions), only considering a single image or an object appeared in one scene view may not be sufficient to generate strong adversarial attacks for the embodied agent.

In this work, we provide the first study of adversarial attacks for embodied agents in dynamic environments, as demonstrated in Figure 1. By exploiting the interaction history in both the temporal and spatial dimensions, our adversarial attacks generate 3D spatiotemporal perturbations. Regarding the temporal dimension, since agents make predictions based on historical observations, we develop a trajectory attention module to explore scene view contributions, which could help to localize 3D objects that appeared with highest stimuli for agents' predictions. Coupled with clues from the temporal dimension, along the spatial dimension, we adversarially perturb the physical properties (*e.g.*, 3D shape, and texture) of the contextual objects that appeared in the most important scene views. Currently, most embodied agents input 2D images transformed and processed from 3D scenes by undifferentiable renderers. To apply the attack using a gradient-based strategy, we replace the undifferentiable renderer with a differentiable one by introducing a neural renderer [19].

To evaluate the effectiveness of our spatiotemporal attacks, we conduct extensive experiments in both the white-box and black-box settings using different models. We first demonstrate that our generated 3D adversarial examples are able to attack the state-of-the-art embodied agent models and significantly outperform other 3D adversarial attack methods. Also, our adversarial perturbations can be transferred to attack the black-box renderer using non-differentiable operations, indicating the applicability of our attack strategy, and the potential of extending it to the physical world. We also provide a discussion of adversarial training using our generated attacks, and a perceptual study indicating that contrary to the human vision system, current embodied agents are mostly more sensitive to object textures rather than shapes, which sheds some light on bridging the gap between human perception and embodied perception.

2 Related Work

Adversarial examples or perturbations are intentionally designed inputs to mislead deep neural networks [35]. Most existing studies address the static scene including 2D images and static 3D scenes.

In the 2D image domain, Szegedy *et al.* [35] first introduced adversarial examples and used the L-BFGS method to generate them. By leveraging the gradients of the target model, Goodfellow *et al.* [14] proposed the Fast Gradient Sign Method (FGSM) which could generate adversarial examples quickly. In addition, Mopuri *et al.* [29] proposed a novel approach to generate universal perturbations for DNNs for object recognition tasks. These methods add perturbations on 2D image pixels rather than 3D objects and fail to attack the embodied agents.

Some recent work study adversarial attacks in the static 3D domain. A line of work [38, 42, 26] used differentiable renderers to replace the undifferentiable one, and perform attacks through gradient-based strategies. They mainly manipulated object shapes and textures in 3D visual recognition tasks. On the other hand, Zhang *et al.* [44] learned a camouflage pattern to hide vehicles from being detected by detectors using an approximation function. Adversarial patches [5, 24] have been studied to perform real-world 3D adversarial attacks. In particular, Liu *et al.* [24] proposed the PS-GAN framework to generate scrawl-like adversarial patches to fool autonomous-driving systems. However, all these attacks mainly considered the static scenes and ignored the temporal information. Our evaluation demonstrates that by incorporating both spatial and temporal information, our spatiotemporal attacks are more effective for embodied tasks.

Another line of work studies adversarial attacks against reinforcement learning agents [13, 21, 18, 31, 23]. These works mainly consider adversarial attacks against reinforcement learning models trained for standard game environments, where the model input only includes the visual observation. For example, most of existing work focuses on single-agent tasks such as Atari [4], while Gleave *et al.* [13] studied adversarial attacks in multi-agent environments. Different from prior work, we focus on tasks related to embodied agents (i.e., EQA and EVR), with richer input features including both vision and language components.

3 Adversarial Attacks for the Embodiment

The embodiment hypothesis is the idea that intelligence emerges in the interaction of an agent with an environment and as a result of sensorimotor activity [33, 8]. To achieve specific goals, embodied agents are required to navigate and interact with the dynamic environment through egocentric vision. For example, in the EQA task, an agent is spawned at a random location in a 3D dynamic environment to answer given questions through navigation and interaction.

3.1 Motivations

Though showing promising results in the virtual environment, the agent robustness is challenged by the emergence of adversarial examples. Most of the agents are built upon deep learning models which have been proved to be weak in the adversarial setting [35, 14]. By performing adversarial attacks to the embodiment, an adversary could manipulate the embodied agents and force them to execute unexpected actions. Obviously, it would pose potential security threats to agents in both the digital and physical world.

From another point of view, adversarial attacks for the embodiment are also beneficial to understand agents' behaviors. As black-box models, most deep-learning-based agents are difficult to interpret. Thus, adversarial attacks provide us with a new way to explore model weakness and blind-spots, which are valuable to understand their behaviors in the adversarial setting. Further, we can improve model robustness and build stronger agents against noises.

3.2 Problem Definition

In this paper, we use 3D adversarial perturbations (adversarial examples) to attack embodied agents in a dynamic environment.

In a **static scenario**, given a deep neural network $\mathbb{F}_\theta$ and an input image $\mathbf{I}$ with ground truth label y, an adversarial example $\mathbf{I}^{adv}$ is the input that makes the model conducted the wrong label

$$\mathbb{F}_\theta(\mathbf{I}^{adv}) \neq y \quad s.t. \quad \|\mathbf{I} - \mathbf{I}^{adv}\| < \epsilon,$$

where $\|\cdot\|$ is a distance metric to quantify the distance between the two inputs $\mathbf{I}$ and $\mathbf{I}^{adv}$ sufficiently small.

For the **embodiment**, an agent navigates the environment to fulfil goals and observe 3D objects in different time steps t. The input image $\mathbf{I}_t$ at time step t for an agent is the rendered result of a 3D object from a renderer $\mathcal{R}$ by $\mathbf{I}_t = \mathcal{R}(\mathbf{x}, \mathbf{c}_t)$. $\mathbf{x}$ is the corresponding 3D object and $\mathbf{c}_t$ denotes conditions at t (*e.g.*, camera views, illumination, *etc.*). To attack the embodiment, we need to consider the agent trajectory in temporal dimension and choose objects to perturb in the 3D spatial space. In other words, we generate adversarial 3D object $\mathbf{x}^{adv}$ by perturbing its physical properties at multiple time steps. The rendered image set $\{\mathbf{I}_1, ..., \mathbf{I}_N\}$ is able to fool the agent $\mathbb{F}_\theta$:

$$\mathbb{F}_\theta(\mathcal{R}(\mathbf{x}_t^{adv}, \mathbf{c}_t)) \neq y \quad s.t. \quad \|\mathbf{x}_t - \mathbf{x}_t^{adv}\| < \epsilon,$$

where t belongs to a time step set we considered.

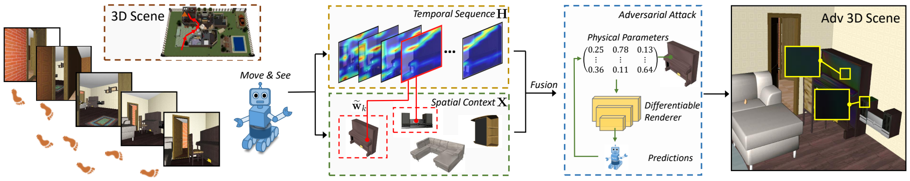


Fig. 2. Our framework exploits interaction histories from both the temporal and the spatial dimension. In the temporal dimension, we develop a trajectory attention module to explore scene view contributions. Thus, important scene views are extracted to help localize 3D objects that appeared with highest stimuli for agents predictions. By conciliating with clues from the temporal dimension, along the spatial dimension, we adversarially perturb the 3D properties (*e.g.*, 3D shape, and texture) of the contextual objects appeared in the most important scene views.

4 Spatiotemporal Attack Framework

In this section, we illustrate our framework to generate 3D adversarial perturbations for embodied agents in the dynamic environment. In Figure 2, we present an overview of our attack approach, which incorporates history interactions from both the temporal and spatial dimensions.

Motivated by the fact that agents make predictions based on historical scene views (trajectory), we attack the 3D objects appeared in scene views containing the highest stimuli to the agent's prediction. In the temporal dimension, we develop a trajectory attention module $\mathcal{A}$ to explore scene view contributions, which directly calculates the contribution weight for each time step scene view $\{\mathbf{I}_1, ..., \mathbf{I}_N\}$ to the agent prediction $\mathbb{F}_\theta$. Given a N-step trajectory, the K most important historical scene views $\mathbf{S}$ are selected by $\mathcal{A}$ to help localize 3D objects that appeared with highest stimuli.

Meanwhile, rather than solely depending on single objects, humans always collect discriminative contextual information when making predictions. By conciliating with clues from the temporal dimension, along the spatial dimension, we adversarially perturb the physical properties $\boldsymbol{\phi}$ of multiple 3D contextual objects $\mathbf{X}$ appeared in the most important scene views. Moreover, to attack physical properties (*i.e.*, 3D shapes and textures), we also employ a differentiable renderer $\mathbb{R}_\delta$ to use the gradient-based attacks.

Thus, by coupling both temporal and spatial information, our framework generates spatiotemporal perturbations to form 3D adversarial examples, which could perform adversarial attacks for the embodiment.

4.1 Temporal Attention Stimulus

To achieve specific goals, embodied agents are required to navigate the environment and make decisions based on the historical observations. Conventional vision tasks, *e.g.*, classification, mainly base on one-shot observation in static images. In contrast, we should consider historical information (trajectory) such as last N historical scene views observed by the agent $\mathbf{H} = \{\mathbf{I}_{t-N}, \mathbf{I}_{t-N+1}, ..., \mathbf{I}_{t-1}\}$, and adversarially perturb the 3D objects that appeared in them. Thus, we can formulate the attack loss:

$$\mathcal{L}_{adv}(\mathbf{H}, y; \mathbb{F}_\theta) = \mathrm{P}(y|\mathbf{H}), \tag{1}$$

where $\mathrm{P}(\cdot|\cdot)$ denotes the prediction probability of the model, and y indicates the ground truth label (*i.e.*, correct answer, object class or action *w.r.t.* question answering, visual recognition and navigation, respectively). To attack agents, the equation above aims to decrease the confidence of the correct class.

There is extensive biological evidence that efficient perception requires both specialized visual sensing and a mechanism to prioritize stimuli, *i.e.*, visual attention. Agents move their eyes towards a specific location or focus on relevant locations to make predictions by prioritizing different scene views [6]. To improve attack abilities, we must design a visual attention module that selects a suitable set of visual features (historical scene views) to perform attack. Inspired by [32], given scene views $\mathbf{H}$, we first compute the gradient of target class y *w.r.t.* normalized feature maps $\mathbf{Z}$ of a specified layer. These gradients flowing back are global average pooled to obtain weight $\mathbf{w}_t$ for the t-th scene view:

$$\mathbf{w}_t = \max(0, \sum_{n=1}^{r} \frac{1}{u \times v} \sum_{j=1}^{v} \sum_{i=1}^{u} \frac{\partial \mathrm{P}(y|\mathbf{H})}{\partial \mathbf{Z}_{i,j}^{n}}), \tag{2}$$

where $u \times v$ represents the size of the feature map, and r indicates total feature map numbers in a specified layer. Then, We normalize each weight according to their mean vector $\boldsymbol{\mu}$ and variance vector $\boldsymbol{\sigma}$:

$$\overline{\mathbf{w}}_t = \frac{\mathbf{w}_t - \boldsymbol{\mu}}{\boldsymbol{\sigma}^2 + \epsilon}, \tag{3}$$

Thus, our trajectory attention module calculates the contribution of each scene view in the trajectory $\mathbf{H}$ towards the model decision for class y:

$$\mathcal{A}(\mathbf{H}, y; \mathbb{F}_\theta) = \langle \overline{\mathbf{w}}_1, ..., \overline{\mathbf{w}}_N \rangle. \tag{4}$$

The weights directly reflect the contribution of observed views at different time steps in the trajectory. Thus, we can further adversarially perturb the 3D objects that appeared in those scene views containing higher weights to execute a stronger attack.

4.2 Spatially Contextual Perturbations

Adversarial attacks in the static scene usually manipulate pixel values in the static image or different frames. In contrast, adversarial attacks for the embodiment require us to perturb the physical properties of 3D objects. Simply, we could randomly choose an object appeared in the most important scene views based on the attention weights to perform attacks. However, when humans look at an object, they always collect a discriminative context for that object [12]. In other words, we concentrate on that object while simultaneously being aware of its surroundings and context. The contextual information enables us to perform much stronger adversarial attacks. As shown in Figure 1, when asking "*What room is the chessboard located in?*", it is better to perturb contextual objects rather than only the target object "chessboard". To answer the question, agent relied on contextual objects (*e.g.*, sofa, laptop, *etc*), that convey critical factors and key features about the answer "living room".

Coupled with the clues from the temporal dimension, we further perturb the 3D contextual objects appeared in the K most important views. Specifically, given K most important scene views selected by our trajectory attention module $\mathbf{S} = \{\mathbf{S}_1, ..., \mathbf{S}_K\}$, we perturb M 3D objects $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_M\}$ appeared in $\mathbf{S}$. Thus, the adversarial attack loss can be formalized as:

$$\mathcal{L}_{adv}(\mathbf{X}, y; \mathbb{F}_\theta, \mathbb{R}_\delta) = \mathrm{P}(y | \mathbf{S}, \mathbb{R}_\delta(\mathbf{X}, \mathbf{c})). \tag{5}$$

Let $\boldsymbol{\phi_m}$ be the 3D physical parameters of object $\mathbf{x}_m$ (*e.g.*, texture, shape, *etc*). With the contribution weight $\overline{\mathbf{w}}$ for the K most important scene views, we add the following perturbation to $\boldsymbol{\phi_m}$:

$$\begin{aligned} \boldsymbol{\Delta\phi_m} = \sum_{k=1}^{K} \mathbb{1}(\mathbf{x}_m \in \mathit{\Phi}(\mathbf{S}_k)) \cdot \overline{\mathbf{w}}_k \cdot \\ \nabla_{\boldsymbol{\phi_m}} \mathcal{L}_{adv}(\mathbf{x}_m, y; \mathbb{F}_\theta, \mathbb{R}_\delta), \end{aligned} \tag{6}$$

where $\mathit{\Phi}(\cdot)$ extracts the objects appeared in scene views.

4.3 Optimization Formulations

Based on the above discussion, we generate 3D adversarial perturbations using the optimization formulation:

$$\begin{aligned}\mathcal{L}(\mathbf{X};\mathbb{F}_\theta,\mathbb{R}_\delta) = \mathrm{E}_{\mathbf{c}\sim\mathbf{C}}\Big[&\mathcal{L}_{adv}(\mathbf{X},y;\mathbb{F}_\theta,\mathbb{R}_\delta,\mathbf{c})+\\ &\lambda\cdot\mathcal{L}_{per}(\mathbf{X},\mathbf{X}^{adv};\mathbb{R}_\delta,\mathbf{c})\Big],\end{aligned} \tag{7}$$

where we append the adversarial attack loss with a perceptual loss:

$$\mathcal{L}_{per}(\mathbf{x},\mathbf{x}_{adv};\mathbb{R}_\delta,\mathbf{c}) = ||\mathbb{R}_\delta(\mathbf{x},\mathbf{c}) - \mathbb{R}_\delta(\mathbf{x}_{adv},\mathbf{c})||, \tag{8}$$

which constrains the magnitude of the total noises added to produce a visually imperceptible perturbation. $\mathbf{C}$ represents different conditions (*e.g.*, camera views, illumination, *etc.*) and λ balances the contribution of each part.

Recent studies have highlighted that adversarial perturbations are ineffective to different transformations and environmental conditions (*e.g.*, illuminations, rotations, *etc*). In the dynamic environment, the viewing angles and environmental conditions change frequently. Thus, we further introduce the idea of *expectation of transformations* [3] to enhance the attack success rate of our perturbations as shown in the expectation of different conditions $\mathbf{C}$ in Eqn (7). Specifically, for each object to attack, we select five positional views one meter away with an azimuth angle uniformly ranging from [0°, 180°] to optimize the overall loss.

It is intuitive to directly place constraints on physical parameters such as the contour or color range of object surfaces. However, one potential disadvantage is that different physical parameters have different units and ranges. Therefore, we constrain the RGB intensity changes in the 2D image space after the rendering process to keep the consistency of the change of different parameters (*i.e.*, shape or texture).

5 Experiments

In this section, we evaluate the effectiveness of our 3D spatiotemporal adversarial attacks against agents in different settings for different embodied tasks. We also provide a discussion of defense with adversarial training, and an ablation study of how different design choices affect the attack performance.

5.1 Experimental Setting

For both EQA and EVR tasks, we use the EQA-v1 dataset [8], a visual question answering dataset grounded in the simulated environment. It contains 648 environments with 7,190 questions for training, 68 environments with 862 questions for validation, and 58 environments with 933 questions for testing. It divides the task into T_{-10}, T_{-30}, T_{-50} by steps from the starting point to the target. We restrict the adversarial perturbations to be bounded by 32-pixel values per frame of size 224×224, in terms of ℓ_∞ norm.

5.2 Evaluation Metrics

To measure agent performance, we use evaluation metrics as in [8, 37, 9]:

- top-1 accuracy: whether the agent's prediction matches ground truth (↑ is better);

- d_T: the distance to the target object at navigation termination (↓ is better);

- d_Δ: change in distance to target from initial to the final position (↑ is better);

- d_{min}: the smallest distance to the target at any point in the episode (↓ is better);

Note that the goal of adversarial attacks is compromising the performance of the embodied agents, *i.e.*, making worse values of the evaluation metrics above.

5.3 Implementation Details

We use the SGD optimizer for adversarial perturbation generation, with momentum 0.9, weight decay 10^{-4}, and a maximum of 60 iterations. For the hyper-parameters of our framework, we set λ to 1, K to 3, and M as the numbers of all contextual objects observed in these frames. For EQA, we generate adversarial perturbations using PACMAN-RL+Q [8] as the target model (we use "PACMAN" for simplicity), and we use Embodied Mask R-CNN [40] as the target model for EVR. In our evaluation, we will demonstrate that the attacks generated against one model could transfer to different models.

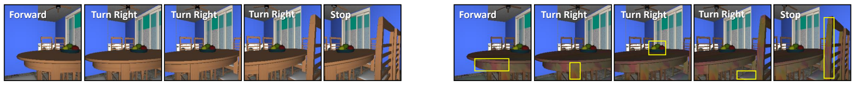
For both EQA and EVR, unless otherwise specified, we generate adversarial perturbations on texture only, *i.e.*, in Equation 6, we only update the parameters corresponding to texture, because it is more suitable for future extension to physical attacks in the real 3D environment. In the supplementary material, we also provide a comparison of adversarial perturbations on shapes, where we demonstrate that with the same constraint of perturbation magnitude, texture attacks achieve a higher attack success rate.

5.4 Attack via a Differentiable Renderer

we first provide the quantitative and qualitative results of our 3D adversarial perturbations on EQA and EVR through our differentiable renderer. For EQA, besides PACMAN, we also evaluate the transferability of our attacks using the following models: (1) NAV-GRU, an agent using GRU instead of LSTM in navigation [37]; (2) NAV-React, an agent without memory and fails to use historical information [8]; and (3) VIS-VGG, an agent using VGG to encode visual information [9]. For EVR, we evaluate the white-box attacks on Embodied Mask R-CNN. As most of the embodied tasks can be directly divided into navigation and problem-solving stages, *i.e.*, question answering or visual recognition, we attack each of these stages. We compare our spatiotemporal attacks to MeshAdv [38] and Zeng *et al.* [42], both of which are designed for the static 3D environment, and thus do not leverage the temporal information.

Table 1. Quantitative evaluation of agent performance on EQA task using different models in clean and adversarial settings (ours, MeshAdv [38] and Zeng *et al.* [42]). Note that the goal of attacks is to achieve a worse performance. We observe that our spatiotemporal attacks outperform the static 3D attack algorithms, achieving higher d_T and d_{min} as well as lower d_Δ and accuracy.

		Navigation									**QA**		
		$\boldsymbol{d_T}$ (↓ is better)			$\boldsymbol{d_\Delta}$ (↑ is better)			$\boldsymbol{d_{min}}$ (↓ is better)			**accuracy** (↑ is better)		
		T_{-10}	T_{-30}	T_{-50}	T_{-10}	T_{-30}	T_{-50}	T_{-10}	T_{-30}	T_{-50}	T_{-10}	T_{-30}	T_{-50}
PACMAN	Clean	1.05	2.43	3.82	0.10	0.45	1.86	0.26	0.97	1.99	50.23%	44.19%	39.94%
	MeshAdv	1.06	2.44	3.90	0.09	0.44	1.78	0.31	1.17	2.33	16.07%	15.34%	13.11%
	Zeng *et al.*	**1.07**	2.46	3.88	**0.08**	0.42	1.80	0.42	1.37	2.43	17.15%	16.38%	14.32%
	Ours	1.06	**3.19**	**5.58**	0.09	**-0.39**	**0.10**	**0.90**	**2.47**	**5.33**	**6.17%**	**4.26%**	**3.42%**
NAV-GRU	Clean	1.03	2.47	3.92	0.12	0.41	1.76	0.34	1.02	2.07	48.97%	43.72%	38.26%
	MeshAdv	1.07	2.50	3.92	0.08	0.38	1.76	0.38	1.28	2.48	17.22%	17.01%	14.25%
	Zeng *et al.*	1.09	2.47	3.87	0.06	0.41	1.81	0.36	1.38	2.51	17.14%	16.56%	15.11%
	Ours	**1.13**	**2.96**	**5.42**	**0.02**	**-0.08**	**0.26**	**0.96**	**2.58**	**4.98**	**8.41%**	**6.23%**	**5.15%**
NAV-React	Clean	**1.37**	2.75	4.17	**-0.22**	0.13	1.51	0.31	0.99	2.08	48.19%	43.73%	37.62%
	MeshAdv	1.05	2.79	4.25	0.10	0.09	1.43	0.32	1.29	2.47	15.36%	14.78%	11.29%
	Zeng *et al.*	1.10	2.79	4.21	0.05	0.09	1.47	0.36	1.59	2.32	15.21%	14.13%	13.29%
	Ours	1.22	**2.85**	**5.70**	-0.07	**0.03**	**-0.02**	**1.06**	**2.59**	**5.47**	**8.26%**	**5.25%**	**5.39%**
VIS-VGG	Clean	1.02	2.38	3.67	0.13	0.50	2.01	0.38	1.05	2.26	50.16%	45.81%	37.84%
	MeshAdv	1.06	2.41	3.67	0.09	0.47	2.01	0.40	1.11	2.52	16.69%	15.24%	15.21%
	Zeng *et al.*	1.06	2.43	3.70	0.09	0.45	1.98	0.44	1.41	2.44	15.13%	14.84%	14.21%
	Ours	**1.18**	**2.83**	**5.62**	**-0.03**	**0.05**	**0.06**	**1.04**	**2.01**	**5.12**	**6.33%**	**4.84%**	**4.29%**



(a) Clean Scene

(b) Adversarial Scene

Fig. 3. Given the question "*What is next to the fruit bowl in the living room?*", we show the last 5 views of the agent for EQA in the same scene with and without adversarial perturbations. The contextual objects perturbed including table, chairs and fruit bowel. The agent gives wrong answers "television" to the question (ground truth: chair) after seeing adversarial textures in subfigure (b). Yellow boxes show the perturbed texture.

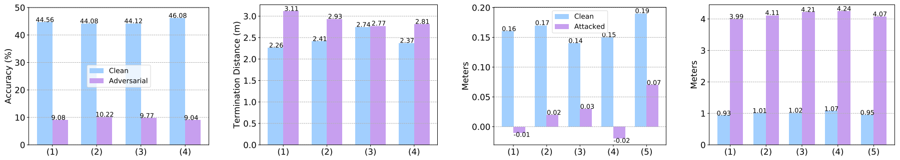
For **question answering** and **visual recognition**, we generate 3D adversarial perturbations using our proposed method on the test set and evaluate agent performance throughout the entire process, *i.e.*, the agent is randomly placed and navigate to answer a question or recognize an object. As shown in Table 1, for white-box attacks, there is a significant drop in question answering accuracy from 50.23%, 44.19% and 39.94% to 6.17%, 4.26% and 3.42% for tasks with 10, 30, and 50 steps, respectively. Further, the visual recognition accuracy drastically decreases from 89.91% to 18.32%. The black-box attacks also result in a large drop in accuracy. The visualization of the last five steps before the agent's decision for EQA is shown in Figure 3. Our perturbations are unambiguous for human prediction but misleading to the agent.

For **navigation**, we generate 3D adversarial perturbations that intentionally stop the agent, *i.e.*, make the agent predict *Stop* during the navigation process. As shown in Table 1, for both white-box and black-box attacks, the values of

d_T and d_{min} significantly increase compared to the clean environment when adding our perturbations, especially for long-distance tasks, *i.e.*, T_{-50}. Further, the values of d_Δ decreases to around 0 after attack, which reveals that agents make a small number of movements to the destination. Also, some d_Δ even become negative, showing that the agent is moving away from the target.

To understand the transferability of attacks, we study attention similarities between models. The results can be found in the Supplementary Material.

In a word, our generated 3D adversarial perturbations achieve strong attack performance in both the white-box and black-box settings for navigation and problem-solving in the embodied environment.



(a) Accuracy (b) d_T (c) d_Δ (d) d_{min}

Fig. 4. Method (1) to (4) represents PACMAN, NAV-GRU, NAV-React and VIS-VGG, respectively. Our framework generates adversarial perturbations with strong transferabilities to non-differentiable renderers.

5.5 Transfer Attack onto a non-differentiable Renderer

Our proposed framework aims to adversarially attack $\mathbb{F}_\theta(\mathbb{R}_\delta(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n))$ by end-to-end gradient-based optimization. In this section, we further examine the potential of our framework in practice, where no assumptions about the non-differentiable renderer are given. By enabling interreflection and rich illumination, non-differentiable renderers can render images at high computational cost, so that the rendered 2D image is more likely to be an estimate of real-world physics. Thus, these experiments are effective to illustrate the transferability of generated adversarial perturbations and their potential in practical scenarios.

Specifically, we use the original non-differentiable renderer $\mathcal{R}$ for EQA-V1, which is implemented on OpenGL with unknown parameters, as the black-box renderer. We first generate 3D adversarial perturbations using our neural renderer $\mathbb{R}_\delta$, then save the perturbed scenes. We evaluate agent performance through the non-differentiable renderer $\mathcal{R}$ on those perturbed scenes to test the transferability of our adversarial perturbations.

As shown in Figure 4, our spatiotemporal attacks can easily be transferred to a black-box renderer. However, our generated adversarial perturbations are less effective at attacking the non-differentiable renderer compared to the neural renderer. Many recent studies have reported that attacking the 3D space is much more difficult than attacking the image space [42, 38]. Further, we believe there are three other reasons for this phenomenon: (1) To generate attacks for the non-differentiable renderer, we first generate 3D adversarial perturbations using a differentiable renderer, then save the perturbed scenes into OBJ, MTL, and JPG files (the required files of the non-differentiable renderer to render a 3D scene) and feed them to the renderer. The information loss comes from the JPG compression process, which may decrease the attack success rate. (2) The parameter difference between $\mathbb{R}_\delta$ and $\mathcal{R}$ may causes some minute rendering differences for the same scenarios. As adversarial examples are very sensitive to image transformations [39, 16], the attacking ability is impaired; (3) The adversarial perturbation generated by optimization-based or gradient-based methods fails to obtain strong transferability due to either overfitting or underfitting [10].

5.6 Generalization Ability of the Attack

In this section, we further investigate the generalization ability of our generated adversarial perturbations. Given questions and trajectories, we first perturb the objects and save the scene. Then, loading the same perturbed scene, we ask agents different questions and change their start points to test their performance.

We first use the same perturbations on **different questions** (denoted as "Q"). We fix the object in questions during perturbation generation and test to be the same. For example, we generate the perturbations based on question *"What is the color of the table in the living-room?"* and test the success rate on question *"What is next to the table in the living-room?"*. Moreover, we use the same perturbations to test agents from **different starting points** (*i.e.*, different trajectories, denoted as "T"). We first generate the perturbations and then test them by randomly spawning agents at different starting points (*i.e.*, random rooms and locations) under the same questions. As shown in Table 2, the attacking ability drops a little compared to the baseline attack (generate perturbation and test at the scene with the same questions and starting point, denoted as "Attack") in both setting with higher QA accuracy but still very strong, which indicates the strong generalization ability of our spatiotemporal perturbations.

	QA accuracy		
	T_{10}	T_{30}	T_{50}
Clean	51.42%	42.68%	39.15%
Attack	6.05%	3.98%	3.52%
Q	10.17%	8.13%	7.98%
T	8.19%	7.26%	7.14%

Table 2. Generalization ability experiments. Our 3D perturbations generalize well in settings using different questions and starting points.

5.7 Improving Agent Robustness with Adversarial Training

Given the vulnerability of existing embodied agents with the presence of adversarial attacks, we study defense strategies to improve the agent robustness. In particular, we base our defense on adversarial training [14, 36, 27], where we integrate our generated adversarial examples for model training.

Training. We train 2 PACMAN models augmented with adversarial examples (*i.e.*, we generate 3D adversarial perturbations on object textures, denoted as AT) or Gaussian noises (denoted as GT), respectively. We apply the common adversarial training strategy that adds a fixed number of adversarial examples in each epoch [14, 1], and we defer more details in the supplementary material.

Testing. We create a test set of 110 questions in 5 houses. Following [14, 17], we add different common noises including adversarial perturbations and Gaussian noises. To conduct fair comparisons, adversarial perturbations are generated in the white-box setting (*e.g.*, for our adversarially trained model, we generate adversarial perturbations against it). The results in Table 3 support the fact that training on our adversarial perturbations can improve the agent robustness towards some types of noises (*i.e.*, higher QA accuracy, and lower d_T).

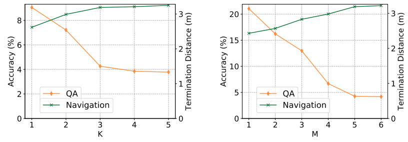
	QA		**Navigation**	
	Adv	Gaussian	Adv	Gaussian
Vanilla	5.67%	22.14%	1.39	1.20
GT	8.49%	32.90%	1.32	1.09
AT	**23.56%**	**38.87%**	**1.17**	**1.01**

Table 3. Agent robustness in scenes with different noises. Adversarial training provides the most robust agent.

5.8 Ablation Study

Next, we present a set of ablation studies to further demonstrate the effectiveness of our proposed strategy through different hyper-parameters K and M, *i.e.*, different numbers of historical scene views and contextual objects considered. All experiments in this section are conducted on T_{-30}.

Historical scene views numbers. As for K, we set K=1,2,3,4,5, with a maximum value of M=5. For a fair comparison, we set the overall magnitude of perturbations to 32/255. As shown in Figure 5 (a), for navigation, we nearly obtain the optimal attack success rate when K=3. The results are similar to the question answering. However, the attack ability does not increase as significantly as that for navigation when increasing K. Obviously, the agent mainly depends on the target object and contextual objects to answer the questions. The contextual objects to be perturbed are quite similar to the increasing number of historical scene views considered.



(a) (b)

Fig. 5. Ablation study with different K and M values in (a) and (b).

Contextual objects numbers. As for M, we set M=1,2,3,4,5,6 and K=3 to evaluate the contribution of the context to adversarial attacks. Similarly, we set the overall magnitude of adversarial perturbations to 32/255 for adversarial attacks with different M values, *i.e.*, perturbations are added onto a single object or distributed to several contextual objects. As shown in Figure 5(b), the attack success rate increases significantly with the increasing of M and converges at around 5. The reason is the maximum number of objects observable in 3 frames is around 5 or 6. Further, by considering the type of questions, we could obtain a deeper understanding about how an agent makes predictions. For questions about location and composition, *e.g.*, "*What room is the <OBJ> located in?*" and "*What is on the <OBJ> in the <ROOM> ?*", the attack success rate using context outperforms single object attack significantly with 4.67% and 28.51%, respectively. However, attacks on color-related questions are only 3.56% and 9.88% after contextual attack and single object attack, respectively. Intuitively, agents rely on different information to solve different types of questions. According to the attention visualization study shown in Figure 6, agents generally utilize clues from contextual objects to answer locational and compositional questions while mainly focus on target objects when predicting their colors.

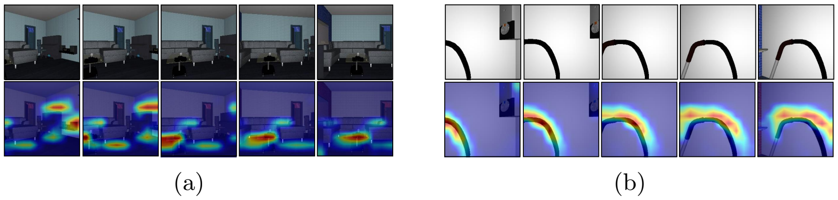


Fig. 6. Visualization of last 5 views and corresponding attention maps. (a) denotes the locational and compositional question, and (b) denotes the color-related question.

6 Conclusion

In this paper, we generate spatiotemporal perturbations to form 3D adversarial examples, which could attack the embodiment. Regarding the temporal dimension, we develop a trajectory attention module to explore scene view contributions, which further help localize 3D objects appeared with highest stimuli. By conciliating with clues from the temporal dimension, along the spatial dimension, we adversarially perturb the physical properties (*e.g.*, texture) of the contextual objects that appeared in the most important scene views. Extensive experiments on the EQA-v1 dataset for several embodied tasks in both the white-box and black-box settings are conducted, which demonstrate that our framework has strong attack and generalization abilities.

Currently, most embodied tasks could only be evaluated in the simulated environment. In the future, we are interested in performing spatiotemporal attacks in real-world scenarios. Using projection or 3D printing, we could bring our perturbations into the real-world to attack a real agent.

Acknowledgement

This work was supported by National Natural Science Foundation of China (61872021, 61690202), Beijing Nova Program of Science and Technology (Z191100001119050), Fundamental Research Funds for Central Universities (YWF-20-BJ-J-646), and ARC FL-170100117.

References

1. Alexey, K., Ian, G., Samy, B.: Adversarial machine learning at scale. In: International Conference on Learning Representations (2017)
2. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420 (2018)
3. Athalye, A., Engstrom, L., Ilyas, A., Kwok, K.: Synthesizing robust adversarial examples. arXiv preprint arXiv:1707.07397 (2017)
4. Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research **47**, 253–279 (2013)
5. Brown, T.B., Mané, D., Roy, A., Abadi, M., Gilmer, J.: Adversarial patch. arXiv preprint arXiv:1712.09665 (2017)
6. Carlone, L., Karaman, S.: Attention and anticipation in fast visual-inertial navigation. IEEE Transactions on Robotics (2018)
7. Chen, W., Zhang, Z., Hu, X., Wu, B.: Boosting decision-based black-box adversarial attacks with random sign flip. In: Proceedings of the European Conference on Computer Vision (2020)
8. Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., Batra, D.: Embodied question answering. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
9. Das, A., Gkioxari, G., Lee, S., Parikh, D., Batra, D.: Neural modular control for embodied question answering. arXiv preprint arXiv:1810.11181 (2018)
10. Dong, Y., Liao, F., Pang, T., Su, H.: Boosting adversarial attacks with momentum. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
11. Gao, L., Zhang, Q., Song, j., Liu, X., Shen, H.: Patch-wise attack for fooling deep neural network. In: European Conference on Computer Vision (2020)
12. Garland-Thomson, R.: Staring: How we look (2009)
13. Gleave, A., Dennis, M., Kant, N., Wild, C., Levine, S., Russell, S.A.: Adversarial policies: Attacking deep reinforcement learning. In: International Conference on Learning Representations (2020)
14. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2014). arXiv preprint arXiv:1412.6572 (2014)
15. Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., Farhadi, A.: Iqa: Visual question answering in interactive environments. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
16. Guo, C., Rana, M., Cisse, M., Van Der Maaten, L.: Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117 (2017)
17. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: International Conference on Learning Representations (2019)
18. Huang, S.H., Papernot, N., Goodfellow, I.J., Duan, Y., Abbeel, P.: Adversarial attacks on neural network policies. arXiv preprint arXiv: 1702.02284 (2017)
19. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
20. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Gordon, D., Zhu, Y., Gupta, A., Farhadi, A.: Ai2-thor: An interactive 3d environment for visual ai. arXiv preprint arXiv:1712.05474 (2017)

21. Kos, J., Song, D.X.: Delving into adversarial attacks on deep policies. arXiv preprint arXiv: 1705.06452 (2017)
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems (2012)
23. Lin, Y.C., Hong, Z.W., Liao, Y.H., Shih, M.L., Liu, M.Y., Sun, M.: Tactics of adversarial attack on deep reinforcement learning agents. In: IJCAI (2017)
24. Liu, A., Liu, X., Fan, J., Ma, Y., Zhang, A., Xie, H., Tao, D.: Perceptual-sensitive gan for generating adversarial patches. In: 33rd AAAI Conference on Artificial Intelligence (2019)
25. Liu, A., Wang, J., Liu, X., Cao, b., Zhang, C., Yu, H.: Bias-based universal adversarial patch attack for automatic check-out. In: European Conference on Computer Vision (2020)
26. Liu, H.T.D., Tao, M., Li, C.L., Nowrouzezahrai, D., Jacobson, A.: Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer (2019)
27. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
28. Mohamed, A.r., Dahl, G.E., Hinton, G.: Acoustic modeling using deep belief networks. IEEE T AUDIO SPEECH (2011)
29. Mopuri, K.R., Ganeshan, A., Radhakrishnan, V.B.: Generalizable data-free objective for crafting universal adversarial perturbations. IEEE T PATTERN ANAL (2018)
30. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against deep learning systems using adversarial examples. arXiv preprint (2016)
31. Pattanaik, A., Tang, Z., Liu, S., Bommannan, G., Chowdhary, G.: Robust deep reinforcement learning with adversarial attacks. In: AAMAS (2018)
32. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: IEEE International Conference on Computer Vision (2017)
33. Smith, L., Gasser, M.: The development of embodied cognition: Six lessons from babies. Artificial life **11**(1-2), 13–29 (2005)
34. Sutskever, I., Vinyals, O., Le, Q.: Sequence to sequence learning with neural networks. NeurIPS (2014)
35. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
36. Tu, Z., Zhang, J., Tao, D.: Theoretical analysis of adversarial learning: A minimax approach. In: Advances in Neural Information Processing Systems (2019)
37. Wijmans, E., Datta, S., Maksymets, O., Das, A., Gkioxari, G., Lee, S., Essa, I., Parikh, D., Batra, D.: Embodied question answering in photorealistic environments with point cloud perception. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
38. Xiao, C., Yang, D., Li, B., Deng, J., Liu, M.: Meshadv: Adversarial meshes for visual recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
39. Xie, C., Wang, J., Zhang, Z., Ren, Z., Yuille, A.: Mitigating adversarial effects through randomization. arXiv preprint arXiv:1711.01991 (2017)
40. Yang, J., Ren, Z., Xu, M., Chen, X., Crandall, D., Parikh, D., Batra, D.: Embodied visual recognition. IEEE International Conference on Computer Vision (2019)

41. Yu, L., Chen, X., Gkioxari, G., Bansal, M., Berg, T.L., Batra, D.: Multi-target embodied question answering. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
42. Zeng, X., Liu, C., Wang, Y.S., Qiu, W., Xie, L., Tai, Y.W., Tang, C.K., Yuille, A.L.: Adversarial attacks beyond the image space. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
43. Zhang, T., Zhu, Z.: Interpreting adversarially trained convolutional neural networks. arXiv preprint arXiv:1905.09797 (2019)
44. Zhang, Y., Foroosh, H., David, P., Gong, B.: Camou: Learning physical vehicle camouflages to adversarially attack detectors in the wild. In: International Conference on Learning Representations (2019)