强化化身主动防御：在对抗性 3D 环境中利用自适应交互实现稳健视觉感知

Xiao Yang, Lingxuan Wu, Lizhong Wang, Chengyang Ying, Hang Su, and Jun Zhu, *Fellow, IEEE*

Abstract—在 3D 环境中，敌对攻击已成为视觉感知系统可靠性的重大威胁，特别是在身份验证和自动驾驶等安全敏感应用中。这些攻击通过利用复杂场景中的漏洞，使用对抗性补丁和 3D 物体来操控深度神经网络（DNN）的预测。现有的防御机制，如对抗性训练和净化，主要采用被动策略来增强鲁棒性。然而，这些方法往往依赖于对敌对策略的预定义假设，限制了它们在动态 3D 环境中的适应性。为了解决这些挑战，我们引入增强体现主动防御（Rein -EAD），这是一种主动防御框架，通过与环境的自适应探索和交互来提高 3D 对抗背景下的感知鲁棒性。通过实现多步目标，平衡即时预测准确性与预测熵最小化，Rein -EAD 优化了跨多步视野的防御策略。此外，Rein -EAD 涉及一种以不确定性为导向的奖励塑造机制，促进了高效的策略更新，从而减少计算开销，并在无需可微分环境的情况下支持实际应用。全面的实验验证了 Rein -EAD 的有效性，在保持不同任务的标准准确性的同时，显著降低了攻击成功率。值得注意的是，Rein -EAD 对未见过的和自适应攻击表现出强大的泛化能力，使其适用于实际的复杂任务，包括三维对象分类、人脸识别和自动驾驶。通过将主动策略学习与具体现场交互相结合，Rein -EAD 确立了一种可扩展且可适应的方法，用于在动态和对抗的三维环境中保护基于 DNN 的感知系统。

Index Terms—Adversarial Robustness, Active Defense, Embodied Learning, Policy Learning

1 介绍

A 对抗性 攻击在 3D 环境中已成为视觉感知系统安全性和可靠性的重大威胁 [?], [?] 。这些攻击利用精心设计的扰动，例如对抗性补丁和 3D 物体，策略性地放置在物理场景中以操纵深度神经网络（DNN）的预测 [?], [?] 。此类漏洞的后果在安全关键领域尤其严重，例如身份验证 [?], [?], [?] 和自动驾驶 [?], [?] ，其中错误的预测可能严重损害系统完整性。因此，确保在对抗条件下的稳健感知对于在现实应用中可靠部署这些系统至关重要。

为了应对这些新出现的威胁，研究人员开发了多种防御策略来增强深度神经网络的鲁棒性。对抗训练 [?], [?], [?] 已经成为一种特别有效的方法 [?] ，其中对抗样本被刻意纳入训练数据以增强模型的韧性。同时，提出了输入预处理技术，如对抗净化，以缓解这些扰动 [?], [?], [?] 。然而，这些方法主要属于被动防御，对未见或自适应攻击 [?], [?] 表现出脆弱，因为它们基于对对手方法的预设。此外，这些策略通常忽略了三维环境中事物的内在物理背景和相关理解，削弱了这些防御在现实物理环境中的有效性。

与被动防御的局限性相比，人类的主动视觉通过迭代优化和错误校正机制，能够轻松检测复杂三维环境中的错位或不一致元素 [?], [?] 。受人类主动视觉的启发，我们近期的开创性工作提出了一种新颖的 具身主动防御 (EAD) 防御框架，该框架融合了一种前瞻性策略网络，以复制这种动态过程 [?] 。EAD 主动将环境信息情境化并利用物体一致性来解决三维环境中未对齐的对抗性补丁。系统通过整合当前和过去的观察，不断优化对场景的理解，形成对环境更全面的表示。这种前瞻性行为使系统能够预测不确定区域并相应调整其动作，从而提高其收集的观察质量。通过协同前瞻性运动和迭代预测，EAD 加强了其对场景的理解，并缓解了对抗性补丁的影响。

尽管具有潜力，当前的 EAD 框架面临几个关键挑战，这些挑战限制了其在现实场景中的有效性和适用性。首先，EAD 的贪心信息探索策略优先考虑的是即时的、单步的信息增益，而不是长期相关性，这导致时间上不一致的行动。这种目光短浅的方法常常使代理重新访问之前探索过的视点，降低了探索效率并增加了错误预测的可能性。此外，EAD 依赖于通过微分环境模型训练主动策略网络，引入了不匹配，特别是在处理不可微的现实世界物理动态时，这限制了框架的实际应用性。此外，通过可微仿真进行学习在计算上是密集的，并且容易出现数值不稳定性，进一步削弱了系统的整体效率。

为了克服现有的局限性并增强防御机制的有效性、适用性和效率，我们提出了加强的 EAD (Rein -EAD) 框架。该框架使代理可以通过试错互动学习来优化长期结果，而不需要可微分的模拟。为了应对探索中的时间不一致性，我们引入了一个积累多步互动的广义目标函数。此目标在多步范围内平衡了预测损失减少与预测熵最小化，使系统能够考虑时间依赖性并优先考虑长期结果而非即时不确定性减少。此外，为了消除可微分约束并确保高效收敛，我们在强化学习中实施了一种面向不确定性的奖励塑形技术。这种方法在每一步提供密集的奖励，引导代理减少感知不确定性并最小化预测误差。通过促进高效更新，这种奖励结构支持即使在复杂环境中也能稳定收敛，提高代理对动态变化的适应性，并在不确定和变化的场景中提高整体性能。最后，为了解决对抗性补丁生成的过拟合和计算负担问题，我们提出了用于生成与攻击无关对抗性补丁的离线对抗性补丁近似（OAPA）。OAPA 系统地表征了来自多种攻击策略的对抗性模式的流形。通过离线提取不同攻击策略中的基本特征，OAPA 在显著降低在线对抗性训练的计算负担的同时，极大提升了在不同攻击中的泛化能力。

• *X. Yang, L. Wu, L. Wang, C. Ying, H. Su and J. Zhu are with Dept. of Comp. Sci. & Tech., Institute for AI, BNRist Center, THBI Lab, Tsinghua-Bosch Joint Center for ML, Tsinghua University, Beijing, China. Email: yangxiao19@tsinghua.org.cn, { wlx23, ycy21 }@mails.tsinghua.edu.cn, wanglizhong99@outlook.com, { suhangss, dcszj }@tsinghua.edu.cn. Xiao Yang and Lingxuan Wu had contributed equally to this work. Corresponding authors: Hang Su; Jun Zhu. Code is available at https://github.com/thu-ml/EmbodiedActiveDefense.*

广泛的实验表明，REIN -EAD 相较于传统的被动防御机制具有显著优势。首先，REIN -EAD 始终优于最先进的防御技术，在各种任务中均能显著降低攻击成功率达 95 %。值得注意的是，REIN -EAD 通过有效利用适合在动态三维环境中检测目标物体的指导性信息，保持甚至提高了标准准确性。其次，与被动方法相比，REIN -EAD 展示了卓越的泛化能力。其与攻击无关的策略使其能够有效防御广泛的对抗性补丁，包括未知和自适应攻击。此外，REIN -EAD 在复杂和真实世界场景中表现出强大的适用性，例如三维物体分类、人脸识别和自动驾驶的物体检测。试错学习范式确保了策略更新的稳定性和效率，使得 REIN -EAD 能够高度适应真实世界任务。

综上所述，我们的贡献如下：

- 首先，我们提出了 REIN -EAD，它将多步累积交互和策略学习整合到一个统一的框架中。该框架通过不确定性导向的奖励塑造优化多步目标，促进时间一致且信息丰富的探索。
- 其次，我们开发了一种对抗者无关的防御策略 OAPA，使得 REIN -EAD 能够防御多种对抗性补丁攻击，包括未见过的和自适应的攻击，而不依赖于对对抗者能力的特定假设。
- 第三，通过大量实验，我们展示了 REIN -EAD 在各种设置中相较于先进的被动防御的卓越效果、针对各种未知和自适应攻击的强大泛化能力，以及适应复杂现实世界场景的能力。

在本节中，我们深入探讨了 3D 环境中对抗性补丁所带来的威胁，并探讨了相应的防御策略。

1.1 对抗性补丁和防御

对抗性补丁最初是为操纵图像的特定区域以误导图像分类器而设计的 [?]，现已显著演变。它们现在能够欺骗包括在三维环境中运行的那些在内的广泛感知模型 [?], [?] [?], [?], [?], [?]。

为了防御这种针对感知模型的对抗补丁攻击，已经提出了多种策略，从经验防御 [?], [?], [?], [?] 到认证防御 [?], [?]。然而，经过关键审查发现，大多数当代防御机制属于我们称之为被动防御的方法。这些方法依赖于从单目观测中获得的信息和预设的对抗策略来减轻基于补丁的威胁。

在被动防御范式中，两种主要方法已获得显著地位。对抗训练 [?], [?], [?] 通过在训练过程中暴露于对抗性样本来增强感知模型，提升监督学习 [?], [?] 和半监督学习 [?] 中对抗性扰动的内在鲁棒性 。或者，对抗净化 [?], [?], [?] 将一个辅助净化器集成到感知流程中。这个净化器首先在观测中识别出对抗性补丁，然后中和或去除这些扰动。修改后的观测随后由模型处理，形成一个两阶段的防御流程，在感知任务开始之前减轻对抗性影响。

虽然被动防御在特定场景中显示出成功，但由于其依赖于静态的、预定义的机制，使其在适应性攻击面前变得脆弱，并限制了其在动态、现实环境中的有效性。相比之下，主动防御策略表现出更具适应性的方式，能够在复杂的 3D 环境中动态地应对不断演变的对抗性威胁。

我们最近的工作引入了一个名为 具身主动防御 (EAD) 的前沿防御框架，该框架采用主动政策网络来模拟这一动态响应过程 [?]。EAD 主动语境化环境信息，并利用对象一致性来解决 3D 环境中的对抗性补丁错位问题，这标志着对抗性防御的一个重大进展。

1.2 具身感知

具身感知范式 [?], [?] 代表了人工智能和计算机视觉领域的一个显著转变。这种方法认为，感知不仅仅是信息接收的被动过程，而是一个主动的、具身的体验，在这个过程中，智能体可以动态地与其环境交互和导航，以优化感知结果并提升任务性能。

具身感知框架展现了显著的多功能性，应用于广泛的计算机视觉任务。在目标检测领域，研究人员利用具身代理主动探索和分析场景，从而开发出更为稳健和具上下文感知的检测系统 [?], [?]。同样地，在 3D 姿态估计领域，具身方法通过允许代理主动寻找最佳的估计视点，实现了更精确和适应性更强的解决方案 [?]。此外，这一范式在推动 3D 场景理解方面也被证明是无价的，具身代理可以在复杂环境中导航以构建全面的空间表示 [?]。

具身感知的力量在于其能够模仿生物视觉系统的主动和探索性特征，使得智能体能够克服静态和被动感知的局限。通过根据环境线索和任务需求动态调整其位置、方向或焦点，具身智能体有可能获取更具信息性和不那么模糊的感官数据，从而提高在不同感知任务中的性能。我们将具身感知与对抗性鲁棒性进行创新整合，为研究开辟了新的方向，这可能导致更具弹性和适应性的感知系统，即使在复杂的对抗性攻击中也能保持高性能。

2 方法论

我们首先在第 2.1 节介绍关于具身主动防御（EAD）的初步内容，该内容利用递归反馈来对抗对抗性补丁。然后，我们在第 2.2 节为 EAD 提供理论分析以作进一步探索。在第 ?? 节，我们提出了结合多步交互和策略学习的 REIN -EAD。它有效增强了防御机制在复杂且真实世界环境中的适应性和弹性。最后，我们在第 2.3 节提供了与攻击者无关的防御措施。

2.1 初步：具身主动防御

考虑一个场景 $x \in \mathcal{X}$ 及其相关的真实标签 $y \in \mathcal{Y}$。感知模型 $f : \mathcal{O} \to \mathcal{Y}$ 旨在基于图像观测 $o_i \in \mathcal{O}$ 预测场景注释 y，其中 o_i 来源于场景 x 并以摄像机的状态为条件 s_i（包括摄像机的位置和视点）。函数 $\mathcal{L}(\cdot)$ 表示一个特定任务的损失函数，例如交叉熵损失。

传统的被动防御策略操作单一观察 o_i 来对抗对抗性补丁，因此未能利用通过主动探索环境可获得的丰富上下文信息 [?], [?]。形式上，一个对抗性补丁 p 被引入到观察 o_i 中，导致感知模型 f 的错误预测。在 3D 场景中生成对抗性补丁 [?] 通常通过优化来实现：

$$\max_{p} \mathbb{E}_{s_i} \mathcal{L}(f(A(p, o_i; s_i)), y), \tag{1}$$

这里 $A(\cdot)$ 根据相机的状态 s_i 将 3D 对抗性补丁 p 投影到 2D 相机观测 o_i 中。当 s_i 被限制为 2D 变换和补丁位置时，这种广义的 3D 表述涵盖 2D 情况，如同 Brown *et al.* [?]。

为了后续分析，我们定义场景 x 的欺骗性对抗补丁集合 $\mathcal{P}_x$ 如下：

$$\mathcal{P}_x = \{p \in [0,1]^{H_p \times W_p \times C} : \mathbb{E}_{s_i} f(A(p, o_i; s_i)) \neq y\}, \tag{2}$$

，其中 H_p 和 W_p 分别表示补丁的高度和宽度。在实践中，近似求解集合 $\mathcal{P}_x$ 涉及使用特定的优化技术 [?], [?] 来解决方程 (1) 中提出的问题。这种通用的公式为分析复杂 3D 环境中对抗补丁的行为和影响提供了稳固的基础。

EAD 框架。我们的最新工作提出了 EAD [?]，这是一种倡导主动场景参与的范式，迭代利用环境反馈来增强对抗贴片攻击的感知系统的鲁棒性。EAD 由两个复发性模型组成，它们模拟了支持主动人类视觉的复杂大脑结构。感知模型 $f(\cdot; \boldsymbol{\theta})$，由 $\boldsymbol{\theta}$ 参数化，被精心设计以便充分利用来自外部世界的时间观测中的丰富上下文信息，促进复杂的视觉感知。在每个时间步 t，该模型巧妙地利用当前观测 o_t 并将其与关于场景的现有内部信念 b_{t-1} 融合，从而通过一种递归模式构建周围

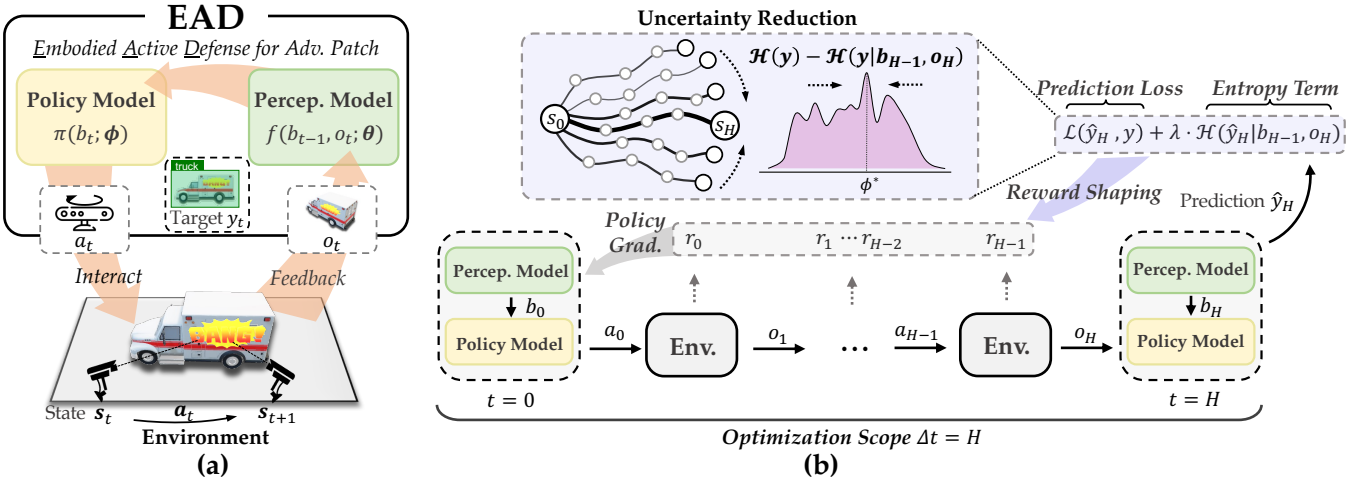

Fig. 1: 对 EAD 和 Rein -EAD 的概述。(a) 在 EAD 中，感知模型使用观测 o_t 和先前的内部信念 b_{t-1} 对环境表示 b_t 进行优化，从而做出与任务相关的预测 y_t 。策略模型基于 b_t 生成动作 a_t ，最小化单步的感知不确定性 $H(y \mid b_{t-1}, o_t)$ 。(b) Rein -EAD 通过累积多步交互来扩展 EAD，在时间范围 H 内平衡预测损失的减少和熵最小化。策略采用无模型强化学习进行学习，每步都有密集的奖励，指导代理进行稳健的决策。

环境 b_t 的增强表示。同时，感知模型生成一个场景注释 $\hat{y}_t$ 如下：

$$\{\hat{y}_t, b_t\} = f(o_t, b_{t-1}; \boldsymbol{\theta})^1. \tag{3}$$

随后的政策模型 $\pi(\cdot;\boldsymbol{\phi})$ ，由 $\boldsymbol{\phi}$ 参数化，用于控制运动的视觉控制。形式上，给定由感知模型细致维持的当前集体环境理解 b_t ，它通过从分布 $\pi(b_t;\boldsymbol{\phi})$ 中采样来推导动作 $\boldsymbol{a}_t$ 。

为了正式描述 EAD 框架在环境中的交互和主动探索（如图 1 所示），我们扩展了部分可观测马尔可夫决策过程（POMDP）框架 [?] 。在场景 x 下的交互过程表示为 $\mathcal{M}(x) \coloneqq \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{O}, \mathcal{Z} \rangle$ 。其中，$\mathcal{S}$ 和 $\mathcal{A}$ 分别表示状态和动作空间。对于 $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$ ，场景 x 下的转换动态遵循马尔可夫性质，满足 $\mathcal{T}(\cdot \mid s, a, x)$ 。由于环境的部分可观察性质，代理不能直接访问状态 s ，而是从观察函数 $\mathcal{Z}(\cdot \mid s, x)$ 中接收到一个观察值 o 。在每个时间步 t ，EAD 根据当前状态 s_t 获得一个观察值 o_t 。这一观察值 o_t 作为关键的环境反馈，通过复杂的感知模型 $f(\cdot;\boldsymbol{\theta})$ ，有助于完善代理对环境 b_t 的理解。EAD 采用的递归感知机制对于保持人类视觉稳定性至关重要 [?], [?] 。与其保持静止并被动吸收观察，EAD 利用策略模型执行从分布 $\pi(b_t;\boldsymbol{\phi})$ 中采样的动作 a_t 。策略模型的引入使 EAD 能够在每个时间步确定最优动作，确保从场景中获取最具信息量的反馈。针对对抗性补丁训练 EAD。为了装备复杂构建的 EAD 模型，我们引入了一种专门设计用于抵御对抗性补丁的学习算法。考虑一个数据分布 $\mathcal{D}$ ，其中包含成对的数据 (x, y) ，我们研究通过未知攻击技术生成的对抗性补丁，这些补丁将观测 o_t 损坏为 $o'_t = A(o_t, p; s_t)$ 。EAD 的主要目标是在存在对抗性补丁威胁时最小化预期损失。因此，EAD 减轻对抗性补丁的学习过程被表述为参数 $\boldsymbol{\theta}$ 和 $\boldsymbol{\phi}$ 的优化问题，如下：

$$\begin{aligned} &\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathbb{E}_{(x,y)\sim\mathcal{D},\tau\sim(\mathcal{M}(x),\pi),t}\Big[\sum_{p\in\mathcal{P}_x}\mathcal{L}(\hat{y}_t, y)\Big], \\ &\text{with } \{\hat{y}_t, b_t\} = f(A(o_t, p; s_t), b_{t-1}; \boldsymbol{\theta}),\ a_t \sim \pi(\cdot \mid b_t; \boldsymbol{\phi}) \\ &\text{s.t.} \quad o_t \sim \mathcal{Z}(\cdot \mid s_t, x), \quad s_t \sim \mathcal{T}(\cdot \mid s_{t-1}, a_{t-1}, x), \end{aligned} \tag{4}$$

，其中 $\tau \coloneqq (o_0, a_0, o_1, \ldots, o_H)$ 表示收集的轨迹，具有长度 H 和概率 $p_{\mathcal{M}(x),\pi}(\tau) = \rho_0(s_0)\prod_{t=0}^{H}\mathcal{T}(s_{t+1} \mid s_t, a_t, x)\pi(a_t \mid b_t;\boldsymbol{\phi})\mathcal{Z}(o_t \mid s_t)$ ；而 $\hat{y}_t$ 表示模型在时间步 t 的预测，遵循 $\{0, 1, 2, ..., H\}$ 上的均匀分布。值得注意的是，损失函数 $\mathcal{L}$ 表现出任务无关的性质，强调了 EAD 框架的超强适应性。这种内在的灵活性保证了 EAD 在广泛的感知任务中提供强大的防御。

1. 我们分别设置 $f_y(o_t, b_{t-1}; \boldsymbol{\theta}) = \hat{y}_t$ 和 $f_b(o_t, b_{t-1}; \boldsymbol{\theta}) = b_t$ 。

2.2 EAD 的理论分析

为了更深入地理解模型的行为，我们进一步研究了方程（4）中 EAD 的广义实例，其中代理使用了 InfoNCE 目标 [?]，简化为：

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathbb{E}_{(x^{(j)},y^{(j)})}\left[\frac{1}{K}\sum_{j=1}^{K}\log\frac{e^{-S(f_y(b_{t-1}^{(j)},o_t^{(j)};\boldsymbol{\theta}),y^{(j)})}}{\frac{1}{K}\sum_{k=1}^{K}e^{-S(f_y(b_{t-1}^{(j)},o_t^{(j)};\boldsymbol{\theta}),y^{(k)})}}\right], \tag{5}$$

，其中 $(x^{(j)}, y^{(j)})_{j=1}^{K}$ 表示从分布 $\mathcal{D}$ 中采样的大小为 K 的数据批次，$S : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ 量化了预测场景注释与真实标签之间的相似性。在具身感知中，这种损失建立了观察与注释之间的跨模态对应关系，如 CLIP [?] 。此外，我们提供方程（5）的信息论解释，以阐明其基本原理。

Theorem 2.1 (Proof in Appendix A.1). *在当前观测 o_t 和场景注释 y 之间的互信息，条件于先前的信念 b_{t-1} ，记作 $I(o_t; y \mid b_{t-1})$ ，我们可以证明我们的目标是此互信息的下界：*

$$\begin{aligned} &\mathbb{E}_{(x^{(j)},y^{(j)})\sim\mathcal{D}}\left[\frac{1}{K}\sum_{j=1}^{K}\log\frac{q_{\boldsymbol{\theta}}(y^{(j)} \mid b_{t-1}^{(j)}, o_t^{(j)})}{\frac{1}{K}\sum_{k=1}^{K}q_{\boldsymbol{\theta}}(y^{(k)} \mid b_{t-1}^{(j)}, o_t^{(j)})}\right] \\ \le &\mathbb{E}_x I(o_t; y \mid b_{t-1}) - \frac{\log(K)}{K} \\ = &\mathbb{E}_x\left[\mathcal{H}(y \mid b_{t-1}) - \mathcal{H}(y \mid b_{t-1}, o_t)\right] - \frac{\log(K)}{K}, \end{aligned} \tag{6}$$

，其中 $q_{\boldsymbol{\theta}}(y \mid o_1, \cdots, o_t)$ 表示变分分布，用于近似真实的条件分布 $p(y \mid o_1, \cdots, o_t)$ ，并使用样本 $(x^{(j)}, y^{(j)})_{j=1}^{K}$ 。

Remark 2.2. 为了阐明派生的条件互信息下界与优化目标之间的关系，我们通过采用等式（5）中的相似性项重新构造变分分布 $q_{\boldsymbol{\theta}}(y \mid b_{t-1}, o_t)$ ，得到 $q_{\boldsymbol{\theta}}(y \mid b_{t-1}, o_t) \coloneqq p(b_{t-1}, o_t)e^{-S(f(b_{t-1},o_t;\boldsymbol{\theta}),y)}$ 。这一重新构造等于负 InfoNCE 目标，如等式（5）中所示。这强调了 EAD 训练过程最大化

条件互信息，引导智能体收集最具信息量的观察 o_t 以确定任务指定的注释 y 。请注意，随着批量大小 K 增加，派生下界的收紧性改善。

方程（6）中的最后一个等式源于互信息等同于条件熵的减少。这表明优化指引策略朝向贪婪的信息探索，定义为：

Definition 2.3 (Greedy Informative Exploration)**.** *贪心的信息探索，由 π^g 表示，指的是一种在任意时间步 t 内选择一个行为 a_t 的行动策略，该行为最大化通过执行行动 a_t 获得的新观测 o_t 所给定的随机变量 y 的条件熵的下降。形式上，*

$$\pi^g = \arg\max_{\pi\in\Pi} \left[\mathcal{H}(y \mid b_{t-1}) - \mathcal{H}(y \mid b_{t-1}, o_t)\right], \tag{7}$$

其中 $\mathcal{H}(\cdot)$ 表示熵，Π 表示涵盖所有可行策略的空间。

通过理论分析，我们证明了在 InfoNCE 目标函数（5）中，最优策略 π^*_ϕ 收敛到一个贪婪的信息探索策略，将互信息与策略改进相结合。信息论的视角揭示，一个训练良好的 EAD 模型自然会采用贪婪的信息策略，利用上下文信息来解决含对抗性块的场景中的高不确定性 [**?**] 。

以往的 EAD 展示了显著的潜力，但在有效性、适用性和效率这三个关键领域仍然面临挑战：(1) 时间不一致性：EAD 的贪婪信息通常产生时间不一致的动作，如图 **??** 所示。这种不一致性源于贪婪策略的短视性质，它们优化即时信息增益而不考虑具身学习中的长期相关性 [**?**], [**?**] 。因此，可能导致智能体返回到先前探索过的视角，减少探索有效性，并可能导致错误的预测，特别是在重访具有显著对抗性影响的视角时 [**?**], [**?**] 。(2) 适用性有限：学习 EAD 依赖可微分动态模型，严重限制了其实际应用。在现实场景中 [**?**], [**?**] ，由于物理系统的复杂性和不可预测性，以可微方式准确建模环境动态通常是不可行的。(3) 计算效率低下和不稳定性：通过可微模拟进行的学习计算需求高且易于出现数值不稳定。该过程需要大量计算以遍历动态模型并推导其梯度。此外，由于模拟动态 [**?**] 提供的梯度质量不足，经常在优化过程中出现梯度消失 [**?**] 。

Definition 2.4 (Accumulative Informative Exploration)**.** *累积信息探索，用 π^* 表示，是指通过执行动作 $a_{0:H}$ 与环境持续互动产生一系列观测 $o_{0:H}$ ，从而最大化减少给定 y 熵的策略。形式化地，*

$$\pi^* = \arg\max_{\pi\in\Pi} \left[\mathcal{H}(y) - \mathcal{H}(y \mid b_{H-1}, o_H)\right]. \tag{8}$$

Remark 2.5. 与定义 2.3 中定义的贪婪信息探索相反，累积信息策略旨在通过与环境的持续互动来最小化目标 y 的不确定性，而不是仅关注单步。贪婪信息探索可以被视为累积信息探索的一种特殊情况，仅限于 $H = 1$ 的单步转换。从长远来看，贪婪信息探索倾向于短视地减小逐步的不确定性，往往导致比累积信息探索次优的结果。

为了实现累积的信息探索，我们提出了一个多步累积交互目标，该目标在 H 步的范围内优化策略，包含鼓励达到最小化预测损失的信念状态并惩罚高熵预测的项：

$$\begin{aligned} &\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathbb{E}_{x,y,\tau} \sum_{p\in\mathcal{P}_x} \Big[\mathcal{L}(\hat{y}_H, y) + \lambda \cdot \mathcal{H}(\hat{y}_H \mid b_{H-1}, o_H)\Big], \\ &\text{with } \{\hat{y}_t, b_t\} = f(A(o_t, p; s_t), b_{t-1}; \boldsymbol{\theta}),\ a_t \sim \pi(\cdot \mid b_t; \boldsymbol{\phi}) \\ &\text{s.t.} \quad o_t \sim \mathcal{Z}(\cdot \mid s_t, x), \quad s_t \sim \mathcal{T}(\cdot \mid s_{t-1}, a_{t-1}, x), \end{aligned} \tag{9}$$

，其中 $\mathbb{E}_{x,y,\tau}$ 是 $\mathbb{E}_{(x,y)\sim\mathcal{D},\tau\sim(\mathcal{M}(x),\pi)}$ 根据公式（4）的缩写，$\mathcal{L}(\hat{y}_H, y)$ 表示第 H 步的预测损失，$\mathcal{H}(\hat{y}_H \mid b_{t-1}, o_t)$ 表示第 H 步预测标签的熵。采样轨迹与公式（4）的分布相同。熵项作为正则项，阻止代理做出具有对抗性例子特征的高熵预测 [**?**] 。

所提出的多步骤交互与累积信息探索的定义一致，因为它试图通过一系列的动作和观察来最小化目标变量的不确定性。通过结合预测损失和熵正则项，这一目标鼓励代理达到信息丰富且稳健的信念状态，从而增强对抗对抗性扰动的能力。然后，我们分析累积信息策略与贪婪策略之间的性能差距，如下所示。

Theorem 2.6 (Informative Policy Efficacy Inequality, Proof in Appendix A.2)**.** *考虑两种策略在时间范围 H 内与环境连续互动：*

- *π^g 代表贪婪的信息策略，产生观测序列 $o^g_{0:H}$ 和信念序列 $b^g_{0:H}$ 。*
- *π^* 表示累积信息策略，生成观测序列 $o^*_{0:H}$ 和信念序列 $b^*_{0:H}$ 。*

在策略 π 下，从时间 0 到 H 的轨迹信息增益定义为变量 y 的熵减少：

$$\Delta\mathcal{H}_\pi := \mathcal{H}(y) - \mathcal{H}\big(y \mid b_{H-1}, o_H\big).$$

假设信念更新函数 $f_b : (b_{t-1}, o_t) \mapsto b_t$ 是双射的。则相对 π^g 而言 π^* 的效力满足以下不等式：

$$\Delta\mathcal{H}_{\pi^*} \geq \Delta\mathcal{H}_{\pi^g}, \tag{10}$$

当且仅当问题表现出最优子结构时，该等式成立。

Remark 2.7. 如这个不等式所示，效能差距主要来自两个因素：(1) 探索问题的复杂性，这使得很难保证一个最佳结构以确保贪婪策略的有效性，以及 (2) 在信念更新过程中信息的丢失，它将先前的信念 b_{t-1} 和当前的观察 o_t 编码进当前信念 b_t 。贪婪策略在通过与环境的持续交互进行信念更新时不考虑信息丢失。其在方程（7）中的表述仅通过在单一步骤中采取行动 a_{t-1} 来确保选择最具信息性的观察 o_t 。

通过扩展在第 2.2 节中讨论的贪婪信息探索，我们可以将方程（5）重新表述为一个多步交互形式。在无限的模型容量和数据样本的假设下，学习到的策略模型 π^*_ϕ 代表了一种累积的信息策略。此外，定理 2.6 正式确立了多步交互的优越性，表明一个训练良好的对比任务多步模型采用累积的信息策略来持续探索环境。该模型利用环境中的一系列上下文信息持续减少对抗性扰动造成的感知不确定性，为方程（9）中的多步累积目标提供了理论基础。

尽管在第 **??** 节中的有效性，REIN -EAD 在现实应用中面临重大挑战。物理系统的不可预测性使得以可微方式建模环境动力学变得不可行。此外，从动态模型微分得到的不可避免的计算开销和由于近似产生的数值不稳定性使得解决方程（9）变得不切实际，特别是在长时间跨度上。这些问题随着计算需求的增长和累积梯度估计误差的增加而加剧，阻碍了其适用性和效率。

为了应对这些挑战并提高我们方法在现实世界中的适用性，我们提出了一种策略学习方法，该方法在强化学习框架中结合了面向不确定性的奖励塑造技术。通过消除对可微分动态模型的需求，我们的方法使智能体能够通过与环境的试错互动直接学习最大化预期累积奖励的策略。这种灵活性允许智能体有效地适应变化的动态或随机环境，这在现实世界的场景中是普遍存在的 [**?**] 。

具体来说，不确定性导向的奖励塑造技术旨在在每一步提供密集的奖励，引导智能体减少感知不确定性并最小化预测误差。这种方法解决了经典稀疏奖励策略 [**?**] 在方程（9）中提到的问题，智能体只能在回合结束时获得奖励。通过结合加权的中间奖励，我们的方法允许智能体获得更细致和信息量更大的反馈。正式地，我们定义密集奖励 r_t 如下（等价性证明在附录 A.3 中）：

$$r_t = \mathcal{L}(\hat{y}_{t-1}, y) - \gamma \cdot \mathcal{L}(\hat{y}_t, y), \quad (t > 0) \tag{11}$$

Algorithm 1 通过策略学习训练 REIN -EAD

Require: Training data $\mathcal{D}$, number of iterations M , number of epochs E , loss function $\mathcal{L}$, perception model $f(\cdot;\boldsymbol{\theta})$, policy model $\pi(\cdot;\boldsymbol{\phi})$.
Ensure: The parameters $\boldsymbol{\theta}, \boldsymbol{\phi}$ of the learned EAD model.

for iteration $\leftarrow 0$ to $M-1$ **do**
  ▷ *Roll-out perception $f(\cdot;\boldsymbol{\theta})$ and policy $\pi(\cdot;\boldsymbol{\phi})$ in the environment.* ◁
  Collect set of augmented trajectory and label pairs $\mathcal{D}_\tau = \{(\tau, y)\}$ by running policy $\pi(\cdot;\boldsymbol{\phi})$ and perception $f(\cdot;\boldsymbol{\theta})$ on $\mathcal{M}(x)$ with $(x, y) \sim \mathcal{D}$, where

$$\tau = (o'_0, b_0, a_0, r_0, o'_1, b_1, \ldots)$$

  Estimate advantages $\hat{A}_t$ using any advantage estimation algorithm
  for epoch $\leftarrow 0$ to $E-1$ **do**
    $\boldsymbol{\phi}_{\text{old}} \leftarrow \boldsymbol{\phi}$
    for all mini-batch $\mathcal{B}_\tau \in \mathcal{D}_\tau$ **do**
      Update $\boldsymbol{\phi}$ by maximizing estimated the PPO-Clip objective $\mathcal{J}_{\text{policy}}(\boldsymbol{\phi})$ defined as:

$$\frac{1}{|\mathcal{B}_\tau|}\sum_{\tau\in\mathcal{B}_\tau}\sum_t \min(R(\boldsymbol{\phi})\hat{A}_t, \text{clip}(R(\boldsymbol{\phi}), 1-\epsilon, 1+\epsilon)\hat{A}_t)$$

      with one-step gradient ascent, where

$$R(\boldsymbol{\phi}) = \frac{\pi(a_t \mid b_t, \boldsymbol{\phi})}{\pi(a_t \mid b_t, \boldsymbol{\phi}_{\text{old}})}$$

      Update $\boldsymbol{\theta}$ by minimize the estimated objective in Eq. (9), namely $\mathcal{J}_{\text{percep}}(\boldsymbol{\theta})$:

$$\frac{1}{|\mathcal{B}_\tau|}\sum_{(\tau,y)\in\mathcal{B}_\tau}\sum_t \mathcal{L}(f(o_t, b_t;\boldsymbol{\theta}), y) + \lambda\cdot\mathcal{H}(f(o_t, b_t;\boldsymbol{\theta}))$$

      with one-step gradient descent.
    end for
  end for
end for

其中 γ 是折扣因子。这种密集的奖励结构激励策略 π 寻求从环境中获得新的观察 o_t 作为反馈。由密集奖励促进的连续反馈循环使得智能体能够有效地适应新情况，并在面临不确定性时做出明智的决策。此外，通过在每一步中公平地分配奖励，我们缓解了探索和信用分配的问题 [?] ，促进更快的收敛和更高效的学习。

对于强化学习骨干网络，我们采用了 Proximal Policy Optimization (PPO) [?] ，因为它具有学习效率高和收敛稳定的特点。PPO 通过基于我们的不确定性导向奖励塑形技术所提供的密集奖励，限制每次迭代中策略变化的幅度，从而实现稳定的策略更新。这个渐进的学习过程确保了智能体保持一个稳定的轨迹，以减少不确定性并最小化预测误差。详细的训练过程见算法 1 。

2.3 针对补丁攻击的对抗无关防御

$\mathcal{P}_x$ 的计算通常需要通过在线对抗训练 [?] 来解决方程 （4 ）中的内部最大化问题。然而，这不仅在计算上昂贵 [?] ，而且由于对敌手的特征刻画假设不足，可能会阻碍模型在不同的、未见过的攻击中泛化的能力 [?] 。

离线对抗性补丁近似 (OAPA)。尽管 USAP 方法 [?] 在给定足够的训练周期下有效地学习到一个最佳的信息策略，但由于需要进行大量采样以确保采样的流形包含足够数量的对抗性样本，它在计算上是昂贵的。为了在保持对抗者无关特性的同时提高采样效率，我们引入了 OAPA，它采用投影梯度技术来近似对抗性补丁流形 $\mathcal{P}_x$ ，在训练 REIN -EAD 模型之前进行。OAPA 通过预生成一组替代补丁 $\tilde{\mathcal{P}} = p_i$ 系统地刻画了对抗性模式的流形，其中每个补丁 p_i 是通过对视觉骨干进行投影梯度上升获得的。对抗性补丁流形的这种离线近似允许 REIN -EAD 模型学习紧凑但具表现力的对抗性模式表示，使其能够有效防御此前未见的攻击。我们的实验证明，执行这种离线近似最大化在开发对广泛攻击具有鲁棒性的模型方面非常有效（参考 Sec. 3 ）。此外，由于这一最大化过程在训练之前离线且仅进行一次，因此它极大地提升了训练效率，使其在竞争中与传统训练方法相当。

3 实验

在本节中，我们验证了 REIN -EAD 在不同任务中的有效性，包括在第 3.1 节的人脸识别、第 3.2 节的 3D 对象分类和第 3.3 节的目标检测。

3.1 人脸识别评估

3.1.1 实验设置

实验环境。为了实现 REIN -EAD 的无约束导航和观测收集，我们为训练和目的构建了一个可操作的模拟环境。为了与特定的视觉任务对齐，我们将状态定义为摄像机的偏航和俯仰的组合，而动作对应于摄像机 [2] 的旋转。此定义建立了转换函数，模拟环境的核心围绕着观测函数，该函数根据摄像机的状态生成二维图像。如 Sec. 2.1 所详述，原始 EAD 的训练需要可微的环境动态。为了实现公平比较，我们首先采用先进的 3D 生成模型 EG3D [?] ，该模型能实现逼真的可微渲染（模拟逼真度详见附录 C.1 ）。为了进一步证明 REIN -EAD 的优越性，我们在相同环境中进行 REIN -EAD 的训练和测试以确保公平比较。测评指标与协议。为了严格验证 REIN -EAD 的有效性，我们在 CelebA-3D 上进行了广泛的实验。我们通过利用 GAN 反演 [?] 和 EG3D [?] ，将 CelebA 中的二维人脸图像精心重构为三维表示。为了评估标准准确性，我们从 CelebA 中抽取 2,000 个测试对，并遵循 LFW [?] 的成熟评估协议。为了全面评估鲁棒性，我们报告在 100 对身份对上的攻击成功率（ASR），并考虑在白盒和黑盒设置下各种攻击方法中的冒充和规避攻击 [?] 。白盒攻击方法包括 MIM [?] 、EoT [?] 、GenAP [?] 和 Face3Dadv（3DAdv） [?] 。在黑盒攻击中，我们采用基于转移的攻击，针对 IResNet-18 CosFace [?], [?] 和 IResNet-50 Softmax 等代理模型与 3DAdv 的配合，以及包括 NAttack [?] 和 RGF [?] 的查询方法。注意，3DAdv 在优化过程中利用了对三维变换的期望，使其在某个 $\pm 15°$ 范围内对三维视角变化具有内在的鲁棒性。更多细节描述在附录 C.1 & C.2 中。

实现细节。为了提取具有区别性的视觉特征，我们采用了预训练的 IResNet-50 ArcFace [?] 作为视觉主干网络，利用其预训练的权重，并在后续训练阶段保持它们冻结。为了对递归感知和策略组件建模，我们采用了 Decision Transformer [?] 的一种变体，该变体有效地处理由视觉主干提取的特征序列以预测 FR 的标准化嵌入。我们在 EAD 中将最大地平线长度默认设置为 $H = 4$。与 EAD 相比，REIN -EAD 消除了对通过时间反向传播 (BPTT) 的依赖，使我们可以在没有 VRAM 限制的情况下将地平线长度扩展到 $H = 16$。此外，我们在 REIN -EAD 中加入了一个额外的 MLP 值头，以促进 PPO 的优势估计。关于 REIN -EAD 的进一步细节可以在附录 C.5 中找到。

防御基准。为了全面评估 REIN -EAD 的有效性，我们将它与各种最新的防御方法进行对比。这些基准包括基于对抗训练的防御遮挡攻击 (DOA) [?] ，以及 基于纯化的方法如 JPEG

2. 为了防止代理通过旋转到隐藏对抗性补丁的角度来“作弊”，在偏航和俯仰上施加了特定的约束。

TABLE 1: 在面部识别上的标准准确率（%）和攻击成功率（%）。† 表示使用对抗训练的方法。 淡蓝色背景的方法 不需要可微分环境，而其他方法则需要。

Method	Acc. (%)	White-box				Transfer-based		Query-based		Adaptive	
		MIM	EoT	GenAP	3DAdv	Cos.	Softmax	NAttack	RGF	BPDA	Worst-case
Impersonation Attack											
Undefended	88.86	100.0	100.0	99.00	89.00	28.00	23.00	100.0	100.0	100.0	100.0
JPEG	89.98	99.00	100.0	99.00	93.00	33.00	33.00	96.00	94.00	99.00	100.0
LGS	83.50	5.10	7.21	33.67	30.61	11.63	6.98	11.63	4.65	38.37	48.83
SAC	86.83	6.06	9.09	67.68	64.64	8.70	9.78	13.04	14.13	48.00	67.68
PZ	87.58	4.17	5.21	59.38	45.83	6.45	9.68	4.30	3.26	89.76	92.86
SAC †	80.55	3.16	3.16	18.94	22.11	11.11	12.36	12.22	14.44	51.73	51.73
PZ †	85.85	3.13	3.16	19.14	27.34	8.24	5.00	10.58	9.41	61.01	98.99
DOA †	79.55	95.50	89.89	96.63	89.89	15.73	17.97	34.83	16.86	89.89	96.63
EAD	90.45	4.12	3.09	5.15	7.21	4.17	5.20	4.12	4.12	8.33	9.38
REIN-EAD	89.03	2.10	1.06	3.15	7.37	2.10	2.08	1.05	2.12	4.21	7.37
Dodging Attack											
Undefended	88.86	100.0	100.0	99.00	98.00	44.00	35.00	96.00	96.00	100.0	100.0
JPEG	89.98	98.00	99.00	95.00	88.00	49.00	45.00	81.00	83.00	100.0	100.0
LGS	83.50	49.47	52.63	74.00	77.89	22.11	21.05	18.95	20.00	78.92	78.92
SAC	86.83	73.46	73.20	92.85	78.57	40.80	36.84	55.26	50.00	65.22	92.85
PZ	87.58	6.89	8.04	58.44	57.14	41.67	28.34	28.33	31.67	88.89	90.00
SAC †	80.55	78.78	78.57	79.59	85.85	47.46	43.54	55.93	62.71	85.02	85.85
PZ †	85.85	6.12	6.25	14.29	20.41	50.88	47.69	56.14	50.87	69.45	98.00
DOA †	79.55	75.28	67.42	87.64	95.51	30.33	31.46	53.93	28.09	95.51	95.51
EAD	90.45	0.00	0.00	2.10	13.68	5.26	7.36	1.05	0.00	22.11	22.11
REIN-EAD	89.03	1.04	2.04	5.15	13.54	4.17	7.29	1.03	0.00	8.16	14.43

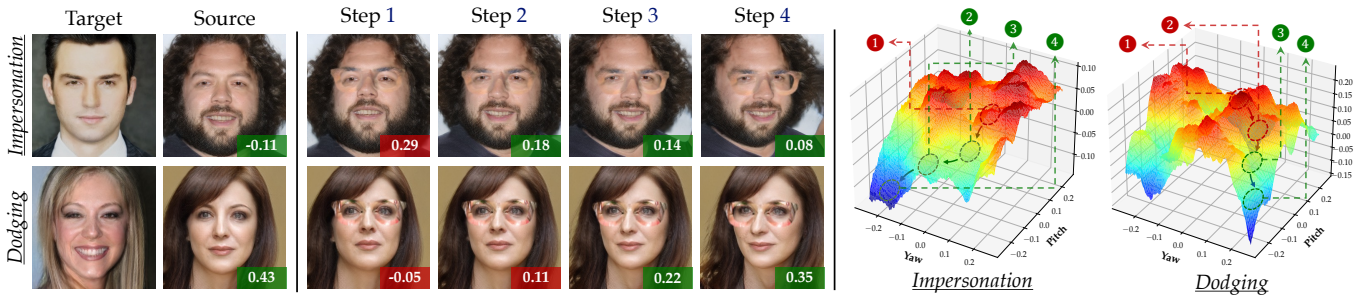

Fig. 2: REIN -EAD 的定性结果。前两列展示了原始图像对，后续列展示了模型进行的交互推理步骤。REIN -EAD 的防御轨迹在损失地形图上绘制，包括相机的航向和俯仰角，以 IResNet-50 ArcFace 作为目标模型 [?] 。使用 3DAdv 生成的对抗眼镜对 3D 视点变化具有鲁棒性。区分正负样本对的计算最佳阈值设定为 0.19，范围为 [-1, 1]。

压缩（JPEG） [?]，局部梯度平滑（LGS） [?]，分割与完成（SAC） [?]，PatchZero（PZ） [?]，无关 Patch 的防御（PAD） [?] 和 DIFFender [?] 。对于 DOA，我们应用矩形 PGD 补丁攻击 [?]，迭代 10 次，步长为 2/255 。需要注意的是，SAC 和 PZ 需要一个补丁分割器来定位对抗补丁的区域。因此，我们使用高斯噪声的补丁来训练分割器，以确保相同的对抗无关设置。此外，我们考虑了 SAC 和 PZ 的增强版本，记作 SAC † 和 PZ †，这些版本涉及使用 EoT 技术生成的对抗补丁进行训练。更多细节在附录 C.4 中提供。

3.1.2 实验结果

的有效性和 REIN -EAD。表 1 展示了在白盒环境下对 标准准确性 和 稳健性能 进行全面评估的结果，其中对抗性补丁大小设置为图像大小的 8 %。值得注意的是，我们的方法在干净准确性和防御效果方面显著优于之前不考虑对抗样本的最新被动技术。例如，REIN -EAD 在两种情况下都显著降低了 3DAdv 的攻击成功率。此外，REIN -EAD 通过将信息贪婪策略扩展为累积策略，还在面对黑盒和自适应攻击时，比 EAD 增强了平均攻击率的降低效果。值得注意的是，REIN -EAD 在标准准确性方面甚至超越了无防御被动模型的表现，有效地通过体现感知实现了稳健性和准确性之间的权衡。此外，我们的方法通过在训练期间结合对抗样本甚至超越了基线。虽然 SAC † 和 PZ † 是通过 EoT [?] 生成的补丁进行训练的，但我们仍然获得了卓越的表现，突出了在主动防御中利用环境反馈的有效性。

图 2 直观地展示了由 REIN -EAD 执行的防御过程。尽管 REIN -EAD 可能最初被对抗性补丁欺骗，但其随后与环境的主动交互逐步增加了正对之间的相似性，并减少了负对之间的相似性。因此，REIN -EAD 通过主动获取额外观察有效地减轻了对抗性幻觉的影响，如相应损失景观中所示。

对自适应攻击的有效性。当确定性和可微分的动态模型可能潜在地使得可以通过 EAD 的整个推理轨迹进行反向传播时，由于随着轨迹长度 H 的增加导致 GPU 内存的快速消耗，计算成本变得不可承受。为了解决这个问题，我们首先采用类似于原始策略的方法，通过计算代理统一超集策略分布上的期望梯度来近似真实梯度。这个近似（BPDA）需要一个优化的补丁来处理多样的动作策略。我们的自适应攻击实现基于 3DAdv [?]，利用 3D 视点变化。此外，我们还评估了更复杂的自适应攻击，包括 1) 使用通过梯度检查点获得的真实梯度攻击整个管道，以及 2) 独立针对感知和策略子模块。表 1 中

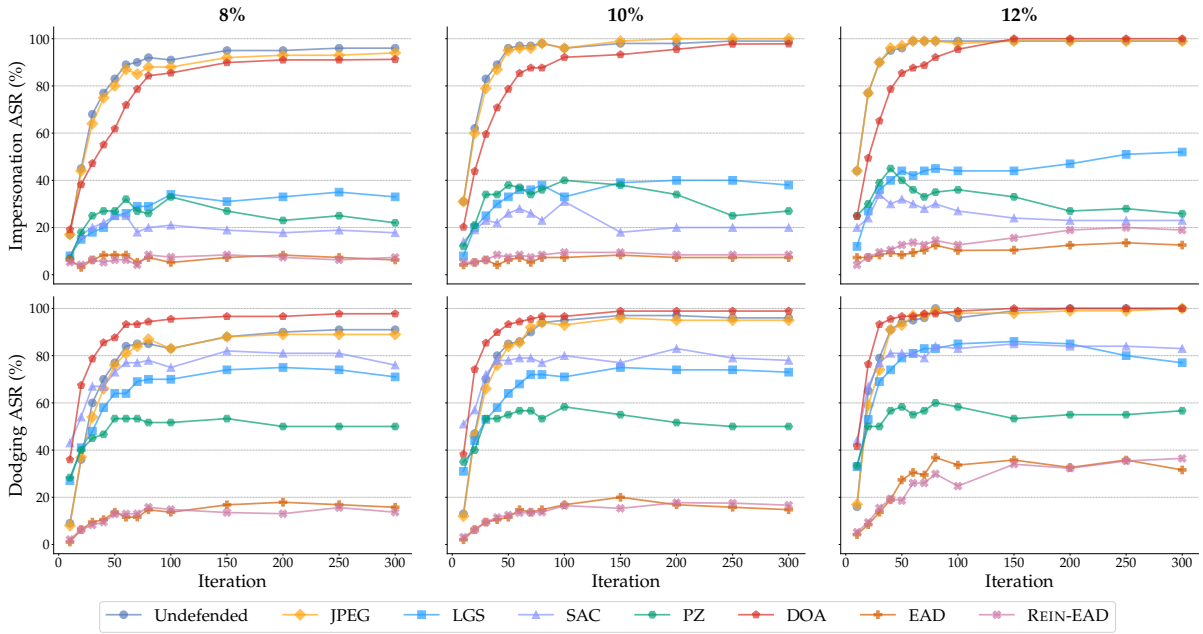

Fig. 3: 对不同敌对补丁大小在不同攻击迭代下的防御方法进行比较评估。请注意，SAC、PZ 和 DOA 涉及对抗训练。

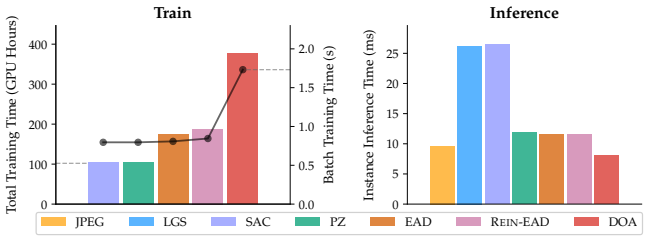

Fig. 4: 防御方法的 计算开销 的比较评估。

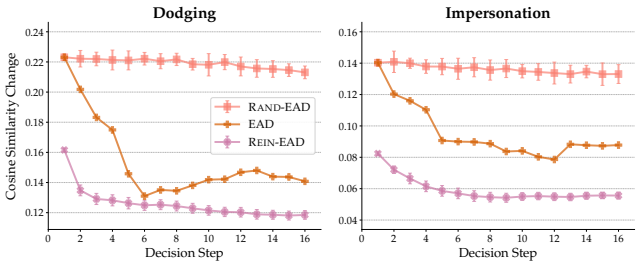

Fig. 5: 不同决策步骤的性能差异。

给出了 Rein -EAD 在两种自适应攻击中的最坏情况表现以进行可靠评估。此外，为了对比其他防御方法，我们报告了它们在一系列自适应攻击下的最坏情况表现。更多细节见附录 C.3 。我们可以看到，Rein -EAD 在面对最强大的自适应攻击时仍保持其鲁棒性。此观察表明 Rein -EAD 的防御能力源于其策略和感知模型的协同集成，而不是依赖于从特定视角中和对抗性补丁的捷径策略。

对 Rein -EAD 的泛化。如表 1 所示，尽管对具体的对手没有预先知识，我们的方法在各种未见过的对抗攻击方法中表现出卓越的泛化能力。这部分归功于 Rein -EAD 的一种内在能力，即能够动态地与其环境互动，使其能够适应和应对新型攻击。此外，我们评估了模型在不同补丁大小和攻击迭代次数范围内的弹性。尽管仅在构成图像 10 % 的补丁上进行训练，但即使在更大补丁尺寸和增加的攻击迭代次数下，Rein -EAD 仍然保持显著低的攻击成功率，如图 3 所示。这种卓越的弹性可归因于 Rein -EAD 主要依赖于环境信息，而不仅仅依赖于假定对手的模式。通过动态更新其感知，Rein -EAD 能够很好地泛化到不同方面。详细信息见附录 C.7 。

计算开销。我们进一步比较了 Rein -EAD 与一些被动防御基准在训练和推理时间方面的计算开销。如图 4 所示，尽管 EAD 所采用的差分渲染在在线训练阶段施加了显著的计算需求，但是 EAD 的总训练时间在纯粹的对抗训练方法 DOA 和部分对抗方法（如 SAC 和 PZ）之间有效地实现了平衡。这种效率主要来源于我们独特的对手无关方法（OAPA），该方法消除了在线生成对抗样本的需要，从而提高了训练效率。尽管 Rein -EAD 的较大视野固有地增加了采样时间，但其训练速度仍然比 EAD 快。这种优势源于被模型无关方法采用的 Rein -EAD，避免了通过反向传播获取策略梯度的计算密集过程，从而提高了整体学习效率。该优势归因于 LGS 和 SAC 对于 CPU 密集型的、基于规则的图像预处理技术的依赖，这不可避免地降低了它们的推理效率。关于计算开销的更多细节在附录 C.6 中提供。

3.1.3 消融研究

反馈的有效性。我们深入研究了反馈的关键作用，即利用全面融合模型反思先前的信念，以实现稳健的性能。在表 2 中，即使仅配备感知模型，Rein -EAD 显著超过了未防御的基线和依赖多视角集成（随机运动）的被动 FR 模型。值得注意的是，多视角集成模型未能对抗最先进的 3DAdv。这一观察结果证实，Rein -EAD 的防御强度并不只是对抗样本在视点变换中脆弱的结果。相反，Rein -EAD 的优越性能归因于其主动探索环境和动态调整感知能力的能力。

地平线长度 H 的影响。我们在图 5 中分析了地平线长度 H 对 Rein -EAD 性能的影响。通过检查受对抗性补丁影响的人脸对之间的相似性变化，我们一贯证明了 Rein -EAD 在减轻对抗性攻击负面影响方面的有效性。具体来说，对于冒充攻击，相似性变化意味着增加，而对于规避攻击则观察到减少。这一趋势表明，在决策过程中，从额外视点累积的信息有效地缓解了由对抗性补丁引发的信息丢失和模型幻觉问题。

策略的效率。我们进一步实证验证了该策略的优越性，通过将 EAD 的表现与两个变体进行比较：集成随机移动策略的 EAD，即 Rand -EAD 和 Rein -EAD。这三种方法共享相同的神经网络架构和参数。图 5 表明，Rand -EAD 在随机探索时无法缓解对抗效应，即使使用了更多的动作。因此，随机策

TABLE 2: 削减模型上的标准准确率和白盒模拟攻击成功率。对于具有随机策略的模型，我们报告了五次独立运行中的平均值和标准差，以确保可靠性。

Category	Component	Acc. (%)	Attack Success Rate (%)				
			MIM	EoT	GenAP	3DAdv	Adaptive
Passive Perception	Undefended	88.86	100.0	100.0	99.00	98.00	98.00
	+ Random Movement	90.38 (± 0.12)	4.17 (± 2.28)	5.05 (± 1.35)	8.33 (± 2.21)	76.77 (± 3.34)	76.77 (± 3.34)
EAD	+ Perception Model	90.22 (± 0.31)	18.13 (± 4.64)	18.62 (± 2.24)	22.19 (± 3.97)	30.77 (± 1.81)	31.13 (± 3.01)
	+ Policy Model	89.85	3.09	4.12	7.23	11.34	15.63
REIN-EAD	+ Multi-steps Interaction	89.02 (± 0.18)	2.12 (± 0.08)	3.19 (± 0.44)	5.33 (± 0.87)	12.77 (± 1.13)	6.38 (± 0.80)
	+ OAPA (FGSM [?])	88.95 (± 0.19)	3.22 (± 0.23)	3.03 (± 0.41)	5.15 (± 1.06)	13.18 (± 1.11)	5.14 (± 0.62)
	+ OAPA (PGD [?])	89.03 (± 0.21)	2.10 (± 0.42)	1.06 (± 0.50)	3.15 (± 1.08)	7.37 (± 1.32)	4.21 (± 0.56)

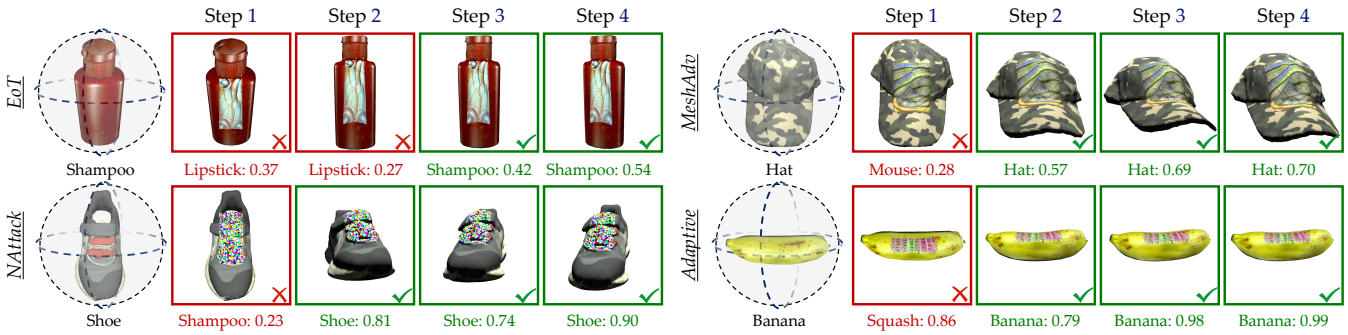

Fig. 6: 动态 OmniObject3D 上 REIN -EAD 的定性结果，其中对抗性补丁在正面视图中占据对象边界框的 20% 。对象分类的状态空间定义为 $[-\frac{\pi}{2}, \frac{\pi}{2}] \times [0, \frac{\pi}{2}]$ ，涵盖了广泛的视角范围。

TABLE 3: 标准精度（%）和攻击成功率（%）在 3D 物体分类上的表现。† 表示使用对抗训练的方法。 浅蓝色背景的方法不需要可微环境，而其他方法则需要。

Method	Acc. (%)	White-box			Transfer-based	Query-based		Adaptive
		MIM	EoT	MeshAdv	Swin-T	RGF	NAttack	BPDA
Undefended	88.17	100.00	93.72	96.19	58.90	68.03	96.48	100.00
JPEG	83.22	99.50	92.44	94.76	43.35	15.73	49.09	100.00
LGS	85.91	18.36	61.91	64.55	33.59	38.96	13.77	93.46
PAD	87.16	25.31	27.14	29.45	18.48	17.81	27.22	90.47
DIFFender	80.20	20.61	54.71	61.40	28.24	14.15	18.41	45.08
SAC †	88.00	10.30	9.53	10.49	9.72	11.15	10.20	57.01
PZ †	88.00	5.34	5.34	7.53	4.29	9.06	4.39	80.65
DOA †	87.33	6.63	6.53	7.30	5.57	4.61	5.96	59.75
REIN-EAD	88.93	3.21	4.15	4.34	3.87	2.26	3.87	28.96

略的探索效率明显低于 REIN -EAD。此外，REIN -EAD 展示了更为稳定和持续的对抗效应削弱能力，这主要归功于其采用了一种累积信息策略，能够持续降低感知不确定性，而不是贪心策略。这一策略有效避免了由时间不一致导致的振荡。

补丁数据的影响。对于 REIN -EAD，RL 在有效探索广阔的补丁空间时所固有的局限性通过结合 OAPA 算法得以解决。我们比较了使用默认 PGD 的 OAPA 和使用 FGSM 对抗样本的变体，后者由于只进行一次梯度步而优化较少。如表 2 所示，OAPA 确保 REIN -EAD 在经过大量训练后达到了卓越的鲁棒性，其中 PGD 的表现优于 FGSM 和其他方法。

策略训练的稳定收敛。为了应对动态三维环境中的主动防御挑战，REIN -EAD 通过两个关键的稳定性特性来增强标准 PPO：一种两阶段的训练方法（离线感知预训练和随后的联合在线训练），加速了收敛；以及整合的监督学习作为正则化干预，用以维持感知能力，同时防止不稳定的策略更新。更多实验详见附录 C.9 。

替代奖励形式。我们进行了额外的消融实验，以比较 REIN-EAD 与面向不确定性的奖励塑形和两种替代奖励形式（即直接熵扣除和二元结果奖励）的表现。在附录 C.10 中提供了更多细节。如表 C。7 所示，我们提出的奖励塑形方法在干净精度和对抗补丁的鲁棒性方面优于其他方法。它采用了一种密集形式，加速了收敛，并指导模型学习能够最大化信息增益以实现准确感知的策略。

3.2 3D 物体分类评估

3.2.1 实验设置

应用于非微分环境。分类任务被广泛应用并且本质上容易受到补丁攻击，由于 3D 数据的稀缺，通常依赖于 3D 数据集合成的渲染技术 [?] 。然而，光栅化的离散性使得动作转换的分析推导不可微分 [?] ，限制了依赖于微分动态模型的 EAD

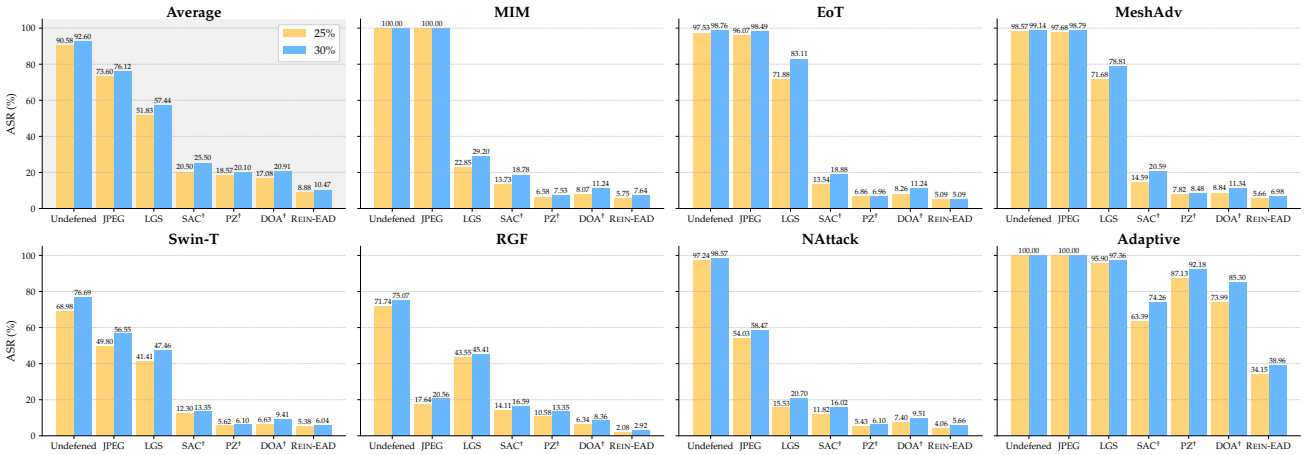

Fig. 7: 评估在具有不同补丁大小的攻击下 3D 对象分类模型的泛化能力。

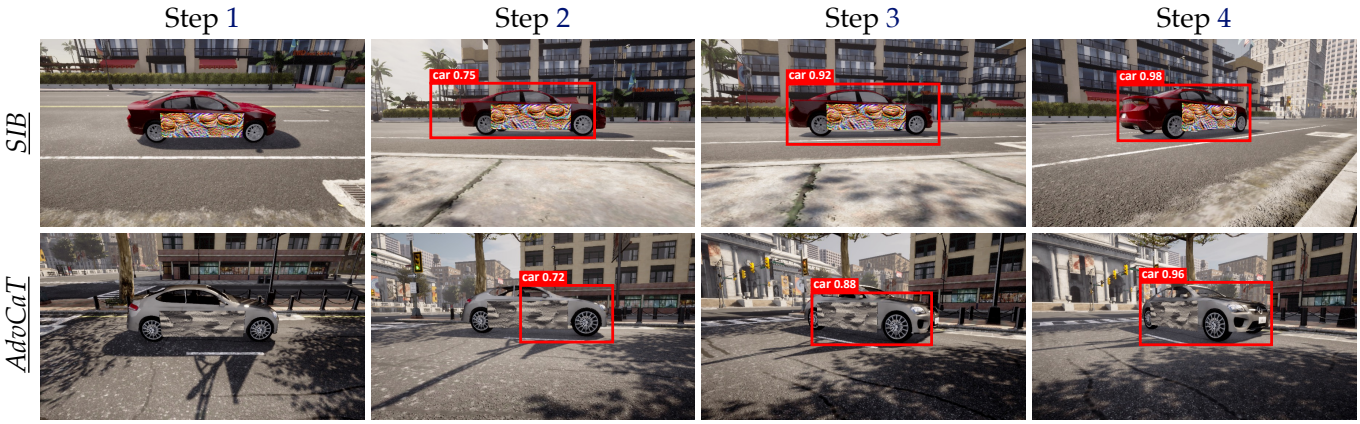

Fig. 8: CARLA 上 REIN -EAD 的定性结果。对抗性补丁覆盖了物体正视图边界框的 25 %。

TABLE 4: REIN -EAD 算法在不同奖励塑形下的性能表现。

Reward	Acc (%)	Attack Success Rate (%)			
		MIM	EoT	GenAP	3DAdv
Entropy Deduction	88.67	3.15	2.11	4.21	11.42
Outcome Reward	88.62	3.22	3.26	5.94	10.86
ours	89.03	2.10	3.15	7.37	4.21

TABLE 5: EG3D 中的目标检测性能。† 表示使用对抗样本进行训练。

Method	Average Precision (%)				
	Clean	EoT	SIB	UAP	AdvCaT
Undefended	88.55	9.06	17.30	20.29	28.38
JPEG	88.40	11.78	12.46	13.11	30.20
LGS	87.81	43.15	34.26	10.01	57.31
SAC	88.55	67.99	69.70	71.64	32.80
PZ	88.55	80.58	81.32	81.87	28.65
SAC †	88.55	70.10	71.08	74.06	40.67
PZ †	88.55	85.31	85.43	83.53	43.36
EAD	92.50	91.61	91.47	91.02	91.34
REIN-EAD	94.26	92.09	91.45	91.14	92.13

的应用。现有的可微分渲染框架 [?], [?], [?], [?] 要么缺乏精度，要么牺牲了渲染质量，无法满足 EAD 的训练要求。为了解决这个问题，我们引入了 REIN-EAD ，一个可以在非微分环境中有效操作的新框架。

动态 OmniObject3D。我们利用了最近提出的 OmniObject3D [?] ，这是最大规模的真实扫描 3D 数据集。OmniObject3D 与经典的 2D 数据集（例如 ImageNet [?] ）共享大量的通用类别，使其特别适合于评估 REIN-EAD 。该环境由 Pytorch3D [?] 和 Gym [?] 建立，能够根据智能体的动作在特定视角渲染对象。我们将此环境称为动态 OmniObject3D。环境建立的详细信息可在附录 D.1 中找到。

在纹理空间中的对抗性。在动态环境中应对对抗性威胁，我们使用 Pytorch3D 通过其差异后向传播管道 [3] 在 3D 网格纹理上实施补丁攻击。该方法能够从指定视角将 3D 对抗物体渲染为 2D 图像，确保在多个视图中对抗性外观的一致性。由于攻击改变了纹理空间，补丁的形状随视角变化，这对防御的概括性构成挑战。MeshAdv [?] 在其差异化渲染中考虑了 3D 变换的期望值，适用于多视角 3D 网格攻击。补丁影响物体前视图中边界框区域的 20 %。更多信息详见附录 D.3 。

实现细节。对于视觉骨干，我们使用了预训练的 Swin Transformer (Swin-S) [?] 在 ImageNet 上的模型，并对其在动态 OmniObject3D 上进行微调。在随后的实验中，Swin-S 的权重被冻结。我们实施了 REIN -EAD，采用类似于 FR 系统中使用的范式，利用 OAPA 算法对 REIN -EAD 进行补丁攻击，这些补丁占据物体边界框的 20 %。更多细节见附录 D.2 。

防御基线。我们采用与 FR 任务相同的防御基线，并对参数进行必要的调整以适应分类设置。值得注意的是，SAC、PZ 和 DOA 涉及对抗训练，这是一种广泛采用的增强稳健性的技术。更多细节见附录 D.4 。

3. Pytorch3D 促进了纹理微分，但不支持用于训练 EAD 的动作转换微分。

TABLE 6: CARLA 中的目标检测模型性能。† 表示使用对抗样本进行训练。

Method	Average Precision (%)				
	Clean	EoT	SIB	UAP	AdvCaT
Undefended	80.97	28.47	28.61	35.85	38.87
JPEG	81.57	36.50	35.80	34.66	37.28
LGS	80.32	74.73	72.64	57.76	51.34
SAC	79.78	27.06	28.55	42.03	37.38
PZ	80.70	62.43	59.35	49.24	37.90
SAC †	79.28	31.68	32.95	30.70	39.09
PZ †	80.91	76.10	75.50	75.24	40.51
REIN-EAD	83.15	82.82	81.97	82.12	82.86

3.2.2 实验结果

REIN -EAD 的效果评估。我们在表格 3 中的 OmniObject3D 测试集上，评估了在各种白盒、黑盒和自适应攻击下，REIN-EAD 及五种基线防御的稳健性。在大多数情况下，JPEG 和 LGS 未能提供有效的防御，而其他基线在白盒和黑盒设置下提供了保护。值得注意的是，REIN -EAD 显著降低了各种未见过的对抗者的攻击成功率，而不影响标准准确性。相比之下，尽管 SAC † 和 PZ † 可以利用 EoT 对抗者的先验知识净化对抗补丁，但由于遮罩特征的损失，它们不如 REIN -EAD。此外，在针对分类器和防御模块组合的更强自适应攻击下，稳健性差距被放大。我们可以看到，被动基线在自适应攻击下的性能显著下降。相反，REIN -EAD 在基线中实现了最强的稳健性。这些结果证明了我们的无模型学习方法的有效性及其在一般不可微分环境中的适用性。图 6 显示，即便最初被对抗者欺骗，REIN -EAD 仍能主动探索和观察环境，以纠正和完善其预测。

是对 REIN -EAD 的推广。我们进一步研究了 REIN -EAD 对不同补丁大小的适应性。具体而言，防御方法使用覆盖边界框 20 % 的补丁进行训练，随后用 25 % 和 30 % 大小的补丁来评估其性能。如图 7 所示，REIN -EAD 在遇到比训练阶段使用的补丁更大的补丁时仍保持其鲁棒性。

3.3 对象检测评价

3.3.1 实验设置

在自动驾驶中的目标检测比人脸验证和物体分类更具挑战性，因为模型必须在复杂背景中区分车辆，并在不可微分的现实环境中准确回归边界框。为了解决这一问题，我们设计了 EAD 和 REIN-EAD 用于目标检测，并在支持可微学习的 EG3D [?] 以及属于自动驾驶研究中使用的真实感环境的 CARLA [?] 上进行评估。

在可微分 EG3D 上的评估。我们采用 EG3D 的可微分生成框架，可以从可控视角生成多样的车辆类型，同时确保 3D 一致性。可微分环境允许验证 EAD 和 REIN -EAD 范式。更多详细信息可以在附录 E.1 中找到。

在不可微分的 CARLA 上的评估。我们在 CARLA 中评估 REIN -EAD 的防御能力，CARLA 是一个实用且不可微分的自动驾驶模拟器。模型的鲁棒性在 41 辆车辆上进行评估，涵盖了 CARLA 中所有可用的资产蓝图。有关 CARLA 环境的详细信息见附录 E.2 。

评估指标和对抗方法。交并比阈值从 50 % 到 95 % 的平均精度（AP@50:95）被用于作为评估指标。对抗方法包括 EoT [?] 、SIB [?] 、UAP [?] 和 AdvCaT [?] ，这些方法旨在发起隐藏攻击，使车辆从检测器中消失。SIB 影响隐藏层和最终层，在 REIN -EAD 之前干扰特征，而 AdvCaT 生成不明显的环境马赛克伪装。所有攻击方法使用 3D 变换的期望值来增强补丁对视点变化的弹性。附在车辆侧面的补丁占据了初始视角的边界框的 25% 。

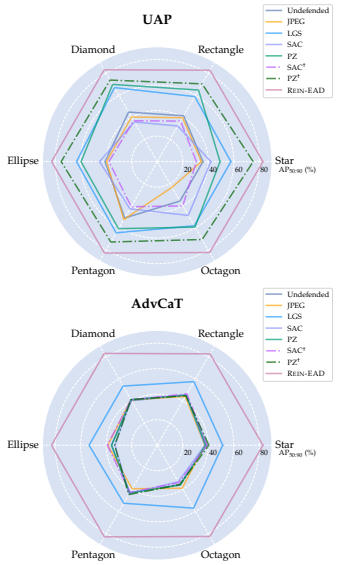

Fig. 9: 在 CARLA 中，比较评估了不同白盒攻击和补丁形状下的目标检测。

实施细节。预训练的 YOLOv5n [?] 被用作视觉骨干，与决策变压器结合以实现 REIN -EAD。有关 REIN -EAD 的更多详细信息可以在附录 E.3 中找到。防御基线遵循与以前任务相同的方法。更多详细信息可以在附录 E.5 中找到。

3.3.2 实验结果

在 EG3D 上的有效性。我们在来自 EG3D 的 200 个测试数据样本上评估了 REIN -EAD 和其他防御基准，面对旨在欺骗 YOLO 检测器的四种白盒攻击。表 5 显示，无防御模型和 JPEG 防御在所有攻击下都被突破。LGS 在 EoT、SIB 和 UAP 环境下性能不佳，但对 AdvCaT 有一定的抵抗力。SAC 和 PZ 在对抗噪声模式攻击时表现良好，增强版本（SAC † 和 PZ † ）的表现略胜一筹。PZ † 接近于干净的平均精度，但在面对隐蔽的 AdvCaT 时失效。REIN -EAD 在清晰和稳健的表现上超越了其他方法，在不同攻击下显著优于被动基准。

在 CARLA 上的有效性。我们在照片般逼真的 CARLA 环境中，以非差分和实际设置下评估了 REIN -EAD 的性能。表格 6 揭示了与其他任务具有相似趋势的观察结果。具体而言，LGS 在 EoT 和 SIB 攻击下表现更好，而 SAC 系列由于其完成机制的严重遮挡（在附录 E.6 中可视化）而明显下降。得益于循环时间反馈，REIN -EAD 在干净和对抗条件下都取得了最佳结果。图 8 展示了 REIN -EAD 在 SIB 和 AdvCaT 攻击下的防御过程，显示了通过交互和环境观察来提高的预测精度和边界框精度。更多的定性结果展示在附录 E.6 中。此外，我们在附录 E.9 中讨论了一些失败案例并分析了它们的影响。

对补丁形状的泛化。我们评估了一个在矩形上训练的 REIN -EAD，使用形状变化但保持固定补丁面积占用的对抗性补丁。正如图 9 所示，我们的模型在清晰和稳健的准确性方面均超过了其他方法，即使面临不同、未遇到的对抗性补丁形状。更多结果见附录 E.8 。这些发现进一步强调了 REIN -EAD 在处理未知对抗性攻击方面的出色泛化能力。

本文中，我们引入了一种新的主动防御框架，即强化体现主动防御（REIN -EAD），该框架可以有效缓解现实世界 3D 环境中的对抗性补丁攻击。REIN -EAD 利用对环境的探索和交互来关联环境信息，并完善对目标对象的理解。它通过多步骤交互来积累时间一致性，以平衡即时预测精度和长期熵最小化。此外，REIN -EAD 包含一个以不确定性为导向的奖励塑形机制，以在不需要可微分环境的情况下提高效率。大

量实验表明，REIN -EAD 显著提高了鲁棒性和泛化能力，并在复杂任务中具有很强的适用性。

Appendix A
证明与附加理论

A.1 定理 3.1 的证明

Proof. 对于一系列观察 $\{o_1, \cdots o_t\}$ 和一个由场景 x 确定的先前维护的信念 b_{t-1}，我们将方程（ 6 ）的左侧展开如下：

$$\begin{aligned}\mathbb{E}_x I(o_t; y \mid b_{t-1}) &= \mathbb{E}_{x,y} \log \frac{p(b_{t-1})p(b_{t-1}, o_t, y)}{p(b_{t-1}, y)p(b_{t-1}, o_t)} \\ &= \mathbb{E}_{x,y} \log \frac{p(y \mid b_{t-1}, o_t)}{p(y \mid b_{t-1})}.\end{aligned} \tag{A. 12}$$

通过在方程（ A. 12 ）的被积函数中引入变分分布 $q_\theta(y \mid o_1, \cdots o_t)$ 作为乘法因子，我们得到：

$$\begin{aligned}\mathbb{E}_x I(o_t; y \mid b_{t-1}) =& \mathbb{E}_{x,y} \log \frac{p(y \mid b_{t-1}, o_t) q_\theta(y \mid b_{t-1} o_t)}{p(y \mid b_{t-1}) q_\theta(y \mid b_{t-1}, o_t)} \\ &= \mathbb{E}_{x,y} \log \frac{q_\theta(y \mid b_{t-1}, o_t)}{p(y \mid b_{t-1})} \\ &+ \mathbb{E}_x D_{\mathrm{KL}}(p(y \mid b_{t-1}, o_t) \| q_\theta(y \mid b_{t-1}, o_t)).\end{aligned} \tag{A. 13}$$

KL 散度的非负性使我们能够为互信息建立一个下界：

$$\begin{aligned}\mathbb{E}_x I(o_t; y \mid b_{t-1}) &\geq \mathbb{E}_{x,y} \log \frac{q_\theta(y \mid b_{t-1}, o_t)}{p(y \mid b_{t-1})} \\ &= \mathbb{E}_{x,y} \log q_\theta(y \mid b_{t-1}, o_t) + \mathcal{H}(y \mid b_{t-1}),\end{aligned} \tag{A. 14}$$

其中 $\mathcal{H}(y \mid b_{t-1})$ 代表在信念 b_{t-1} 条件下的 y 的条件熵，且方程（ A. 14 ）是众所周知的 Barber 和 Agakov 界限 [?]。我们通过选择一个包含评论器 $\mathcal{E}\theta(bt-1, o_t, y)$ 并按数据密度 $p(b_{t-1}, o_t)$ 缩放的基于能量的变分族进行：

$$q_\theta(y \mid b_{t-1}, o_t) = \frac{p(y \mid b_{t-1})}{Z(b_{t-1}, o_t)} e^{\mathcal{E}_\theta(b_{t-1}, o_t, y)}, \tag{A. 15}$$

其中 $Z(b_{t-1}, o_t) = \mathbb{E}y e^{\mathcal{E}\theta(b_{t-1}, o_t, y)}$。将方程（ A. 15 ）中定义的分布代入方程（ A. 14 ）得到：

$$\begin{aligned}&\mathbb{E}_x I(o_t; y \mid b_{t-1}) \\ &\geq \mathbb{E}_{x,y} \log q_\theta(y \mid b_{t-1}, o_t) + \mathcal{H}(y \mid b_{t-1}) \\ &= \mathbb{E}_{x,y}[\mathcal{E}_\theta(b_{t-1}, o_t, y)] - \mathbb{E}_x[\log Z(b_{t-1}, o_t)],\end{aligned} \tag{A. 16}$$

这代表了 Barber 和 Agakov 界限的非规范化版本。应用任何 $g(b_{t-1}, o_t) > 0$ 的不等式 $\log Z(b_{t-1}, o_t) \leq \frac{Z(b_{t-1}, o_t)}{g(b_{t-1}, o_t)} + \log[g(b_{t-1}, o_t)] - 1$，并且当 $g(b_{t-1}, o_t) = Z(b_{t-1}, o_t)$ 时界限变得紧密，我们得到了一个称为 Barber 和 Agakov 互信息下界的可处理非规范化版本的可处理上界：

$$\begin{aligned}&\mathbb{E}_{x,y}[\mathcal{E}_\theta(b_{t-1}, o_t, y)] - \mathbb{E}_x[\log Z(b_{t-1}, o_t)] \\ &\geq \mathbb{E}_{x,y}[\mathcal{E}_\theta(b_{t-1}, o_t, y)] - \mathbb{E}_x\left[\frac{\mathbb{E}_y e^{\mathcal{E}_\theta(b_{t-1}, o_t, y)}}{g(b_{t-1}, o_t)}\right] \\ &- \mathbb{E}_x \log[g(b_{t-1}, o_t)] + 1 \\ &= 1 + \mathbb{E}_{x,y}\left[\log \frac{e^{\mathcal{E}_\theta(b_{t-1}, o_t, y)}}{g(b_{t-1}, o_t)}\right] - \mathbb{E}_x\left[\frac{\mathbb{E}_y e^{\mathcal{E}_\theta(b_{t-1}, o_t, y)}}{g(b_{t-1}, o_t)}\right].\end{aligned} \tag{A. 17}$$

为了减小方差，我们利用来自 $\mathcal{D}$ 的多个样本 $\{x^{(j)}, y^{(j)}\}j = 1^K$ 来实现互信息的低方差高偏差估计。对于来自不同场景 $(x^{(j)}, y^{(j)})$ $(j \neq i)$ 的观察轨迹，具有与 $x^{(i)}$ 和 $y^{(i)}$ 无关的注释 $\{y^{(j)}\}j = 1, j \neq i^K$，我们有：

$$g(b_{t-1}, o_t) = g(b_{t-1}, o_t; y^{(1)}, \cdots, y^{(K)}). \tag{A. 18}$$

这使我们能够利用额外的样本 $\{x^{(j)}, y^{(j)}\}j = 1^K$ 来构建函数 $Z(bt-1, o_t)$ 的蒙特卡洛估计：

$$\begin{aligned}g(b_{t-1}, o_t; y_1, \cdots y_K) =& m(b_{t-1}, o_t; y^{(1)}, \cdots y^{(K)}) \\ =& \frac{1}{K} \sum_{j=1}^{K} e^{\mathcal{E}_\theta(y^{(j)} | b_{t-1}, o_t)}.\end{aligned}$$

当对 K 样本估计界限时，方程（ A. 17 ）的最后一项缩减为常数值 1：

$$\begin{aligned}&\mathbb{E}_x\left[\frac{\mathbb{E}_{y^{(1)}, \cdots, y^{(K)}} e^{\mathcal{E}_\theta(b_{t-1}, o_t, y)}}{m(b_{t-1}, o_t; y^{(1)}, \cdots y^{(K)})}\right] \\ =& \mathbb{E}_{x_1}\left[\frac{\frac{1}{K}\sum_{j=1}^{K} e^{\mathcal{E}_\theta(b_{t-1}, o_t^{(1)}, y^{(j)})}}{m(b_{t-1}, o_t^{(1)}; y^{(1)}, \cdots y^{(K)})}\right] = 1.\end{aligned} \tag{A. 19}$$

将方程（ A. 19 ）应用于方程（ A. 17 ）并对 K 个样本求平均（将每项中的 $x^{(1)}$ 重新索引为 $x^{(j)}$），我们准确地恢复了 Oord *et al.* [?] 提出的互信息下界：

$$\begin{aligned}&1 + \mathbb{E}_{x^{(j)}, y^{(j)}}\left[\log \frac{e^{\mathcal{E}_\theta(b_{t-1}^{(j)}, y^{(j)})}}{g(b_{t-1}^{(j)}; y^{(1)}, \cdots, y^{(K)})}\right] \\ &- \mathbb{E}_{x^{(j)}}\left[\frac{\mathbb{E}_{y^{(j)}} e^{\mathcal{E}_\theta(b_{t-1}^{(j)}, o_t^{(j)}, \hat{y}^{(j)})}}{g(b_{t-1}^{(j)}, o_t^{(j)}; y^{(1)}, \cdots, ^{(K)})}\right] \\ &= \mathbb{E}_{x^{(j)}, y^{(j)}}\left[\log \frac{e^{\mathcal{E}_\theta(b_{t-1}^{(j)}, o_t^{(j)}, y^{(j)})}}{g(b_{t-1}^{(j)}, o_t^{(j)}; y^{(1)}, \cdots, y^{(K)})}\right] \\ &= \mathbb{E}_{x^{(j)}, y^{(j)}}\left[\log \frac{e^{\mathcal{E}_\theta(b_{t-1}^{(j)}, o_t^{(j)}, y^{(j)})}}{\frac{1}{K}\sum_{\hat{y}^{(j)}} e^{\mathcal{E}_\theta(b_{t-1}^{(j)}, o_t^{(j)}, \hat{y}^{(j)})}}\right].\end{aligned} \tag{A. 20}$$

在方程（ A. 12 ）中将被积函数乘以和除以 $\frac{p(y|b_{t-1}^{(j)}, o_{t-1})}{Z(b_{t-1}^{(j)}, o_t)}$，并从括号中提取 $\frac{1}{K}$ 使方程转变为：

$$\mathbb{E}_{x^{(j)}, y^{(j)}}\left[\log \frac{q_\theta(b_{t-1}^{(j)}, o_t^{(j)}, y^{(j)})}{\sum_{\hat{y}^{(j)}} q_\theta(b_{t-1}^{(j)}, o_t^{(j)}, \hat{y}^{(j)})}\right] + \log(K). \tag{A. 21}$$

因此，我们得到

$$\begin{aligned}&\mathbb{E}_{(x^{(j)}, y^{(j)}) \sim \mathcal{D}}\left[\frac{1}{K}\sum_{j=1}^{K} \log \frac{q_{\boldsymbol{\theta}}(y^{(j)} \mid b_{t-1}^{(j)}, o_t^{(j)})}{\frac{1}{K}\sum_{k=1}^{K} q_{\boldsymbol{\theta}}(y^{(k)} \mid b_{t-1}^{(j)}, o_t^{(j)})}\right] \\ \leq& \mathbb{E}_x I(o_t; y \mid b_{t-1}) - \frac{\log(K)}{K}.\end{aligned} \tag{A. 22}$$

给定一个场景注解 y，我们通过时间步 t 时的条件熵来量化注解 y 的不确定性，条件熵以给定观测序列 $\{b_{t-1}, o_t\}$ 的 y 表示为 $\mathcal{H}(y \mid b_{t-1}, o_t)$。在以下部分，我们演示了条件互信息 $I(o_t; y \mid b_{t-1})$ 等价于条件熵的减少：

$$\begin{aligned}&I(o_t; y \mid b_{t-1}) \\ =& I(y; b_{t-1}, o_t) - I(y; b_{t-1}) \\ =& [\mathcal{H}(y) - I(y; b_{t-1})] - [\mathcal{H}(y) - I(y; b_{t-1}, o_t)] \\ =& \mathcal{H}(y \mid b_{t-1}) - \mathcal{H}(y \mid b_{t-1}, o_t).\end{aligned}$$

这里推导中的第一个等式来自 Kolmogorov 身份 [?]。 □

定理 3.1 使我们能够在方程（ 6 ）的互信息和定义 3.3 中定义的贪婪信息探索之间建立一个联系。因此，我们可以推断出优化方程（ 5 ）中 InfoNCE 目标的 EAD 策略模型与贪婪信息探索原则之间的关系。

A.2 定理 3.7 的证明

Proof. 我们的目标是证明，当假设信念更新函数 f_b 是双射时，累积信息策略 π^* 在减少 y 的熵方面的效果大于或等于贪心信息策略 π^g 。回顾一下，在给定策略 π 下，从时间 0 到 H 的信息增益定义为 y 的熵减少：

$$\Delta\mathcal{H}_\pi = \mathcal{H}(y) - \mathcal{H}(y \mid b_{H-1}, o_H),$$

其中 b_{H-1} 和 o_H 分别是时间 $H-1$ 和 H 在策略 π 下的信念和观察结果。具体地，对于两种策略：

$$\Delta\mathcal{H}_{\pi^*} = \mathcal{H}(y) - \mathcal{H}(y \mid b^*_{H-1}, o^*_H),$$
$$\Delta\mathcal{H}_{\pi^g} = \mathcal{H}(y) - \mathcal{H}(y \mid b^g_{H-1}, o^g_H).$$

我们首先考虑每个时间步的累积信息增益。对于每个时间步 $t=1$ 到 H ，定义策略 π 下的增量信息增益为：

$$\begin{aligned}\Delta\mathcal{H}^\pi_t =& \mathcal{H}(y \mid b^\pi_{t-1}, o^\pi_t) - \mathcal{H}(y \mid b^\pi_t, o^\pi_{t+1}) \\ =& \left[\mathcal{H}(y \mid b^\pi_{t-1}, o^\pi_t) - \mathcal{H}(y \mid b^\pi_t)\right] \\ &+ \left[\mathcal{H}(y \mid b^\pi_t) - \mathcal{H}(y \mid b^\pi_t, o^\pi_{t+1})\right]. \end{aligned} \tag{A. 23}$$

因为 f_b 是双射的，每个观察 o_t 唯一地决定了后续的信念 b_t ，反之亦然。这种双射性确保了信念和观察之间的映射可以无损地保留关于 y 的信息：

$$\mathcal{H}(y \mid b^\pi_{t-1}, o^\pi_t) - \mathcal{H}(y \mid b^\pi_t) = 0. \tag{A. 24}$$

因此，我们得到

$$\Delta\mathcal{H}_\pi = \sum_{t=1}^{H} \Delta\mathcal{H}^\pi_t = \sum_{t=1}^{H} \left[\mathcal{H}(y \mid b^\pi_t) - \mathcal{H}(y \mid b^\pi_t, o^\pi_{t+1})\right]. \tag{A. 25}$$

为此，我们比较这两种策略的轨迹信息增益：

$$\begin{aligned}&\Delta\mathcal{H}_{\pi^*} - \Delta\mathcal{H}_{\pi^g} \\ &= \left(\sum_{t=1}^{H} \Delta\left[\mathcal{H}(y \mid b^*_t) - \mathcal{H}(y \mid b^*_t, o^*_{t+1})\right]\right) \\ &\quad - \left(\sum_{t=1}^{H} \Delta\left[\mathcal{H}(y \mid b^g_t) - \mathcal{H}(y \mid b^g_t, o^g_{t+1})\right]\right).\end{aligned}$$

因为 π^* 最大化 $\Delta\mathcal{H}$ 考虑到 H 步的轨迹，我们得到：

$$\begin{aligned}&\left(\sum_{t=1}^{H} \Delta\left[\mathcal{H}(y \mid b^*_t) - \mathcal{H}(y \mid b^*_t, o^*_{t+1})\right]\right) \\ &- \left(\sum_{t=1}^{H} \Delta\left[\mathcal{H}(y \mid b^g_t) - \mathcal{H}(y \mid b^g_t, o^g_{t+1})\right]\right) \geq 0,\end{aligned} \tag{A. 26}$$

等式成立当且仅当对于所有 t ， $\Delta\mathcal{H}^*_t = \Delta\mathcal{H}^g_t$ ，这种情况恰好在问题表现出最优子结构时发生。在这种情况下，贪心策略 π^g 天然地如同累积策略 π^* 一样有效地积累信息增益。 □

A.3 奖励的推导

给定密集奖励 r_t 的定义来自方程 (11)，我们代入到轨迹奖励 $\mathcal{R}(\tau)$ 的表达式中：

$$\begin{aligned}\mathcal{R}(\tau) =& \sum_{t=1}^{H} \gamma^{t-1}\left[\mathcal{L}(\hat{y}_{t-1}, y) - \gamma\cdot\mathcal{L}(\hat{y}_t, y)\right] \\ =& \sum_{t=1}^{H} \gamma^{t-1}\mathcal{L}(\hat{y}_{t-1}, y) - \sum_{t=1}^{H} \gamma^t\mathcal{L}(\hat{y}_t, y).\end{aligned} \tag{A. 27}$$

注意到，第二个求和现在从 $t=1$ 到 $t=H$ ，这与第一个求和的索引相差一个。因此，这两个求和可以如下对齐：

$$\begin{aligned}\mathcal{R}(\tau) =& \gamma^0\mathcal{L}(\hat{y}_0, y) + \gamma^1\mathcal{L}(\hat{y}_1, y) + \cdots + \gamma^{H-1}\mathcal{L}(\hat{y}_{H-1}, y) \\ &- \left(\gamma^1\mathcal{L}(\hat{y}_1, y) + \gamma^2\mathcal{L}(\hat{y}_2, y) + \cdots + \gamma^H\mathcal{L}(\hat{y}_H, y)\right).\end{aligned}$$

由于该级数的望远特性，所有中间项都会抵消：

$$\mathcal{R}(\tau) = \mathcal{L}(\hat{y}_0, y) - \gamma^H\mathcal{L}(\hat{y}_H, y). \tag{A. 28}$$

因此，累积折扣奖励 $\mathcal{R}(\tau)$ 相当于初始损失减去折扣后的最终损失。由于初始损失 $\mathcal{L}(\hat{y}_0, y)$ 仅由初始状态分布 ρ_0 决定，并且与策略无关，因此在策略优化期间它可以被视为一个常数。这就完成了证明，这种形式的奖励与方程 (9) 中的目标是一致的。

Appendix B
模拟环境的实验细节

环境动态。我们正式将状态 $s_t = (h_t, v_t)$ 定义为在时刻 t 相机的偏航角 $h_t \in \mathbb{R}$ 和俯仰角 $v_t \in \mathbb{R}$ 的组合，而动作被定义为由 $a_t = (\Delta h, \Delta v)$ 表示的连续旋转。因此，转移函数表示为 $T(s_t, a_t, x) = s_t + a_t$ ，表明下一个状态是通过将动作的旋转加到当前状态上获得的。观察函数使用 3D 生成模型重新表述，记为 $O(s_t, x) = \mathcal{R}(s_t, x)$ ，其中 $\mathcal{R}(\cdot)$ 表示一个渲染器（例如，一个 3D 生成模型或图形引擎），根据由状态 s_t 确定的相机参数生成 2D 图像观察 o_t 。

在计算图形中，渲染器的详细公式表示为：

$$o_t = \mathcal{R}'(\boldsymbol{E}_t, \boldsymbol{I}, x), \tag{B. 29}$$

，其中 $\boldsymbol{E}_t \in \mathbb{R}^{4\times4}$ 表示由状态 s_t 决定的相机外部矩阵，而 $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ 是预定义的相机内在矩阵。为了利用渲染器生成二维图像，我们需要根据状态 s_t 计算相机的外部矩阵 $\boldsymbol{E}_t$ 。假设使用右手坐标系和列向量，我们有：

$$\boldsymbol{E}_t = \begin{bmatrix}\boldsymbol{R}_t & \boldsymbol{T} \\ \boldsymbol{0} & 1\end{bmatrix}, \tag{B. 30}$$

，其中 $\boldsymbol{R}_t \in \mathbb{R}^{3\times3}$ 是由 s_t 确定的旋转矩阵，$\boldsymbol{T} \in \mathbb{R}^{3\times1}$ 是不变的平移向量。偏航 h_t 和俯仰 v_t 的旋转矩阵表示为：

$$\boldsymbol{R}^y(h_t) = \begin{bmatrix}\cos(h_t) & 0 & \sin(h_t) \\ 0 & 1 & 0 \\ -\sin(h_t) & 0 & \cos(h_t)\end{bmatrix}, \tag{B. 31}$$

$$\boldsymbol{R}^x(v_t) = \begin{bmatrix}1 & 0 & 0 \\ 0 & \cos(v_t) & -\sin(v_t) \\ 0 & \sin(v_t) & \cos(v_t)\end{bmatrix}. \tag{B. 32}$$

组合旋转 $\boldsymbol{R}_t$ 是通过将各个旋转矩阵相乘得到的：$\boldsymbol{R}_t = \boldsymbol{R}^y(h_t) \times \boldsymbol{R}^x(v_t)$ 。然后，完整的外部矩阵被构建为：

$$\boldsymbol{E}_t = \begin{bmatrix}\boldsymbol{R}^y(h_t) \times \boldsymbol{R}^x(v_t) & \boldsymbol{T} \\ \boldsymbol{0} & 1\end{bmatrix}. \tag{B. 33}$$

应用函数。在实验中，对抗性贴图被附在一个平面表面上，例如用于人脸识别的眼镜和用于物体检测的广告牌。通过利用已知的世界坐标系中对抗性贴图的角坐标，采用外部矩阵 $\boldsymbol{E}_t$ 和内部矩阵 $\boldsymbol{K}$ 来渲染包含对抗性贴图的图像观测。遵循 Zhu *et al.* [?] 描述的 3D 贴图投影过程来构建应用函数。投影矩阵 $\boldsymbol{M}_{\text{3d-2d}} \in \mathbb{R}^{4\times4}$ 被指定为：

$$\boldsymbol{M}_{\text{3d-2d}} = \begin{bmatrix}\boldsymbol{K} & \boldsymbol{0} \\ \boldsymbol{0} & 1\end{bmatrix} \times \boldsymbol{E}_t. \tag{B. 34}$$

这个过程是可微分的，使得可以优化对抗性补丁。

TABLE C。1 : CelebA-3D 的定量评估。图像尺寸为 112×112 。

	PSNR↑	SSIM↑	LPIPS↓	ID↑
CelebA-3D	21.28	.7601	.1314	.5771

综上所述，提出了一种确定性环境模型，该模型适用于所有实验环境（例如，EG3D、CARLA）：

$$\begin{aligned} \text{State} \quad & s_t = (h_t, v_t) \in \mathbb{R}^2, \\ \text{Action} \quad & a_t = (\Delta h, \Delta v) \in \mathbb{R}^2, \\ \text{Transition Function} \quad & T(s_t, a_t, x) = s_t + a_t, \\ \text{Observation Function} \quad & Z(s_t, x) = \mathcal{R}(s_t, x). \end{aligned}$$

不同任务的模拟之间的主要区别在于可行的视点区域。这些区域在每个任务的实现部分中有详细说明，具体见附录 C ，?? & D 。

Appendix C
人脸识别实验细节

C.1 CelebA-3D

我们使用 EG3D 的 GAN Inversion 的非官方实现（https://github.com/oneThousand1000/EG3D-projector ），利用默认参数将 CelebA 数据集 [?] 中的 2D 图像转换为 3D 潜在表示 w^+ 。作为 3D 生成模型的先验，我们利用了在 FFHQ 数据集上预训练的 EG3D 模型，这些模型已在 https://catalog.ngc.nvidia.com/orgs/nvidia/teams/research/models/eg3d 上正式发布。为了减少计算开销，我们放弃了 EG3D 的超分辨率模块，并使用其神经渲染器直接渲染分辨率为 112×112 像素的 RGB 图像。

我们对重构的 CelebA-3D 数据集进行了全面评估，以评估其质量。图像质量通过原始图像与同一视角下由 EG3D 生成的图像之间的 PSNR、SSIM 和 LPIPS 来衡量。这些指标提供了一个多方面的评估，用以衡量重构图像相对于其 2D 对照的大致逼真程度。此外，我们引入了一种修改后的身份一致性（ID）指标，稍微偏离了在 [?] 中提出的那个，用以评估重构的 3D 脸与其原始 2D 脸之间的身份一致性。我们的 ID 指标计算了由随机相机位置渲染的脸部视图与其 CelebA 中原始图像间的平均 ArcFace [?] 余弦相似性分数。

表格 C。1 中展示的结果表明，学到的关于 FFHQ 的 3D 先验能够实现非常高质量的单视图几何重建。因此，我们重建的 CelebA-3D 数据集展示出高图像质量，与原始 2D 形式具有足够的身份一致性，使其适合后续实验。图 C。1 展示了一组选定的重建多视角人脸。

CelebA-3D 数据集结合了广泛使用的 CelebA 数据集提供的丰富标注，可以通过 https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html 访问

C.2 攻击细节

针对人脸识别系统的对抗性攻击中的伪装和规避。在针对人脸识别系统的对抗性攻击中，两个主要目标是伪装和规避。伪装涉及操控图像以欺骗面部识别算法，使其错误地将一个人识别为另一个人。规避则是通过对图像进行战略性修改来防止面部识别系统的准确识别。这些修改旨在要么完全阻止面部识别系统检测到人脸，要么模糊检测到的人脸与其在系统数据库中的真实身份之间的关联。这些子任务的性质突显了它们对人脸识别系统的安全性和完整性所构成的深刻挑战。

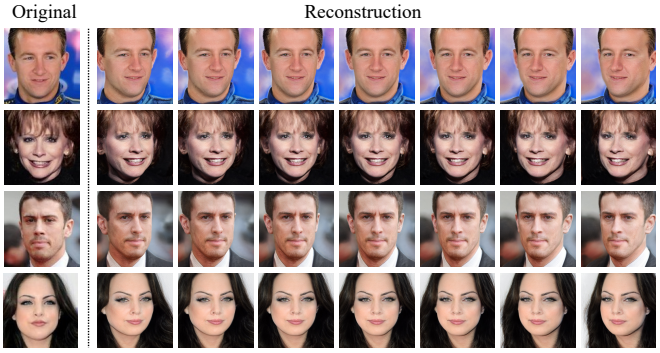

Fig. C。1 : 第一列展示了来自 CelebA 的源人脸，随后的列展示了使用 EG3D 从逆向 w^+ 生成的多视图人脸。每个渲染人脸图像的尺寸是 112×112 。

像素空间中的攻击。动量迭代方法（MIM）[?] 和转变期望（EoT）[?] 代表了在 RGB 像素空间内改进对抗性补丁的最新技术。MIM 通过在优化过程中引入动量来增强对抗样本的生成，从而促进这些样本在不同模型和设置中的可转移性。相比之下，EoT 使用多种变换方法，如旋转和照明变化，以提高攻击在物理条件下的鲁棒性。为了确保最佳性能，我们按照文献 [?] 中的建议参数，将迭代次数设置为 $N = 150$ ，学习率设置为 $\alpha = 1.5/255$ ，衰减因子设置为 $\mu = 1$ 。这些参数在所有实验条件下始终保持一致。此外，EoT 的采样频率设定为 $M = 10$ ，以平衡计算效率和攻击效果。

在生成模型的潜在空间进行攻击。GenAP [?] 和 Face3DAdv [?] 代表了将对抗性补丁的关注点从像素空间转移到生成对抗网络（GAN）潜在空间的开创性方法。通过利用生成能力，GAN 开发出能够欺骗面部识别系统的对抗性补丁，并且考虑到 3D 变化。此外，我们在 EG3D 的潜在空间中使用 Adam 优化器 [?] 进行补丁优化，学习率为 $\eta = 0.01$ ，迭代次数为 $N = 150$ 。采样频率设定为 $M = 10$ 。

C.3 自适应攻击的细节

针对防御基线的自适应攻击。为了对无参数、基于净化的防御机制如 JPEG 和 LGS 发动自适应攻击，我们采用 Athalye 提出的反向传递可微分近似 (BPDA) *et al.* [?] 。该方法假设每个防御机制的输出接近原始输入。对于 SAC 和 PZ 的自适应攻击，我们使用他们的官方实现 [?] ，通过阈值操作期间的反向传播整合直接传递估计器 (STE) [?] 。

自适应攻击使用统一上集策略。在针对 REIN -EAD 的自适应攻击中，我们利用统一上集近似来处理策略模型。因此，我们有替代策略

$$\tilde{\pi} := \mathcal{U}(h_{\min}, h_{\max}) \times \mathcal{U}(v_{\min}, v_{\max}), \tag{C. 35}$$

其中 $a_t \in [h_{\min}, h_{\max}] \times [v_{\min}, v_{\max}]$ ，而 $[h_{\min}, h_{\max}]$ ，$[v_{\min}, v_{\max}]$ 分别表示水平旋转（偏航）和垂直旋转（俯仰）的预定义可行区域。优化目标如下所示，为了清晰起见，给出简化的顺序表示 [4] ：

$$\begin{aligned} \max_{p} \quad & \mathbb{E}_{s_0 \sim \rho_0, a_i \sim \tilde{\pi}} \mathcal{L}(\overline{y}_\tau, y), \\ \text{with} \quad & \{\overline{y}_\tau, b_\tau\} = f(\{A(o_i, p; s_0 + \sum_{j=0}^{i-1} a_j)\}_{i=0}^{\tau}; \boldsymbol{\theta}) \\ \text{s.t.} \quad & p \in [0,1]^{H_p \times W_p \times C}, \end{aligned} \tag{C. 36}$$

4. 为简化说明，本节中递归推理过程按顺序呈现。

，其中 ρ_0 表示初始状态 s_0 的分布，$\mathcal{L}$ 是任务特定的损失函数。

针对子模块的自适应攻击。端到端攻击可能并不总是最有效的策略，特别是对于那些具有复杂前向传递的防御措施而言。通常，针对最薄弱的组件进行攻击就足够了。因此，我们提出了两种独立的自适应攻击：一种针对感知模型，另一种针对策略模型。对感知模型的攻击旨在生成一个对抗性补丁，从而破坏内部信念 b_t [?] 。该攻击的优化目标是最大化被破坏的信念 $b\tau$ 与良性信念 $b\tau^+$ 之间的欧几里得距离，具体表述如下：

$$\begin{aligned} \max_{p} \quad & \mathbb{E}_{s_0\sim\rho_0, a_i\sim\tilde{\pi}} \|b_\tau - b_\tau^+\|_2^2, \\ \text{with} \quad & \{\overline{y}_\tau, b_\tau\} = f(\{A(o_i, p; s_0 + \sum_{j=0}^{i-1} a_j)\}_{i=1}^{\tau}; \boldsymbol{\theta}), \\ & \{\overline{y}_\tau^+, b_\tau^+\} = f(\{o_i\}_{i=0}^{\tau}; \boldsymbol{\theta}), \\ \text{s.t.} \quad & p \in [0,1]^{H_p \times W_p \times C}. \end{aligned} \tag{C. 37}$$

针对策略模型的攻击，目标是创建一个对抗性补丁，使策略模型输出零动作 $a_i = \pi(b_i; \boldsymbol{\phi}) = 0$ ，从而使模型保持不变状态 $s_i = s_1$ 并生成错误预测 $\overline{y}_\tau$ 。虽然在策略输出作为约束的情况下，原问题可能具有挑战性，但我们使用拉格朗日松弛方法将约束纳入目标中，并解决以下问题：

其中 $c > 0$ 是一个常数，它产生一个对抗样本，确保模型输出为零操作。

对整个管道的自适应攻击。由于随着轨迹长度增加，GPU 内存的快速消耗，通过反向传播攻击模型变得不可行（例如，4 步需要近 90 GB 的视频内存）。为了解决这个问题，我们使用梯度检查点 [?] 来减少内存消耗。通过选择性地重新计算由 H 步推理过程定义的计算图的一部分，而不是存储它们，这种技术在一定的额外计算成本下有效地降低了内存成本。使用这种方法，我们成功地沿着 4 步轨迹使用 NVIDIA RTX 3090 Ti 攻击了整个管道，但仍然无法将其扩展到攻击 16 步 Rein -EAD。

关于实现，我们采用与 Face3DAdv 相同的超参数，并考虑在附录 C.5 中定义的动作范围。对于针对策略模型的自适应攻击中的常数 c ，我们使用二分搜索根据 Carlini *et al.* [?] 来识别最佳值，发现 $c = 100$ 最有效。此外，这些自适应攻击的评估结果和分析详见附录 C.7 ，主要结果在表 C。6 中。

C.4 防御的细节

JPEG 压缩。我们将质量参数设置为 75 。

局部梯度平滑。我们采用 https://github.com/fabiobrau/local_gradients_smoothing 的实现方法，并保持 [?] 中声明的默认超参数。

分割和完成。我们在 https://github.com/joellliu/SegmentAndComplete 处使用官方实现，并使用由 EoT [?] 和 USAP 技术分别优化的对抗性补丁进行重新训练。我们采用 [?] 中声明的相同超参数和训练过程，除了先前的补丁大小，我们将其按比例调整为输入图像的大小。

Patchzero。对于 Patchzero [?] ，我们直接使用 SAC 的训练好的 patch 分割器，因为它们的训练流程几乎相同。

防御遮挡攻击。DOA 是一种基于对抗训练的方法。我们采用 DOA 训练范式，对相同的预训练 IResNet-50 和训练数据进行微调，代码在 https://github.com/P2333/Bag-of-Tricks-for-AT 中，与 Rein -EAD 相同。至于超参数，我们采用了 [?] 中的默认超参数，并按比例缩放了训练补丁大小。

TABLE C。2 : 用于人脸识别的 Rein -EAD 的超参数。

Hyper-parameter	Value
Lower bound for horizontal rotation ($h_{\min}$)	-0.35
Upper bound for horizontal rotation ($h_{\max}$)	0.35
Lower bound for vertical rotation ($v_{\min}$)	-0.25
Upper bound for vertical rotation ($v_{\max}$)	0.25
Ratio of patched data (r_{patch})	0.4
Training epochs for offline phase ($\text{lr}_{\text{offline}}$)	50
learning rate for offline phase ($\text{lr}_{\text{offline}}$)	10^{-3}
batch size for offline phase (b_{offline})	64
Learning rate for online phase ($\text{lr}_{\text{online}}$)	2.5×10^{-4}
Batch size for online phase (b_{online})	256
Return attenuation factor (γ)	0.95
Updates per iteration (n)	2

C.5 实现细节

模型细节。在人脸识别的背景下，我们实现了 Rein -EAD 模型，该模型由一个预训练的人脸识别特征提取器和一个决策转换器组成，具体来说，我们选择 IResNet-50 作为视觉骨干提取特征。对于每个时间步长 $t > 0$ ，我们使用 IResNet-50 将当前维度为 $112 \times 112 \times 3$ 的观察输入映射为一个维度为 512 的嵌入向量。然后将该嵌入向量与先前提取的维度为 $(t-1) \times 512$ 的嵌入序列连接，从而形成用于决策转换器输入的观察嵌入时间序列（$t \times 512$ ）。决策转换器随后输出用于推理的时间融合人脸嵌入以及预测的动作。对于训练过程，我们进一步使用线性投影层将融合的人脸嵌入映射为 logits。为了简化起见，我们直接使用 Softmax 损失函数进行模型训练。

在实验中，我们使用在 MS1MV3 上预训练的带有 ArcFace 边距的 IResNet-50 [?] 从 `InsightFace` [?] ，可在 https://github.com/deepinsight/insightface/tree/master/model_zoo 获得。

课程训练。观察到当从头同时训练感知和策略模型时，训练会遇到相当大的不稳定性。一个主要的问题是，感知模型在其初始训练阶段无法提供准确的监督信号，这导致策略网络生成不合理的动作并阻碍整个学习过程。为缓解此问题，我们首先使用从随机动作策略获得的帧独立训练感知模型，即离线阶段。一旦感知模型达到稳定性能，我们继续进入在线阶段，联合训练感知和策略网络，使用算法 1，从而确保它们的有效协调与学习。

同时，在第一阶段使用预先收集的数据进行离线学习被证明比通过从环境中交互收集数据的在线学习要显著更高效。通过将训练过程分为离线和在线两个不同的阶段，我们大大提高了训练效率并降低了计算成本。

训练细节。为了训练 Rein -EAD 进行人脸识别，我们从 CelebA-3D 的训练集中随机抽取来自 2,500 个不同身份的图像。我们采用之前展示过的分阶段训练模式，其超参数列在表 C。2 中。

C.6 计算开销

本节评估我们的方法在面部识别系统中与其他被动防御基线相比的计算开销。性能评估是在 NVIDIA GeForce RTX 3090 Ti 和 AMD EPYC 7302 16 核处理器上进行的，使用的训练批量大小为 64。SAC 和 PZ 需要训练一个分割器来识别补丁区域，包括两个阶段：首先对预生成的对抗图像进行初始训练，然后进行自对抗训练 [?], [?] 。DOA 是一种基于对抗训练的方法，需要对特征提取器进行重新训练 [?] 。此外，Rein -EAD 的训练包括离线和在线阶段，不涉及对抗训练。

如表 C。3 所示，尽管在在线训练阶段微分渲染需要显著的计算需求，但我们 REIN -EAD 模型的总训练时间在纯对抗训练方法 DOA 和部分对抗方法如 SAC 和 PZ 之间有效平衡。这种效率主要源于我们独特的 USAP 方法，它避免了生成对抗样本的需要，从而提高了训练效率。在模型推理方面，我们的 REIN -EAD 与 PZ 和 DOA 相比，展示了优于 LGS 和 SAC 的速度。这是因为后者方法需要 CPU 密集型的规则基础图像预处理，降低了它们的推理效率。

关于详细训练，REIN -EAD 模型按照附录 C.5 中的配置进行了训练。离线训练使用了 4 块 NVIDIA GeForce RTX 3090 Ti 显卡，大约持续了 2.5 小时（150 分钟）。由于强化学习对采样的要求增加，在线训练阶段延长至约 12 小时（728 分钟），并使用了 8 块 NVIDIA GeForce RTX 3090 Ti 显卡。

关于其关键组件的计算成本，端到端推理管道在面部识别任务下每个实例达到 11.5 毫秒的速度。感知模型占据此处理时间的 98.4%（IResNet50 主干：8.13 毫秒；因果变换器：3.19 毫秒），而轻量级策略 MLP 仅需 0.18 毫秒。此外，离线补丁近似（OAPA）阶段发生在训练之前，不会增加训练或推理阶段的计算负担。整个大约 50,000 个实例的训练集在大约 20 分钟的可接受时间内通过批处理使用八个 NVIDIA RTX 3090 GPU 进行处理。结果表明，在推理期间，感知模型占据了 REIN-EAD 大部分的计算成本。这些发现表明，未来关于优化 REIN-EAD 计算效率的工作应该主要关注于感知模型。

C.7 更多评估结果

不同补丁大小下的评估。为了进一步评估 REIN -EAD 模型在不同补丁大小和攻击方法下的普适性，我们进行了实验，包括模拟攻击和躲避攻击。这些攻击与表 1 所示的设置有相似之处。尽管补丁大小不同，表 C。4 和表 C。5 中的结果与表 1 中显示的结果非常相似。这种一致性进一步支持了 REIN -EAD 模型在解决未见过的攻击方法和适应不同补丁大小方面的适应性。

不同自适应攻击下的评估。如表 C。6 所示，我们使用 USP 的原始自适应攻击比追踪 EAD 的真实策略更加有效（整体而言）。这可能归因于阻碍优化的消失或爆炸梯度 [?] 。我们的策略通过对均匀策略分布的期望计算方法有可能缓解这一问题。同时，结果重申了 REIN -EAD 在各类自适应攻击下的鲁棒性。进一步显示，REIN -EAD 的防御能力源自于其策略和感知模型的协同集成，促进了战略性观测的收集，而不是学习一种捷径策略以从特定视角中和对抗性补丁。

C.8 更多定性结果

不同版本 SAC 的定性比较。SAC 是一种基于预处理的方法，它采用分割模型来检测补丁区域，然后通过“形状完成”技术将预测区域扩展为更大的方块，并去除可疑区域 [?] 。如图 C。2 所示，增强版 SAC 在诸如人脸识别等场景中表现出优越的分割性能，但无意中增加了遮蔽眼睛和鼻子等重要面部特征的可能性。这导致人脸识别模型正确识别个体的能力下降，从而影响其躲避攻击的性能。

C.9 策略训练的稳定收敛

REIN -EAD 在标准的 PPO 算法中融入了几项关键设计选择，以促进策略训练过程中的稳定收敛。首先，为了增强训练的稳定性和效率，我们避免从一开始就同时训练感知模型和策略模型。早期的感知模型提供不可靠的指导，导致不稳定的策略行动并阻碍进展。我们的解决方案包括两个阶段：初始的离线阶段中，感知模型在来自随机策略的数据上独立训练直到稳定，接着是在线阶段，在这一阶段中两个网络联合训练。这种分离使学习更加顺畅，并且显著提高了资源效率，因为离线预训练比通过交互直接进行在线学习要更快和对计算

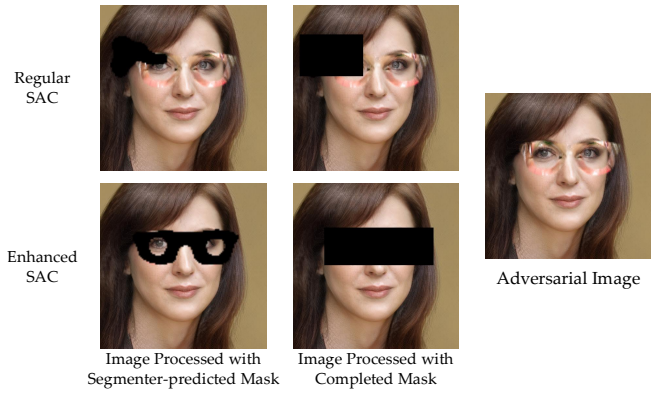

Fig. C。2：使用不同数据训练的 SAC 的定性结果。第一列展示了由常规 SAC 处理的对抗性图像，常规 SAC 是用填充了高斯噪声的补丁训练的，而后续列展示了由增强型 SAC 处理的图像。对抗性补丁是由 3DAdv 生成的，占据了图像的 8% 。

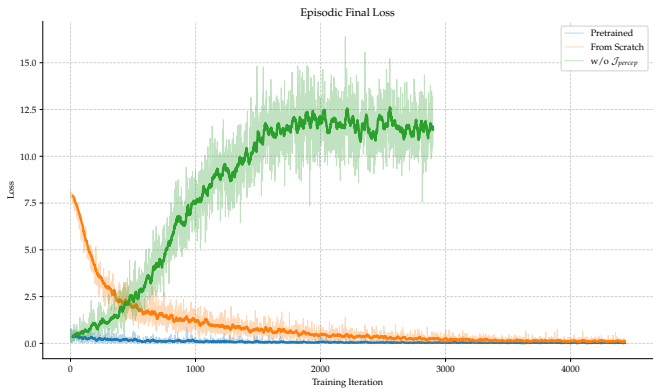

Fig. C。3： 三个模型配置在训练迭代过程中的情节最终损失。

资源的需求更少。如图 C。3 所示，预训练的 EAD 模型（$Pretrained$）比未经过预训练的模型（$FromScratch$）表现出更好的起点，并在相同迭代次数的训练中达到 10^{-2} 的损失水平。其次，除了强化学习目标之外，我们还引入了来自于真实感知标注的监督学习信号。考虑到主动防御自然地拓展了被动感知，当在感知模型与策略模型共同训练时，保留原始的监督学习目标是很直观的。此外，这种监督信号还作为一种正则化形式，维持 EAD 模型具备相当的感知能力，防止仅仅由于潜在的噪声 RL 奖励信号导致的过大或有害的变化。它引导策略朝向已知的有效行为，进一步增强稳定性。如图 C。3 所示，未正则化的曲线（$w/o\ J_{percep}$）迅速上升，剧烈波动，并未能收敛。

C.10 替代奖励塑造

直接熵减少。为了明确鼓励直接减少预测目标的不确定性，我们将每一步的奖励定义为熵项的加权减少：

$$\hat{r}_t = \mathcal{H}(\hat{y}_{t-1} \mid b_{t-1}) - \gamma \cdot \mathcal{H}(\hat{y}_t \mid b_{t-1}, o_t). \quad \text{(C. 38)}$$

二进制结果奖励。作为一种稀疏且无偏的基准，二进制结果奖励用于指示代理是否成功完成任务。在主动感知的情况下，我们将二进制结果奖励定义为目标超出预定义阈值的预测概率。形式上，它被定义为

$$\hat{r}_t = \mathbb{I}(\hat{y}_t = y), \quad \text{(C. 39)}$$

TABLE C。3：人脸识别中不同防御方法的计算开销比较。我们在 NVIDIA GeForce RTX 3090 Ti 和 AMD EPYC 7302 16 核处理器上报告防御的训练和推理时间，其中训练批次大小为 64 。

Method	# Params (M)	Parametric Model	Training Epochs	Training Time per Batch (s)	Total Training Time (GPU hours)	Inference Time per Instance (ms)
JPEG	-	non-parametric	-	-	-	9.65
LGS	-	non-parametric	-	-	-	26.22
SAC	44.71	segmenter	50 + 10	0.152 / 4.018	104	26.43
PZ						11.88
DOA	43.63	feature extractor	100	1.732	376	8.10
EAD	57.30	Perception & Policy Model	50 + 50	0.595 / 1.021	175	11.51
REIN-EAD	57.52		50 + 40	0.595 / 1.096	188	

TABLE C。4：在不同贴片大小的情况下，对人脸识别模型的白盒攻击成功率（%）。† 表示使用对抗样本训练的方法。

Method	8 %				10 %				12 %			
	MIM	EoT	GenAP	3DAdv	MIM	EoT	GenAP	3DAdv	MIM	EoT	GenAP	3DAdv
Impersonation Attack												
Undefended	100.0	100.0	99.00	98.00	100.0	100.0	100.0	99.00	100.0	100.0	100.0	99.00
JPEG	99.00	100.0	99.00	93.00	100.0	100.0	99.00	99.00	100.0	100.0	99.00	99.00
LGS	5.10	7.21	33.67	30.61	6.19	7.29	41.23	36.08	7.21	12.37	61.85	49.48
SAC	6.06	9.09	67.68	64.64	1.01	3.03	67.34	63.26	5.05	4.08	69.70	66.32
PZ	4.17	5.21	59.38	45.83	2.08	3.13	60.63	58.51	4.17	3.13	60.63	58.33
SAC †	3.16	3.16	18.94	22.11	2.10	3.16	21.05	16.84	3.16	4.21	15.78	18.95
PZ †	3.13	3.16	19.14	27.37	2.11	3.13	20.00	30.53	5.26	5.26	18.95	28.42
DOA †	95.50	89.89	96.63	89.89	95.50	93.26	100.0	96.63	94.38	93.26	100.0	100.0
EAD	4.12	3.09	5.15	7.21	3.09	2.06	4.17	8.33	3.09	5.15	8.33	10.42
REIN-EAD	2.10	1.06	3.15	7.37	1.04	1.06	6.32	9.47	3.15	4.21	7.37	15.79
Dodging Attack												
Undefended	100.0	100.0	99.00	89.00	100.0	100.0	100.0	95.00	100.0	100.0	100.0	99.00
JPEG	98.00	99.00	95.00	88.00	100.0	100.0	99.00	95.00	100.0	100.0	100.0	98.00
LGS	49.47	52.63	74.00	77.89	48.93	52.63	89.47	75.78	55.78	54.73	100.0	89.47
SAC	73.46	73.20	92.85	78.57	80.06	78.57	92.85	91.83	76.53	77.55	92.85	92.92
PZ	6.89	8.04	58.44	57.14	8.04	8.04	60.52	65.78	13.79	12.64	68.49	75.71
SAC †	78.78	78.57	79.59	85.85	81.65	80.80	82.82	86.73	80.61	84.69	87.87	87.75
PZ †	6.12	6.25	14.29	20.41	7.14	6.12	21.43	25.51	11.22	10.20	24.49	30.61
DOA †	75.28	67.42	87.64	95.51	78.65	75.28	97.75	98.88	80.90	82.02	94.38	100.0
EAD	0.00	0.00	2.10	13.68	2.11	1.05	6.32	16.84	2.10	3.16	12.64	34.84
REIN-EAD	1.04	2.04	5.15	13.54	1.03	2.02	8.16	14.58	6.18	5.15	13.54	34.02

其中 $\mathbb{I}(\cdot)$ 是指示函数。当感知模型使用概率建模时，该模型将从之前的信念 b_{t-1} 和观察 o_t 映射到目标 y 上的分布，一旦目标的后验概率超过置信阈值 κ ，我们就认为该情景是成功的。因此，奖励为

$$\hat{r}_t \;=\; \mathbb{I}\big(f(y \mid b_{t-1}, o_t; \boldsymbol{\theta}) > \kappa\big). \qquad \text{(C. 40)}$$

在实现中，我们选择 $\kappa = 0.95$ 用于训练期间的面部识别。

评估结果。如表 C。7 所示，我们提出的奖励塑造方法在干净准确性和抵抗对抗性补丁的鲁棒性方面均优于其他方法。直接熵减少鼓励策略选择能减少不确定性的动作；然而，由于它不利用真实标签，它可能无意中支持信心充足但不正确的预测。因此，该方法容易受到更强的对抗性攻击，比如 3DAdv，导致更高的攻击成功率。同时，由于其稀疏性，二进制结果奖励需要更多迭代才能收敛，在相同训练迭代次数下性能相对较弱。此外，由于它缺乏不确定性知情引导，其主动信息获取能力有限，无法有效应对对抗性补丁。相比之下，我们的奖励塑造采用了密集公式，加速了收敛并引导模型无偏地学习一个能最大化信息增益的策略，以实现准确的感知。此外，我们展示了不同奖励设计下 REIN-EAD 的损失收敛行为，如图 C。4 所示。二进制结果奖励方法表现出显著的不稳定性和缓慢、次优的收敛，最终在高损失值处趋于平稳。相比之下，直接熵减少方法和我们提出的奖励塑造技术都实现了更快且更稳定的收敛，两者的损失在大约 4,000 次训练迭代内接近于零值。

OmniObject3D 的原始版本可以在 https://omniobject3d.github.io 访问。由于数据集是原始的，并且未对分类任务的任何部分进行区分，我们对其进行预处理。对于遵循长尾分布的数据集，我们删除了实例少于 10 的类别，因为它们对我们的实验贡献不大，并按 4 到 1 的比例在每个类别中分割训练和测试数据。最终的数据集具有 176 个类别，其中训练有 4409 个对象，测试有 1192 个对象。我们使用 Pytorch3D (https://github.com/facebookresearch/pytorch3d) 作为模拟引擎，因为它提供了用于批量渲染的高效 API，以及实现对对象纹理的对抗攻击的微分管道。原始 OmniObject3D 数据集中的网格有数百万个面，并且其规模差异显著。为了降低计算开销并促进批量渲染，我们使用 Pymeshlab (https://pymeshlab.readthedocs.io) 来处理网格，将面数简化到 10,000 ，并在不影响任何渲染质量的情况下标准化比例。

TABLE C。5：对具有不同补丁大小的人脸识别模型进行黑箱攻击的成功率 (%)。† 表示使用对抗样本训练的方法。

Method	8 %				10 %				12 %			
	Cos.	Softmax	NAttack	RGF	Cos.	Softmax	NAttack	RGF	Cos.	Softmax	NAttack	RGF
Impersonation Attack												
Undefended	28.00	23.00	100.00	100.00	41.00	36.00	100.00	100.00	55.00	45.00	100.00	100.00
JPEG	33.00	33.00	96.00	94.00	42.00	40.00	98.00	98.00	62.00	48.00	99.00	99.00
LGS	11.63	6.98	11.63	4.65	12.79	11.63	9.30	3.49	15.12	17.44	9.30	4.65
SAC	8.70	9.78	13.04	14.13	13.04	11.97	14.13	15.22	17.39	13.04	13.04	17.39
PZ	6.45	9.68	4.30	3.26	5.38	7.63	4.30	4.31	8.60	5.52	4.30	4.15
SAC †	11.11	12.36	12.22	14.44	10.00	19.10	13.33	15.56	12.22	17.98	15.56	14.44
PZ †	8.24	5.00	10.58	9.41	10.59	3.75	9.41	8.25	11.77	5.00	9.41	8.25
DOA †	15.73	17.97	34.83	16.86	15.73	14.61	32.58	15.72	16.85	17.98	31.46	10.11
EAD	4.17	5.20	4.12	4.12	5.21	5.21	3.09	3.08	7.29	5.21	2.06	4.12
Rein-EAD	2.10	2.08	1.05	2.10	4.21	3.16	3.12	2.10	5.26	8.33	4.21	4.21
Dodging Attack												
Undefended	44.00	35.00	96.00	96.00	53.00	43.00	100.00	99.00	68.00	61.00	100.00	100.00
JPEG	49.00	45.00	81.00	83.00	58.00	51.00	94.00	96.00	71.00	72.00	99.00	99.00
LGS	22.11	21.05	18.95	20.00	22.11	21.05	20.00	16.84	29.48	31.58	20.00	24.21
SAC	40.80	36.84	55.26	50.00	47.37	44.74	53.95	52.63	50.00	50.00	59.21	47.37
PZ	41.67	28.34	28.33	31.67	38.33	36.67	28.33	26.67	41.67	28.33	30.00	30.00
SAC †	47.46	43.54	55.93	62.71	44.07	50.00	47.46	66.10	44.07	46.77	44.07	57.63
PZ †	50.88	47.69	56.14	50.87	47.37	52.30	50.88	56.14	43.86	53.84	52.63	49.12
DOA †	30.33	31.46	53.93	28.09	30.33	31.46	61.80	25.85	31.46	34.83	62.93	23.60
EAD	5.26	7.36	1.05	0.00	4.21	6.31	0.00	0.00	10.52	16.84	0.00	3.16
Rein-EAD	4.17	7.29	1.03	0.00	6.25	8.42	1.03	2.06	12.50	13.54	5.15	3.06

TABLE C。6：对 EAD 和 Rein -EAD 的自适应攻击评估。带有 USP 的列表示通过在均匀超集策略（USP）上优化预期梯度得到的结果。感知和策略分别代表针对单个子模块的自适应攻击。整体则表示通过沿整体（4 步）轨迹的梯度随梯度检查攻击模型。

Method	Attack Success Rate (%)			
	USP	Perception	Policy	Overall
Impersonation Attack				
EAD	8.33	1.04	9.38	7.29
Rein-EAD	4.21	2.11	2.17	-
Dodging Attack				
EAD	22.11	10.11	16.84	15.79
Rein-EAD	8.16	3.06	2.06	-

TABLE C。7：不同奖励整形下 Rein -EAD 的性能表现。

Reward	Acc (%)	Attack Success Rate (%)			
		MIM	EoT	GenAP	3DAdv
Entropy Deduction	88.67	3.15	2.11	4.21	11.42
Outcome Reward	88.62	3.22	3.26	5.94	10.86
ours	89.03	2.10	3.15	7.37	4.21

图像通过硬朗的 Phong 着色器以视场 60 和分辨率 256×256 使用透视相机渲染。我们使用 Gym [?] 将 Pytorch3D 扭曲为 Rein -EAD 的环境。

C.11 实现细节

模型详情。针对动态 OmniObject3D 上进行的实验，我们结合 Swin Transformer 和 Decision Transformer 实现了 Rein -EAD 分类。我们使用了来自 PyTorch Image Models（https://github.com/huggingface/pytorch-image-models ）的预训练

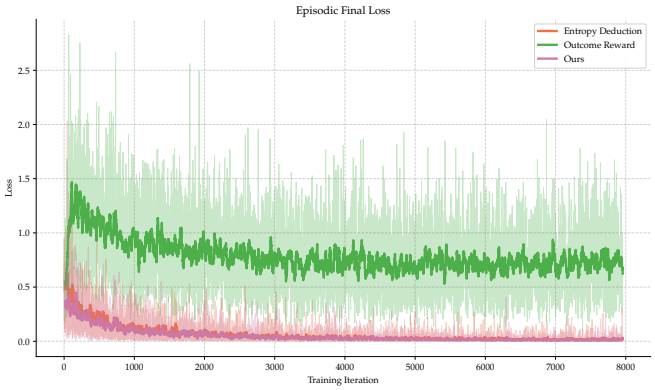

Fig. C。4：REIN-EAD 在不同奖励下的损失收敛情况。

Swin-Small Transformer，并在动态 OmniObject3D 的训练集上通过 176 个类别的头进行微调。为了实现 Rein -EAD，我们将 Swin Transformer 的头替换为一个时间融合模块和一个 Decision Transformer。在每个步骤 $t > 0$，从 Swin-Small 主干提取的长度为 768 的特征嵌入被输入到时间融合模块，在那里它会与之前提取的维度为 $(t-1) \times 768$ 的观察序列相连接，形成一个时间视觉特征序列。这个序列然后由 Decision Transformer 进行时间融合，输出一个优化后的特征嵌入。最后，通过浅层 MLP 实现的特征解码器、动作解码器和价值解码器被用来将嵌入解码成优化后的标签、预测动作和价值。动作解码器的输出是一个预测的视角。在训练阶段，我们用它作为均值，从具有固定方差的多变量高斯分布中采样进行 RL 探索，而在测试中，我们直接用它作为实际的动作值。

训练细节。我们采用类似于 Rein -EAD 的两阶段训练范式用于人脸识别。在离线阶段，我们冻结 Swin-Small 主干，使用随机动作策略训练决策变换器和特征解码器以收集观测。在在线阶段，我们使用算法 1 来结合策略网络的训练。我们

TABLE C。8 : REIN -EAD 用于目标分类的超参数。

Hyper-parameter	Value
Lower bound for horizontal rotation ($h_{\min}$)	$-90°$
Lower bound for horizontal rotation ($h_{\max}$)	$90°$
Lower bound for vertical rotation ($v_{\min}$)	$0°$
Upper bound for vertical rotation ($v_{\max}$)	$90°$
Ratio of patched data (r_{patch})	0.8
Training epochs for offline phase (N_{offline})	100
Learning rate for offline phase ($\text{lr}_{\text{offline}}$)	2×10^{-4}
Batch size for offline phase (b_{offline})	128
Total Episodes for online phase	$150,000$
Learning rate for online phase ($\text{lr}_{\text{online}}$)	1×10^{-4}
Batch size for online phase (b_{online})	128
Return attenuation factor (γ)	0.95
Updates per iteration (n)	2

使用 PGD 近似替代补丁集，称为 OAPA，这在单视图下攻击 Swin-Small 主干。学习率和迭代次数分别为 $\alpha = 8/255$ 、$N = 30$ ，与 DOA † 一致。注意，REIN -EAD 的 OAPA 训练是离线的，仅进行一次，而 DOA † 在每次迭代中生成一个替代补丁。用于物体分类的 REIN -EAD 的超参数如表 C。8 所示。

在纹理空间中进行 3D 环境下的攻击。我们实施对抗性攻击以误导分类器输出错误的标签。在分类任务中的攻击方法类似于面部识别系统中的方法，但在纹理空间中实现。我们利用 Pytorch3D 的可追踪渲染计算图，直接在对象的遮罩纹理上生成对抗性补丁，这适合各种类别对象的非平面表面，并确保多视图一致性。由于在分类任务的背景下对象形状差异很大，为了确保补丁可以完全贴附在对象上，我们将补丁的大小设置为对象包围盒的 20% ，并将其定位在包围盒的中心。用于鲁棒性测试的对抗样本是在测试数据集上生成的，而这些数据集对于 REIN -EAD 和其他防御基准都是未知的。

白盒攻击。我们使用 MIM [?] 作为单视图白盒对抗项，并通过动量项进行了增强。MIM 的衰减因子设定为 $\mu = 1.0$ ，与 FR 攻击一致。多视图威胁由 EoT [?] 和 MeshAdv [?] 建立。EoT 采用了一种二维批量数据增强方法，包括平移、旋转和翻转渲染图像，并在图像空间中平均梯度。相反，MeshAdv 在 REIN -EAD 的动作范围内进行一批 3D 变换，并通过差异渲染管道将梯度反向传播到纹理空间，使其对视图变化更具鲁棒性。对于白盒攻击，我们将学习率和迭代次数设置为 $N = 100$ 和 $\alpha = 8/255$ 。EoT 和 MeshAdv 的采样频率设定为 $M = 128$ 。

黑箱攻击。对于基于查询的攻击，我们使用 RGF [?] 和 N 攻击 [?] 。我们将最大查询次数设置为 $N = 10000$ ，采样频率设置为 $M = 100$ 。RGF 和 N 攻击的学习率分别为 $\alpha = 0.05$ 和 $\alpha = 0.1$ 。对于基于迁移的对手攻击，我们利用 MeshAdv，其参数与白盒版本相同，并微调一个 Swin-Tiny Transformer 作为代理模型来发动迁移攻击。

自适应攻击。在人脸识别设置中，我们使用 BPDA [?] 技术自适应攻击 JPEG 和 LGS，使用 STE [?] 技术攻击 SAC † 和 PZ † 。对于 REIN -EAD，我们采用统一超集策略来发起自适应攻击，该策略在 FR 任务中被证明最有效。

C.12 防御细节

JPEG [?] 和 LGS [?] 的实现与 FR 任务一致。对于 SAC † [?] 和 PZ † [?] ，我们使用相同的代码通过 EoT [?] 优化的对抗补丁在动态 OmniObject3D 的训练集上重新训练补丁分割器。对于 DOA † [?] ，我们遵循其训练范式，在动态 OmniObject3D 的训练集上微调与 REIN -EAD 使用相同的 Swin-Small Transformer 骨干网络。具体而言，我们利用学习率为 $\alpha = 8/255$ 和迭代次数为 $N = 30$ 的 PGD 进行对抗训练，并使用基于梯度的方法搜索补丁位置，候选数为 $C = 10$ ，如其论文中所述。用于训练 SAC † 、PZ † 和 DOA † 的补丁占据了对象边界框的 20% ，这与 REIN -EAD 相同。

Appendix D
目标检测的实验细节

对于目标检测，我们在 https://catalog.ngc.nvidia.com/orgs/nvidia/teams/research/models/eg3d 上使用 ShapeNet Cars 的预训练 EG3D 模型来生成分辨率为 256×256 的多视图汽车图像。我们生成了具有不同外观的 1000 辆汽车，并将数据划分为 800 训练数据和 200 测试数据。记录潜在因素，以确保在所有实验中每辆车的身份保持一致。由于生成图像的背景为空白，我们能够自动标注边界框。用于 REIN -EAD 的在线环境由 Gym- [?] 封装。

D.1 关于 CARLA 的详细信息

我们使用 CARLA 0.9.14 [?] 进行一个更复杂的目标检测实验。训练数据涵盖了 CARLA 提供的所有 41 种不同的车辆模型。对于每个模型，我们生成不同颜色版本的车辆，并在 Town10 的不同背景下收集多视角样本，这些样本作为训练模型和对抗样本的离线数据集。REIN -EAD 的训练和测试过程在 CARLA 中在线进行，这使得智能体可以在动作范围内探索每一个可能的视角。在测试实验期间，我们使用所有 41 种车辆，并将它们放置在与训练集不同的位置。我们使用 CARLA 提供的 Python API 以 256×444 的分辨率自动收集数据并标注标签。我们使用 Gym [?] 将 CARLA 包装为 REIN -EAD 的在线环境。

D.2 实现细节

模型细节。在 EG3D 上的目标检测任务中，我们实现了结合 YOLOv5n 和 Decision Transformer 的 REIN -EAD。我们使用了官方实现的预训练 YOLOv5n（https://github.com/ultralytics/yolov5 ）并在每个环境的训练集上将其微调为单类别检测模型。对于每个时间步 $t > 0$ ，给定维度为 $256 \times 256 \times 3$ 的当前观测作为输入，利用第二个 Cross Stage Partial Networks 输出的维度为 $32 \times 32 \times 64$ 的特征图。我们发现这一层的特征图对于 REIN -EAD 在计算上是高效的，因为 YOLOv5n 的后期阶段形成了一个大规模的连接特征金字塔。然后将这些特征图重塑为维度为 64×1024 的序列。为了将其与先前提取的观测序列 $(t-1) \times 64 \times 1024$ 连接起来，我们提供一个视觉特征的时间序列作为 Decision Transformer 的输入，它输出时间融合的视觉特征序列 64×1024 和预测的动作。对于 REIN -EAD，额外使用了一个值解码器来估计优势值。为了预测目标检测中所需的边界框和目标性得分，我们将时间融合的视觉序列重塑回其原始形状，并将其用作 YOLOv5n 后期阶段的输入。

训练细节。我们采用与先前任务中 REIN -EAD 类似的训练范式，并根据表格 D。9 和表格 D。10 设置 REIN -EAD 的目标检测超参数

D.3 攻击的细节

多视角隐藏攻击用于车辆目标检测。在单类别车辆目标检测任务中，攻击者的目标是将补丁放置在车辆上，使其从目标检测器中消失。补丁放置在车辆的一侧，占据边界框的 25% 。我们遵循为 YOLO [?] 设计的对抗性损失来最小化目标分数，

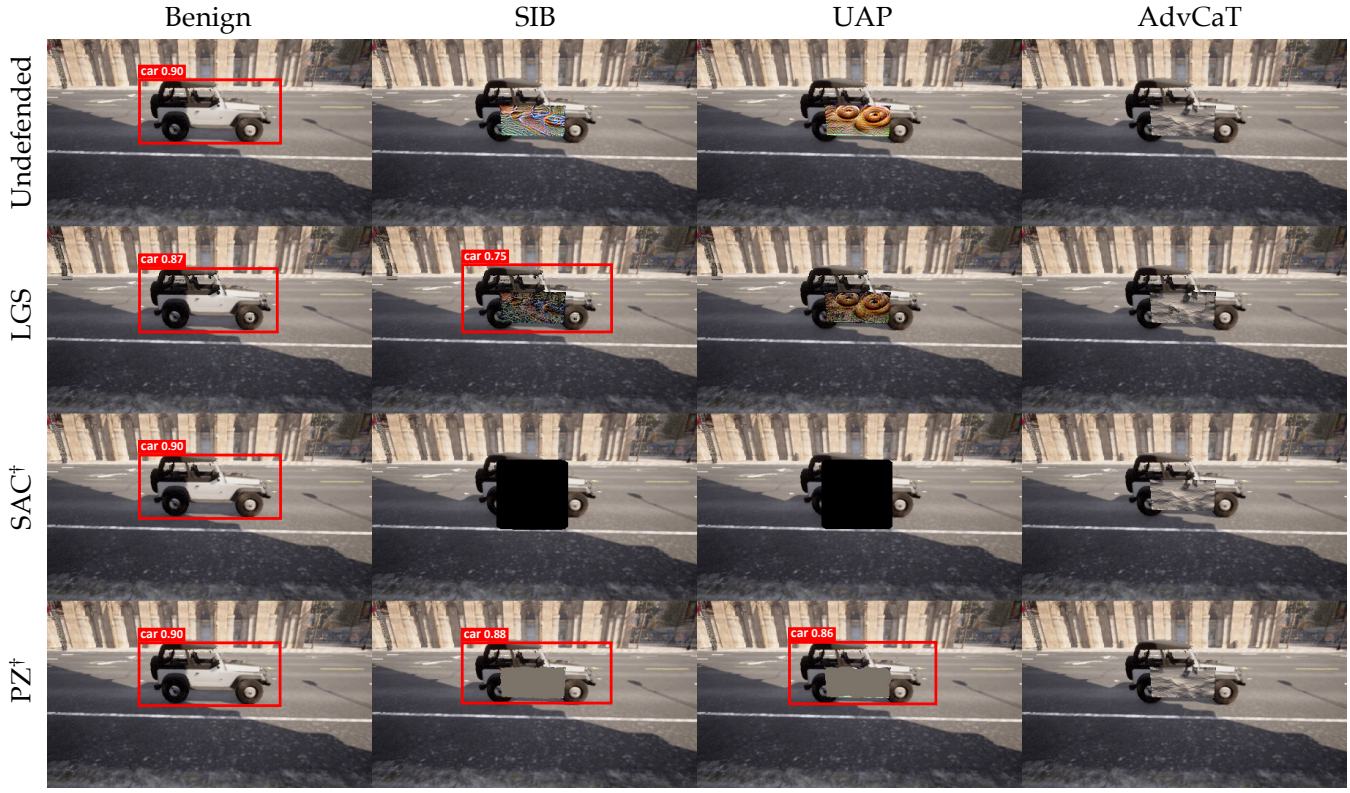

Fig. D。5 : CARLA 上的防御基线可视化。† 表示方法通过对抗样本进行训练。

TABLE D。9 : REIN -EAD 在 EG3D 上进行目标检测的超参数。

Hyper-parameter	Value
Lower bound for horizontal rotation ($h_{\min}$)	$-60°$
Upper bound for horizontal rotation ($h_{\max}$)	$60°$
Lower bound for vertical rotation ($v_{\min}$)	$0°$
Upper bound for vertical rotation ($v_{\max}$)	$30°$
Ratio of patched data (r_{patch})	0.4
Training epochs for offline phase (N_{offline})	50
Learning rate for offline phase ($\text{lr}_{\text{offline}}$)	2×10^{-4}
Batch size for offline phase (b_{offline})	128
Total episodes for online phase (N_{online})	$10,000$
Learning rate for online phase ($\text{lr}_{\text{online}}$)	1×10^{-4}
Batch size for online phase (b_{online})	64
Return attenuation factor (γ)	0.95
Updates per iteration (n)	2

TABLE D。10 : CARLA 上进行目标检测的 REIN -EAD 超参数。

Hyper-parameter	Value
Lower bound for horizontal rotation ($h_{\min}$)	$-60°$
Upper bound for horizontal rotation ($h_{\max}$)	$60°$
Lower bound for vertical rotation ($v_{\min}$)	$5°$
Upper bound for vertical rotation ($v_{\max}$)	$35°$
Ratio of patched data (r_{patch})	0.4
Training epochs for offline phase ($\text{lr}_{\text{offline}}$)	50
Learning rate for offline phase ($\text{lr}_{\text{offline}}$)	2×10^{-4}
Batch size for offline phase (b_{offline})	128
Total Episodes for online phase	$10,000$
Learning rate for online phase ($\text{lr}_{\text{online}}$)	1×10^{-4}
Batch size for online phase (b_{online})	64
Return attenuation factor (γ)	0.95
Updates per iteration (n)	2

从而实现隐藏攻击。为了增强多视角的鲁棒性，我们在每次迭代中使用一批具有不同视角的图像训练对抗补丁。采样频率设置为 100 。

像素空间中的攻击。EoT 作为一个基线多视角对手被使用，它结合了上述视图变换的期望。为了实现更广泛的攻击，我们利用通用对抗扰动（UAP）[?] 为数据集中所有车辆生成一个单一的对抗补丁，使其能够隐藏任何车辆不被检测器发现。两种方法的学习率和迭代次数设置为 $N = 500$ 和 $\alpha = 8/255$ 。

隐藏层中的攻击。当 EAD 模块插入 YOLOv5n 的中间阶段时，我们实现了 SIB [?] ，它攻击隐藏层的特征。我们将 EAD 模块的输入设定为需要扰动的特征，并通过特征干扰增强损失最大化对抗性特征与原始特征之间的差异。物体性损失和特征损失的系数分别是 $\mu_1 = 0.5$ 和 $\mu_2 = 0.5$ 。学习率和迭代次数与 EoT 和 UAP 保持一致。

潜在空间攻击。为了实现对人类和检测器都不易察觉的贴图攻击，我们采用了对抗伪装纹理（AdvCaT） [?] ，其利用了 Voronoi 图和 Gumbel-softmax 技巧生成带有控制点和潜在种子的语义多边形伪装。我们使用 K-means 聚类（4 个聚类）从环境中提取基础颜色作为伪装使用。我们将控制点和潜在种

子的学习率分别设置为 $\alpha_1 = 0.0005$ 和 $\alpha_2 = 0.005$ 。迭代次数为 $N = 200$ 。

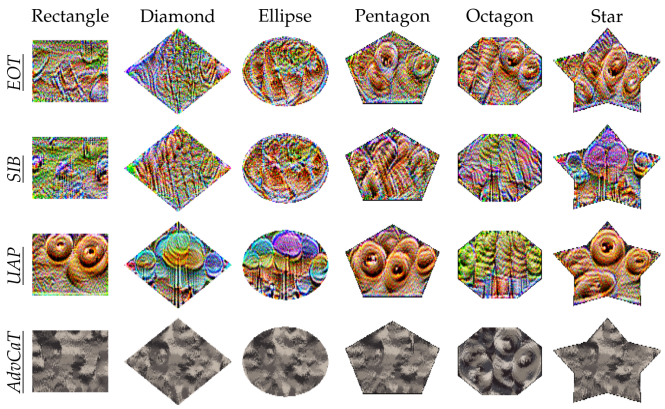

Fig. D。6 : 在 CARLA 上生成补丁在不同攻击和形状下的可视化。

D.4 防御细节

JPEG [?] 和 LGS [?] 的实现与 FR 任务一致。对于 SAC † [?] 和 PZ † [?]，我们在相应环境的训练集上重新训练补丁分段器，使用 EoT 优化的对抗性补丁，采用与 FR 任务相同的代码。用于训练 SAC † 和 PZ † 的补丁占据车辆边界框的 25%，这与用于训练 Rein -EAD 的补丁相同。

我们在图 D。5 中展示了基线方法的定性防御结果。LGS 的平滑机制相对较弱，无法完全消除对抗性噪声。相比之下，SAC † 和 PZ † 的分割骨干可以很好地区分和去除带噪声模式的贴片攻击，但未能检测到环境镶嵌的 AdvCaT。此外，SAC † 的补全机制导致过度遮挡，从而使后续检测任务复杂化。

D.5 在 CARLA 上可视化补丁

我们生成了菱形、椭圆形、五边形、八边形和星形的补丁。这些补丁对于 Rein -EAD 和其他防御基线都是看不见的。这些补丁的可视化显示在图 D。6 中。

更多关于 EoT 和 SIB 攻击与不同补丁形状的对比评估在图 ?? 中进行了说明。

D.6 失败案例

失败案例揭示了 REIN-EAD 的几个显著局限性。首先，该系统容易受到策略性放置的对抗性补丁的攻击，这些补丁会遮挡关键的对象特征。如图 D。7 所示，当对抗性补丁遮挡住绿色玩偶的面部区域时，模型一致地将该对象误分类为西兰花。这表明，关键辨别特征的遮挡会损害模型获取足够信息以进行准确识别的能力。其次，该框架在同时遭受对抗性攻击和自然分布外干扰时表现出性能下降。虽然多步探索机制通常有效，但在这种复合不确定性情境下，它未能解决信号冲突。一个代表性例子，如图 D。7 所示，在处理被咬过的苹果时，表现出持续的预测不确定性，此时咬痕引起的形状变形与对抗性扰动相互作用。这些限制凸显了未来研究的潜在领域。潜在方向包括开发优先考虑重要视觉区域的补丁放置策略，以及结合更多样化的 3D 对象数据集进行训练，以增强模型对遮挡和分布外样本的鲁棒性。

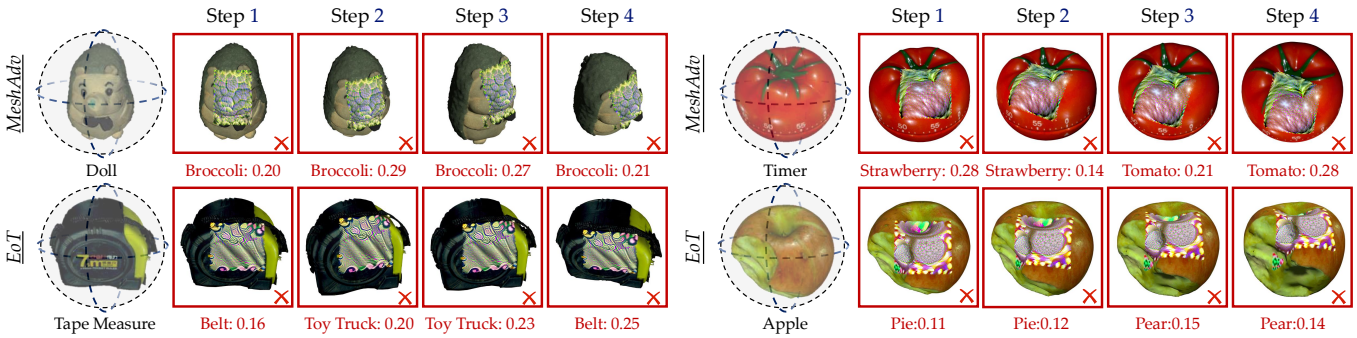

Fig. D。7：在动态 OmniObject3D 上采用 REIN-EAD 进行故障案例的可视化，攻击性补丁在前视图中占据了对象边界框的 20 %。