计算机研究与发展
Journal of Computer Research and Development
https://crad.ict.ac.cn
62(8): 2111−2124, 2025

针对自动驾驶系统目标检测器的迁移隐蔽攻击方法

郑君豪[1,2] 蔺琛皓[1,2] 赵正宇[1,2] 贾子羿[1,2] 吴黎兵[3] 沈 超[1,2]

[1](西安交通大学网络空间安全学院 西安 710049)

[2](智能网络与网络安全教育部重点实验室(西安交通大学) 西安 710049)

[3](武汉大学国家网络安全学院 武汉 430072)

(2193412684@stu.xjtu.edu.cn)

Towards Transferable and Stealthy Attacks Against Object Detection in Autonomous Driving Systems

Zheng Junhao[1,2], Lin Chenhao[1,2], Zhao Zhengyu[1,2], Jia Ziyi[1,2], Wu Libing[3], and Shen Chao[1,2]

[1](*School of Cyber Science and Engineering*, *Xi'an Jiaotong University*, *Xi'an* 710049)

[2](*Key Laboratory for Intelligent Networks and Network Security* (*Xi'an Jiaotong University*), *Ministry of Education*, *Xi'an* 710049)

[3](*School of Cyber Science and Engineering*, *Wuhan University*, *Wuhan* 430072)

Abstract Deep learning-based object detection algorithms have been widely applied, while recent research indicates that these algorithms are vulnerable to adversarial attacks, causing detectors to either misidentify or miss the target. Nonetheless, research focusing on the transferability of adversarial attacks in autonomous driving is limited, and few studies address the stealthiness of such attacks in this scenario. To address these limitations in current research, an algorithmic module to enhance attack transferability is designed by drawing an analogy between optimizing adversarial examples and the training process of machine learning models. Additionally, through employing style transfer techniques and neural rendering, a transferable and stealthy attack (TSA) method is proposed and implemented. Specifically, the adversarial examples are first repeatedly stitched together and combined with masks to generate the final texture, which is then applied to the entire vehicle surface. To simulate real-world conditions, a physical transformation function is used to embed the rendered camouflaged vehicle into realistic scenes. Finally, the adversarial examples are optimized using a designed loss function. Simulation experiments demonstrate that the TSA method surpasses existing methods in attack transferability and exhibits a certain level of stealthiness in appearance. Furthermore, physical domain experiments validate that the TSA method maintains effective attack performance in real-world scenarios.

Key words adversarial attacks; autonomous driving; object detection; transferability; stealthiness; neural rendering

摘 要 基于深度学习的目标检测算法已广泛应用,与此同时最近的一系列研究表明现有的目标检测算法容易受到对抗性攻击的威胁,造成检测器失效.然而,聚焦于自动驾驶场景下对抗攻击的迁移性研究较少,并且鲜有研究关注该场景下对抗攻击的隐蔽性.针对现有研究的不足,将对抗样本的优化类比于机器

收稿日期: 2024-02-21; **修回日期**: 2024-10-22

基金项目: 国家重点研发计划项目(2021YFB3100700); 国家自然科学基金项目(T2341003, 62006181, 62161160337, 62132011, U21B2018, U20A20177, 62206217); 陕西省重点研发计划项目(2023-ZDLGY-38, 2021ZDLGY01-02)

This work was supported by the National Key Research and Development Program of China (2021YFB3100700), the National Natural Science Foundation of China (T2341003, 62006181, 62161160337, 62132011, U21B2018, U20A20177, 62206217), and the Key Research and Development Program of Shaanxi Province (2023-ZDLGY-38, 2021ZDLGY01-02).

通信作者: 蔺琛皓(linchenhao@xjtu.edu.cn)

学习模型的训练过程,设计了提升攻击迁移性的算法模块.并且通过风格迁移的方式和神经渲染(neural rendering)技术,提出并实现了迁移隐蔽攻击(transferable and stealthy attack, TSA)方法.具体来说,首先将对抗样本进行重复拼接,结合掩膜生成最终纹理,并将其应用于整个车辆表面.为了模拟真实的环境条件,使用物理变换函数将渲染的伪装车辆嵌入逼真的场景中.最后,通过设计的损失函数优化对抗样本.仿真实验表明,TSA方法在攻击迁移能力上超过了现有方法,并在外观上具有一定的隐蔽性.此外,通过物理域实验进一步证明了TSA方法在现实世界中能够保持有效的攻击性能.

关键词 对抗攻击;自动驾驶;目标检测;迁移性;隐蔽性;神经渲染

中图法分类号 TP391

DOI: 10.7544/issn1000-1239.202440097 **CSTR:** 32373.14.issn1000-1239.202440097

目前自动驾驶在工业界迎来蓬勃发展,但结构性的技术缺陷也导致自动驾驶系统面临算法安全威胁.例如,2016年在美国的佛罗里达州,一辆特斯拉汽车由于未能在强烈日光照射下检测到白色货车,导致自动驾驶系统失效,引起了致命交通事故.正常情况下发生这种事件的概率极低,但可怕的是攻击者能够利用算法漏洞欺骗自动驾驶系统从而以极低的成本复制此类事件.例如,通过修改车辆的纹理特征,使自动驾驶的目标检测模型失效,恶意引发交通事故.因此,对自动驾驶领域对抗攻击的研究迫在眉睫.

针对机器学习模型的对抗攻击由Szegedy等人[1]首次提出.此后Goodfellow等人[2]提出一步构造对抗样本的快速梯度符号法(fast gradient sign method, FGSM),国内外科研工作者[3-14]也相继提出了一系列的对抗攻击方法.早期对抗攻击主要实施在数字域,可以攻击分类任务或者目标检测任务.Madry等人[3]提出投影梯度下降(project gradient descent, PGD)攻击,通过迭代随机扰动来找到可以欺骗模型的对抗样本.与FGSM不同,PGD做多次迭代,步长较小,每次迭代均将扰动修剪到规定范围内[15].Kurakin等人[7]提出基本迭代方法(basic iterative method, BIM),通过多次攻击迭代,找到最小的能够误导分类器的扰动,可以生成更隐蔽的对抗样本,同时攻击效果也得到提升.

与上述需要扰动输入图片所有像素的对抗攻击方法不同,采用对抗补丁的攻击方法仅需扰动小部分区域面积的像素,代表方法有对抗补丁攻击[16](adversarial-patch-based attack, DPatch).DPatch常用于攻击目标检测器,可同时攻击边界框回归和对象分类,其首先在原图片上人为设置一个真值(ground truth, GT),将带有对抗补丁的输入传进检测器得到预测结果,然后将预测结果和该真值对比来计算损失函数,再使用损失函数计算梯度并回传更新对抗补丁,最终得到的对抗补丁便具有攻击检测器的能力.此外,Brown等人[17]提出一种在现实世界中创建通用的、健壮的、有针对性的对抗图像补丁的方法.Lang等人[18]提出一种能在复杂环境中成功隐藏和错误分类车辆的方法.

由于补丁仅需扰动部分区域,因此其在物理域下实施的成本相对更低.然而相比于数字域,物理域下的攻击需要考虑的影响因素更多、难度更大.在物理域下实现对抗攻击,需要将对抗补丁打印出来,并由相机拍摄成图片输入到深度学习模型中.Wu等人[19]提出了隐藏人的对抗补丁生成方法,生成的补丁作为装饰图案打印在衣服上,能够有效地欺骗检测器对人的检测.Hu等人[20]提出了全覆盖式的对抗攻击,将生成的对抗补丁重复拼接从而覆盖整个衣服,使攻击在不同角度下均有效.Chen等人[21]提出了鲁棒物理扰动算法.Huang等人[22]提出通用物理伪造(universal physical camouflage, UPC)的方法,使用虚拟仿真技术训练生成通用的对抗补丁.

针对自动驾驶场景下,现有的研究通常采用伪造纹理的方式,即通过可微分神经渲染器,将对抗补丁以类似涂鸦的形式覆盖在目标物体上.相比于常见的2维对抗补丁,3维的伪装纹理能够覆盖更大面积,在不同观察角度下均有更稳定的攻击效果.Wang等人[23]提出了双注意力抑制(dual attention suppression, DAS)攻击,通过抑制模型和人类注意力来生成鲁棒的、迁移的物理对抗性迷彩,并采用包含语义信息的图案作为攻击纹理,以提升攻击隐蔽性.Wang等人[24]利用神经渲染实现了全涂装式对抗(full-coverage attack, FCA)纹理攻击,通过直接优化3维物体的全身外表面纹理,为多场景、多角度的物理攻击提供了一种高效的解决方案.

然而,现有的研究方法对于特定的物理场景并不鲁棒,特别是对于多视角、远距离和部分遮挡的情况.首先,对抗补丁通常粘贴在平面物体上,因此基

于补丁的方法不适合粘贴在3维车辆上攻击检测器，攻击效果不如伪装纹理稳定. 其次，此前基于伪装纹理的方法[22-23]仅在3维车辆模型的部分区域绘制对抗性纹理，例如侧门或车顶部，限制了攻击性能. 尽管FCA纹理攻击[24]是全覆盖的伪装纹理攻击方法，但是它的纹理优化框架限制了攻击目标，生成的对抗性纹理仅能实施在与训练时所用车辆模型同类别的汽车上. 对于重复同一模式的纹理攻击方法[25-27]，其训练时生成单独的对抗样本，在实际攻击时才将样本重复拼接并覆盖整个车辆表面，因此本质上仍是对抗补丁优化，在覆盖整个车辆表面时会不可避免地受到车辆表面弯曲的影响，不如伪装纹理方法鲁棒. 此外，现有攻击没有考虑到自动驾驶场景下的攻击模型往往是黑盒模型，因此能否攻击成功不仅需要足够鲁棒，更需要具备足够的迁移能力. 另外现有研究并没有关注攻击的隐蔽性，对抗样本的可疑外观容易引人注意.

因此本文针对现有研究的不足，设计了一种攻击自动驾驶系统中目标检测模块的迁移隐蔽攻击(transferable and stealthy attack, TSA)方法. 具体而言，我们考虑到对抗性纹理需要具备通用性，因此将初始纹理通过重复拼接的方式生成不同车辆类型的整车纹理，并采用可微分渲染器，将生成的整车纹理绘制到3维车辆模型表面. 然后我们应用物理变换将渲染的3维车辆投影到不同的环境场景中，以获得逼真的对抗图像. 最后使用我们设计的损失函数将对抗性纹理的生成建模为优化问题. 通过这种生成的对抗性伪装，涂装车辆可以在物理场景中保持对抗性. 其中，为了提高对抗性纹理的攻击迁移性，我们采用了3种改进算法的策略，包括物理变换、权重衰减和纹理暂退，使得利用单一模型优化生成的纹理能够攻击多种检测模型. 为了提高对抗性纹理的攻击隐蔽性，我们在设计损失函数时引入了风格迁移的思想，在优化时参考风格图片的特征，使得生成的伪装纹理不会引起人类观察者的怀疑，达到隐蔽的效果. 本文的贡献有3点：

1)提出了针对自动驾驶系统目标检测器的迁移隐蔽攻击方法TSA，在攻击迁移性和隐蔽性上超过此前方法.

2)使用TSA方法生成的纹理具有通用性，能够在不同汽车类型上使用.

3)通过仿真实验，我们验证了攻击的迁移性和隐蔽性，并使用量化指标进行评估. 通过物理域实验，我们验证了TSA方法在物理世界中仍能保持有效的攻击性能.

1 相关工作

对抗攻击(adversarial attack)是指通过施加小幅扰动或者输入来对机器学习模型产生假的输出或错误分类的一种攻击方式，根据攻击实施的环境可以分为数字域攻击和物理域攻击. 本节主要介绍隐蔽性对抗攻击的相关工作以及在自动驾驶领域对抗攻击的研究现状，并介绍了神经渲染的基础知识.

1.1 自动驾驶场景下的对抗攻击

攻击自动驾驶场景下的目标检测器需要满足多方面的要求：通用性、迁移性、隐蔽性. 通用性指的是当目标在不同的背景、光照、距离和角度下对抗样本均能保持攻击能力；迁移性指的是目标检测器的模型未知，对抗样本具有迁移攻击能力，对于不同模型的检测器均有攻击效果；隐蔽性指的是对抗样本不能引起人的注意和怀疑，保证攻击行为不被人发现的同时攻击效果不受影响.

目前攻击自动驾驶场景下目标检测器的方法主要是通过仿真实现[23-24,28]. DAS攻击[23]使用Carla仿真器采集一定数量的图片制作训练和测试数据集. 首先获取训练集中背景图片的相机位姿，使用神经渲染(neural rendering)[29]的方式将对抗补丁覆盖在可微分的3维网格表面，然后以相同的相机位姿将网格渲染成2维的图片，并结合掩膜获取对抗图片. 另外，针对攻击的迁移性，DAS攻击同时抑制模型注意力和人类注意力. 具体地，将不同模型之间共享的注意区域从目标物体上转移到背景上，同时参考人类的视觉习惯，减少纹理中较为可疑的扭曲，保留初始的语义图案形状，从而获得鲁棒的、自然的对抗纹理.

与DAS攻击不同的是，FCA攻击[24]和涂层对抗伪装(coated adversarial camouflage, CAC)攻击[28]利用神经渲染实现全车身涂装式的对抗纹理攻击，通过直接优化3维汽车的表面纹理，并将纹理覆盖在全车身表面，在不同观察角度和相机距离下均保持有效的攻击能力. CAC攻击结合不同的背景图片，将3维纹理渲染并粘贴在背景图片中，通过控制渲染的参数，如相机位姿、汽车距离以及位置等，提高攻击的鲁棒性以及不受角度和距离的影响.

1.2 隐蔽性对抗攻击

由于对抗样本的像素在优化的时候没有过多约束，因此色彩通常较为鲜艳且没有特定的语义信息，虽然有较好的攻击能力，但容易引起被攻击者的警觉.

考虑到对攻击鲁棒性的要求较高,在扰动大小方面做约束来提高隐蔽性是不合适的,这会影响到攻击效果,因此只能从对抗样本的语义信息上入手.目前隐蔽性对抗攻击的研究均期望生成一个具有强语义信息的对抗样本,比如直接使用编码器生成对抗补丁,如自然物理对抗补丁(naturalistic physical adversarial patch, NPAP)攻击[30],采用生成对抗网络(generative adversarial network, GAN)[31]生成具有对抗攻击效果的小狗图案.另有方法[32]在训练优化的过程中添加约束,使得对抗样本近似于大众所熟知的动画图案,或者使用风格迁移的方法,将对抗样本伪装成自然现象,比如积雪等[33].

上述隐蔽性对抗攻击方法仅局限于2维的补丁优化过程,无法用于3维的纹理优化.在自动驾驶场景下,目前研究攻击隐蔽性的代表方法为DAS攻击.DAS攻击主要考虑2点,即语义信息和可疑畸变.参考文献[34],人类在观察事物并对其做出判断时会更关注物体形状,因此可以通过更好地保留种子内容块(可以选用一种常见的图案,比如笑脸)的形状,来保证人类注意力更集中在形状所带来的语义信息上,而不去怀疑图案的可疑性.针对可疑畸变,DAS攻击在优化过程中添加了额外的平滑度约束,使得生成的图案的相邻像素之间尽可能过渡平滑,减少了可疑的颜色畸变.

1.3 神经渲染

目前常用的3维渲染中,有使用光追渲染的方法,也有采用栅格化的方法,其中后者使用更多.栅格化的方法指的是将3维场景中的物体进行着色并投影到成像平面形成2维图形的过程.神经渲染[29]为一种可微分的栅格化渲染,将栅格化过程看作一个可以传递梯度的过程,从而可以对3维模型直接优化.在栅格化渲染过程中,3维模型通常表述为3个部分,即顶点、面片和纹理.顶点表示3维网格模型各顶点的3维空间向量,如图1所示,左边是一个茶壶模型,右边是三角形面片放大图,其中的空心圆点即为顶点,围成的各个三角形即为面片,而面片的像素即为面片的纹理值.

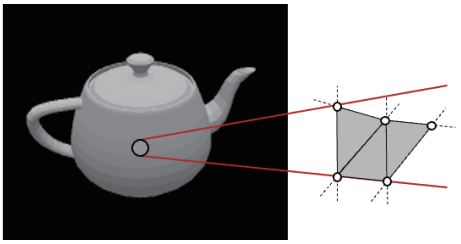


Fig. 1 Teapot model

图1 茶壶模型

神经渲染与传统的栅格化渲染的区别在于其能够进行可微分渲染,即渲染出的图片对3维模型的各参数均可微,因此可以利用渲染图片设计损失函数,然后通过梯度反向传播计算并更新3维模型的参数,包括3维模型的顶点位置、面片纹理,甚至相机的位姿参数.然而,此前自动驾驶场景下的对抗攻击方法往往将目标物体的面片纹理作为优化目标,通过梯度回传直接去计算并更新整体的纹理,即3维模型每一个三角形面片的像素,这带来了以下问题.首先是面片纹理的边界问题,相邻的2个面片在视觉上是直接邻接的,但是在优化空间中并不相邻,在优化空间更新的时候无法考虑到相邻面片的像素变化,因此即使利用平滑度损失函数也无法避免面片之间颜色过渡不自然的问题;另外这种方式存在一定的局限性,面片之间难以联结起来,导致此前研究攻击迁移性和隐蔽性的方法难以在此应用,一定程度上限制了纹理攻击的迁移性和隐蔽性.

2 迁移隐蔽对抗攻击方法

本节首先详细介绍了自动驾驶场景下如何实现对抗纹理攻击,分析了此前方法的缺陷所在并做出了适当的改进.在此基础上,本文将对抗样本的优化类比于机器学习模型的训练过程,设计了提升攻击迁移性的算法模块,并且采用风格迁移的方式,最终提出TSA方法.方法框架如图2所示.

2.1 问题定义

对抗攻击的训练通常包含以下步骤:采集足够多的图片数据作为训练数据集;初始化对抗样本;在每个训练小批次时读取一定数量的数据集图片,将对抗样本和数据集图片输入设计好的变换函数,制作成对抗图片;将对抗图片输入目标神经网络,利用预测结果计算损失函数,然后使用损失函数计算并回传梯度以更新对抗样本,重复前2个步骤,直至达到训练目标即可结束训练.

训练数据集记为$\{X,Y,\Theta_c\}$,其中X,Y分别表示背景图片和目标车辆的检测框真值,Θ_c表示相机位姿的参数,用于设置渲染器中相机位置和姿态.记3维模型网格为$\boldsymbol{M}_G$,其表面纹理为$\boldsymbol{T}$,本文使用的仿真器为神经渲染器,记为R.使用仿真器R并结合参数Θ_c,即可将带有纹理$\boldsymbol{T}$的3维模型在指定相机位姿下投影成2维的图像$\boldsymbol{O}\in\mathbb{R}^{H\times W\times 3}$,其中,$H$,$W$表示渲染的2维图像的高和宽,$\boldsymbol{O}$可表示为

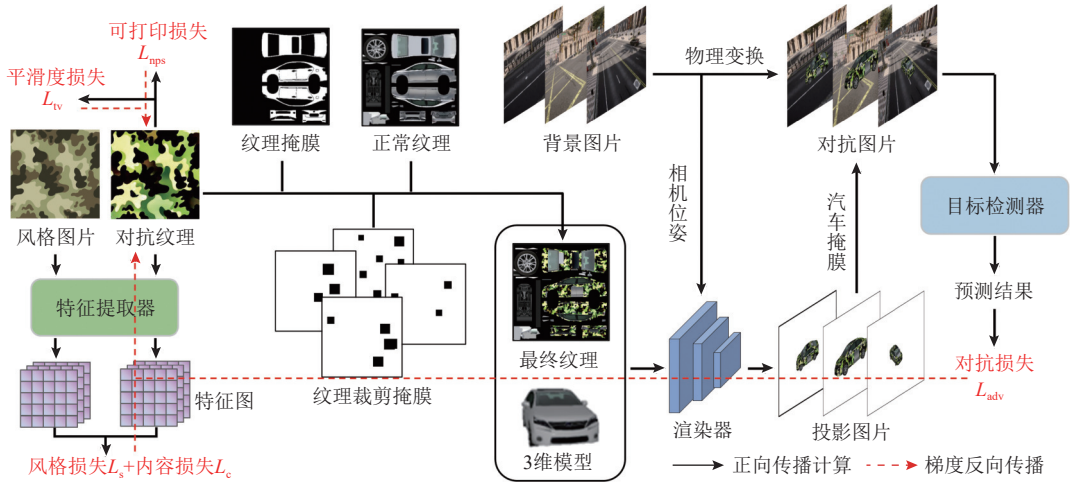


Fig. 2 TSA method framework

图 2 TSA 方法框架

$$\boldsymbol{O}=R(\boldsymbol{M}_{\mathrm{G}},\boldsymbol{T};\Theta_{\mathrm{c}}). \tag{1}$$

此前方法均直接使用 3 维模型的部分面片纹理作为对抗样本，如使用 3ds Max 软件读取 3 维网格模型，手动选择可以作为对抗样本的面片纹理并记录其序号，如图 3 所示，红色部分区域即为用于对抗攻击的纹理. 然而这些纹理互不相干，给攻击带来一定的问题（如 1.3 节所述）.

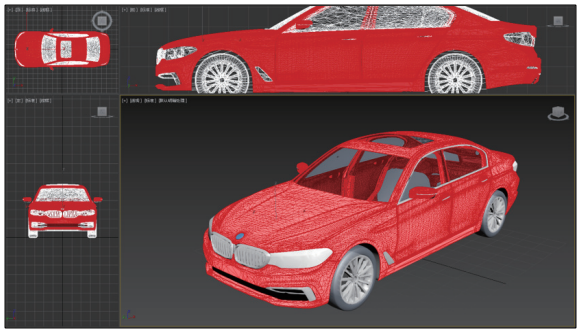


Fig. 3 Texture selection

图 3 纹理选择

因此我们利用 3ds Max 软件的“UVW 贴图”修改器将 3 维纹理进行 UV 展开，铺平成 2 维图片. 图 2 中的正常纹理即为 3 维模型初始纹理的展开图，记为 $\boldsymbol{T}_{\mathrm{n}}$. 与此前方法不同的是，我们可以直接在纹理图片的 2 维平面上选择可以粘贴对抗样本的区域，获取纹理掩膜，记为 $\boldsymbol{m}_{\boldsymbol{T}}$. 对抗纹理记为 $\boldsymbol{T}_{\mathrm{adv}}$，将对抗纹理在边缘处重复拼接至 3 维模型的纹理大小，记重复拼接函数为 $repeat(\cdot)$. 因此用于渲染的最终纹理 $\boldsymbol{T}$ 可以表示为

$$\boldsymbol{T}=repeat(\boldsymbol{T}_{\mathrm{adv}})\odot\boldsymbol{m}_{\boldsymbol{T}}+\boldsymbol{T}_{\mathrm{n}}\odot(1-\boldsymbol{m}_{\boldsymbol{T}}), \tag{2}$$

此处的⊙表示矩阵像素相乘. 这里，我们没有将铺平的全车身纹理直接作为对抗样本，是考虑到这种做法会限制对抗样本的使用范围. 由于有纹理掩膜的存在，在不能粘贴对抗纹理的部分区域，对抗纹理无法计算损失函数（因为掩膜在该区域的值为 0），使得该部分的纹理无法训练，因此最终获取的纹理仅能使用在训练时所用的车型上，难以迁移到其余车型上. 因此我们采用重复拼接函数，使得纹理能充分训练，即使将对抗纹理贴在其余的车型上也能成功攻击目标模型.

为了模拟物理环境，我们将渲染出的 2 维的汽车图片裁剪出来粘贴到不同背景环境的仿真图片中. 数据集中的仿真背景图片为 X，其对应的汽车掩膜（mask）为 $\boldsymbol{m}_{\mathrm{c}}$，因此对抗图片 I 可以表示为

$$\boldsymbol{I}=X\odot(1-\boldsymbol{m}_{\mathrm{car}})+O\odot\boldsymbol{m}_{\mathrm{car}}. \tag{3}$$

然后将对抗图片 $\boldsymbol{I}\in\mathbb{R}^{H\times W\times 3}$ 输入目标检测器，记为 F，得到检测结果 $\{\boldsymbol{b}_i\}=F(I;\theta_F)$. 其中 θ_F 为检测器的网络权重，在攻击纹理的训练过程中保持不变. $\boldsymbol{b}_i$ 表示检测结果，包括检测框和类别概率，可以表示为 $\boldsymbol{b}_i=\left(b_x,b_y,b_{\mathrm{w}},b_{\mathrm{h}},b_{\mathrm{obj}},\boldsymbol{b}_{\mathrm{cls}}\right)$. 其中 b_x，b_y 为检测框的中心坐标，b_{w}，b_{h} 为检测框的宽和高，b_{obj} 为检测框中包含前景物体的置信度，$\boldsymbol{b}_{\mathrm{cls}}$ 为检测框中目标的类别概率向量. 本文所攻击的检测器均由 coco 数据集[35] 预训练，因此输出的分类类别数为 80，即 $\boldsymbol{b}_{\mathrm{cls}}$ 为一个 80 维的向量.

本文的攻击方式是通过更新 3 维汽车模型表面的纹理，使其具有攻击检测器的能力，因此目标类别选用“car”即汽车. 这是符合自动驾驶场景需求的，因为在自动驾驶车辆的目标检测器前，存在各种不同的环境，比如停车场、十字路口、城市街道等，这些环境中出现最多的物体即为汽车，因此攻击自动驾

驶车辆的检测器最便捷的方式便是将攻击纹理粘贴在路边的汽车上. 我们利用检测器的预测结果$\{\boldsymbol{b}_i\}$计算损失函数, 通过梯度回传即可训练纹理, 即

$$\boldsymbol{T}_{\text{adv}}^{*}=\arg\max_{\boldsymbol{T}_{\text{adv}}} J(F(I;\theta_F),Y), \tag{4}$$

此处$\boldsymbol{T}_{\text{adv}}^{*}$表示最优的攻击纹理, $J(\cdot,\cdot)$表示代价函数. 通过优化式(4), 最终可以得到对抗攻击纹理.

2.2 损失函数设计

在TSA方法的训练过程中, 损失函数起到十分重要的作用, 本文设计的损失函数综合考虑纹理的攻击性能和隐蔽性.

纹理的攻击性能依靠对抗损失(adversarial loss)函数. 由于攻击的目标模型为目标检测器, 从模型输出的角度出发, 对抗损失函数需要包括3个部分: 首先是针对交并比(intersection over union, IoU)的损失, 记为$L_{\text{adv}}^{\text{iou}}$; 然后针对检测器对前景物体检测的损失, 记为$L_{\text{adv}}^{\text{obj}}$; 最后一部分是对类别的攻击, 保证即使检测器能够检测出前景物体并定位汽车, 也不能正确地识别, 该部分损失记为$L_{\text{adv}}^{\text{cls}}$. 因此最终的对抗损失函数可以表示为

$$L_{\text{adv}}=L_{\text{adv}}^{\text{iou}}+L_{\text{adv}}^{\text{obj}}+L_{\text{adv}}^{\text{cls}}. \tag{5}$$

下面详细介绍各损失项的计算. 首先是交并比损失$L_{\text{adv}}^{\text{iou}}$. 交并比表示检测框的真值和预测值之间的重叠区域, 可以直接使用检测结果结合检测框真值进行计算. 计算公式如式(6)所示:

$$L_{\text{adv}}^{\text{iou}}=\sum_{i=1}^{N} IoU(\boldsymbol{b}_i,\boldsymbol{b}_{\text{gt}}), \tag{6}$$

其中$\boldsymbol{b}_i$表示第i个检测框的预测结果, N为用于计算损失函数的检测框数量, $\boldsymbol{b}_{\text{gt}}$表示检测框的真值. 这里检测框的数量是经过2次筛选后得到的, 因为本文主要攻击的类别为汽车, 不需要影响其余类别物体的检测, 因此可以使用类别概率的最大索引去进行第1次筛选. 如果当前检测框的最大置信度类别为汽车, 就将其保留下来, 否则抛弃. 第2次筛选是为了去除前景物体概率较小的检测框. 我们设置了一个阈值, 保留前景概率和汽车类别概率的乘积大于阈值的部分检测框, 仅需将该乘积用于计算损失函数. 通过最小化$L_{\text{adv}}^{\text{iou}}$, 可以将目标物体周围的多个检测框与真值的交并比缩小至阈值以下, 即可在检测的过程中将其过滤, 从而实现攻击的目的.

然后是前景物体概率损失项$L_{\text{adv}}^{\text{obj}}$, 它表示当前检测框中前景物体存在的置信度. 经过式(6)的筛选, 剩余的检测框都是对汽车的检测, 并且置信度较高, 因此将筛选后所有的检测框的前景物体概率求和, 即前景物体概率损失, 表示为

$$L_{\text{adv}}^{\text{obj}}=\sum_{i=1}^{N} b_{i,\text{obj}}. \tag{7}$$

最后一部分是类别损失项$L_{\text{adv}}^{\text{cls}}$, 它表示检测框内物体的类别概率, 本文攻击的类别为汽车, 因此该损失项由汽车的类别概率组成, 计算公式如式(8)所示, 其中$b_{i,\text{car}}$表示类别向量$\boldsymbol{b}_i$中类别为汽车的概率分量. 通过最小化类别损失项, 可以使得检测器误分类, 即使能正确地定位也不能正确地分类出目标汽车.

$$L_{\text{adv}}^{\text{cls}}=\sum_{i=1}^{N} b_{i,\text{car}}. \tag{8}$$

为了保证攻击的隐蔽性, 本文从2个角度着手. 首先是尽量减少颜色的畸变, 让纹理的颜色过渡尽可能地平滑, 因此采用全变分(total variation)距离计算平滑度损失(smoothness loss), 记为L_{tv}, 计算公式如式(9)所示, 其中$x_{i,j}$表示对抗纹理图片于坐标(i,j)处的像素值. 通过最小化平滑度损失项, 能够有效地降低相邻像素之间的不连续性, 使生成的纹理更平滑, 达到自然的效果.

$$L_{\text{tv}}=\sum_{i,j}\sqrt{(x_{i,j}-x_{i,j+1})^2+(x_{i,j}-x_{i+1,j})^2}. \tag{9}$$

提高隐蔽性的另一个措施是利用风格迁移(style transfer)[36]的方法, 将对抗样本伪装成自然图案的风格. 考虑到自动驾驶场景, 涂装在汽车上的图案可以是具有语义信息的图案, 比如笑脸或者动画角色等图案, 也可以是装饰性的迷彩图案等, 如图4所示.



Fig. 4 Cars with decorative patterns

图4 带有装饰图案的汽车

风格迁移指的是将一张图像的风格迁移到另一张图像上, 产生一张混合两者风格的新图像. 具体来说, 将风格图片$\boldsymbol{s}$和目标图片的格拉姆矩阵[36]做差值, 使用差值的均方误差作为风格损失函数L_{s}, 在训练的过程中使得两者相关性逐渐增强, 即可让目标图片学习到风格图片的风格样式. 此处目标图片即为待优化的对抗纹理图片$\boldsymbol{T}_{\text{adv}}$. L_{s}的计算公式可以表示为

$$L_{\text{style}}=\sum_{l\in S_l}|g(\mathrm{F}_l(\mathrm{T}_{\text{adv}}))-g(\mathrm{F}_l(\mathrm{s}))|_2^2, \tag{10}$$

其中S_l表示神经网络中需要用于计算风格损失项的特征层序号，F_l表示特征提取器，通常采用 VGG19[37] 或者 ResNet52[38] 网络. 除了风格损失，还需要使用内容损失L_{content}作为约束，保证在训练的过程中目标图片和内容图片 $\boldsymbol{c}$ 的内容相近，将初始的语义信息保留下来，可以表示为

$$L_{\text{content}}=\sum_{l\in C_l}|F_l(\boldsymbol{T}_{\text{adv}})-F_l(\boldsymbol{c})|_2^2, \tag{11}$$

其中C_l表示神经网络中需要用于计算内容损失项的特征层序号. 对比笑脸或动画角色的图案，迷彩图案的风格更为明显和统一，用于风格迁移的效果会更好，因此我们选用迷彩图案作为风格图片，考虑到生成的目标图片应尽可能接近于初始的迷彩图案，因此将迷彩图案同样作为内容图片，如图 5 所示为迷彩图案和贴有迷彩图案的 3 维汽车模型.



Fig. 5 Camouflage patterns and a 3D car model with camouflage patterns

图 5 迷彩图案和贴有迷彩图案的 3 维汽车模型

最后，物理域对抗攻击不得不考虑可实施性问题，即生成的对抗样本是否能够在现实中打印出来. 这是因为数字域的颜色通常采用 0~255 像素的表示方式，而现实中的打印机往往只能打印几种固定的颜色组合，记为 P，所以需要设计额外的可打印损失函数[39]，使得优化生成的补丁颜色和颜色集合 P 中的颜色相近，以尽量减小打印误差，记为L_{nps}，可由下式计算：

$$L_{\text{nps}}=\sum_{i,j}\min_{\boldsymbol{p}\in P}|\boldsymbol{p}-x_{i,j}|. \tag{12}$$

综上所述，最终的损失函数为上述各损失项的加权和，记为L_{total}，如式(13)所示：

$$L_{\text{total}}=(L_{\text{s}}+L_{\text{c}})+(L_{\text{tv}}+L_{\text{nps}})+\lambda L_{\text{adv}}, \tag{13}$$

其中，λ表示对抗损失项的权重，用于调整攻击性能和隐蔽性的比重.

2.3 提升迁移性的设计

2.1 节和 2.2 节分别详细介绍了攻击的基本实施流程以及损失函数设计，实验证明上述方法能够有效地攻击目标检测器，造成检测器对目标车辆的误检或漏检，但是在攻击的迁移性上仍有待提高，对黑盒模型的攻击效果还有所欠缺，因此受文献 [40] 启发，本节针对攻击迁移性采用 3 个改进算法模块.

此前关于黑盒攻击的研究主要有 2 种方式：基于查询的和基于迁移的. 基于查询的方法允许黑盒攻击能够获取目标模型的输出信息，然而自动驾驶场景下的目标检测模型无法提供查询信息，因此仅能参考基于迁移的攻击方法. 参考文献 [40] 的思路，可以将对抗样本优化的过程类比于模型训练的过程，因为两者在本质上有多种相似之处：两者都需要训练数据集和都需要将数据集输入到神经网络模型中以及优化的过程都是通过损失函数回传梯度去更新待优化变量. 综合看来，可以尝试将模型训练所用的策略使用在对抗样本优化中，例如将数据增强和防止过拟合的正则化方式对应到纹理优化中，我们提出了 3 种改进算法模块，分别为：物理变换、权重衰减和纹理暂退.

首先对应数据增强的是物理变换，这里我们采用变换期望（expectation over transformation, EOT）函数[41]，该函数能够有效地提升对抗样本的鲁棒性和迁移性. 具体地，在训练过程中，加入多种模拟物理域的变换，例如明亮度变换，能够使得对抗样本在不同光照条件下保持稳定的攻击效果. 表 1 展示了各种变换类型和参数范围，以及变换模拟的目标. 因此我们在对抗图片的裁剪中加入物理变换，式(3)可以重写为式(14)，其中$EoT(\cdot)$表示变换期望函数.

$$I=EoT(X\odot(1-M_{\text{car}})+O\odot M_{\text{car}}). \tag{14}$$

Table 1 Transformation Distribution

表 1 变换分布

变换类型	参数值	目标
随机噪声	±0.1	拍摄、打印噪声
旋转/(°)	±20	相机拍摄模拟
明亮度变换	±0.1	光照模拟
对比度变换	±0.2	相机参数模拟
随机裁剪	[0.7, 1.0]	拍摄/遮挡模拟
仿射变换	0.7	视角/变形模拟
缩放	[0.25, 1.25]	距离/尺寸模拟

除了借鉴数据增强外，TSA 方法还借鉴了模型训练时防止过拟合的措施. 过拟合指的是模型过度适应训练数据，无法泛化到未知测试数据，是机器学习模型训练过程中常见的问题. 对抗样本的训练与之类似，优化迭代次数越多，针对训练模型的攻击效

果越好, 而对黑盒模型的效果越差, 因此完全可以认为对抗补丁的训练同样存在过拟合的问题. 在这里, 我们采用 2 种措施: 权重衰减和纹理暂退(dropout). 权重衰减使用动态衰减的学习率, 如式(15)所示, lr 为学习率, μ为衰减因子, ε_1, ε_2为控制学习率更新的阈值, L_t为第 t 代的损失值.

$$lr \Leftarrow lr \times \mu, \quad \text{if } (L_t - L_{t-1}) < \varepsilon_1 \text{ and } \frac{(L_t - L_{t-1})}{L_t} < \varepsilon_2. \tag{15}$$

纹理暂退表现为训练过程中对对抗纹理的随机裁剪, 随机减小纹理在对抗图片中的比重, 如图 2 中的纹理掩膜$\boldsymbol{m}_{\mathrm{c}}$所示, 黑色部分为纹理裁剪的区域, 该部分的对抗纹理由正常纹理替代. 在每次迭代时生成随机数量和尺寸的矩形, 用于表示纹理裁剪的掩膜. 因此式(2)应改进为

$$\boldsymbol{T} = repeat(\boldsymbol{T}_{\mathrm{adv}}) \odot \boldsymbol{m}_{\boldsymbol{T}} \odot \boldsymbol{m}_{\mathrm{c}} + \boldsymbol{T}_{\mathrm{n}} \odot (1 - \boldsymbol{m}_{\boldsymbol{T}} \odot \boldsymbol{m}_{\mathrm{c}}). \tag{16}$$

因此, 最终的攻击算法如算法 1 所示.

算法 1. TSA 方法.

输入: 训练数据集$\{X, Y, \Theta_{\mathrm{c}}\}$, 纹理掩膜$\boldsymbol{m}_{\boldsymbol{T}}$, 纹理裁剪掩膜$\boldsymbol{m}_{\mathrm{c}}$, 汽车掩膜 M_{car}, 正常纹理$\boldsymbol{T}_{\mathrm{n}}$, 3 维模型$\boldsymbol{M}_{\mathrm{G}}$, 神经渲染器 R, 目标检测器 F, 检测器权重θ_F, 风格图片 $\boldsymbol{s}$, 内容图片 $\boldsymbol{c}$;

输出: 对抗攻击纹理$\boldsymbol{T}_{\mathrm{adv}}$.

① 初始化对抗纹理: $\boldsymbol{T}_{\mathrm{adv}} \leftarrow \boldsymbol{s}$;

② for 最大迭代次数 do

③ 选择批次$\boldsymbol{x} \in X$, $\boldsymbol{m}_{\mathrm{car}} \in M_{\mathrm{car}}$, $\boldsymbol{\theta}_{\mathrm{c}} \in \Theta_{\mathrm{c}}$;

④ 随机生成纹理裁剪掩膜$\boldsymbol{m}_{\mathrm{c}}$;

⑤ 计算最终纹理: $\boldsymbol{T} = repeat(\boldsymbol{T}_{\mathrm{adv}}) \odot \boldsymbol{m}_{\boldsymbol{T}} \odot \boldsymbol{m}_{\mathrm{c}} + \boldsymbol{T}_{\mathrm{n}} \odot (1 - \boldsymbol{m}_{\boldsymbol{T}} \odot \boldsymbol{m}_{\mathrm{c}})$;

⑥ 获取渲染图片: $\boldsymbol{o} = R(\boldsymbol{M}_{\mathrm{G}}, \boldsymbol{T}; \boldsymbol{\theta}_{\mathrm{c}})$;

⑦ 计算对抗图片: $\boldsymbol{i} = EoT(\boldsymbol{x} \odot (1 - \boldsymbol{m}_{\mathrm{car}}) + \boldsymbol{o} \odot \boldsymbol{m}_{\mathrm{car}})$;

⑧ 获取预测结果: $\{\boldsymbol{b}_i\} \leftarrow F(\boldsymbol{i}; \theta_F)$;

⑨ 使用式(13)计算损失函数;

⑩ 更新对抗纹理$\boldsymbol{T}_{\mathrm{adv}}$;

⑪ 使用式(15)调整学习率;

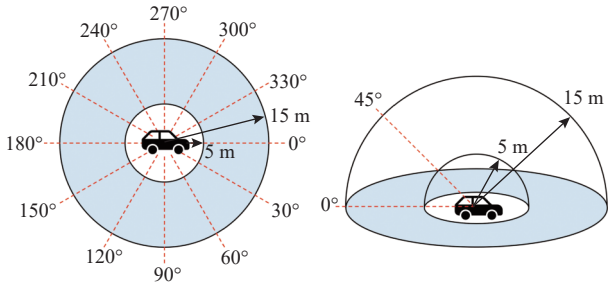
⑫ end for

3 实验结果及分析

3.1 数据集

本文的攻击目标是自动驾驶场景下的目标检测器, 由于物理域的数据采样难度大且成本较高, 并且不易进行对抗攻击的训练, 因此我们仿照文献 [23] 的实验设置, 使用 Carla 仿真器[42] 采样数据.

我们随机选择一个地图(Carla 中的 Town10HD), 并随机选择 60 个生成位置来生成训练数据集和验证数据集, 并选择 40 个生成位置来生成测试数据集. 对于每个位置, 我们使用不同的相机位姿共捕获 108 张仿真背景图片, 相机的位姿参数由图 6 所示, 距离范围为 5~15 m, 俯仰角范围为 0~45°, 偏航角范围为 0~360°. 因此一共采样训练数据集和验证数据集 6 480 张, 测试数据集 4 320 张. 由于 TSA 方法生成的对抗纹理可用于不同车型, 因此测试集随机使用 3 种汽车类型进行评估, 以验证攻击的鲁棒性.



(a) 俯视图 (b) 侧视图

Fig. 6 Positions and poses of camera

图 6 相机位姿

3.2 实验设置

训练采用 Adam 优化器[43], 设置初始学习率为 0.01, 衰减因子μ为 0.8, 阈值ε_1, ε_2均为 1×10^{-4}[40], 最大迭代次数为 15, 对抗损失项的权重系数λ为 2. 物理变换的参数如表 1 所示. 纹理的尺寸为 224×224, 纹理裁剪掩膜生成的矩形边长为 20~50, 数量为 3~8. 输入目标检测器的图片尺寸为 640×640, 训练时批次大小(batch size)为 6, GPU 为 NVIDIA RTX3090 24 GB. 目标检测器选择目前主流的多种应用类型, 包括基于深度卷积神经网络的 Yolov3[44], Yolov7[45], SSD[46], FR(faster R-CNN)[47], MR(mask R-CNN)[48], 以及基于 Transformer[49] 的 DDTR[50], PVT[51]. 对比的代表性攻击方法包括同样是使用神经渲染和仿真实现的 DAS[23], FCA[24], CAMOU[25], ER[27]. 另外, 测试正常纹理的检测结果作为攻击效果的参考.

3.3 鲁棒性评估

我们在 Yolov3 上训练了 TSA 方法和此前攻击方法的攻击纹理, 并在多个检测器上对比攻击性能, 评测指标选用目标检测器的检测指标, 即汽车类别的平均精度(average precision, AP), AP 越小则攻击效果越好. 如表 2 所示, 我们对比了此前所有将神经渲染用于攻击自动驾驶场景目标检测器的方法, 同时测试了正常纹理下汽车的 AP@0.5. 从表 2 中可以看出, TSA 方法在几乎所有模型上均获得了最好的攻击效果, 证明了在仿真环境中检测器依然具有正常

Table 2 Attack Effect of Each Method for Different Detectors

表 2 各方法针对不同检测器的攻击效果 %

攻击方法	基于深度卷积神经网络					基于 Transformer	
	Yolov3*	Yolov7	SSD	FR	MR	DDTR	PVT
正常纹理	90.61	93.89	92.12	88.47	92.36	88.51	91.06
DAS[23]	81.27	86.83	78.39	82.61	85.20	85.33	86.57
FCA[24]	73.69	78.24	71.60	72.83	79.43	76.80	81.61
CAMOU[25]	55.20	59.45	62.08	63.48	58.79	40.51	65.72
ER[27]	46.16	52.63	65.74	**51.06**	55.21	37.06	69.55
TSA（本文）	**43.25**	**46.71**	**45.92**	52.64	**53.10**	**31.85**	**51.07**

注：带有*的为白盒模型. 黑体数值表示最优值.

的检测性能. 另外，我们注意到 DAS 和 FCA 生成的纹理攻击性能相对较差，这是因为这 2 种方法没有考虑物理条件对攻击的影响. 而 CAMOU 和 ER 尽管在训练时考虑了物理变换，但是由于两者采用的纹理为 16×16 的像素块，优化空间有限，且均为白盒攻击，无法在黑盒模型上取得更好的攻击效果，因此效果受到一定的影响. 相比之下，TSA 采用了物理变换的同时，考虑了迁移攻击的设置，并且待优化的纹理空间更大，能获取更优的解. 图 7 展示了纹理涂装在 3 种车型上的效果，在不同的角度和距离下，纹理涂装均能有效地攻击目标检测器，造成误检或漏检.

另外，我们总结了在不同的相机位姿下攻击性能的对比情况，如图 8 所示，AP@0.5 为所有检测模型的平均结果，TSA 在不同的相机位姿下均保持相对稳定. 此外，我们观察到相机的角度参数比相机的距离参数更能影响攻击效果. 这是因为在不同的角度下，攻击纹理覆盖的区域面积发生变化，某些角度下覆盖面积较小，因此攻击效果会有所下降.

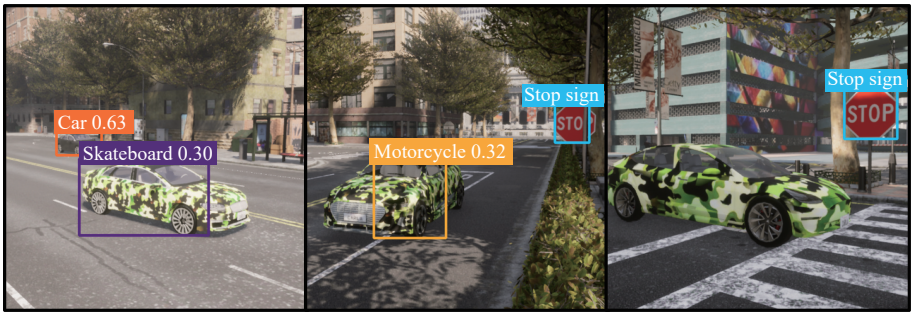


Fig. 7 Attack effect of TSA method

图 7 TSA 方法攻击效果

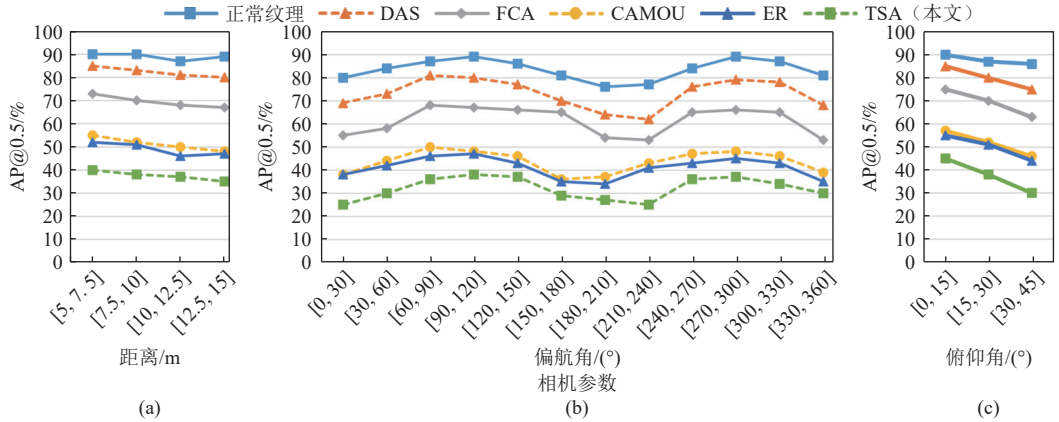


Fig. 8 Attack effect at different positions and poses of camera

图 8 不同相机位姿下的攻击效果

由于在自动驾驶场景下，道路上通常不止目标车辆，还有行人、其余的车辆或者路边的树木等，因此衡量对抗纹理被遮挡情况下的攻击效果是有必要的. 根据遮挡区域面积占汽车总面积的比例的不同，分为 3 个等级进行测试，分别为小（10%~30%）、中（30%~50%）、大（50%~70%）. 由于遮挡位置是随机的，不便使用 AP 进行衡量，因此这里选用攻击成功率（attack success rate, ASR）评估攻击的性能，当检测器误检或漏检目标汽车时则视为攻击成功，使用百分数表示. 检测器采用 Yolov3，实验记录如表 3 所示，

攻击效果如图 9 所示, 可以看出遮挡几乎不影响纹理的 ASR, 而当遮挡过大时 ASR 会升高(升高的原因是过多的遮挡不仅遮住了攻击纹理, 而且遮住了汽车本身的特征, 比如轮胎、车窗等, 反而使检测器检测失败). 相比之下, TSA 的攻击效果优于此前的方法, 这得益于在训练过程中的纹理暂退操作模拟了缺失部分纹理的攻击情况, 提高了攻击鲁棒性.

Table 3 ASRs of Each Method with Partial Occlusion

表 3 各方法在部分遮挡时的 ASR %

攻击方法	遮挡比例		
	小	中	大
正常纹理	0.21	0.65	1.33
DAS[23]	12.12	13.55	15.43
FCA[24]	19.38	21.77	23.24
CAMOU[25]	56.97	57.63	59.63
ER[27]	58.35	60.57	63.15
TSA (本文)	**70.06**	**72.89**	**74.65**

注: 黑体数值表示最优值.



Fig. 9 Attack effect with partial occlusion

图 9 部分遮挡时的攻击效果

最后, 考虑到正常道路上车辆种类较多, 因此对抗性纹理的攻击通用性是值得关注的. 通过实验, 我们探究了现有工作和 TSA 方法在较常见的 3 种汽车 Lincoln MKZ2017, Audi Etron, Tesla Model 3 上的攻击效果. 由于 DAS[23] 和 FCA[24] 这 2 种方法产生的纹理仅能使用在训练时所用的 Audi Etron 汽车上, 因此不具备通用性. 检测器采用 Yolov3, 实验记录如表 4 所示. 实验结果表明, 3 种纹理攻击在不同汽车上均能保持攻击效果, 几乎不受汽车类型的影响. 但相比之下, TSA 方法的攻击效果优于此前的方法, 这得益于 TSA 的优化框架采用了多种改进措施, 大幅提升了攻击迁移性和鲁棒性.

Table 4 ASRs of Each Method on Various Cars

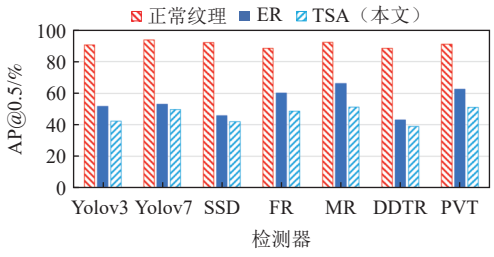
表 4 各方法在不同汽车上的 ASR %

攻击方法	汽车种类		
	Lincoln	Audi	Tesla
正常纹理	0.19	0.23	0.20
CAMOU[25]	55.32	58.15	56.07
ER[27]	57.61	59.24	58.39
TSA (本文)	**69.45**	**71.83**	**70.04**

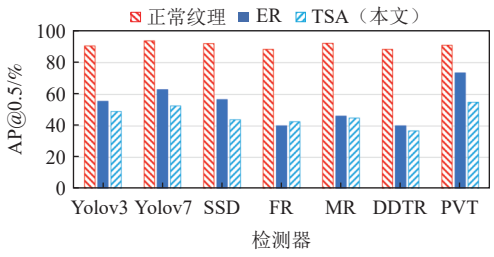
注: 黑体数值表示最优值.

3.4 迁移性评估

在 3.2 节中, 我们使用 Yolov3 检测器[44] 训练了 TSA 方法和此前攻击方法的攻击纹理, 并对比攻击性能, 如表 2 所示. 其中, 在 Yolov3 上的攻击结果反映了白盒攻击性能, 而在其余的模型上的攻击结果则反映了纹理的黑盒攻击性能. 对比之下, TSA 方法在 Yolov3 上的攻击效果略优于此前的攻击方法, 而在其余的黑盒模型上的攻击效果基本明显优于此前方法, 因此表明 TSA 具有一定的攻击迁移能力. 为了验证这个结论, 本节在另外的检测器上重新训练纹理并进行黑盒攻击实验, 这里选择此前方法中最优的 ER[27] 进行对比. 具体来说, 我们在 SSD[46], FR[47] 检测器上分别重新训练了 TSA 和 ER 方法, 并在相同的实验设置下测试黑盒攻击性能, 记录如图 10 所示. 从图 10 可以看出, 2 种方法对白盒模型的攻击效果差别不大, 而在黑盒模型上 TSA 明显优于 ER.



(a) SSD检测器



(b) FR检测器

Fig. 10 Assessment for attack transferability

图 10 攻击迁移性评估

3.5 隐蔽性评估

在自动驾驶场景, 纹理攻击的隐蔽性与汽车所

处的环境、自然光照以及观察者注意力有关[52]. 攻击者无法控制前 2 个因素，因此只能设法构造能够分散人类观察者注意力的纹理图案. 此前自动驾驶场景下的隐蔽性纹理攻击代表性工作为 DAS[23]，其试图通过纹理中突出的笑脸图案分散人的注意力，并控制纹理尽量平滑，以达到隐蔽性攻击的效果. 相比之下，TSA 方法更关注于整体纹理的风格统一，采用风格迁移的方式学习、模仿大众较为常见的伪装图案，以躲避观察者的关注. 图 11 展示了 DAS 和 TSA 方法生成的纹理样式，由于个人的主观评价不一定准确，因此我们参考文献 [52]，使用双先验对齐（dual prior alignment, DPA）网络[52] 对攻击纹理进行量化评估. DPA 网络充分学习了人在自然程度推理中的评价标准和凝视行为，能够将纹理的自然程度量化为确切的指标，即平均观点分数（mean opinion score, MOS），模拟人对纹理自然程度的评分，取值范围为 0~5，数值越大表示自然程度越高，即攻击越自然. 如图 11 所示，正常纹理和初始伪装纹理的自然程度都较高，相比之下，TSA 的攻击纹理虽有所下降，但比 DAS 纹理的 MOS 指标更高，因此更为自然隐蔽.



(a) 正常纹理 (MOS为3.93) (b) 伪装纹理 (MOS为3.17) (c) TSA纹理 (MOS为2.81) (d) DAS纹理 (MOS为2.30)

Fig. 11 Comparison of stealthiness

图 11 隐蔽性比较

3.6 物理域实验评估

对于物理域攻击，我们使用多个检测器验证了 TSA 生成纹理的攻击有效性. 由于成本条件的限制，参考文献 [23−24] 的做法，我们使用 HP CP5225dn 彩色打印机将 TSA 生成的纹理打印下来，并粘贴到 22∶1 等比例缩小的特斯拉 model Y 的模型小车表面，在不同距离和角度下测试攻击效果. 相机角度设置如 3.1 节所述，由于采用的目标物体为等比例汽车模型，因此将相机距离设置为前述的 1/22. 拍摄采用红米 k60 原装相机，共拍摄 144 张图片. 粘贴纹理前后的图片如图 12 所示，粘贴攻击纹理后汽车模型能够有效地欺骗目标检测器，导致检测器对汽车模型漏检或误检. 采用 ASR 作为攻击评估指标，实验结果如表 5 所示. 可以看出，TSA 方法生成的纹理能够有效地攻击各类检测器，使得检测器不能识别和定位目标物体.

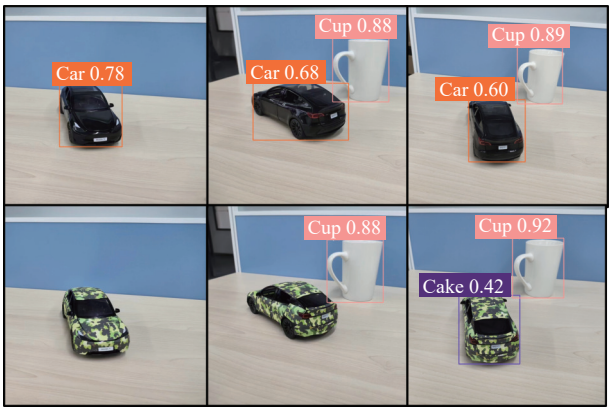


Fig. 12 Physical-world attack effect

图 12 物理域攻击效果

Table 5 Evaluation for Physical-World Attack Effect

表 5 物理域攻击效果评估 %

攻击方法	检测器类别			
	Yolov3	SSD	FR	DDTR
正常纹理	1.39	2.08	1.39	2.78
TSA（本文）	**83.33**	**87.50**	**79.86**	**89.58**

注：黑体数值表示最优值.

3.7 消融实验

为了充分评估 TSA 方法，我们使用消融实验分别探究了各算法模块的作用. 如表 6 所示，我们将不同的算法模块组合，使用 Yolov3 检测器优化并生成攻击纹理，用汽车的 AP@0.5 表示攻击效果，因此 Yolov3 的 AP@0.5 表示了白盒攻击性能，其余检测器的 AP@0.5 表示了黑盒攻击性能，即攻击迁移性. 如表 6 中结果所示，当添加了算法模块后，SSD 等 3 种检测器的 AP 均有所下降，表明了 3 种算法模块均对攻击迁移性有一定的帮助. 并且当加入物理变换后，在 Yolov3 上的白盒攻击性能也有所提升，这是因为变换提高了攻击的鲁棒性，能够有效地抵御物理域中光线环境以及相机位置等因素对攻击带来的影响.

Table 6 AP@0.5 of Different Modules Combination

表 6 不同模块组合的 AP@0.5 %

算法模块	检测器类别			
	Yolov3	SSD	FR	DDTR
TSA（本文）	56.12	59.74	60.38	52.71
+物理变换	43.18	52.27	57.06	43.52
+权重衰减	42.13	49.36	53.79	38.04
+纹理暂退	43.25	45.92	52.64	31.85

4 结 论

本文重点关注自动驾驶领域中针对目标检测器的对抗攻击,考虑到自动驾驶场景下的背景、光照条件等因素复杂多变,攻击目标模型结构未知,并且当前常用方法的攻击补丁或纹理容易让人起疑,本文提出了针对自动驾驶场景下迁移的、隐蔽的TSA方法.本文通过仿真实验测试,证明了TSA方法所训练出的纹理具有迁移的攻击能力,并且在外观上具有一定的隐蔽性.使用物理域实验证明了TSA方法能够在物理世界中保持有效的攻击性能.最后通过消融实验论证了本文所提出的算法模块的必要性.

本文提出的TSA方法的主要目的是为逃避检测器识别,是一种无目标攻击.相比之下,有目标攻击更值得关注,即将车辆攻击为行人、路牌等,但是TSA方法没有相关的改进模块,仍存在一定的局限性.未来的工作主要有4个方面:1)针对有目标攻击,设计改进的算法模块;2)针对所提出的攻击方法设计一个可行的、低成本和有效的防御措施和方法;3)深入学习自动驾驶系统其余的探测结构,进一步学习自动驾驶决策系统的结构原理,研究更深层次的、更系统化和多模块化的攻击方法;4)学习自动驾驶仿真技术,测试攻击方法对自动驾驶系统决策的影响.

作者贡献声明:郑君豪负责相关文献的收集、方法设计并完成论文撰写;蔺琛皓负责方法设计和论文修订;赵正宇负责方法修改和论文修订;贾子羿负责文献调研;吴黎兵和沈超指导论文撰写、提出修改建议、校对并最终审核论文.

参 考 文 献

[1] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks[J]. arXiv preprint, arXiv: 1312.6199, 2014

[2] Goodfellow I, Shlens J, Szegedy C. Explaining and harnessing adversarial examples[J]. arXiv preprint, arXiv: 1412.6572, 2015

[3] Madry A, Makelov A, Schmidt L, et al. Towards deep learning models resistant to adversarial attacks[J]. arXiv preprint, arXiv: 1706.06083, 2018

[4] Kurakin A, Goodfellow I, Bengio S. Adversarial examples in the physical world[J]. arXiv preprint, arXiv: 1607.02533, 2017

[5] Carlini N, Wagner D. Towards evaluating the robustness of neural networks[C]//Proc of the 38th IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2017: 39−57

[6] Moosavi-Dezfooli S, Fawzi A, Fawzi O, et al. Universal adversarial perturbations[C]//Proc of the 35th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 86−94

[7] Kurakin A, Goodfellow I, Bengio S. Adversarial machine learning at scale[C/OL]//Proc of the 5th Int Conf on Learning Representations. 2017[2023-10-09]. https://openreview.net/forum?id=BJm4T4Kgx

[8] Wiyatno R, Xu Anqi. Maximal jacobian-based saliency map attack[J]. arXiv preprint, arXiv: 1808.07945, 2018

[9] Zhao Zhengyu, Zhang Hanwei, Li Renjue, et al. Revisiting transferable adversarial image examples: Attack categorization, evaluation guidelines, and new insights[J]. arXiv preprint, arXiv: 2310.11850, 2023

[10] Zhao Zhengyu, Liu Zhuoran, Larson M. On success and simplicity: A second look at transferable targeted attacks[C]//Proc of the 35th Advances in Neural Information Processing Systems. La Jolla, CA: NIPS, 2021: 6115−6128

[11] Zhao Zhengyu, Liu Zhuoran, Larson M. Towards large yet imperceptible adversarial image perturbations with perceptual color distance[C]//Proc of the 38th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 1036−1045

[12] Zhao Zhengyu, Liu Zhuoran, Larson M. Adversarial image color transformations in explicit color filter space[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 3185−3197

[13] Yang Yulong, Lin Chenhao, Li Qian, et al. Quantization aware attack: Enhancing transferable adversarial attacks by model quantization[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 3265−3278

[14] Zheng Junhao, Lin Chenhao, Sun Jiahao, et al. Physical 3D adversarial attacks against monocular depth estimation in autonomous driving[C]//Proc of the 42nd IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2024: 24452−24461

[15] Jin Ran. Research and implementation of text adversarial example generation method[D]. Beijing: Beijing University of Posts and Telecommunications, 2021 (in Chinese)
(靳然. 文本对抗样本生成技术的研究与实现[D]. 北京:北京邮电大学, 2021)

[16] Liu Xin, Yang Huanrui, Liu Ziwei, et al. Dpatch: An adversarial patch attack on object detectors[J]. arXiv preprint, arXiv: 1806.02299, 2018

[17] Brown T B, Man D, Roy A, et al. Adversarial patch[J]. arXiv preprint, arXiv: 1712.09665, 2017

[18] Lang Dapeng, Chen Deyun, Shi Ran, et al. Attention-guided digital adversarial patches on visual detection[J]. Security and Communication Networks, 2021, 2021: 1−11

[19] Wu Zuxuan, Lim S, Davis L S, et al. Making an invisibility cloak: Real world adversarial attacks on object detectors[C]//Proc of the 16th European Conf on Computer Vision. Berlin: Springer, 2020: 12349−12366

[20] Hu Zhanhao, Huang Siyuan, Zhu Xiaopei, et al. Adversarial texture for fooling person detectors in the physical world[C]//Proc of the 40th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2022: 13307−13316

[21] Chen S, Cornelius C, Martin J, et al. Shapeshifter: Robust physical adversarial attack on faster R-CNN object detector[C]//Proc of the

European Conf on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. Berlin: Springer, 2019: 52−68

[22] Huang Lifeng, Gao Chengying, Zhou Yuyin, et al. Universal physical camouflage attacks on object detectors[C]//Proc of the 38th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 720−729

[23] Wang Jiakai, Liu Aishan, Yin Zixin, et al. Dual attention suppression attack: Generate adversarial camouflage in physical world[C]//Proc of the 39th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 8565−8574

[24] Wang Donghua, Jiang Tingsong, Sun Jialiang, et al. FCA: Learning a 3D full-coverage vehicle camouflage for multi-view physical adversarial attack[C]//Proc of the 36th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2022: 2414−2422

[25] Zhang Yang, Hassan F, Philip D, et al. CAMOU: Learning physical vehicle camouflages to adversarially attack detectors in the wild[C/OL]//Proc of the 7th Int Conf on Learning Representations. 2019[2023-11-05]. https://openreview.net/forum?id=SJgEl3A5tm

[26] Xiao Chaowei, Yang Dawei, Li Bo, et al. Meshadv: Adversarial meshes for visual recognition[C]//Proc of the 37th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 6891−6900

[27] Wu Tong, Ning Xuefei, Li Wenshuo, et al. Physical adversarial attack on vehicle detector in the carla simulator[J]. arXiv preprint, arXiv: 2007.16118, 2020

[28] Duan Yexin, Chen Jialin, Zhou Xingyu, et al. Learning coated adversarial camouflages for object detectors[C]//Proc of the 31st Int Joint Conf on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann, 2021: 891−897

[29] Kato H, Ushiku Y, Harada T. Neural 3D mesh renderer[C]//Proc of the 36th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 3907−3916

[30] Hu Y, Kung B, Tan D S, et al. Naturalistic physical adversarial patch for object detectors[C]//Proc of the 20th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2021: 7848−7857

[31] Creswell A, White T, Dumoulin V, et al. Generative adversarial networks: An overview[J]. IEEE Signal Processing Magazine, 2018, 35(1): 53−65

[32] Tan Jia, Ji Nan, Xie Haidong, et al. Legitimate adversarial patches: Evading human eyes and detection models in the physical world[C]// Proc of the 29th ACM Int Conf on Multimedia. New York: ACM, 2021: 5307−5315

[33] Duan Ranjie, Ma Xingjun, Wang Yisen, et al. Adversarial camouflage: Hiding physical-world attacks with natural styles[C]//Proc of the 38th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 1000−1008

[34] Liu Aishan, Huang Tairan, Liu Xianglong, et al. Spatiotemporal attacks for embodied agents[C]//Proc of the 16th European Conf on Computer Vision. Berlin: Springer, 2020: 122−138

[35] Lin T, Maire M, Belongie Se, et al. Microsoft coco: Common objects in context[C]//Proc of the 13th European Conf on Computer Vision. Berlin: Springer, 2014: 740−755

[36] Gatys L A, Ecker A S, Bethge M. Image style transfer using convolutional neural networks[C]//Proc of the 34th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 2414−2423

[37] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint, arXiv: 1409.1556, 2015

[38] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition[C]//Proc of the 34th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 770−778

[39] Sharif M, Bhagavatula S, Bauer L, et al. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition[C]//Proc of the 23rd ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2016: 1528−1540

[40] Huang Hao, Chen Ziyan, Chen Huanran, et al. T-sea: Transfer-based self-ensemble attack on object detection[C]//Proc of the 41st IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2023: 20514−20523

[41] Athalye A, Engstrom L, Ilyas A, et al. Synthesizing robust adversarial examples[C]//Proc of the 35th Int Conf on Machine Learning. New York: ACM, 2018: 284−293

[42] Alexey D, German R, Felipe C, et al. CARLA: An open urban driving simulator[J]. arXiv preprint, arXiv: 1711.03938, 2017

[43] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint, arXiv: 1412.6980, 2014

[44] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint, arXiv: 1804.02767, 2018

[45] Wang C, Bochkovskiy A, Liao H. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[J]. arXiv preprint, arXiv: 220702696, 2022

[46] Liu Wei, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector[C]//Proc of the 14th European Conf on Computer Vision. Berlin: Springer, 2016: 21−37

[47] Ren Shaoqing, He Kaiming, Girshick Ross, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 39(6): 1137−1149

[48] He Kaiming, Gkioxari G, Dollr P, et al. Mask R-CNN[C]//Proc of the 14th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2017: 2980−2988

[49] Vaswani A, Shazeer N, Parmar N. Attention is all you need[J]. arXiv preprint, arXiv: 1706.03762, 2017

[50] Zhu Xizhou, Su Weijie, Lu Lewei, et al. Deformable DETR: Deformable transformers for end-to-end object detection[J]. arXiv preprint, arXiv: 2010.04159, 2020

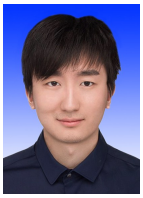
[51] Wang Wenhai, Xie E, Li Xiang, et al. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions[C]// Proc of the 20th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2021: 548−558

[52] Li Simin, Zhang Shuning, Chen Gujun, et al. Towards benchmarking and assessing visual naturalness of physical world adversarial attacks[C]//Proc of the 41st IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2023: 12324−12333



Zheng Junhao, born in 2002. PhD candidate. His main research interests include artificial intelligence security and autonomous driving test.

郑君豪, 2002 年生. 博士研究生. 主要研究方向为人工智能安全、自动驾驶测试.



Lin Chenhao, born in 1989. PhD, professor, PhD supervisor. His main research interests include artificial intelligence security, adversarial machine learning, deepfake, and identity authentication.

蔺琛皓, 1989 年生. 博士，教授，博士生导师. 主要研究方向为人工智能安全、对抗机器学习、深度伪造、身份认证.



Zhao Zhengyu, born in 1992. PhD, professor, PhD supervisor. His main research interests include artificial intelligence security and privacy, and adversarial machine learning.

赵正宇, 1992 年生. 博士，教授，博士生导师. 主要研究方向为人工智能安全与隐私、对抗机器学习.



Jia Ziyi, born in 2003. Master candidate. His main research interests include artificial intelligence security and autonomous driving test.

贾子羿, 2003 年生. 硕士研究生. 主要研究方向为人工智能安全、自动驾驶测试.



Wu Libing, born in 1972. PhD, professor, PhD supervisor. Member of CCF. His main research interests include wireless sensor networks, network management, and distributed computing.

吴黎兵, 1972 年生. 博士，教授，博士生导师. CCF 会员. 主要研究方向为无线传感器网络、网络管理、分布式计算.



Shen Chao, born in 1985. PhD, professor, PhD supervisor. Member of CCF. His main research interests include trusted artificial intelligence, artificial intelligence security, and network security.

沈　超, 1985 年生. 博士，教授，博士生导师. CCF 会员. 主要研究方向为可信人工智能、人工智能安全、网络安全.