

自动驾驶模拟测试平台——ADEPT

王森^{1*} 盛朱恒^{1*} 徐经纬^{1†} 陈韬略² 朱俊俊¹ 张舒慧¹ 姚远¹ 马晓星¹
¹ 南京大学计算机软件新技术国家重点实验室 ² 伦敦大学伯贝克学院计算机科学系

【摘要】 近年来, 自动驾驶系统 (Autonomous Driving System, ADS) 的有效质量保证方法引起了越来越多研究人员的兴趣。在本文中, 我们提出了一个新的测试平台 ADEPT, 它为基于深度神经网络 (Deep Neural Network, DNN) 的 ADS 提供了实用且全面的测试方法。ADEPT 基于虚拟模拟器 CARLA 实现, 提供了场景构建、ADS 接入、测试执行和记录测试等众多基础功能。ADEPT 具有两种不同的自动驾驶测试场景生成策略。第一, 我们基于现实生活中的事故报告, 利用自然语言处理技术制造丰富的驾驶场景。第二, 我们考虑 ADS 的反馈信息, 使用物理上鲁棒的对抗攻击, 从而生成闭环的攻击测试场景。后续的实验证实了该平台的有效性。

【关键词】 软件测试, 深度神经网络, 自动驾驶, 测试用例生成, 测试平台

1 引言

业内越来越关注自动驾驶系统的可靠性和安全性, 尤其是在使用深度神经网络作为其核心组件的情况下。测试作为保证 ADS 质量的最有效的方法之一, 在其开发过程中有着不可或缺的作用。在真实环境中使用真实的自动驾驶汽车进行测试是很自然的想法, 但这一想法显然既不经济也不安全。特别地, 在真实环境下重现各种类型的事故 (例如, 涉及在恶劣天气条件的事故), 这在技术上具有挑战性, 并且其成本在经济上也难以接受。一个更合理且有效的替代方案, 是在模拟环境下测试 ADS, 其中驾驶场景和物理逻辑 (即汽车动力学、交通规则和天气条件) 都建立在虚拟世界中。

在本文中, 我们提出了一个平台, ADEPT (Autonomous Driving tEsting PlatForm), 为 ADS 提供类似于现实世界场景的综合测试。我们可选的基础模拟器包括 CARLA^[1] 和 AirSim^[2]。鉴于 CARLA 拥有丰富的 API 和活跃的社区支持, 我们选择将我们的系统基于它来实现。ADEPT 支持各种功能, 包括场景构建、ADS 接入、测试执行和记录等。对于一个测试框架来说, 最重要的功能是测试用例生成, 在我们的设想中, 这一平台会通过构建特定场景的形式生成测试用例, 并接收 ADS 所做出反应, 将结果展示在虚拟场景中。自动驾驶这一应用场景所固有的复杂性产生了巨大的场景搜索空间, 而详尽的搜索显

然是不切实际的。此外, 从现实世界的角度来看, 寻找特殊测试用例生成方法容易产生不符合实际且意义不大的场景, 例如, 障碍物突然出现在路中间的场景, 这是没有意义的场景, 因为这种情况很少发生。

在 ADEPT 中, 我们引入了两种策略来有效地生成更真实的测试用例。第一种策略, 我们观察到危险情况与自动驾驶缺陷之间的联系, 我们利用这些危险情况将场景搜索引向更容易触及 ADS 缺陷的区域。为此, 我们使用自然语言处理 (Natural Language Processing, NLP) 技术从现实的交通事故报告中提取可能会触发 ADS 缺陷的危险因素, 然后将其用于生成测试场景。第二种策略, 我们采用 DNN 的对抗样本生成测试场景。鉴于 DNN 在 ADS 中的重要地位 (例如, 在决策模块中), 以及它们在面对对抗样本 (一种不易察觉的恶意扰动) 时的脆弱性, 生成测试场景并不困难。然而, 对抗样本的有效性会取决于各种物理条件 (例如, 照明、模糊和视角), 这一因素导致简单的测试场景容易失效。此外, 诱使 ADS 犯错的单个对抗攻击样本通常是不够的, 因为 ADS 可以从这些一次性的、零星的错误中恢复。在 ADEPT 中, 我们生成了一系列鲁棒的对抗样本, 这些样本将 ADS 的反馈考虑在内, 从而生成闭环地测试场景。

我们的代码发布在<https://github.com/shengzh-o0o0/ADEPT>, 视频发布在<https://www.bilibili.com/video/BV17G411A7nv/>。

2 系统介绍

图 1 描绘了 ADEPT 的三层架构, 即模拟器内核、ADS 测试库和场景引擎。

模拟器内核层包括几个直接基于 Unreal Engine (虚幻引擎) 和 Carla 构建的关键组件, 为自动驾驶测试提供特定操作, 例如交互式相机。图中红色框中的模块是我们开发的, 而蓝色框中的模块是由 CARLA 或虚幻引擎提供。

ADS 测试库层包括三个关键模块。

- 通信模块: 负责外部对象 (例如, 驾驶模型或攻击算法) 与场景内部对象 (例如汽车、人类和电子广告牌) 之间的交互。该模块旨在封装用于开发和记录专业自动驾驶测试场景的基本的 CARLA API, 因为通过 CARLA 的原生 API 在被测试对象和模拟器之间进行数据访问和传输

*同等贡献作者

†通讯作者

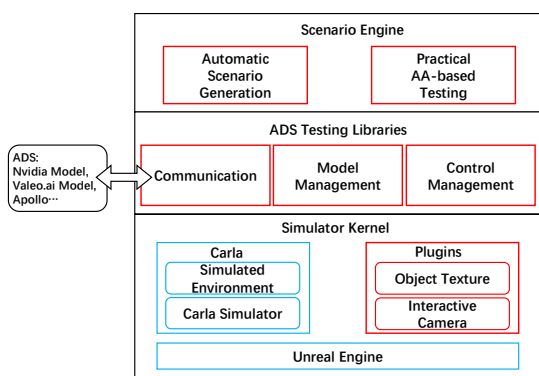


图 1: 平台框架

的接口复杂, 开发效率较低。

- 模型管理模块: 负责管理 ADS 和汽车, 例如, 将 ADS 连接到 ADEPT 并将所需的传感器放置在被测试的汽车上。
- 控制管理模块: 在线部署生成的场景, 例如, 利用重定向到平台对象的虚拟相机来实时调整物体表面纹理。

在通信模块接入自动驾驶方面, 我们尝试了三种不同层级的 ADS, 包括 NVIDIA 基于 CNN 的转向模型^[3] (L2 级别驾驶辅助), 2020 CARLA 自动驾驶挑战赛 Camera-Only 组第一名 Valeo.ai^[4] (L3 级别自动驾驶), Apollo* (L4 级别自动驾驶), 都可以连接成功并完成驾驶工作。

场景引擎层实现了多种方法有效地生成可能触发 ADS 缺陷的真实场景。目前, 有两种方法。

基于事故报告的测试场景生成, 这一部分可以获取的自然语言形式的交通事故报告转换为一种中间表示, 即场景描述语言。我们使用 Scenic^[5] 语言作为中间表示。Scenic 提供的工具可以同时加载场景并连接到 ADS 进行测试。

我们使用自然语言处理的方法来自动转换事故报告。在技术上, 我们使用自然语言大模型 GPT-3 所提供的问答技术。我们定义了一组问题以及对应的 Scenic 代码模板。首先, 将报告文本和预定义问题组合发送到 GPT-3^[6]。根据 GPT-3 的答案, 我们从预定义的模板中选择与事故报告描述相对应的模板, 并填写模板的重要缺失信息。具体来说, 我们询问了事故的位置 (包括事故发生在丁字路口、十字路口还是非路口道路)、时间、天气、以及涉事的两辆车之间的相对位置关系 (包括车道的左右关系、距离的前后关系等)。

基于对抗攻击的闭环测试, 对抗样本 (Adversarial Examples) 揭示了 DNN 的弱点, 而同时, DNN 在现代 ADS 中被广泛采用。因此可以很容易地利用对抗样本来测试 ADS。然而, 在现实世界条件下, 一些物理条件 (例如照明、模糊、视角等) 可能会严重降低对抗样本的有效性, 尤其是在汽车行驶时。此外, 复

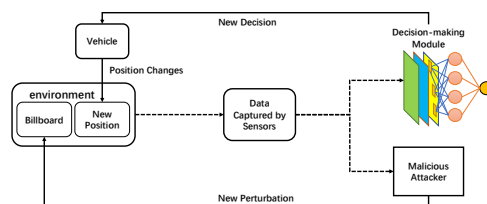


图 2: 闭环劫持

杂的 ADS 能够从过去的干扰中恢复过来, 进一步削弱了单一攻击的有效性。对此, 我们提出了两种策略来解决这些问题。

策略 1: 我们生成在“物理”上更鲁棒的对抗样本。基于对像素色彩偏差的观察, 我们发现 RGB 相机拍摄的图像的真实像素色彩值和被拍摄对象的原始色彩值之间的颜色可以通过非线性变换来描述。我们根据从测试平台收集的像素色彩变化数据, 采用轻量级神经网络来拟合这种非线性变换。同时, 由于在被攻击车辆的 ADS 的决策间隔期间, 视角没有显著变化, 因此在我们的测试平台上, 由车辆运动引起的物理干扰因素主要包括模糊和视角的微小变化。在这里, 我们利用期望转换方法 (Expectation Over Transformation, EOT^[7]), 它将各种转换抽象为分布, 并将可能的转换在输入上做并行的预处理, 然后将所有输出的期望作为最终的优化对象。迭代的每一步都以上一步的更新值作为输入。

策略 2: 我们设计了一系列对抗样本, 通过将反馈结合到连续攻击中来实现车辆的连续欺骗。图 2 描绘了图 1 的“基于对抗攻击的闭环测试”引擎, 其中反馈是指恶意攻击模块观察到的受害车辆的位置和姿势变化。它利用反馈从被攻击车辆的角度确定攻击需求, 进行量身定制的攻击。借助对抗性攻击算法, 恶意攻击模块计算出为了干扰被攻击车辆, 下一帧需要显示的内容。

为此, 恶意攻击模块考虑了车辆的动态模型, 即每一小段时间的实际转向角度和行驶距离。我们使用了纯追踪算法 (Pure Pursuit^[8]) 来控制被攻击车辆追踪移动目标点, 使得车辆遵循规定的轨迹。目标曲线由多个点组成, 纯追踪算法计算出将车辆从当前位置移动到目标位置的转向角。最终, 将估计的转向角作为期望的攻击结果。

3 实验评估

在本节中, 我们评估 ADEPT 的两个场景引擎。

基于事故报告的测试场景生成。为了选择合适的实验对象 (ADS) 进行评估, 我们选择了一个端到端的 ADS, 它是 2020 年 CARLA 挑战赛 Camera-Only 组的第一名获得者^[4], 同时, 我们选用 NHTSA Crash Viewer 数据集^[9]中的事故报告作为文本输入。

我们使用 20 份交通事故报告生成 365 个测试用例。ADEPT 首先将每个事故报告转换为多份中间 Scenic 代码。然后, ADEPT 从每份 Scenic 代码中生成多个测试场景, 并在报告中未提及的描述方面略

*<https://github.com/ApolloAuto/apollo>

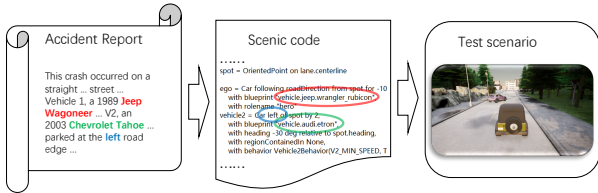


图 3: 基于事故报告的测试场景生成样例

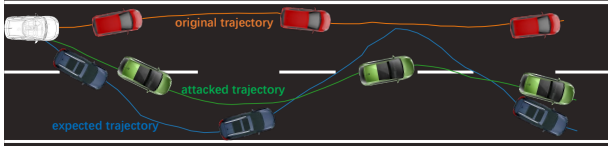


图 4: 轨迹劫持场景

有变化。测试用例的生成依赖于 Scenic 中预定义的环境模板。在评估中,我们选择了两个典型的事故地点,分别是一个在路口的事故以及在非路口区域的事故。如表 1 所示,一共生成了 73 份有效的 Scenic 代码,根据车辆事故风险等级标准,生成的 365 个测试用例在模拟环境中引发 133 起事故,其中 54 起涉及严重危险。

图 3 显示了一个例子。使用定制的 GPT-3 从自然语言事故描述中提取有价值的信息(例如,汽车数量、车辆品牌、位置、速度和方向)以生成 Scenic 代码。然后,基于 Scenic 代码生成 ADS 测试用例。由于 Carla 和 Scenic 不支持雪佛兰 Tahoe 车型,所以此时工具会随机选择一辆相似车型,例如奥迪 Etron 作为车辆 2。

基于对抗攻击的闭环测试。为了评估闭环对抗攻击的测试,我们选择 NVIDIA 基于 CNN 的转向模型^[3]作为被攻击车辆的核心决策模块,该模型接收前视角的 RGB 图像作为输入并输出-1.0 至 1.0 之间的浮点数表示转向角。NVIDIA 模型的目标是沿着车道直线行驶,而不会发生越过黄线或进入人行道等违规行为。在实验过程中,被攻击车辆被设置为在直路上保持恒定速度。为了展示 ADEPT 的能力,我们评估了两种 ADS 测试场景,即轨迹跟踪和行人碰撞。

第一个场景中,控制管理模块在路边放置一个广告牌。然后通过提供笛卡尔坐标序列的方式来预先确定一个预设的期望轨迹。如图 4 所示,红线代表原始路线,蓝线代表预设路线,而绿线代表受害汽车在实验过程中所遵循的真实轨迹。我们真实的劫持成功并误导了被攻击车辆行驶了危险的 S 路线,我们注意到图中的绿线比预期的要滞后一点,这是因为对抗攻击只在当前帧上运行,而实际的被攻击的帧在稍后的时间才会发生。

第二个场景中,如图 5 所示,我们安装了一个广告牌,行人在路边的受害车辆前面行走。我们不断地干扰汽车,使它与行人发生碰撞(测试人员在实验过程中对行人进行了操纵)。如随附的视频所示,被攻击车辆被有效地劫持了,并且被动态误导,使得它跟随并最终撞到了移动的行人。



(a) 通过广告牌操纵 ADS 车 (b) 碰撞发生在闭环操作之后

图 5: 与行人发生碰撞的场景

4 相关工作

Abrecht 等人^[10]将所有 ADS 测试分为四个级别,级别 1,直接测试单独的 DNN 模型;级别 2,将硬件传感器纳入到协同测试的范围;级别 3,在级别 2 的基础上继续将其他外部环境因素纳入考量;级别 4,在闭环环境中测试 ADS。我们的测试平台实现了级别 4 测试能力。

AC3R^[11]是第一种通过从事事故报告中自动重建车祸来测试 ADS 的方法。但是 AC3R 从一个空的世界构建场景,忽略了背景物体(比如建筑、绿化带等)的构建,测试起来可能不太现实。此外,它不能处理长报告。

除了从事事故报告文本生成测试场景外,也有其他相关工作,Nguyenet 等人^[12]选择从事事故草图生成测试场景,Xinxin 等人^[13]选择从事事故视频生成测试场景。除了生成 3D 测试场景之外,Holland 等人^[14]和 Goss 等人^[15]还试图从事事故报告或数据生成 2D 平面场景,用于测试 ADS 的预测和规划模块。

由于自动驾驶的复杂性和不可预测性,定义和发现关键场景成为关键挑战^[16]。Gladisch 等人^[17]将问题形式化为基于搜索的测试问题。为了优化搜索过程,Klischat 等人^[18]使用进化算法;Althoff 等人^[19]通过分析场景的可达性来减少搜索空间;SAMOTA^[20]结合代理辅助优化和多目标搜索,有效地生成测试场景;SALVO^[21]通过从现有地图生成测试场景来减少搜索空间。在 SBST Tool Competition 2021^[22]中,参赛者就如何搜索可能导致车祸的道路布局展开竞争。Gambi 等人^[11]尝试根据事故数据库生成场景。Scenic^[5]和 Paracosm^[23]提供语言和工具来手动定义和构建关键场景。

DNN 的对抗样本已在相关工作中得到广泛研究。典型的对抗样本生成算法包括 CW^[24]、PGD^[25]等。将对抗样本用于测试的简单应用可以归类为 L1 测试。RP2^[26]和 Shapeshifter^[27]通过修改对象的表面纹理而不是图像像素来实现对抗攻击。Deepbillboard^[28]和 PhysGAN^[29]在现实世界中用广告牌拍摄视频,可以归类为 L3 测试。Patel 等人^[30]在半闭环环境中实现 ADS 的伪 L4 测试,其中对抗样本的物理效果是通过处理数字图像来制作的。

事故地点类型	报告	Scenic 代码	场景	安全	危险	
					高风险	低风险
非路口	11	46	230	153/66.52%	48/20.87%	29/12.61%
V1&V2 同向	8	40	200	130/65.00%	43/21.50%	27/13.50%
V1&V2 反向	3	6	30	23/76.67%	5/16.67%	2/6.67%
路口	9	27	135	79/58.52%	31/22.96%	25/18.52%
总计	20	73	365	232/63.56%	79/21.64%	54/14.79%

表 1: 基于事故报告生成的测试场景中 ADS 的测试结果

5 结论

我们实现了一个新的 ADS 测试平台 ADEPT, 它基于虚拟环境进行测试, 并提供众多测试功能。ADEPT 的显著特点包括两点, 即从事故报告中推导出现实场景, 以及基于 DNN 的对抗样本制作具有物理意义的闭环测试场景。

参考文献

- [1] DOSOVITSKIY A, ROS G, CODEVILLA F, et al. Carla: An open urban driving simulator[C]//Conference on robot learning. PMLR, 2017: 1-16.
- [2] SHAH S, DEY D, LOVETT C, et al. Airsim: High-fidelity visual and physical simulation for autonomous vehicles[C]//Field and service robotics. Springer, 2018: 621-635.
- [3] BOJARSKI M, DEL TESTA D, DWORAKOWSKI D, et al. End to end learning for self-driving cars[A]. 2016.
- [4] TOROMANOFF M, WIRBEL E, MOUTARDE F. End-to-end model-free reinforcement learning for urban driving using implicit affordances[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 7153-7162.
- [5] FREMONT D J, KIM E, DREOSSI T, et al. Scenic: A language for scenario specification and data generation[A]. 2020.
- [6] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.
- [7] ATHALYE A, ENGSTROM L, ILYAS A, et al. Synthesizing robust adversarial examples [C]//ICML. 2018.
- [8] COULTER R C. Implementation of the pure pursuit path tracking algorithm[C]//1992.
- [9] NHTSA. Nhtsa crash viewer[EB/OL]. 2016. <http://crashviewer.nhtsa.dot.gov/>, Last accessed on 2022-5-24.
- [10] ABRECHT S, GAUERHOF L, GLADISCH C, et al. Testing deep learning-based visual perception for automated driving[J]. ACM Transactions on Cyber-Physical Systems (TCPS), 2021, 5(4): 1-28.
- [11] GAMBI A, HUYNH T, FRASER G. Generating effective test cases for self-driving cars from police reports[C]//Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2019: 257-267.
- [12] NGUYEN V, GAMBI A, AHMED J, et al. Crisce: Towards generating test cases from accident sketches[C]//2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). IEEE, 2022: 339-340.
- [13] XINXIN X, FEI L, XIANGBIN W. Csg: Critical scenario generation from real traffic accidents[C]//2020 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2020: 1330-1336.
- [14] HOLLAND J C, SARGOLZAEI A. Verification of autonomous vehicles: Scenario generation based on real world accidents[C]//2020 SoutheastCon: volume 2. IEEE, 2020: 1-7.
- [15] GOSS Q, ALRASHIDI Y, AKBAS M İ. Generation of modular and measurable validation scenarios for autonomous vehicles using accident data[C]//2021 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2021: 251-257.

- [16] ZHANG X, TAO J, TAN K, et al. Finding critical scenarios for automated driving systems: A systematic mapping study[J]. IEEE Transactions on Software Engineering, 2022.
- [17] GLADISCH C, HEINZ T, HEINZEMANN C, et al. Search-based testing in automated driving control applications[C]//2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2019: 26-37.
- [18] KLISCHAT M, ALTHOFF M. Generating critical test scenarios for automated vehicles with evolutionary algorithms[C]//2019 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2019: 2352-2358.
- [19] ALTHOFF M, LUTZ S. Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles[C]//2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 1326-1333.
- [20] UL HAQ F, SHIN D, BRIAND L. Efficient online testing for dnn-enabled systems using surrogate-assisted and many-objective optimization[C]//Proceedings of the 44th International Conference on Software Engineering (ICSE' 22). ACM, 2022.
- [21] NGUYEN V, HUBER S, GAMBI A. Salvo: Automated generation of diversified tests for self-driving cars from existing maps[C]//2021 IEEE International Conference on Artificial Intelligence Testing (AITest). IEEE, 2021: 128-135.
- [22] PANICHELLA S, GAMBI A, ZAMPETTI F, et al. Sbst tool competition 2021[C]//2021 IEEE/ACM 14th International Workshop on Search-Based Software Testing (SBST). IEEE, 2021: 20-27.
- [23] MAJUMDAR R, MATHUR A, PIRRON M, et al. Paracosm: A language and tool for testing autonomous driving systems[A]. 2019.
- [24] CARLINI N, WAGNER D. Towards evaluating the robustness of neural networks[C]//2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017: 39-57.
- [25] MADRY A, MAKELOV A, SCHMIDT L, et al. Towards deep learning models resistant to adversarial attacks[A]. 2017.
- [26] EYKHOLT K, EVTIMOV I, FERNANDES E, et al. Robust physical-world attacks on deep learning visual classification[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 1625-1634.
- [27] CHEN S T, CORNELIUS C, MARTIN J, et al. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector[C]//Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2018: 52-68.
- [28] ZHOU H, LI W, KONG Z, et al. Deepbillboard: Systematic physical-world testing of autonomous driving systems[C]//2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE). IEEE, 2020: 347-358.
- [29] KONG Z, GUO J, LI A, et al. Physgan: Generating physical-world-resilient adversarial examples for autonomous driving[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 14254-14263.
- [30] PATEL N, KRISHNAMURTHY P, GARG S, et al. Adaptive adversarial videos on road-side billboards: Dynamically modifying trajectories of autonomous vehicles[C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019: 5916-5921.