软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007486] [CSTR: 32375.14.jos.007486]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向整车系统的自动驾驶安全测试研究综述*

任睿晗[1], 杨 超[1], 杨 凯[1], 张柏迪[1], 张晓东[2], 王利娟[1], 马建峰[1]

[1](西安电子科技大学 网络与信息安全学院, 陕西 西安 710071)

[2](西安电子科技大学 计算机科学与技术学院, 陕西 西安 710071)

通信作者: 杨超, E-mail: chaoyang@xidian.edu.cn

摘 要: 自动驾驶系统能够产生极大的经济效益、安全效益和社会效益, 受到工业界和学术界的格外关注, 逐渐被深入研究, 普及应用. 然而, 引入此类复杂生态系统会产生新的安全问题, 威胁到行人的生命安全, 影响到现有的法律体系. 因此, 在自动驾驶系统实装、自动驾驶车辆上路、自动驾驶行业商业化落地前, 必须通过仿真测试、准入审核、试点运营等多种途径验证自动驾驶系统. 当前对模块安全研究的总结已经成熟, 但仍然缺乏对整车安全研究的归纳整理工作. 因此, 系统性地分析面向整车系统的自动驾驶安全测试研究, 全面回顾当前的主流工作. 首先, 概述自动驾驶系统结构和仿真测试的基本流程, 梳理近 6 年整车系统安全测试领域的文献, 并依托于通用的测试框架形成面向整车系统的自动驾驶安全测试框架. 其次, 基于上述框架提炼出当前工作 5 类核心研究问题, 即关键场景生成、测试充分性、对抗样本生成、测试优化和测试预言, 并详细地分析和整理每类问题的关键技术、研究现状、发展脉络, 归纳当前研究常用的评价指标和对比方法. 最后, 总结各个研究方向面临的严峻挑战, 并展望未来的研究机遇, 思考潜在的解决方案.

关键词: 自动驾驶系统; 整车系统安全; 仿真测试

中图法分类号: TP311

中文引用格式: 任睿晗, 杨超, 杨凯, 张柏迪, 张晓东, 王利娟, 马建峰. 面向整车系统的自动驾驶安全测试研究综述. 软件学报. http://www.jos.org.cn/1000-9825/7486.htm

英文引用格式: Ren RH, Yang C, Yang K, Zhang BD, Zhang XD, Wang LJ, Ma JF. Survey on Vehicle System Safety Testing Research for Autonomous Driving. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7486.htm

Survey on Vehicle System Safety Testing Research for Autonomous Driving

REN Rui-Han[1], YANG Chao[1], YANG Kai[1], ZHANG Bai-Di[1], ZHANG Xiao-Dong[2], WANG Li-Juan[1], MA Jian-Feng[1]

[1](School of Cyber Engineering, Xidian University, Xi'an 710071, China)

[2](School of Computer Science and Technology, Xidian University, Xi'an 710071, China)

Abstract: Autonomous driving systems (ADSs) have gained significant attention from both industry and academia due to their substantial economic, safety, and societal benefits, leading to in-depth research and the gradual popularization of applications. However, the introduction of such complex ecosystems can give rise to new safety issues that threaten the lives of pedestrians and impact the existing legal system. Therefore, it is imperative to validate ADSs through various methods such as simulation testing, access reviews, and pilot operations before the implementation and commercialization of ADSs. While the research on module safety has matured, there is still a lack of comprehensive research and organization regarding the safety of complete vehicle systems. Therefore, this study systematically analyzes vehicle system safety testing for ADSs and comprehensively reviews the current mainstream work. First, the architecture of ADSs and the basic procedure of simulation testing are outlined. The literature on vehicle system safety testing over the past six years is

* 基金项目: 国家自然科学基金 (6223000226, 62125205); 陕西省技术创新引导计划 (2024QCY-KXJ-171); 中国电子科技集团第三十六研究所科研项目 (HX01202310065); 中央高校基本科研业务费专项资金 (ZYTS24138)
收稿时间: 2024-04-25; 修改时间: 2024-12-08; 采用时间: 2025-06-11; jos 在线出版时间: 2025-09-10

reviewed. Based on a universal testing framework, an autonomous driving safety testing framework tailored for vehicle systems is developed. Second, five core research issues are identified based on the aforementioned framework, namely critical scenario generation, test adequacy, adversarial sample generation, test optimization, and test oracle. A detailed analysis and organization of the key technologies, research status, and development context for each issue are provided. The commonly used evaluation metrics and comparative methods in current research are also summarized. Finally, the severe challenges faced by various research directions are summarized, and future research opportunities are anticipated, along with potential solutions.

Key words: autonomous driving system (ADS); vehicle system safety; simulation testing

1 引 言

随着人工智能技术的应用, 近年来自动驾驶系统 (autonomous driving system, ADS) 蓬勃发展, 传统车企和互联网公司都投入了大量的人力物力进行研发, 如图 1 所示. 例如, 百度研制了开源自驾系统 Apollo, 并实装在长沙的无人出租车上, 运营车辆达到 100 辆[1]; 华为与多家车企合作, 提供智驾解决方案, 推出问界 M9、极狐阿尔法 S 等多种车型[2]. 自动驾驶系统广泛应用, 一方面能够减少交通事故, 提升道路安全水准; 另一方面能够缓解交通压力, 减少碳排放, 贯彻可持续发展理念.

(a) 百度 Apollo Moon (b) 谷歌 Waymo (c) 苏宁行龙一号

图 1 自动驾驶部署实例

然而, 自动驾驶系统是复杂的网络物理系统, 软件的工程规模极其庞大, 其设计和实现可能具备安全隐患, 严重威胁其他车辆、驾驶员和行人的安全. 随着低水平自动驾驶在汽车上广泛部署, 各类安全事故频发. 例如, 2018 年, Uber 运营的自动驾驶 SUV 在美国亚利桑那州撞击了一名横穿马路的行人并致其死亡[3]; 2021 年, 蔚来 ES8 在辅助驾驶功能下未能识别高速路段上的施工车辆, 产生严重的碰撞事故, 导致驾驶员离世[4]. 为了避免安全事故发生, 业界提出了预期功能安全标准 (ISO 21448[5])、功能安全标准 (ISO 26262[6]) 等多项国际标准指导自动驾驶系统的研发、功能验证和准入审核, 规范自动驾驶系统的安全验证流程. 开发人员也通过冗余的安全组件、智能化辅助系统、测试评估框架增强系统的健壮性.

目前, 自动驾驶安全研究受到了研究人员的广泛关注, 一系列高价值、强可用性的方案被提出, 模块安全研究和整车系统安全研究是两个主要方向. 前者旨在离线、独立的检测任务完成状况, 判断单个模块输出的正确性; 后者旨在结合仿真器和测试技术验证自动驾驶系统整体运行状态. 模块安全研究是自动驾驶安全研究的基础, 已经进行了多年的深入探索, 在感知模块[7,8]、规划模块[9,10]、预测模块[11]等部分都取得显著的成果. 例如, Eykholt 等人[7,8]将对抗扰动从数字域迁移到物理域, 设计了抗噪声的物理对抗样本, 成功欺骗目标检测模型, 是物理世界对抗攻击工作的里程碑. 然而, 伴随着全栈仿真器和配套工具逐渐完善, 近年来整车系统安全领域的研究热度持续攀升. 由于自动驾驶系统构造复杂, 形式化验证等技术难以对整体建模和分析, 不适用于研究整车系统安全问题. 因此, 当前主要的研究方法是安全测试, 通过测试整车系统挖掘潜在的跨层漏洞, 暴露出新的安全问题. 同时, 该方向将车辆动力学模型纳入考量, 更贴近现实、更符合研究预期.

为了深入分析研究现状, 研究人员系统性地总结了自动驾驶安全领域的工作, 为后续研究提供理论支持和方法指导. 模块安全领域研究历史悠久且成果丰硕, 存在完善的综述工作. 例如, Garcia 等人[12]和 Tang 等人[13]分别对开源自动驾驶系统和高级辅助驾驶系统各模块的漏洞展开分析, 全面调查了开源社区中的漏洞提交和修复报告, 总结了漏洞的成因、症状和影响范围, 辅助后续漏洞定位和修复工作. 朱向雷等人[14]以自动驾驶系统结构为核心, 依次总结了针对感知、决策和辅助系统等目标的研究现状, 指导了自动驾驶安全测试工作的展开. Tang 等人[15]不仅全面梳理了自动驾驶系统各模块的研究现状和测试方法, 还分析了对辅助驾驶系统和简单驾驶模型的安全研

究, 并且总结了仿真测试与混合现实测试工作的差距, 提供了对自动驾驶系统测试技术体系丰富、全面的认知视角. 相比之下, 整车系统安全领域的综述工作较少. 戴嘉润等人[16]首次调研了应用于自动驾驶仿真测试领域的模糊测试技术, 揭露了种子场景生成、事故分类与事故归因工作的不足, 并提出对应的优化方案, 缩小了仿真模糊测试框架各环节的技术差距.

模块安全领域的综述文献全面且成熟, 然而, 整车系统安全领域的综述工作仍处于起步阶段, 尚未有研究人员使用整体视角对现有工作进行清晰明确、全面系统的总结. 文献 [14] 虽然部分内容涉及整车测试, 但涵盖的文献属于安全研究的早期阶段, 缺乏对最新进展的分析. 文献 [16] 只总结了在仿真测试中应用模糊测试的工作, 并没有纳入其他漏洞挖掘和安全验证方法. 此外, 他们重点关注种子场景生成和事故分类归因, 没有深入分析事故挖掘方法. 综上, 当前整车系统安全领域的综述涵盖的文献数量少、范围窄, 且不包含最新研究进展, 无法满足研究人员在该领域深入学习和研究的迫切需求. 同时, 尚未有综述从完整测试流程的视角展开分析, 无法使研究人员对整车系统测试领域形成全方位、多层次、结构化的认知. 因此, 有必要为整车系统安全领域的测试技术撰写综述文献, 弥补现有综述的不足.

本文以集成了感知决策控制功能的自动驾驶系统整体为研究对象. 通过在软件工程、安全、汽车等领域的高水平会议和期刊中搜集文献, 确定了 2018–2023 年期间与面向整车系统的自动驾驶安全测试研究相关的代表性工作, 共计 39 篇. 从图 2 可知, 该领域的文献数量逐年上升, 并且在近两年热度极高, 论文数量占收集文献总数的 71.8%. 围绕这些文献, 本文总结了安全测试工作的技术路线, 并以研究问题为导向, 将这些工作划分为关键场景生成、测试充分性、对抗样本生成、测试优化和测试预言 (test oracle) 这 5 类. 通过对各个研究问题进行深入讨论和细致整理, 本文对比了不同主题下研究工作的优缺点和研究趋势, 展望了未来可能面临的挑战与机遇.

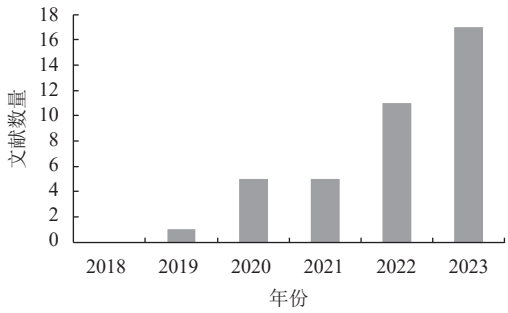

图 2　2018–2023 年面向整车系统的自动驾驶安全测试研究代表性研究统计

本文的贡献归纳如下.

(1) 汇总并分析了近 6 年整车系统安全测试领域的文献, 将其测试方案对应到通用的测试框架中, 形成面向整车系统的自动驾驶安全测试框架, 辅助研究人员建立对该领域测试工作全流程、多维度、体系化的认知.

(2) 基于上述框架提炼出 5 类核心研究问题, 系统地梳理现有工作的技术体系和评估方法, 深入比较其创新性和局限性, 为深化面向整车系统的自动驾驶安全测试研究提供强有力的支撑.

(3) 基于对研究现状的剖析, 揭示出潜在的现实挑战和研究机遇, 为面向整车系统的自动驾驶安全测试研究提供独特的思路和见解, 对于自动驾驶的研发测试工作具有重要现实意义.

本文第 2 节概述自动驾驶系统结构、仿真测试架构等相关背景知识. 第 3 节总结面向整车系统的安全测试框架. 第 4 节基于上述框架总结出 5 类研究问题, 并详细分析整车系统安全领域的研究文献. 第 5 节总结当前研究常用的评价指标和对比方法. 第 6 节基于研究现状, 剖析和探讨现实挑战与研究机遇. 最后, 第 7 节总结本文工作.

2 背　景

在第 2.1 节介绍了自动驾驶系统的概念和基本结构, 分为模块化和端到端两个研究分支. 在第 2.2 节介绍了自

动驾驶系统的仿真测试架构, 解释了仿真器和测试场景.

2.1 自动驾驶系统结构

2.1.1 概 述

美国汽车工程师学会提出了 SAE J3016[17]标准, 将自动驾驶技术分为 6 个级别, 从 level 0 无自动驾驶到 level 5 完全自动驾驶. 其中, level 1–level 2 倾向于辅助驾驶员决策和判断, 属于高级驾驶辅助系统, 已经广泛部署在汽车上. 典型代表有车道偏离预警系统、自适应巡航系统、前方碰撞预警系统等. level 3–level 4 倾向于在一定范围内的驾驶自动化, 偏向于常规认知中的自动驾驶系统, 同时也是本文的主要研究对象. 目前该方向仍在深入研究, 还未大规模地落地部署. 同时, 早期对自动驾驶技术的研究聚焦于高级驾驶辅助系统[18,19], 随着开源自动驾驶系统逐渐成熟, 以及仿真器等下游工具链的完善, 现在的工作更关注高水平的自动驾驶系统. 目前存在两个研究分支, 分别是模块化系统和端到端系统.

2.1.2 模块化系统

模块化系统将自动驾驶任务分解为多个子任务, 交由不同的模块分别处理, 每个子任务又可以细分为多个模型, 如图 3 所示. 例如, 感知模块包括了目标检测模型、交通信号灯识别模型等. 每个模型都需要独立开发, 训练参数, 迭代优化, 最终串接处理数据, 实现自动驾驶任务. 典型的开源模块化自动驾驶系统有 Apollo[20]和 Autoware[21]. 其架构一般包括感知、规划和控制这 3 个部分, 如图 3 所示.

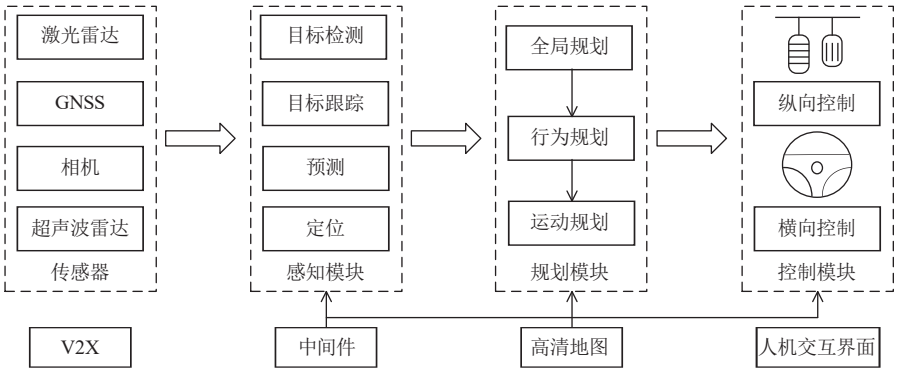

图 3 模块化自动驾驶系统架构

(1) 感知模块接收相机、激光雷达、惯性测量单元等传感器数据, 通过深度学习技术融合和处理数据, 实现交通灯识别、目标跟踪、轨迹预测、定位等任务, 感知和理解外部环境.

(2) 规划模块根据感知模块的输出规划车辆路线, 通过 3 个步骤完成任务: 全局规划, 负责在地图上根据起点和终点规划出一条可行的路线; 行为规划, 负责做出符合交通法规的高级驾驶决策, 如巡航、跟车等; 运动规划, 通过考虑安全、效率和舒适度等因素, 生成最优的局部规划, 如确定速度和转向角.

(3) 控制模块使用控制算法向油门、方向盘等执行器传输控制信号, 驱使车辆沿着规划模块输出的轨迹运动, 实现横向控制和纵向控制. 常用的控制算法有比例积分微分 (PID)[22]算法和模型预测控制 (MPC)[23]算法.

除上述主要部分外, 还有一些辅助组件, 如人机交互界面、高精地图、V2X 等. 模块化设计的自动驾驶系统的优点是各模块基于规则处理任务, 通过规则约束实现了最小安全保障. 但是, 此类设计方案属于流水线架构, 上层模型的错误输出会传播到后续模型, 微小的偏差不断累计, 最终可能造成级联故障, 影响正常任务的执行.

2.1.3 端到端系统

端到端系统将传感器感知的数据、导航命令输入一个预训练的深度神经网络, 直接输出控制信号或规划轨迹. 端到端系统仅通过一个模型即可实现模块化系统的大部分功能, 如图 4 所示. 该模型可能包括多个子模型, 但所有模型联合训练, 使用一致的优化目标[24]. 最早的端到端自动驾驶车辆 AVLINN 出现于 1989 年, Pomerleau[25]设计了一个 3 层全连接神经网络实现简单的自动驾驶任务. 2023 年的 CVPR 最佳论文颁发给 UniAD[26], 一个全栈

的端到端方案. 目前, 端到端系统的主要设计方法是模仿学习或强化学习技术, 典型代表是英伟达的 DAVE-2[27]、comma.ai 的 OpenPilot[28].

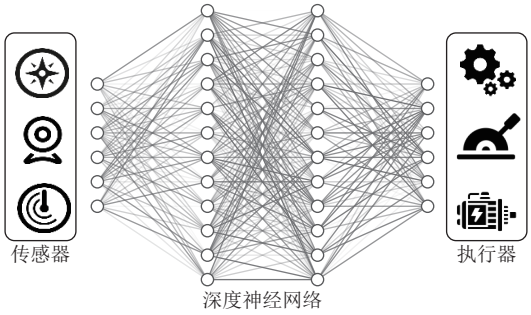

图 4　端到端自动驾驶系统架构

相对来说, 模块化系统在实际生产中应用更多, 端到端系统的研究尚处于起步阶段. 主要原因有两点: 首先, 端到端系统可解释性差, 无法阐释智能决策的原因, 在出现错误时难以定位缺陷和评估修复. 并且由于缺乏规则限制, 端到端系统无法保证 100% 的安全性, 没有安全下界. 其次, 训练端到端系统的深度神经网络模型需要大量准确标注的完整数据, nuScenes[29]等早期的开源视觉数据集效果较差, 目前尚无高质量、多模态的训练数据集. 而针对模块化系统数据的收集和标注技术较为成熟, 也存在高质量、高影响力的开源数据集. 因此, 端到端自动驾驶系统并未广泛应用, 仍然有待深入研究.

2.2　自动驾驶仿真测试架构

目前常用的测试方法分为 3 种: 道路测试[30]、封闭场地测试[31]和仿真测试. 前两种方法难以遍历复杂的真实环境条件, 应用较少. 因此, 仿真测试成为安全研究的主流. 本文主要关注仿真测试下的整车系统安全研究.

自动驾驶仿真测试使用计算机软件构建真实的物理环境, 模拟道路测试面对的路况信息, 并接入自动驾驶系统控制车辆模型, 在参数化的条件组合下运行, 以挖掘自动驾驶系统的缺陷. 仿真测试用驾驶场景代替了行驶里程, 可以灵活配置各种场景参数, 大量生成现实中的稀缺场景和危险场景. 因此, 仿真测试的成本更低, 安全性和效率更高, 能够为自动驾驶系统的部署实装和量产应用提供保障. 图 5 展示了仿真测试的架构体系, 预定义的测试场景在仿真器中渲染处理, 仿真器通过通信接口与自动驾驶系统连通, 转发控制流和数据流信息, 进而使自动驾驶系统在测试场景中运行.

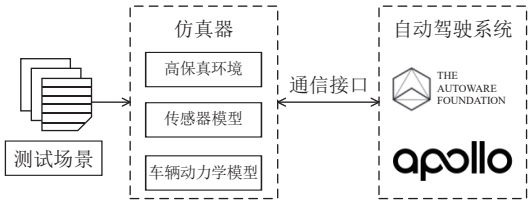

图 5　自动驾驶仿真测试架构

2.2.1　仿真器

理想的仿真器包括高保真的仿真环境、精确的传感器模型和先进的车辆动力学模型[32], 如图 5 所示.

环境仿真技术将现实对象的存在特性和运动过程精准还原到测试场景中, 包括光照、天气等自然环境和交通标志、道路基础设施等交通环境两类. 高保真的环境能够为自动驾驶系统的传感器提供更丰富、更具可用性的输入数据, 使仿真测试更接近现实中的道路测试. 但是, 由于成本和技术的限制, 仿真与现实始终存在差距. 部分研究人员试图缩小二者的距离, 开发更真实、更灵活、更便捷的仿真工具. 例如, 用于在各种照明条件下测试感知模块的光照仿真平台 LightSim[33], 利用多个大语言模型协同工作的场景编辑平台 ChatSim[34].

传感器仿真技术将真实传感器的参数、特性和工作过程数字化, 创建对应的虚拟模型, 其精度决定了感知的

精度. 该技术面临着以下挑战: 首先, 传感器种类繁多, 参数和工作原理各不相同. 因此, 不存在适用于所有传感器的通用模型, 必须为每类设备单独设计和定制. 其次, 现实中传感器的工作过程会受到光学、声学等多种随机噪声的干扰, 因此在建模时需要加入人工设计的误差, 如车载相机的畸变系数、车载激光雷达的扫描角误差等. 然而, 此类误差难以量化和标准化, 成为传感器仿真技术的“痛点”之一. 最后, 仿真需要获取传感器详细的底层数据, 然而出于保密要求和商业考虑, 厂商通常不会开放此类数据, 这使得研究人员遭遇额外的阻碍.

车辆动力学仿真技术将物理车辆的动力学特性抽象为数学模型, 实现出一个能够反应实际工作状态并适用于闭环测试的虚拟车辆. 动力学模型决定了自动驾驶系统控制算法表现的准确性, 进而影响测试结果的可用性. 但汽车的零件数量极多, 参数和特性极其复杂, 研究人员无法拟合所有硬件, 只能精简零件数量, 模拟核心和关键部件, 平衡建模精度和仿真成本.

根据适用条件和设计目标, 可以将现有仿真器分为 3 类: 全栈仿真器、车辆仿真器和交通流仿真器. 全栈仿真器是随自动驾驶技术发展而逐渐兴起的新类型, 典型代表是 LGSVL[35]和 CARLA[36], 二者都是学术界常用的开源仿真器, 支持主流的开源自动驾驶系统. 车辆仿真器是传统车企的主要工具, 典型代表是 CarMaker[37]和 CarSim[38], 能够构造精确的车辆模型, 优化车辆机械结构的参数, 促进新车型的开发. 交通流仿真器的典型代表是 SUMO[39]和 PTV Vissim[40], 优势在于能够仿真大规模的交通流, 模拟多个智能体间的交互行为. 鉴于不同仿真器有各自的侧重点, 联合仿真成为一大发展趋势. 大多数仿真器都提供了与其他仿真器连接的接口, 研究人员可以根据测试需求和测试条件挑选和搭配, 优化仿真效果.

通信接口负责连接自动驾驶系统和仿真器, 实现二者之间的通信链路. 一方面, 通信接口将自动驾驶系统做出的控制命令发送给仿真器, 使车辆根据反馈信息调整决策; 另一方面, 通信接口将仿真器中的地图、车辆状态、轨迹点等数据传输给自动驾驶系统, 迭代更新输入数据, 为其行为决策提供依据.

2.2.2 测试场景

仿真测试常用的测试用例是场景, 表征一段时间内的驾驶环境, 包括静态环境、动态对象及其行为[41]. 静态环境由天气、道路结构、障碍物、交通信号和标志等元素组成. 动态对象包括自动驾驶控制的车辆、其他车辆和行人. 一般将前者称为自车, 将后者称为非玩家角色 (non-player character, NPC) 或背景车辆. 动态对象与环境、动态对象之间频繁交互, 产生跟车、变道等驾驶行为. 上述 3 类对象均可由大量参数表示, 其集合构成了场景的配置参数空间, 关键场景即容易导致交通事故、晕动症等问题的配置参数组合.

研究人员将场景分为 3 个层次, 功能场景、逻辑场景和具体场景[42]. 功能场景也被称为抽象场景, 用自然语言描述场景中存在的实体及其关系. 逻辑场景从功能场景中提取配置参数空间, 并约束每个参数的取值范围. 具体场景被定义为通过搜索或采样算法, 计算出逻辑场景中各参数的具体值, 描述成一组可实现的测试场景. 例如, 功能场景是, 自车换道并超过 NPC1, 随后跟随在 NPC2 后行驶; 逻辑场景是, 定义自车和所有 NPC 的位置、速度、车道等参数的取值范围; 具体场景是, 确定自车的速度是 20 km/h, 从 1 车道换到 2 车道, 换道后的速度是 30 km/h, 以及其余 NPC 的具体参数.

为了将测试场景翻译为机器可执行的脚本, 研究人员开发了场景描述语言. 该类语言能够满足实验的定制化需求, 并与自动驾驶系统和仿真器解耦合, 可移植性强. 例如, AVUnit[43]集成了两种特定于自动驾驶领域的描述性语言, SCENEST 建模场景中的 NPC 和天气等元素, AVSpec 使用信号时序逻辑公式描述正确驾驶规范. ScenoRITA[44]定义了障碍物的位置、形状、种类和驾驶行为, 将其实现为完全可变的编码表示, 并借助高精地图构造的有向图自动化生成 NPC 轨迹. 鉴于编写场景脚本的工作复杂繁琐, Deng 等人[45]利用大语言模型 (large language model, LLM) 代替人类专家, 使用 GPT-4 提取并解析交通规则中蕴含的信息, 随后基于仿真器 API 搜索场景参数, 为每条交通规则生成对应的测试场景.

3 面向整车系统的自动驾驶安全测试框架

本文基于通用的测试框架, 提炼出所有测试方案的共性和个性特征, 总结出面向整车系统的自动驾驶安全测试框架. 通过框架提取出核心研究问题, 以此为基础对整车系统安全领域的研究文献进行全面分析. 该框架将通用

的测试框架具体化, 融入核心研究问题, 能够简单直接地对比现有文献的研究重点, 使研究人员对整车系统安全测试的工作流程和技术体系形成全视角、多维度的认知. 同时, 该框架能够启发研究人员将传统软件安全的测试方法和研究思路迁移到整车系统安全测试领域, 指导开展具体的测试工作. 框架主要包括测试用例生成、测试用例执行、测试结果验证这 3 部分, 如图 6 所示.

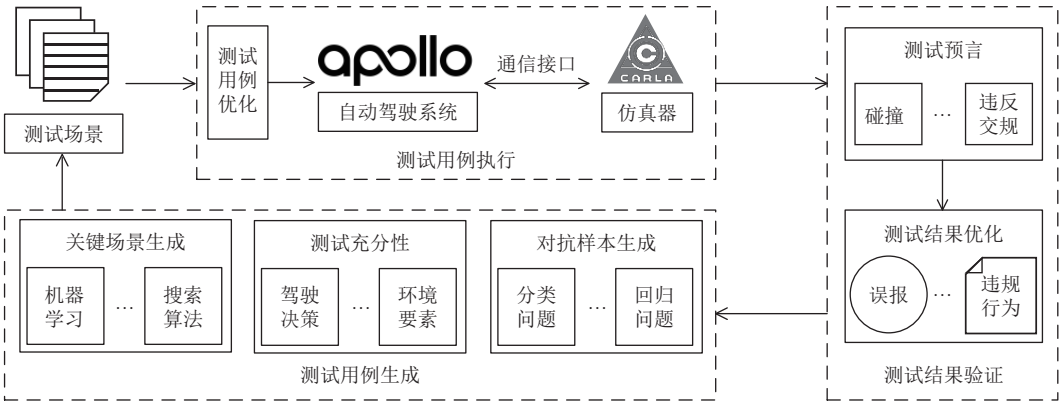

图 6 面向整车系统的自动驾驶安全测试框架

3.1 基本工作过程

面向整车系统的自动驾驶安全测试框架的基本工作过程是: (1) 将测试场景和被测自动驾驶系统导入仿真器, 使自动驾驶系统控制场景中的代理运行. (2) 收集执行过程中的状态数据, 根据测试预言检测是否出现非预期的危险行为, 并筛选误报和真正的危险行为. (3) 计算覆盖率度量或优化算法的目标函数, 指导测试用例生成过程. 迭代执行上述 3 个步骤, 在海量的场景中找出引发问题的场景, 测试自动驾驶系统的安全性. 值得注意的是, 并非每个工作都涉及上述所有环节, 一些研究可能不考虑覆盖率度量或测试结果优化.

3.2 测试用例生成

安全测试框架中最重要的环节是测试用例生成, 即测试场景生成. 研究人员设计了多种场景生成方法. 一方面, 可以通过真实驾驶数据集进行已知场景的重建[46], 进而泛化出更多场景; 另一方面, 可以通过各种场景空间搜索技术生成未知场景, 为解决危险场景长尾分布问题提供方案.

由于场景空间中参数的数量极多, 且取值是连续的, 所以难以穷举所有测试场景. 针对此问题, 本文总结了 3 种测试用例生成思路, 即关键场景生成、测试充分性和对抗样本生成. (1) 关键场景生成, 通过设计出合适的引导度量和优化算法, 逐步提高测试场景的关键度. (2) 测试充分性, 通过聚类等算法将所有场景抽象为多种类型, 用少量具有代表性的测试场景近似整个场景空间. (3) 对抗样本生成, 研究人员在图像或点云中添加对抗扰动, 攻击自动驾驶系统模型应用的人工智能算法, 影响系统的决策和控制.

3.3 测试用例执行

测试用例优化是测试用例执行环节的可选步骤, 负责在执行前对测试用例集进行筛选和精简, 以过滤同质化的测试用例, 保留更高质量、更关键的测试用例. 具体的执行过程已在第 2.2 节中详细说明, 本节不再赘述.

3.4 测试结果验证

测试结果验证环节包括测试预言和测试结果优化两部分. 测试预言用于判断测试结果是否符合预期, 区分被测系统的正确和错误行为[47]. 一方面, 可以采取形式化规约等方法, 将现实世界的规则形式化为任务约束, 根据状态数据判断测试结果, 校验系统设计需求和安全性是否满足. 另一方面, 一些问题难以构造有效的测试预言, 需要使用蜕变测试 (metamorphic testing)、差分测试 (differential testing) 等技术验证结果. 在检测出引发违规行为的测试场景后, 部分研究人员会优化测试结果, 过滤假阳性的测试结果, 只保留自动驾驶系统承担责任的违规行为. 最

终, 执行结果和判断信息会反馈给测试用例生成环节, 指导生成关键度更高、覆盖面更全的测试场景.

4 面向整车系统的自动驾驶安全测试研究现状

目前的综述工作通常以测试方法作为分类依据, 将采用同一类技术方案的工作统一阐述. 然而, 随着安全研究的逐渐深入, 针对同一问题已经提出了多样化的解决方案, 使得当前的分类方法难以适应技术发展的需求, 甚至会割裂研究问题之间的相关性, 打断研究进展的连贯性, 使研究人员无法直观认知到某项研究问题的历史脉络和发展趋势. 因此, 本文基于第 3 节总结的整车系统安全测试框架, 从子环节中提炼出核心研究问题, 并以此为基础将研究工作划分为以下 5 类: 关键场景生成、测试充分性、对抗样本生成、测试优化和测试预言. 这种分类方法将研究工作从技术层面下沉细化到问题层面, 将关注的焦点从具体技术手段转移到核心问题本身, 从而串联起一个连贯而完整的测试流程. 同时, 该分类方法细致地梳理了不同测试阶段研究问题的需求、特征和对应的技术方案, 从而清晰的展现出自动驾驶安全测试中整车系统领域的发展轨迹, 为探寻和发掘潜在的安全问题开辟了新思路. 具体分类结构如图 7 所示.

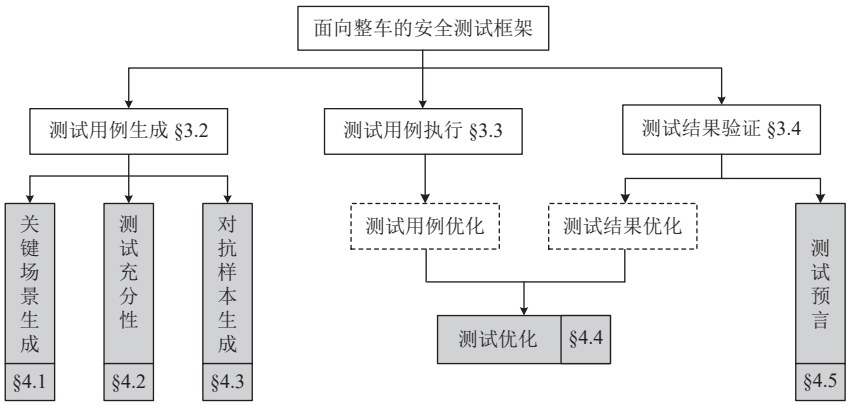

图 7 面向整车系统的自动驾驶安全测试研究文献分类

各分类的主要目标如下: 关键场景生成分类下的文献关注如何快速生成危险的场景参数集合, 以诱发自动驾驶错误行为; 测试充分性分类下的文献关注如何用少量测试场景近似全部输入空间, 充分验证所有可能的场景参数配置, 同时避免重复执行产生相似结果的输入; 对抗样本生成分类下的文献关注如何在测试环境中添加扰动, 生成对抗样本, 破坏模型输出, 进而影响整车系统的安全性; 测试优化分类下的文献关注如何提高测试结果的准确率和真实性, 以及如何提高测试场景的验证效率; 测试预言分类下的文献关注如何深入挑战自动驾驶系统在现实中正常运行的能力.

4.1 关键场景生成

仿真测试中场景数量是无限的, 且大多数场景都无法威胁到自动驾驶系统, 只有极少数关键场景才是最重要的. 因此, 需要有方向、有指导的缩小场景空间, 定位风险最大的测试场景. 基于这一共识, 研究人员提出不同的场景生成策略, 在场景空间中搜索关键场景. 同时, 需要将现实中对自动驾驶系统有挑战性的因素纳入到测试场景中. 测试场景的参数数量越多, 触发漏洞的可能性也越高. 测试场景中常见的对象包括 NPC、天气、静态障碍物等, 近来也有研究人员建模了道路结构和水坑. 图 8 展示了该类别下工作的场景生成方法和场景参数.

生成关键场景是披露自动驾驶系统缺陷的核心, 也是测试用例生成环节的重要组成部分. 基本流程是: (1) 构建逻辑场景, 明确参数及其取值范围. (2) 通过搜索或优化算法确定场景的具体参数值, 在仿真器中执行. (3) 根据执行结果计算目标函数, 指导生成算法的优化方向. (4) 重复上述步骤, 最终得到高质量、有挑战性的测试场景. 本节根据生成方法将文献分为两类, 基于搜索的测试方法[48–56]和机器学习方法[57–61]. 表 1 对比了该分类下文献的场景参数和场景生成策略.

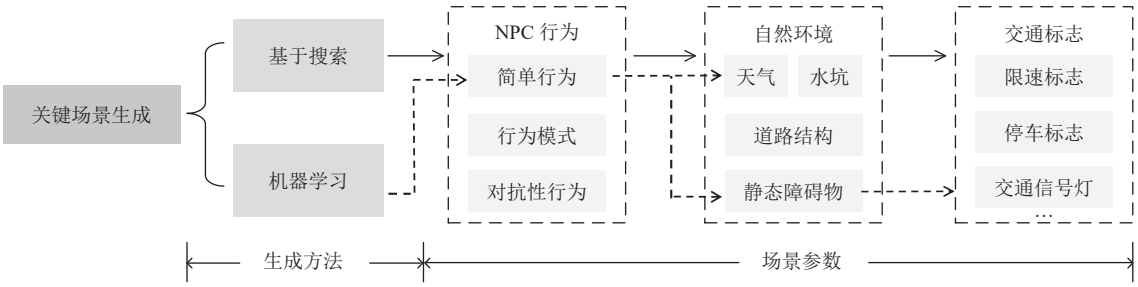

图 8 关键场景生成分类

表 1 关键场景生成分类下的工作对比

文献	分类	场景参数	场景生成策略	ADS
[48]	基于搜索测试方法	NPC速度和变道策略	最小化自车和NPC的距离	Apollo
[49]		NPC行为	最小化自车和NPC的距离	Apollo
[50]		4种NPC行为序列	以场景的风险性、对抗性和多样性为目标函数进行多目标优化	Apollo
[51]		抽象行为模式	添加指定行为模式的NPC以增加场景的关键度	Apollo
[52]		NPC对抗性行为	根据两个时刻间攻击车和自车的相对位置变化确定NPC行为的采样范围	Autoware
[53]		NPC行为、天气和水坑	用驾驶质量分数评估场景并引导参数优化方向	Autoware
[54]		NPC行为和道路结构	以场景覆盖率、驾驶难度和关键度为目标函数选择帕累托最优解	合作伙伴提供
[55]		NPC行为和静态障碍物	使用执行路径与目标攻击位置的控制流和数据流距离引导模糊测试框架	Apollo、Autoware
[56]		NPC行为、天气、交通标志	计算车辆轨迹与交规约束间的差距引导场景空间搜索	Apollo
[57]	机器学习方法	NPC行为、天气和静态障碍物	使用安全距离与当前距离计算碰撞概率, 构建DQN算法的奖励函数	Apollo
[58]		NPC行为、天气和静态障碍物	使用碰撞时间指标构建DQN算法的奖励函数	Apollo
[59]		NPC行为、天气和照明条件	为每个安全需求建立目标函数, 表述为多目标搜索问题	Transfuser[62]
[60]		NPC行为和天气	使用神经网络分类器预测场景导致违规的置信度分数	Apollo
[61]		NPC行为、天气和交通标志	用测试场景及其鲁棒值训练GFlowNet并采样	Apollo

4.1.1 基于搜索的测试方法

研究人员将基于搜索的测试方法应用于自动驾驶领域, 并优化搜索过程, 加速关键场景生成. 通常, 场景生成问题被表述为场景参数的高维空间搜索问题, 利用单目标优化或多目标优化算法找到最优解.

驾驶环境中最常见的对象是 NPC, 其参数包括位置、速度、机动行为等. 机动行为使 NPC 围绕自车进行连续、复杂的运动, 测试自动驾驶系统的交互和处理能力. 简单的机动只考虑速度和转向角的变化; 高级机动组合多个原子机动, 形成机动序列. 复杂的机动序列能够在不同场景间迁移测试, 灵活修改部分参数, 自由度更高[63].

研究人员首先分析和测试了 NPC 与自车的交互. 例如, AV-Fuzzer[48]构建了包括少量 NPC 的简单直道的场景, 使用遗传算法变异 NPC 的速度和变道策略, 以自车和 NPC 的距离为目标函数, 使用轮盘赌策略选择高质量的测试场景, 检测到 5 种违规行为. Sun 等人[49]改进了 AV-Fuzzer 使用的遗传算法, 使用高斯变异、多点交叉和锦标赛选择策略, 提高了局部搜索的效率, 检测到更多数量和种类的安全违规行为. 上述研究只针对简单的驾驶行为, MOSAT[50]对其进行组合, 形成 4 种根据位置和概率触发的行为模式, 通过多目标遗传算法 NSGA-II 引导生成关键场景, 不仅覆盖了 AV-Fuzzer 的实验结果, 还多检测到 6 种违规行为.

鉴于真实环境中交通事故可能是多车交互造成的, CRISCO[51]从真实轨迹数据集中抽象出容易导致交通事故的车辆行为模式, 并基于挖掘的行为模式构建初始抽象场景, 求解约束以实例化具体场景, 通过逐步添加指定行为模式的 NPC 增加场景的关键度, 提升碰撞潜力. 除了测试自动驾驶系统对正常驾驶操作的反应能力, 部分研究还

关注对恶意驾驶行为的鲁棒性. ACERO[52]根据两个时刻间攻击车和自车的相对位置变化方向约束采样范围, 生成多条候选命令并逐一执行, 从中选择违规概率最大的命令. 框架迭代上述过程, 生成一系列对抗性控制命令, 使 NPC 沿对抗轨迹运动, 破坏了 6 类正常的驾驶任务.

上述研究强调自车与其他车辆在地图中产生的交互行为, 关注交通环境对自动驾驶系统的影响, 而不重视天气、时间等自然环境因素. 因此, 部分研究人员同时考虑了车辆交互和自然环境. 例如, DriveFuzz[53]额外考虑了水坑和天气的影响, 计算转向不足、急转弯等多个指标, 组合成驾驶质量分数, 指导搜索场景配置空间, 检测到 Autoware 上 17 个安全漏洞. EvoScenario[54]将高速公路上不同类型的路段和 NPC 机动相结合, 实现了车道拓展、缩减、合并、分离这 4 类连接路段, 并将基本路段和连接路段串联组合, 利用遗传算法和多目标搜索策略生成关键的测试场景, 成功在多样化的道路结构下发现 4 类新的安全问题. 除此之外, 一些研究人员使用静态障碍物搭建了自然环境中出现概率较小的场景进行测试. Wan 等人[55]运用逆向思维, 发现通过精心设计物体的摆放位置, 能够迫使车辆永久静止或不执行决策任务, 挑战自动驾驶系统的行为规划组件. 他们将其称为过于保守的语义拒绝服务漏洞, 并提出白盒模糊测试框架 PlanFuzz, 以执行路径与目标攻击位置的控制流和数据流距离为反馈, 检测在遵守安全约束的条件下自动驾驶系统是否具备完成任务的能力.

尽管现有场景描述语言提供了对车辆状态和行为、天气、道路结构等对象的形式化描述, 但并没有考虑到车灯、喇叭等提供辅助功能的对象, 以及限速、停车线等交通标志对象, 阻碍了复杂规则纳入测试框架的过程. 针对上述问题, LawBreaker[56]制定了面向驾驶员的场景描述语言, 使用信号时序逻辑公式将交通规则形式化, 使测试中的交通环境更加真实. 同时, 研究人员使用遗传算法生成测试场景, 并通过比较当前状态违反交规的程度引导场景空间搜索, 最终发现 14 条法规被违反, 首次验证了自动驾驶系统遵守交通法规的能力.

4.1.2 机器学习方法

基于搜索的测试方法生成的测试场景没有充分考虑环境中代理的动态行为, 因此部分研究人员结合机器学习算法生成关键场景.

一些研究人员使用强化学习技术学习场景参数配置. 例如, DeepCollision[57]将场景中天气、静态障碍物、动态障碍物的配置问题表述为马尔可夫决策过程, 使用 DQN 算法学习容易使自车发生碰撞事故的配置参数, 并用安全距离与当前距离计算碰撞概率, 构建奖励函数, 检测到 40 个独特的碰撞事故. RLTester[58]拓展了上述工作中环境配置参数的数量, 并采用碰撞时间 (TTC) 指标构建奖励函数, 检测到 192 个独特的碰撞事故. 上述方案只针对单个测试目标, 即自车与障碍物是否发生碰撞, 在验证多个测试目标时成本较高, 效率较低. 为了弥补上述不足, 研究人员结合强化学习算法和多目标优化思路, 将多种安全需求的违规检测表述为多目标搜索问题. 例如, MORLOT[59]利用 Q 学习算法生成 NPC 行为、天气等环境参数序列, 并为每个安全目标生成独立的 Q 表, 根据每一轮迭代中奖励值最高的目标选择执行动作, 兼顾和平衡了多个测试目标, 发现了不同类型的违规行为.

另一些研究人员训练神经网络生成测试场景. 例如, AutoFuzz[60]框架基于神经网络设计了场景选择和变异策略. 选择策略使用分类器预测场景导致独特交通违规的置信度分数, 并迭代训练神经网络; 变异策略利用投影梯度下降策略, 反向传播神经网络的梯度, 为置信度分数较低的场景添加微小扰动. 受生成式流网络 (generative flow network, GFlowNet)[64]的启发, ABLE[61]改进了文献 [56] 的工作, 利用测试场景及其对交通规则的鲁棒值训练 GFlowNet, 并结合领域知识和主动学习算法更新迭代模型, 采样出高质量且多样化的测试场景, 以测试自动驾驶系统遵守交通规则的能力. 实验结果表明 ABLE 比文献 [56] 平均多检测到 21% 的违规行为.

4.1.3 小 结

由该分类下的文献可知, 自 AV-Fuzzer 为起点, 研究人员开始使用高保真的仿真器测试自动驾驶系统, 观察其在场景中的表现, 并提出多种场景生成策略. 随后, 研究人员逐渐将人工智能引入场景生成工作, 并增加场景中可测试的对象及其参数. 基于搜索的方法和机器学习是目前研究中主流的场景生成算法, 研究人员结合引导度量缩小输入空间, 并在测试场景中纳入更多可参数化的对象, 使多种因素充分交互、复杂约束相互碰撞, 制造出更具挑战性的驾驶环境, 以充分测试自动驾驶系统整体.

4.2 测试充分性

早期的自动驾驶测试技术承接软件测试领域的思想和方法, 将自动驾驶系统视为一种软件系统, 使用覆盖率

指标衡量测试充分性. 基本思想是, 如果能够测试自动驾驶系统面对所有类型环境下的全部决策行为, 即可认为其设计和功能是完善的. 由于自动驾驶项目代码量庞大, 因此指向软件内部的代码覆盖率不再适用. 研究人员结合测试场景多样性和自动驾驶系统行为, 提出了新的覆盖率度量, 指导测试场景生成, 全面充分测试自动驾驶系统. 本小节根据覆盖率度量将文献分为驾驶决策[65–67]和环境要素[68–71]两类. 表 2 总结了该分类下工作的覆盖方法.

表 2　测试充分性分类下的工作对比

文献	分类	覆盖方法	ADS
[65]	驾驶决策	添加不同位置的静态NPC使行驶路线覆盖更多区块	Apollo
[66]		计算抽象轨迹间的距离衡量轨迹相似度	Apollo
[67]		根据原始参数及其突变体的执行结果判断影响决策的参数集	Autonomoose
[68]	环境要素	将交汇处划分成不同路径类型, 并通过添加静态NPC测试路径规划	Apollo
[69]		将交汇处分类并为每类的代表生成场景, 测试系统对动态NPC的反应能力	Apollo
[70]		对天气、道路和自车行为3类输入进行组合测试, 覆盖所有抽象场景	Apollo
[71]		通过可达性分析确定车辆的物理交互区域, 并用向量集抽象表达	BeamNG.AI

4.2.1　驾驶决策

本分类的目标是覆盖所有类型的驾驶决策, 测试自动驾驶系统在不同场景下的分析和处理能力. 驾驶决策包括了变道、超车、转弯等行为, 一段时间内决策的组合形成轨迹. 当前研究通常使用环境参数和系统参数两种因素影响驾驶决策, 如图 9 所示.

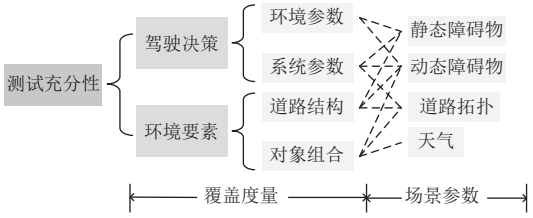

图 9　测试充分性分类

一些研究方案通过环境参数影响驾驶决策. 驾驶环境包括地图上的静态障碍物和动态障碍物, 自动驾驶系统与其交互, 做出对应的决策. 当改变环境参数时, 自车的行驶轨迹会发生变化, 即每一时刻的决策会动态变化. 例如, ASF[65]使用模糊测试变异 NPC 在测试场景中的位置, 以挑战自车的规划能力, 通过计算自车轨迹经过的区块与全部区块的比值得到轨迹覆盖率, 引导产生更复杂的驾驶行为. 实验结果表明, 与随机模糊测试器和 AV-Fuzzer 相比, 该工作能够覆盖更多区块. 但是, 其覆盖率度量的精度受限于区块面积, 难以实现细粒度的决策覆盖. 因此, 研究人员提出通过比较抽象轨迹判断决策的覆盖程度. 例如, BehAVExplor[66]计算了同一时间尺度内两个抽象轨迹的汉明距离, 以此衡量轨迹间的相似度, 从而将没有造成违规但差异较大的轨迹保留在种子集合中, 保证自车决策的多样性, 最终检测到 16 种独特的违规行为.

除此之外, 部分研究方案通过系统参数影响驾驶决策. 自动驾驶系统的参数与决策行为密切相关, 决定了其激进与保守程度. 例如, 自车与其他车辆应当保持的最小距离, 自车在交叉路口范围内的速度限制等. Laurent 等人[67]提出参数覆盖率, 对比规划器中的原始参数及其突变体的执行结果, 根据车辆轨迹、轨迹的安全性和舒适性指标的差距, 分析参数与决策的对应关系. 通过覆盖参数集, 验证自动驾驶系统的所有驾驶决策.

4.2.2　环境要素

本分类的目标是覆盖所有类型的环境要素. 环境中的对象包括天气、道路结构、障碍物等, 将其排列组合并实例化后, 场景数量呈几何级增长. 因此, 必须将所有输入对象分析归纳为抽象类型, 用少量的场景近似全部场景空间. 根据测试的输入对象, 可以将该类工作分为道路结构和对象组合两类, 如图 9 所示.

一些研究人员关注覆盖地图中所有交汇处的道路结构, 随后迭代生成关键场景. 例如, CROUTE[68]将地图建模

为带标记的 Petri 网, 分析交汇处的道路拓扑关系并进行聚类. 针对每一种道路结构, 该方法逐步添加静态障碍物, 测试自动驾驶系统的路径规划能力. ATLAS[69]根据地图拓扑中形状、交通灯、车道数量将交汇处分类, 并从每类中选择出代表性路段, 减小了地图中的测试范围. 与 CROUTE 不同, ATLAS 使用遗传算法生成多个动态 NPC, 使测试场景更加复杂和多样. 实验结果表明, 与随机采样相比, ATLAS 减少了 29.1% 的测试用例.

另一些研究人员考虑环境中各种输入对象的组合. 例如, ComOpT[70]对天气、道路和自车行为这 3 类输入进行组合测试, 覆盖所有的抽象场景, 进而实例化参数生成具体场景, 在潜在碰撞位置生成特定运动轨迹的代理, 扰动自动驾驶系统的行为. 该方法比随机生成方法多检测到 105 个违规行为. 部分研究不仅考虑环境中的对象, 还关注车辆行为对系统工作状态的影响. PhysCov[71]通过可达性分析确定了车辆的物理交互区域, 并用向量集抽象表达该区域. 通过已知向量与潜在全部向量的比值计算环境状态覆盖率, 指导生成具有不同特征的测试场景.

4.2.3 小 结

为了测试自动驾驶系统, 研究人员迁移应用了软件测试的覆盖率思想. 由于自动驾驶软件代码结构和数据交互复杂, 研究人员放弃使用代码覆盖率. 直观上, 通过构建更完善的场景库可以满足充分测试的要求. 但现实中场景是长尾分布的, 伴随着突发情况或新元素的排列组合, 不断出现新的未知场景. 因此, 直接验证场景覆盖率的难度很高, 当前研究通过覆盖驾驶决策或环境要素侧面证明测试充分性. 由于自动驾驶系统内部丰富的状态信息能够帮助研究人员理解任务实现逻辑, 所以最近的工作也将车辆状态和参数纳入考量[67,71].

4.3 对抗样本生成

由于自动驾驶系统逐渐应用更多人工智能算法, 部分研究生成对抗样本测试其安全性. 由图 3 可知, 自动驾驶系统的感知环节接收来自相机的图片数据和激光雷达的点云数据, 进行数据处理、融合和分析, 提供对外部环境的理解. 对抗性测试会生成对抗性的图片和点云, 并注入到仿真环境中, 破坏 AI 模型的输出结果. 因此, 当前存在大量欺骗感知模块的工作. 然而, 这些工作可能无法在现实环境中对自动驾驶系统造成严重危害. Wang 等人[72]通过理论分析和实验证明得出结论, 大部分针对感知模块的对抗攻击工作在闭环测试中效果很差, 无法导致整车系统的状态发生偏移, 组件级攻击在系统层面通常无效. 因此, 本文主要关注能够引起系统级行为偏差的对抗样本生成工作, 即以感知环节为攻击入口, 破坏自动驾驶系统整体的安全性. 为此, 需要将感知下游的规控组件、被控车辆模型和驾驶环境都纳入考虑, 在仿真环境中部署扰动, 衡量驾驶模型对攻击的鲁棒性.

基于攻击针对的任务类型, 本节将对抗样本生成工作分为两类, 分类问题[72–77]和回归问题[78–83]. 分类问题负责为对象建立离散的标签, 回归问题用于预测未来趋势和走向. 例如, 在目标检测模型中, 目标识别属于分类任务, 目标跟踪属于回归任务. 表 3 对比了不同工作的攻击入口、对抗样本生成方法和对系统层面的影响.

表 3 对抗样本生成分类下的工作对比

文献	分类	攻击入口	对抗样本生成方法	对系统层面影响	ADS
[72]	分类问题	相机	系统模型和优化算法	碰撞或违反交通规则	①
[73]		相机	网格搜索	闯红灯或紧急制动	Apollo
[74]		激光雷达	优化算法和全局抽样	紧急制动或永久静止	Apollo
[75]		激光雷达	优化算法和遗传算法	紧急制动或不规则变道	Apollo
[76]		多传感器融合	优化算法	碰撞	Apollo
[77]		多传感器融合	在视锥体范围生成欺骗点	碰撞或紧急制动	Apollo
[78]	回归问题	相机	网格搜索和贝叶斯优化	车道违规行为或碰撞	②
[79]		相机	优化算法	转向错误或碰撞	DriveNet[84]
[80]		相机	优化算法	偏离车道或碰撞	DAVE-2
[81]		相机	神经网络和优化算法	碰撞或紧急制动	Apollo
[82]		相机	优化算法	沿攻击轨迹行驶	②
[83]		相机	状态自适应和优化算法	偏离预期轨迹或碰撞	DAVE-2

注: ① 代表自动驾驶系统由多个部分组合, 包括传统目标检测器、基于卡尔曼滤波的多目标跟踪器、Apollo的规划模块、PID控制器和Stanley控制器; ② 代表基于条件模仿学习的深度学习导航模型[85]

4.3.1 分类问题

分类问题通常应用于目标识别模型中. 根据输入源和攻击入口的差异, 将针对分类问题的安全研究工作划分为 3 类, 即攻击相机、攻击激光雷达和攻击多传感器融合架构, 如图 10 所示.

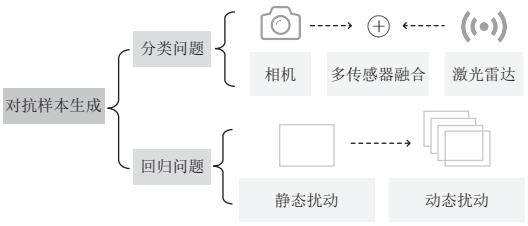

图 10 对抗样本生成分类

相机是重要的传感器, 价格低廉, 应用广泛, 以特斯拉为代表的一批厂商正在探索纯视觉方案. 但是, 相机一般只提供二维视角, 无法对外部环境形成立体的认知. 一些工作对以相机为输入源的模型进行安全研究. 交通信号灯是城市环境的典型特征之一, Yan 等人[73]研究了交通信号灯识别系统. 他们基于信号灯图像搜索出攻击成功率最大的激光参数集, 以在相机捕获的图像中创建彩色条纹, 误导模型的识别过程, 进而通过实验证明对抗样本可能使汽车闯红灯或紧急制动. 为了增强对抗样本在不同距离和视角下的鲁棒性, Wang 等人[72]提出了一种可以嵌入现有对抗攻击工作的系统框架 SysAdv, 设计了新的对象尺寸分布, 并根据控制模型选取合适的采样范围, 在实验中使自动驾驶系统错误识别停车标志和行人, 平均提高了对象逃逸攻击 70% 的成功率.

激光雷达提供了三维视角, 能够描述物体的形状和纹理信息, 但是容易受到天气干扰, 且成本较高. 一些工作对以激光雷达为输入源的模型进行安全研究. Cao 等人[74]分析了 Apollo 中基于激光雷达的目标检测过程, 得出结论: 传统的对抗攻击方法可以更改目标检测模型的输出, 却无法在感知模块的工作过程生成虚假物体. 随后, 他们向激光雷达注入激光脉冲, 添加少量点云, 欺骗目标检测模型, 使其在车辆前方检测到不存在的障碍物, 在注入 60 个欺骗点时有 75% 攻击成功率, 最终导致自动驾驶车辆紧急制动或永久静止. 但是, 该方法需要在攻击车上放置激光雷达, 攻击的隐蔽性较弱. 为了提高攻击的可行性, Yang 等人[75]对激光雷达及其目标检测模型进行仿真, 设计出误导模型检测结果的障碍物. 通过在路边放置该对抗样本生成欺骗性点云, 使系统误检测为存在车道入侵事件, 导致自动驾驶车辆紧急制动或不规则变道.

多传感器融合算法能够综合不同传感器的优点和缺点, 通过多个输入源对外部环境进行实时检测, 使感知结果的鲁棒性更强, 是当前厂商的主流解决方案. 当前对该算法的基本共识是, 在非所有输入源同时受到攻击的情况下, 感知模块的输出是可靠的. 因此, 为了验证该算法的安全性, 研究人员试图设计出能够干扰所有输入源的对抗样本. Cao 等人[76]分析了同时攻击了相机和激光雷达的可行性, 并将其建模为优化问题, 逐步生成对抗性的物理世界对象. 由仿真实验可知, 对抗样本会引发障碍物检测模型产生漏报, 导致碰撞等交通事故, 攻击成功率远高于基于遗传算法生成扰动的基线实验. 上述研究需要使用逆向工程等方法预先获取模型知识, 属于白盒方法, 局限性较大. Hallyburton 等人[77]则提出一种黑盒方法——视锥体攻击. 他们将攻击车检测框与受害车传感器连接形成的几何范围称为视锥体, 使用激光在视锥体内部注入虚假点. 在保持相机和激光雷达语义一致性的前提下, 破坏了二者的融合架构, 诱导感知模型产生误报或漏报, 损害自动驾驶功能. 该攻击方法对于各种 LiDAR 欺骗防御技术的平均攻击成功率超过 90%, 验证了该方法的有效性.

4.3.2 回归问题

回归问题负责预测车辆和行人的未来运动轨迹, 研究人员致力于干扰模型的预测值. 根据对抗样本的类型, 将该部分的工作划分为静态扰动和动态扰动这 2 类, 如图 10 所示.

最简单的对抗样本只修改单帧图片, 部署静态扰动. Boloor 等人[78]在车道上覆盖由长宽、颜色和旋转角度等多个参数表征的对抗性线条, 通过网格搜索和贝叶斯优化算法搜索输入空间, 结合目标函数逐步提高对抗样本的攻击成功率. 在以绝对转向角差为目标函数时, 平均攻击成功率超过 90%. Pavlitskaya 等人[79]选取了特定环境、天

气和时间下的场景进行测试, 利用基于雅可比的显著性映射方法 (Jacobian saliency map algorithm, JSMA)[86]和投影梯度下降法 (projected gradient descent, PGD)[87]生成对抗图像扰动, 并注入到仿真环境中车道旁边的广告牌中, 迫使车辆发生转向错误, 甚至导致碰撞事故. Wu 等人[80]同样以转向角为目标, 设计出通用性对抗扰动, 并为车道旁边的良性对象注入, 改变模型的转向预测, 攻击成功率远高于随机噪声方案.

为了生成针对性更强、实时性更强的对抗样本, 研究人员设计出涉及连续多帧的动态扰动, 结合车辆状态自适应生成对抗样本. Jha 等人[81]在自动驾驶系统上部署恶意软件, 并利用前馈神经网络模型选择合理的干扰时机, 连续修改多帧像素, 使感知模块错误计算车辆和行人的轨迹. 实验结果表明, 为在系统层面引发安全事故, 至少需要持续修改 14 帧涉及行人的图片或 48 帧涉及车辆的图片. 上述方法需要入侵车载系统, 接管汽车的传感器源, 攻击过程较为复杂, 一些研究人员试图降低攻击难度. 借助路边的广告牌, Patel 等人[82]提出一种白盒对抗性攻击方法, 根据车辆相对于广告牌的姿态信息动态生成对抗图像, 逐步修正车辆的方向和速度, 最终控制车辆的驾驶轨迹, 该方法能够使转向角偏差 90°以上. von Stein 等人[83]耦合车辆轨迹仿真与对抗样本生成过程, 利用 PGD 算法的思想自适应的生成对抗扰动, 使车辆执行恶意机动行为, 相比基准实验提升了 20.7% 的成功率.

4.3.3 小 结

该分类下的工作逐渐拓展测试目标, 从针对相机延展到针对激光雷达, 从针对单一传感器延展到针对多传感器融合方案; 扰动类型也由单个静态扰动升级为自适应的动态扰动. 随着研究的深度和广度不断提升, 对抗测试的可行性逐渐增加, 隐蔽性逐渐增强, 成功率也逐渐提高.

4.4 测试优化

测试优化是现有研究关注的方向之一, 有利于提高测试的速度和精度. 根据优化的阶段, 本节将测试优化工作划分为两类, 即测试结果优化[88–90]和测试用例优化[91–93]. 表 4 总结了各工作的具体优化方法.

表 4 测试优化分类下的工作对比

文献	分类	优化方法	ADS
[88]	测试结果优化	多代理仿真使自动驾驶系统实例控制场景中的每一辆车	Apollo
[89]		模糊测试生成静态障碍物, 蜕变测试筛选出误报	Apollo
[90]		训练了一个多模态模型筛选出误报	Apollo
[91]	测试用例优化	使用代理模型替代仿真器搜索关键场景	Pylot
[92]		使用因果模型筛选出可能引发违规的测试场景	Pylot
[93]		对现有数据集做测试约减和优先级排序	Apollo

4.4.1 测试结果优化

测试场景中的背景车辆通常由 PID 等简单算法控制, 仅根据预先规定的速度和驾驶策略行驶, 可能不会遵守交通规则. 其智能性和自主性较差, 鲁莽的驾驶行为造成了大量误报. 除此之外, 仿真器的环境建模与现实世界有偏差, 传感器建模与真实传感器也存在差距. 例如, 真实传感器可能被随机噪声影响, 仿真中传感器建模一般是理想的; 即使在建模时加入噪声, 也难以确定是否符合现实分布. 因此, 测试结果不仅存在误报, 还可能无法在实车上复现. 为了使测试结果更加精确, 研究人员提出了新的场景执行和验证方案, 如图 11 所示.

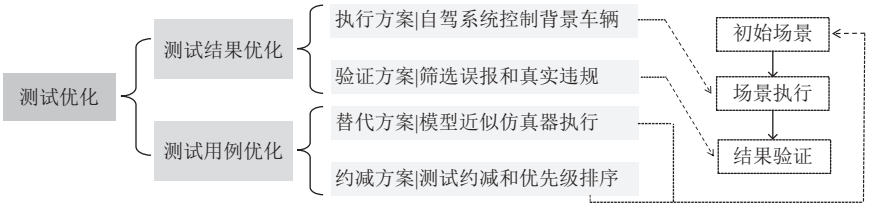

图 11 测试优化分类

一些研究人员修改场景执行方案, 不使用仿真器默认的简单控制器操纵背景车辆, 而是由自动驾驶系统控制. DoppelTest[88]使用自动驾驶系统的多个实例控制场景中的每一辆车, 保证每辆车都有足够复杂的逻辑对其他交通

参与者的行为做出合理决策. 实验结果表明, 74.5% 的违规场景由自动驾驶系统承担主责, 而使用非智能 NPC 的基线实验仅为 1.1%. 该方法避免了低智能 NPC 对测试结果的影响, 但拓展性不强, 无法对现有测试方法进行补充. 为了控制 NPC, DoppelTest 提出了一套订阅方案, 难以整合到现有测试框架中.

一些研究人员设计场景验证方案, 筛选出真正的违规场景和误报. MFT[89]使用模糊测试在 AV 的驾驶轨迹上随机生成静态障碍物, 构建蜕变关系检查碰撞事故是否发生变化, 从而区分出系统故障和无法避免的碰撞, 检测自动驾驶系统面对突发情况的容错能力, 最终筛选出 28.2% 的违规场景. Zhou 等人[90]构建了一个包括事故视频和描述文本的多模态数据集, 并基于 X-CLIP 模型训练了一个多模态模型. 该模型集成在测试框架 CollVer 中, 能够筛选出自车承担事故主要责任的违规场景. 实验结果表明, 模型的查准率为 82.2%, 查全率为 77.9%.

4.4.2 测试用例优化

执行过程是整车系统测试与模块测试的主要区别之一. 模块测试接收图片、轨迹等简单数据, 验证单一模型功能, 执行速度块、效率高; 整车系统测试需要使用仿真器执行和验证大量测试场景, 时间成本较高, 效率较低. 因此, 研究人员亟需加速仿真测试, 优化测试用例. 具体方案有两种, 分别是替代方案和约减方案, 如图 11 所示.

替代方案使用各种模型完成测试场景的初次筛选, 去除无效场景, 保留违规潜力更大的关键测试场景. 例如, SAMOTA[91]训练了一个代理模型近似仿真器的执行结果, 评估测试场景的违规程度, 从而只在仿真测试中验证最大概率发生事故的场景. CART[92]从驾驶记录中推断输入与输出之间的因果关系, 并形式化为因果模型, 输入是测试场景的参数, 输出是自动驾驶表现的行为与预定义的错误行为之间的距离. 随后用因果推理查询模型, 估算场景参数的执行结果, 并在仿真器中运行可能性最大的场景. 与 SAMOTA 相比, CART 在相同的测试时间内能够发现更多违规场景, 其生成的测试集多样性也更高.

现有软件更新迭代速度快, 为了保证新的功能不会引入安全问题, 需要进行回归测试. 由于不断复用测试数据, 测试用例集合的规模不断增加, 冗余度上升, 且存在测试场景的同质化问题. 约减方案使用选择、约减和优先级排序方法解决上述问题. 例如, STRAP[93]接收录制的真实驾驶数据集, 依据数据片段的相似性约减长段的驾驶记录, 并利用驾驶场景的特征覆盖率和稀有度对剩余片段进行优先级排序, 加速回归测试, 约简后的测试集平均能发现原测试集中 98.8% 的故障. 测试约减技术已经较为成熟[94−96], 但多针对车道保持等简单场景, 仍然需要研究人员将应用领域从高级驾驶辅助系统迁移到自动驾驶系统.

4.4.3 小 结

误报检测、多代理仿真研究在传统的软件测试领域已取得长足进展, 但在自动驾驶领域才刚刚起步, 仍需深入研究. 仿真加速、测试约减技术已经被应用于测试高级辅助驾驶系统, 但是研究对象较为简单, 需要研究人员进行拓展研究和方案迁移. 虽然测试优化技术无法发现自动驾驶系统的缺陷, 但是研究人员可以将其整合到安全测试框架中, 提高缺陷检测效率和准确率, 辅助漏洞挖掘过程.

4.5 测试预言

测试预言用于判断测试结果是否符合预期, 评估自动驾驶系统的表现和性能. 道路测试中的测试预言, 如行驶里程和脱离接管率, 无法应用于仿真测试; 模块安全研究中的测试预言, 如转向角偏离程度和终点预测误差, 只能用于判断单个模块的执行结果, 无法准确衡量整车系统的安全性. 因此, 研究人员需要设计精确、合适的测试预言并证明其有效性.

在整车系统安全测试领域, 常用的测试预言有发生碰撞、保持静止等. 最近有研究人员整合了以往工作中提出的跨越车道线、违反限速等涉及交通规则的测试预言, 利用考虑交规的场景描述语言分析了自动驾驶系统遵守交通规则的能力[56]. 由于测试预言服务于测试场景的有效性验证, 属于场景生成和执行步骤后的一个辅助环节. 因此, 上述 4 个分类已经涉及了相关工作, 本节不再赘述.

4.6 总 结

以面向整车系统的自动驾驶安全测试框架为分类支撑, 上述文献覆盖了近年来该领域的主流研究工作. 最核心的研究工作围绕测试用例生成方法展开, 相关研究时间跨度大、思路方法多、需求迫切, 技术体系也不断融合

提升. 此外, 为提高测试的效率和准确率, 各种测试优化方案受到关注. 综上所述, 当前研究正呈现出以下趋势: 从手工定义简单场景, 过渡到利用人工智能构建复杂场景; 从只关注漏洞挖掘的效果, 过渡到效率与效果并重; 从发生碰撞行为等简单的测试预言, 过渡到处理交通规则间复杂逻辑的测试预言.

5 面向整车系统的自动驾驶安全测试评估

5.1 常用评价指标

研究人员使用多种评价指标评估实验的效果和效率. 由于部分指标服务于研究工作的特定需求, 适用性有限, 因此本文仅总结了具备较强通用性的评价指标, 涉及的文献如表 5 所示.

表 5 常用的评价指标

评价指标	涉及文献
安全性指标	[48−60,61−66,68−70,88−93]
多样性指标	[48,50,51,92]
效率指标	[48,50,51,54,57−59,68,69,90,91,93]
攻击效果指标	[72,73−78,80,81,83]

(1) 安全性指标. 发现更多自动驾驶系统的安全问题是测试工作最主要的目标之一. 为了衡量研究方案的有效性, 研究人员统计了测试结束后揭露的安全问题数量, 以及经过归纳分类后的安全问题类型数量. 此外, 部分研究人员还使用“独特安全问题数量”作为评价指标, 具体定义根据不同文献的研究需求而有所差异. 例如, 文献 [55] 将其定义为导致该安全问题的代码级决策逻辑与其他安全问题不同; 文献 [60] 则将其定义为诱发该安全问题的场景与其他场景的部分参数配置不同.

(2) 多样性指标. 多样化的场景有助于暴露自动驾驶系统中更多类型的安全缺陷, 轨迹距离是典型的多样性评价指标. 轨迹距离指标计算不同驾驶场景中 NPC 轨迹之间的距离, 以此衡量场景的多样性. 此外, 文献 [92] 还使用了测试集直径 (test set diameter, TSD) 作为评估测试集多样性的指标.

(3) 效率指标. 该类指标用于衡量测试的时间成本, 比较方案的运行效率. 典型代表包括发现首个或首类安全问题所需时间或所需仿真次数, 发现所有类型安全问题所需时间或所需仿真次数, 测试用例缩减数量等.

(4) 攻击效果指标. 此类指标以攻击成功率为主, 包括相对成功率、平均成功率等指标. 例如, 相对成功率指相对于基准实验所提升的攻击成功率. 此类指标主要被对抗样本生成分类下的工作采用, 如果感知模型未完成正常的任务, 则视为攻击成功.

5.2 常用对比方法

为了验证所提方案的有效性, 研究人员选择多种比较基准实施对比实验, 包括随机方案、消融方案等. 本节列举了现有工作中常用的对比方法, 涉及的文献如表 6 所示.

(1) 随机方案. 在研究的早期阶段, 研究人员采取随机算法作为基准进行比较, 这是最简单的对比方案. 例如, AV-Fuzzer 使用随机生成场景的模糊测试器作为对比方案; 文献 [76] 使用随机噪声扰动模型完成对比实验.

(2) AV-Fuzzer. 文献 [48] 设计了一个模糊测试器 AV-Fuzzer, 这是首个完整测试了自动驾驶系统的研究成果, 其测试方法和实验设计指导了后续多项工作, 部分研究工作选择将其作为基准比较实验效果. 值得注意的是, AV-Fuzzer 框架最初在仿真器 Lgsvl 上实现. 然而, 该平台现在已经停止提供相关服务, 研究人员需要在新平台上重新实现相关算法, 以实施有效的实验对比和分析.

(3) 消融方案. 消融实验在机器学习领域被广泛用于评估模型中不同组件对整体的重要程度. 基于这一概念, 本文将“消融方案”定义为: 通过移除或替换测试框架的部分组件, 验证这些组件在测试过程中的作用, 进而评估它们对于整体的贡献度. 实施此类实验能够证明组件在测试框架中的重要性. 例如, 文献 [53] 禁用了驾驶质量反馈引擎, 发现在无指导的情况下, 发现的安全问题数量下降了 47%.

表 6 常用的对比方法

对比方法	涉及文献
随机方案	[48,54,56−59,65,66,68−70,76,78,80,81,88,91,92]
AV-Fuzzer	[49−54,60,65,66,90]
消融方案	[50−53,55,60,61,66,72,81,90−93]

5.3 总 结

在评价指标方面, 关键场景生成分类下的工作更关注安全性指标和效率指标, 试图更快的生成更多危险场景; 对抗样本生成分类下的工作则选择攻击成功率作为衡量攻击效果的标准. 测试充分性分类和测试优化分类下的工作更倾向于结合具体实验设置独特的评价指标, 以证明覆盖率度量或优化工作的有效性.

在对比方法方面, 早期阶段研究人员通常采用随机方案和 AV-Fuzzer 实施对比实验. 随着研究工作的逐渐深入, AV-Fuzzer 不再成为首选方案, 研究人员更倾向于根据研究的主要问题选择更匹配的对比方案. 此外, 采取消融方案已经成为一种有效的策略, 以证明组件的重要性.

6 挑战与机遇

本节结合自动驾驶安全领域的主要研究方向, 提出了面向整车系统的安全测试研究面临的现实挑战和研究机遇, 如表 7 所示.

表 7 整车系统安全测试研究的挑战与机遇

研究方向	现实挑战	研究机遇
关键场景生成	低置信度的测试场景	基于真实性的场景生成技术
测试充分性	通用性受限的覆盖率度量	覆盖分析框架设计
对抗样本生成	离线的模型验证方案	感知模型闭环测试
测试优化	低智能的交通参与者	智能代理规控方案

6.1 现实挑战

(1) 低置信度的测试场景. 现实世界的输入空间具有几何级别的参数量, 且存在小概率事件, 如前车随意丢弃的异物、蓄意冲出马路的行人等. 而局限于成本和效率, 仿真测试中场景空间的参数量较少, 无法拟合真实环境. 同时, 仿真器中的汽车模型和传感器模型可能与真实对象存在差距, 影响测试结果的有效性. 除此之外, 现有研究的测试场景基于专家知识构建, 由算法自动化生成, 与驾驶数据集脱钩. 此类测试场景缺乏合理性和真实性, 导致其中的约束条件可能超越车辆的动力学极限, 安全事故无法避免, 与自动驾驶系统无关. 因此, 现有的工作中测试场景的置信度存疑, 亟需研究人员设计出更有效的场景生成方法.

(2) 通用性受限的覆盖率度量. 现有研究通过覆盖率度量衡量测试充分性, 研究人员根据各自的研究目标定义了不同的覆盖率度量, 或对相同的度量使用不同的术语, 没有统一和规范的标准. 尽管 Tahir 等人[97]调研了自动驾驶领域的覆盖率研究, 并将其分为场景覆盖率、情景覆盖率和需求覆盖率这 3 类. 但他们的工作从宏观视角出发, 使用宽泛的概念定义覆盖率. 但是, 实际测试中的覆盖率随测试目标的不同而发生改变, 不存在可以套用的通用框架. 由此导致在面对新问题或新条件时, 迁移成熟的解决方案十分困难, 阻碍了研究人员拓展测试充分性领域的研究深度和广度.

(3) 离线的模型验证方案. 在自动驾驶系统中, 深度学习模型逐渐被引入用于分析和处理数据, 尤其是在感知模块. 研究人员生成对抗样本验证模型的正确性和可靠性. 然而, 目前大部分工作仅针对单独的模型实施离线测试, 而没有在仿真环境中进行闭环的在线测试. 在线测试能够提供控制输出对感知输入的反馈, 在实际环境中评估模型的性能和表现. 相比之下, 离线测试方案缺少上述反馈机制, 无法持续地发起针对性攻击, 难以全面验证模型的安全性. 因此, 在对整车系统进行安全分析时, 此类方法的效果并不理想, 亟需闭环测试方案.

(4) 低智能的交通参与者. 为了充分测试自动驾驶系统与 NPC 的交互能力, 研究人员使用场景描述语言构建了存在多个 NPC 的测试场景, 涉及各种类型的背景车辆. 然而, 由于仿真器缺少相关功能, 现有解决方案尚未实现背景车辆与外部环境的有效交互, 即无法根据环境的反馈实时控制背景车辆的驾驶行为. 因此, 背景车辆的智能性不足, 无法处理突发状况, 测试结果中误报占比极高. 例如, 自车切入背景车辆所在车道, 而背景车辆维持固定速度, 于自车的侧后方发生追尾. 上述事故的原因是背景车辆没能对切入的车辆做出合适的反应, 不应该将责任归咎于自动驾驶系统. 此外, 现有工作关注的 NPC 主要是场景中的背景车辆, 较少考虑在测试中纳入具有不同运动方式的行人对象. 因此, 亟需研究人员从误报产生的源头展开分析, 提高 NPC 的智能性并丰富其类型.

6.2 研究机遇

(1) 基于真实性的场景生成技术. 场景生成技术是当前研究的热点之一, 在安全测试框架中占据核心地位. 尽管研究人员设计了各种算法引导生成关键场景, 但无法确认测试结果能否在现实中复现, 仍然需要进一步验证. 为了提高测试场景的置信水平, 一方面可以使用基于数据驱动的场景生成技术, 从驾驶数据集中提取驾驶习惯和特征, 以真实数据为支撑, 构建更合理的场景; 另一方面, 需要研究仿真世界与现实世界的一致性问题[98], 评估和量化二者的差距, 提高仿真器中环境、传感器和车辆的建模精度.

(2) 覆盖分析框架设计. 覆盖率量化了自动驾驶系统的测试充分性, 有助于估算测试结束的时机, 判断软件测试是否完备. 为了设计并实现通用的覆盖分析框架, 工业界和学术界应当达成产学共促、统一标准的共识, 进行长期的合作、交流与讨论, 使研究人员认识、理解并商讨出有效的覆盖率度量, 以及基于覆盖率的通用测试框架. 例如, 未来的覆盖率度量应当综合考虑汽车自身状态和外部环境, 增强在不同测试方案之间的可移植性.

(3) 感知模型闭环测试. 闭环测试在漏洞挖掘能力、测试全面性等方面都存在优势, 值得深入研究. 然而, 该方案面临着一些难点. 首先是在测试工具方面, 开环测试针对单个模型进行研究, 而闭环测试需要在仿真器中运行庞大的自动驾驶系统, 这意味着测试效率受到限制. 其次是在测试数据方面, 开环测试利用开源数据集验证对抗样本的有效性, 而闭环测试需要将对抗样本融入测试场景, 在仿真环境中验证, 增加了时间和经济成本. 最后是在测试方法方面, 开环测试只需要设计优化算法欺骗感知模型, 而闭环测试必须结合规划和控制环节设计对抗样本生成策略, 综合考虑模型之间的关联性. 综上所述, 未来的工作可以基于这 3 个方面进行改进, 研发通信效率更高、操作规范更简单的测试工具, 结合自动驾驶全流程研究对抗样本生成策略, 构建体系化的测试方案.

(4) 智能代理规控方案. 背景车辆的智能程度限制了与其他代理的交互能力, 随着测试要求逐渐提高, 简单的控制模型无法满足复杂的测试需求, 设计具备复杂逻辑的规控系统是当前研究的重要之一. 人工智能等新兴技术的发展提供了新的研究思路. 例如, 利用强化学习或大语言模型训练一个驾驶模型, 使背景车辆的决策方法更接近驾驶员. 多代理测试也是可行的解决方案, 用自动驾驶系统控制所有背景车辆, 避免了低智能性对测试的干扰. 除此之外, 智能化和对抗性的行人对象也能使测试场景更具挑战性. 部分文献深入分析了更真实的行人模型, 如 Muktadir 等人[99]建模了乱穿马路的行人模型, 研究人员可以将此类建模方案集成到现有测试框架中.

7 总 结

自动驾驶系统将人工智能算法集成到复杂生态系统中, 完成数据处理、环境感知、决策控制等关键任务, 同时也引入了长尾场景分布、对抗样本等新的安全问题. 针对当前面临的安全挑战, 本文深入研究并整理分析了面向整车系统的自动驾驶安全测试研究的历史工作和最新进展. 首先, 通过研讨现有研究并融入通用测试框架, 形成面向整车系统的自动驾驶安全测试框架. 其次, 基于上述框架总结出现有工作的 5 类核心研究问题, 并深入对比每类问题的关键技术、研究现状和发展脉络. 此外, 还对现有研究中广泛使用的评价指标和对比方法进行了分析和总结. 最后, 给出了整车系统安全测试研究领域可能存在的现实挑战与未来研究机遇, 展望了更真实、更通用、更智能的解决方案. 伴随着自动驾驶算法在车辆上开始部署实装, 未来从整体视角开展的安全测试研究将更加深入, 本文希望能够对此有所帮助和启发.

References

[1] Kilgore T. Baidu debuts Robotaxi ride hailing service in China, using self-driving electric taxis. 2019. https://www.marketwatch.com/story/baidu-debuts-robotaxi-ride-hailing-service-in-china-using-self-driving-electric-taxis-2019-09-26

[2] Version 2024: Intelligent automotive solution 2030. 2021. https://www-file.huawei.com/-/media/corp2020/pdf/giv/2024/intelligent_automotive_solution_whitepaper_2030_en.pdf

[3] McFarland M. Uber self-driving car kills pedestrian in first fatal autonomous crash. 2018. https://money.cnn.com/2018/03/19/technology/uber-autonomous-car-fatal-crash/index.html

[4] Warrier M. Nio ES8 accident rekindles concerns over safety with smart EVs in China. 2021. https://www.benzinga.com/government/21/08/22504236/nio-es8-accident-rekindles-concerns-over-safety-with-smart-evs-in-china

[5] ISO. ISO 21448. Road vehicles—Safety of the intended functionality. 2022. https://cdn.standards.iteh.ai/samples/77490/d9843a45e11947e0aa79aaf2f00b65a8/ISO-21448-2022.pdf

[6] ISO. ISO 26262-1. Road vehicles—Functional safety—Part 1: Vocabulary. 2018. https://cdn.standards.iteh.ai/samples/68383/4e26ddadc54a4198bed652afe29669fa/ISO-26262-1-2018.pdf

[7] Eykholt K, Evtimov I, Fernandes E, Li B, Rahmati A, Xiao CW, Prakash A, Kohno T, Song D. Robust physical-world attacks on deep learning visual classification. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 1625–1634. [doi: 10.1109/CVPR.2018.00175]

[8] Eykholt K, Evtimov I, Fernandes E, Li B, Rahmati A, Tramer F, Prakash A, Kohno T, Song D. Physical adversarial examples for object detectors. In: Proc. of the 12th USENIX Workshop on Offensive Technologies (WOOT). Baltimore: USENIX Association, 2018. 1–10.

[9] Althoff M, Lutz S. Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles. In: Proc. of the 2018 IEEE Intelligent Vehicles Symp. (IV). Changshu: IEEE, 2018. 1326–1333. [doi: 10.1109/IVS.2018.8500374]

[10] Klischat M, Althoff M. Generating critical test scenarios for automated vehicles with evolutionary algorithms. In: Proc. of the 2019 IEEE Intelligent Vehicles Symp. (IV). Paris: IEEE, 2019. 2352–2358. [doi: 10.1109/IVS.2019.8814230]

[11] Cao YL, Xiao CW, Anandkumar A, Xu DF, Pavone M. AdvDO: Realistic adversarial attacks for trajectory prediction. In: Proc. of the 17th European Conf. on Computer Vision (ECCV). Tel Aviv: Springer, 2022. 36–52. [doi: 10.1007/978-3-031-20065-6_3]

[12] Garcia J, Feng Y, Shen JJ, Almanee S, Xia Y, Chen QA. A comprehensive study of autonomous vehicle bugs. In: Proc. of the 42nd IEEE/ACM Int'l Conf. on Software Engineering (ICSE). Seoul: ACM, 2020. 385–396. [doi: 10.1145/3377811.3380397]

[13] Tang SC, Zhang ZY, Tang J, Ma L, Xue YX. Issue categorization and analysis of an open-source driving assistant system. In: Proc. of the 2021 IEEE Int'l Symp. on Software Reliability Engineering Workshops (ISSREW). Wuhan: IEEE, 2021. 148–153. [doi: 10.1109/ISSREW53611.2021.00057]

[14] Zhu XL, Wang HC, You HM, Zhang WH, Zhang YY, Liu S, Chen JJ, Wang Z, Li KQ. Survey on testing of intelligent systems in autonomous vehicles. Ruan Jian Xue Bao/Journal of Software, 2021, 32(7): 2056–2077 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6266.htm [doi: 10.13328/j.cnki.jos.006266]

[15] Tang SC, Zhang ZY, Zhang Y, Zhou JX, Guo Y, Liu S, Guo SJ, Li YF, Ma L, Xue YX, Liu Y. A survey on automated driving system testing: Landscapes and trends. ACM Trans. on Software Engineering and Methodology, 2023, 32(5): 124. [doi: 10.1145/3579642]

[16] Dai JR, Li ZR, Zhang WQ, Zhang Y, Yang M. Simulation-based fuzzing for autonomous driving systems: Landscapes, challenges and prospects. Journal of Computer Research and Development, 2023, 60(7): 1433–1447 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.202330156]

[17] Society of Automotive Engineers (SAE) International. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. 2021. https://ca-times.brightspotcdn.com/54/02/2d5919914cfe9549e79721b12e66/j3016-202104.pdf

[18] Ben Abdessalem R, Nejati S, Briand LC, Stifter T. Testing advanced driver assistance systems using multi-objective search and neural networks. In: Proc. of the 31st IEEE/ACM Int'l Conf. on Automated Software Engineering (ASE). Singapore: ACM, 2016. 63–74. [doi: 10.1145/2970276.2970311]

[19] Klück F, Zimmermann M, Wotawa F, Nica M. Genetic algorithm-based test parameter optimization for ADAS system testing. In: Proc. of the 19th Int'l Conf. on Software Quality, Reliability and Security (QRS). Sofia: IEEE, 2019. 418–425. [doi: 10.1109/QRS.2019.00058]

[20] Baidu Apollo GitHub repository. 2024. https://github.com/ApolloAuto/apollo

[21] Autoware. https://autoware.org

[22] Johnson MA. PID control technology. In: PID Control. London: Springer, 2005. 1–46. [doi: 10.1007/1-84628-148-2_1]

[23] Camacho EF, Bordons C. Introduction to model predictive control. In: Model Predictive Control. 2nd ed., London: Springer, 2007. 1–11. [doi: 10.1007/978-0-85729-398-5_1]

[24] Chen L, Wu PH, Chitta K, Jaeger B, Geiger A, Li HY. End-to-end autonomous driving: Challenges and frontiers. arXiv:2306.16927,

2024.

[25] Pomerleau DA. ALVINN: An autonomous land vehicle in a neural network. In: Proc. of the 2nd Int'l Conf. on Neural Information Processing Systems (NIPS). Cambridge: MIT Press, 1988. 305–313.

[26] Hu YH, Yang JZ, Chen L, Li KY, Sima CH, Zhu XZ, Chai SQ, Du SY, Lin TW, Wang WH, Lu LW, Jia XS, Liu Q, Dai JF, Qiao Y, Li HY. Planning-oriented autonomous driving. In: Proc. of the 2023 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). Vancouver: IEEE, 2023. 17853–17862. [doi: 10.1109/CVPR52729.2023.01712]

[27] Bojarski M, Del Testa D, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel LD, Monfort M, Muller U, Zhang JK, Zhang X, Zhao J, Zieba K. End to end learning for self-driving cars. arXiv:1604.07316, 2016.

[28] Commaai. commaai/openpilot. 2024. https://github.com/commaai/openpilot

[29] Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan Y, Baldan G, Beijbom O. NuScenes: A multimodal dataset for autonomous driving. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). Seattle: IEEE, 2020. 11618–11628. [doi: 10.1109/CVPR42600.2020.01164]

[30] Scenario-based open road testing. 2022. https://report.asam.net/scenario-based-open-road-testing

[31] State Administration for Market Regulation, Standardization Administration of the People's Republic of China, SAC. GB/T 43119-2023 Technical requirements for construction of closed test site for automatic driving. 2023 (in Chinese). https://www.chinesestandard.net/PDF.aspx/GBT43119-2023

[32] Zhou JX, Zhang Y, Guo SJ, Guo Y. A survey on autonomous driving system simulators. In: Proc. of the 2022 IEEE Int'l Symp. on Software Reliability Engineering Workshops (ISSREW). Charlotte: IEEE, 2022. 301–306. [doi: 10.1109/ISSREW55968.2022.00084]

[33] Pun A, Sun G, Wang JK, Chen Y, Yang Z, Manivasagam S, Ma WC, Urtasun R. LightSim: Neural lighting simulation for urban scenes. arXiv:2312.06654, 2023.

[34] Wei YX, Wang Z, Lu YF, Xu CX, Liu CX, Zhao H, Chen SH, Wang YF. Editable scene simulation for autonomous driving via collaborative LLM-agents. In: Proc. of the 2024 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). Seattle: IEEE, 2024. 15077–15087. [doi: 10.1109/CVPR52733.2024.01428]

[35] Rong GD, Shin BH, Tabatabaee H, Lu Q, Lemke S, Možeiko M, Boise E, Uhm G, Gerow M, Mehta S, Agafonov E, Kim TH, Sterner E, Ushiroda K, Reyes M, Zelenkovsky D, Kim S. LGSVL simulator: A high fidelity simulator for autonomous driving. In: Proc. of the 23rd IEEE Int'l Conf. on Intelligent Transportation Systems (ITSC). Rhodes: IEEE, 2020. 1–6. [doi: 10.1109/ITSC45102.2020.9294422]

[36] Dosovitskiy A, Ros G, Codevilla F, Lopez A, Koltun V. CARLA: An open urban driving simulator. arXiv:1711.03938, 2017.

[37] IPG. CarMaker. 2024. https://ipg-automotive.com/en/products-solutions/software/carmaker

[38] CarSim. 2024. https://www.carsim.com/products/carsim/index.php

[39] Krajzewicz D, Hertkorn G, Wagner P, Rössel C. SUMO (simulation of urban mobility) an open-source traffic simulation. In: Proc. of the 4th Middle East Symp. on Simulation and Modelling (MESM). Sharjah: 2002. 183–187.

[40] PTV. Vissim. 2024. https://www.ptvgroup.com/en/solutions/products/ptv-vissim

[41] Ding WH, Xu CJ, Arief M, Lin HH, Li B, Zhao D. A survey on safety-critical driving scenario generation—A methodological perspective. IEEE Trans. on Intelligent Transportation Systems, 2023, 24(7): 6971–6988. [doi: 10.1109/TITS.2023.3259322]

[42] Menzel T, Bagschik G, Maurer M. Scenarios for development, test and validation of automated vehicles. In: Proc. of the 2018 IEEE Intelligent Vehicles Symp. (IV). Changshu: IEEE, 2018. 1821–1827. [doi: 10.1109/IVS.2018.8500406]

[43] Zhou Y, Sun Y, Tang Y, Chen YQ, Sun J, Poskitt CM, Liu Y, Yang ZJ. Specification-based autonomous driving system testing. IEEE Trans. on Software Engineering, 2023, 49(6): 3391–3410. [doi: 10.1109/TSE.2023.3254142]

[44] Huai YQ, Almanee S, Chen YTY, Wu XF, Chen QA, Garcia J. ScenoRITA: Generating diverse, fully mutable, test scenarios for autonomous vehicle planning. IEEE Trans. on Software Engineering, 2023, 49(10): 4656–4676. [doi: 10.1109/TSE.2023.3309610]

[45] Deng Y, Yao JH, Tu Z, Zheng X, Zhang MS, Zhang TY. TARGET: Automated scenario generation from traffic rules for testing autonomous vehicles via validated LLM-guided knowledge extraction. arXiv:2305.06018, 2025.

[46] Montanari F, Stadler C, Sichermann J, German R, Djanatliev A. Maneuver-based resimulation of driving scenarios based on real driving data. In: Proc. of the 2021 IEEE Intelligent Vehicles Symp. (IV). Nagoya: IEEE, 2021. 1124–1131. [doi: 10.1109/IV48863.2021.9575441]

[47] Barr ET, Harman M, McMinn P, Shahbaz M, Yoo S. The oracle problem in software testing: A survey. IEEE Trans. on Software Engineering, 2015, 41(5): 507–525. [doi: 10.1109/TSE.2014.2372785]

[48] Li GP, Li YR, Jha S, Tsai T, Sullivan M, Hari SKS, Kalbarczyk Z, Iyer R. AV-Fuzzer: Finding safety violations in autonomous driving systems. In: Proc. of the 31st IEEE Int'l Symp. on Software Reliability Engineering (ISSRE). Coimbra: IEEE, 2020. 25–36. [doi: 10.1109/ISSRE5003.2020.00012]

[49] Sun LL, Huang S, Zheng CY, Bai TT, Hu Z. Test case generation for autonomous driving based on improved genetic algorithm. In: Proc. of the 23rd IEEE Int'l Conf. on Software Quality, Reliability, and Security (QRS). Chiang Mai: IEEE, 2023. 272–278. [doi: 10.1109/QRS60937.2023.00035]

[50] Tian HX, Jiang Y, Wu GQ, Yan JR, Wei J, Chen W, Li S, Ye D. MOSAT: Finding safety violations of autonomous driving systems using multi-objective genetic algorithm. In: Proc. of the 30th ACM Joint European Software Engineering Conf. and Symp. on the Foundations of Software Engineering. Singapore: ACM, 2022. 94–106. [doi: 10.1145/3540250.3549100]

[51] Tian HX, Wu GQ, Yan JR, Jiang Y, Wei J, Chen W, Li S, Ye D. Generating critical test scenarios for autonomous driving systems via influential behavior patterns. In: Proc. of the 37th IEEE/ACM Int'l Conf. on Automated Software Engineering. Rochester: ACM, 2022. 46. [doi: 10.1145/3551349.3560430]

[52] Song RY, Ozmen MO, Kim H, Muller R, Celik ZB, Bianchi A. Discovering adversarial driving maneuvers against autonomous vehicles. In: Proc. of the 32nd USENIX Security Symp. Anaheim: USENIX Association, 2023. 2957–2974.

[53] Kim S, Liu M, Rhee J, Jeon Y, Kwon Y, Kim CH. DriveFuzz: Discovering autonomous driving bugs through driving quality-guided fuzzing. In: Proc. of the 2022 ACM SIGSAC Conf. on Computer and Communications Security (CCS). Los Angeles: ACM, 2022. 1753–1767. [doi: 10.1145/3548606.3560558]

[54] Tang SC, Zhang ZY, Zhou JX, Zhou Y, Li YF, Xue YX. EvoScenario: Integrating road structures into critical scenario generation for autonomous driving system testing. In: Proc. of the 34th IEEE Int'l Symp. on Software Reliability Engineering (ISSRE). Florence: IEEE, 2023. 309–320. [doi: 10.1109/ISSRE59848.2023.00054]

[55] Wan ZW, Shen JJ, Chuang J, Xia X, Garcia J, Ma JQ, Chen QA. Too afraid to drive: Systematic discovery of semantic dos vulnerability in autonomous driving planning under physical-world attacks. In: Proc. of the 29th Annual Network and Distributed System Security Symp. (NDSS). San Diego: Internet Society, 2022. [doi: 10.14722/ndss.2022.24177]

[56] Sun Y, Poskitt CM, Sun J, Chen YQ, Yang ZJ. LawBreaker: An approach for specifying traffic laws and fuzzing autonomous vehicles. In: Proc. of the 37th IEEE/ACM Int'l Conf. on Automated Software Engineering. Rochester: ACM, 2022. 62. [doi: 10.1145/3551349.3556897]

[57] Lu CJ, Shi YZ, Zhang HH, Zhang M, Wang TX, Yue T, Ali S. Learning configurations of operating environment of autonomous vehicles to maximize their collisions. IEEE Trans. on Software Engineering, 2023, 49(1): 384–402. [doi: 10.1109/TSE.2022.3150788]

[58] Lu CJ. Test scenario generation for autonomous driving systems with reinforcement learning. In: Proc. of the 45th IEEE/ACM Int'l Conf. on Software Engineering: Companion Proc. (ICSE-Companion). Melbourne: IEEE, 2023. 317–319. [doi: 10.1109/ICSE-Companion58688.2023.00086]

[59] Ul Haq F, Shin D, Briand LC. Many-objective reinforcement learning for online testing of DNN-enabled systems. In: Proc. of the 45th IEEE/ACM Int'l Conf. on Software Engineering (ICSE). Melbourne: IEEE, 2023. 1814–1826. [doi: 10.1109/ICSE48619.2023.00155]

[60] Zhong ZY, Kaiser G, Ray B. Neural network guided evolutionary fuzzing for finding traffic violations of autonomous vehicles. IEEE Trans. on Software Engineering, 2023, 49(4): 1860–1875. [doi: 10.1109/TSE.2022.3195640]

[61] Zhang XD, Zhao W, Sun Y, Sun J, Shen YL, Dong XW, Yang ZJ. Testing automated driving systems by breaking many laws efficiently. In: Proc. of the 32nd ACM SIGSOFT Int'l Symp. on Software Testing and Analysis. Seattle: ACM, 2023. 942–953. [doi: 10.1145/3597926.3598108]

[62] Prakash A, Chitta K, Geiger A. Multi-modal fusion Transformer for end-to-end autonomous driving. In: Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). Nashville: IEEE, 2021. 7073–7083. [doi: 10.1109/CVPR46437.2021.00700]

[63] Neis N, Beyerer J. Literature review on maneuver-based scenario description for automated driving simulations. In: Proc. of the 2023 IEEE Intelligent Vehicles Symp. (IV). Anchorage: IEEE, 2023. 1–8. [doi: 10.1109/IV55152.2023.10186545]

[64] Bengio Y, Lahlou S, Deleu T, Hu EJ, Tiwari M, Bengio E. GFlowNet foundations. arXiv:2111.09266, 2023.

[65] Hu ZS, Guo SJ, Zhong ZY, Li K. Coverage-based scene fuzzing for virtual autonomous driving testing. arXiv:2106.00873, 2021.

[66] Cheng MF, Zhou Y, Xie XF. BehAVExplor: Behavior diversity guided testing for autonomous driving systems. In: Proc. of the 32nd ACM SIGSOFT Int'l Symp. on Software Testing and Analysis. Seattle: ACM, 2023. 488–500. [doi: 10.1145/3597926.3598072]

[67] Laurent T, Klikovits S, Arcaini P, Ishikawa F, Ventresque A. Parameter coverage for testing of autonomous driving systems under uncertainty. ACM Trans. on Software Engineering and Methodology, 2023, 32(3): 58. [doi: 10.1145/3550270]

[68] Tang Y, Zhou Y, Wu FH, Liu Y, Sun J, Huang WL, Wang G. Route coverage testing for autonomous vehicles via map modeling. In: Proc. of the 2021 IEEE Int'l Conf. on Robotics and Automation (ICRA). Xi'an: IEEE, 2021. 11450–11456. [doi: 10.1109/ICRA48506.2021.9560890]

[69] Tang Y, Zhou Y, Zhang TW, Wu FH, Liu Y, Wang G. Systematic testing of autonomous driving systems using map topology-based scenario classification. In: Proc. of the 36th IEEE/ACM Int'l Conf. on Automated Software Engineering (ASE). Melbourne: IEEE, 2021.

1342–1346. [doi: 10.1109/ASE51524.2021.9678735]

[70] Li CW, Cheng CH, Sun TT, Chen YH, Yan RJ. ComOpT: Combination and optimization for testing autonomous driving systems. In: Proc. of the 2022 IEEE Int'l Conf. on Robotics and Automation (ICRA). Philadelphia: IEEE, 2022. 7738–7744. [doi: 10.1109/ICRA46639.2022.9811794]

[71] Hildebrandt C, von Stein M, Elbaum S. PhysCov: Physical test coverage for autonomous vehicles. In: Proc. of the 32nd ACM SIGSOFT Int'l Symp. on Software Testing and Analysis. Seattle: ACM, 2023. 449–461. [doi: 10.1145/3597926.3598069]

[72] Wang NF, Luo YP, Sato T, Xu KD, Chen QA. Does physical adversarial example really matter to autonomous driving? Towards system-level effect of adversarial object evasion attack. In: Proc. of the 2023 IEEE/CVF Int'l Conf. on Computer Vision (ICCV). Paris: IEEE, 2023. 4389–4400. [doi: 10.1109/ICCV51070.2023.00407]

[73] Yan C, Xu ZJ, Yin ZY, Ji XY, Xu WY. Rolling colors: Adversarial laser exploits against traffic light recognition. In: Proc. of the 31st USENIX Security Symp. Boston: USENIX Association, 2022. 1957–1974.

[74] Cao YL, Xiao CW, Cyr B, Zhou YM, Park W, Rampazzi S, Chen QA, Fu K, Mao ZM. Adversarial sensor attack on LiDAR-based perception in autonomous driving. In: Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security. London: ACM, 2019. 2267–2281. [doi: 10.1145/3319535.3339815]

[75] Yang KC, Tsai T, Yu HG, Panoff M, Ho TY, Jin YE. Robust roadside physical adversarial attack against deep learning in LiDAR perception modules. In: Proc. of the 2021 ACM Asia Conf. on Computer and Communications Security. ACM, 2021. 349–362. [doi: 10.1145/3433210.3453106]

[76] Cao YL, Wang NF, Xiao CW, Yang DW, Fang J, Yang RG, Chen QA, Liu MY, Li B. Invisible for both camera and LiDAR: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. In: Proc. of the 2021 IEEE Symp. on Security and Privacy (SP). San Francisco: IEEE, 2021. 176–194. [doi: 10.1109/SP40001.2021.00076]

[77] Hallyburton RS, Liu YP, Cao YL, Mao ZM, Pajic M. Security analysis of camera-LiDAR fusion against black-box attacks on autonomous vehicles. In: Proc. of the 31st USENIX Security Symp. Boston: USENIX Association, 2022. 1903–1920.

[78] Boloor A, Garimella K, He X, Gill C, Vorobeychik Y, Zhang X. Attacking vision-based perception in end-to-end autonomous driving models. Journal of Systems Architecture, 2020, 110: 101766. [doi: 10.1016/j.sysarc.2020.101766]

[79] Pavlitskaya S, Ünver S, Zöllner JM. Feasibility and suppression of adversarial patch attacks on end-to-end vehicle control. In: Proc. of the 23rd Int'l Conf. on Intelligent Transportation Systems (ITSC). Rhodes: IEEE, 2020. 1–8. [doi: 10.1109/ITSC45102.2020.9294426]

[80] Wu H, Yunas S, Rowlands S, Ruan WJ, Wahlström J. Adversarial driving: Attacking end-to-end autonomous driving. In: Proc. of the 2023 IEEE Intelligent Vehicles Symp. (IV). Anchorage: IEEE, 2023. 1–7. [doi: 10.1109/IV55152.2023.10186386]

[81] Jha S, Cui SK, Banerjee S, Cyriac J, Tsai T, Kalbarczyk Z, Iyer RK. ML-driven malware that targets AV safety. In: Proc. of the 50th Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks (DSN). Valencia: IEEE, 2020. 113–124. [doi: 10.1109/DSN48063.2020.00030]

[82] Patel N, Krishnamurthy P, Garg S, Khorrami F. Overriding autonomous driving systems using adaptive adversarial billboards. IEEE Trans. on Intelligent Transportation Systems, 2022, 23(8): 11386–11396. [doi: 10.1109/TITS.2021.3103441]

[83] Von Stein M, Shriver D, Elbaum S. DeepManeuver: Adversarial test generation for trajectory manipulation of autonomous vehicles. IEEE Trans. on Software Engineering, 2023, 49(10): 4496–4509. [doi: 10.1109/TSE.2023.3301443]

[84] Hubschneider C, Bauer A, Weber M, Zöllner JM. Adding navigation to the equation: Turning decisions for end-to-end vehicle control. In: Proc. of the 20th IEEE Int'l Conf. on Intelligent Transportation Systems (ITSC). Yokohama: IEEE, 2017. 1–8. [doi: 10.1109/ITSC.2017.8317923]

[85] Codevilla F, Müller M, López A, Koltun V, Dosovitskiy A. End-to-end driving via conditional imitation learning. In: Proc. of the 2018 IEEE Int'l Conf. on Robotics and Automation (ICRA). Brisbane: IEEE, 2018. 4693–4700. [doi: 10.1109/ICRA.2018.8460487]

[86] Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. In: Proc. of the 2016 IEEE European Symp. on Security and Privacy (EuroS&P). Saarbruecken: IEEE, 2016. 372–387. [doi: 10.1109/EuroSP.2016.36]

[87] Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. arXiv:1706.06083, 2019.

[88] Huai YQ, Chen YTY, Almanee S, Ngo T, Liao X, Wan ZW, Chen QA, Garcia J. Doppelgänger test generation for revealing bugs in autonomous driving software. In: Proc. of the 45th IEEE/ACM Int'l Conf. on Software Engineering (ICSE). Melbourne: IEEE, 2023. 2591–2603. [doi: 10.1109/ICSE48619.2023.00216]

[89] Han JC, Zhou ZQ. Metamorphic fuzz testing of autonomous vehicles. In: Proc. of the 42nd IEEE/ACM Int'l Conf. on Software Engineering Workshops. Seoul: ACM, 2020. 380–385. [doi: 10.1145/3387940.3392252]

[90] Zhou JX, Tang SC, Guo Y, Li YF, Xue YX. From collision to verdict: Responsibility attribution for autonomous driving systems testing. In: Proc. of the 34th IEEE Int'l Symp. on Software Reliability Engineering (ISSRE). Florence: IEEE, 2023. 321–332. [doi: 10.1109/ISSRE59848.2023.00062]

[91] Haq FU, Shin DW, Briand L. Efficient online testing for DNN-enabled systems using surrogate-assisted and many-objective optimization. In: Proc. of the 44th Int'l Conf. on Software Engineering. Pittsburgh: ACM, 2022. 811–822. [doi: 10.1145/3510003.3510188]

[92] Giamattei L, Guerriero A, Pietrantuono R, Russo S. Causality-driven testing of autonomous driving systems. ACM Trans. on Software Engineering and Methodology, 2024, 33(3): 74. [doi: 10.1145/3635709]

[93] Deng Y, Zheng X, Zhang MS, Lou GN, Zhang TY. Scenario-based test reduction and prioritization for multi-module autonomous driving systems. In: Proc. of the 30th ACM Joint European Software Engineering Conf. and Symp. on the Foundations of Software Engineering. Singapore: ACM, 2022. 82–93. [doi: 10.1145/3540250.3549152]

[94] Birchler C, Ganz N, Khatiri S, Gambi A, Panichella S. Cost-effective simulation-based test selection in self-driving cars software with SDC-scissor. In: Proc. of the 2022 IEEE Int'l Conf. on Software Analysis, Evolution and Reengineering (SANER). Honolulu: IEEE, 2022. 164–168. [doi: 10.1109/SANER53432.2022.00030]

[95] Birchler C, Khatiri S, Bosshard B, Gambi A, Panichella S. Machine learning-based test selection for simulation-based testing of self-driving cars software. Empirical Software Engineering, 2023, 28(3): 71. [doi: 10.1007/s10664-023-10286-y]

[96] Birchler C, Khatiri S, Derakhshanfar P, Panichella S, Panichella A. Single and multi-objective test cases prioritization for self-driving cars in virtual environments. ACM Trans. on Software Engineering and Methodology, 2023, 32(2): 28. [doi: 10.1145/3533818]

[97] Tahir Z, Alexander R. Coverage based testing for V&V and safety assurance of self-driving autonomous vehicles: A systematic literature review. In: Proc. of the 2020 IEEE Int'l Conf. on Artificial Intelligence Testing (AITest). Oxford: IEEE, 2020. 23–30. [doi: 10.1109/AITEST49225.2020.00011]

[98] Stocco A, Pulfer B, Tonella P. Mind the gap! A study on the transferability of virtual versus physical-world testing of autonomous driving systems. IEEE Trans. on Software Engineering, 2023, 49(4): 1928–1940. [doi: 10.1109/TSE.2022.3202311]

[99] Muktadir GM, Whitehead J. Adversarial jaywalker modeling for simulation-based testing of autonomous vehicle systems. In: Proc. of the 2022 IEEE Intelligent Vehicles Symp. (IV). Aachen: IEEE, 2022. 1697–1702. [doi: 10.1109/IV51971.2022.9827422]

附中文参考文献

[14] 朱向雷, 王海弛, 尤翰墨, 张蔚珩, 张颖异, 刘爽, 陈俊洁, 王赞, 李克秋. 自动驾驶智能系统测试研究综述. 软件学报, 2021, 32(7): 2056–2077. http://www.jos.org.cn/1000-9825/6266.htm [doi: 10.13328/j.cnki.jos.006266]

[16] 戴嘉润, 李忠睿, 张琬琪, 张源, 杨珉. 面向无人驾驶系统的仿真模糊测试: 现状、挑战与展望. 计算机研究与发展, 2023, 60(7): 1433–1447. [doi: 10.7544/issn1000-1239.202330156]

[31] 国家市场监督管理总局, 国家标准化管理委员会. GB/T 43119-2023 自动驾驶封闭测试场地建设技术要求. 2023. https://www.chinesestandard.net/PDF.aspx/GBT43119-2023

作者简介

任睿晗, 博士生, 主要研究领域为自动驾驶系统安全.

杨超, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为智能无人系统安全, 密码工程.

杨凯, 博士生, 主要研究领域为自动驾驶系统安全测试.

张柏迪, 硕士生, 主要研究领域为自动驾驶功能安全测试, 自动驾驶对抗性攻击.

张晓东, 博士, 副教授, CCF 专业会员, 主要研究领域为软件工程, 软件安全性, 形式化方法.

王利娟, 博士, 讲师, 主要研究领域为机器学习, 智能计算, 无人机自主智能控制.

马建峰, 博士, 教授, CCF 会士, 主要研究领域为无线网络安全, 移动智能系统安全.