



本科毕业设计(论文)



题 目: 湖南工商大学学位论文

学生姓名: 罗明贵

学 号: 2123020028

专 业: 人工智能

班 级: 智能 2101 班

指导老师: 王海东 讲师

人工智能与先进计算学院

2024 年 12 月

湖南工商大学本科毕业设计诚信声明

本人郑重声明：所呈交的本科毕业设计是本人在指导老师的指导下，独立进行研究工作所取得的成果，成果不存在知识产权争议，除文中已经注明引用的内容外，本设计不含任何其他个人或集体已经发表或撰写过的作品成果。对本设计做出重要贡献的个人和集体均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名:罗明贵

日期:2024年12月26日

摘要

随着人工智能和自动驾驶技术的迅猛发展，传统的测试与验证方法已无法满足复杂多变的驾驶环境需求。数字孪生技术作为一种新兴的仿真手段，通过构建与现实世界相对应的虚拟模型，为自动驾驶系统提供了高效、安全的测试平台。数字孪生能够实时反映物理实体的状态，支持对车辆动态、环境变化等多方面的精准模拟，从而为自动驾驶算法的训练与优化提供丰富的数据支持。强化学习作为一种自我学习的智能算法，能够在不断变化的环境中优化决策过程。将数字孪生与强化学习相结合，不仅可以加速自动驾驶系统的开发与验证，还能提升其在复杂场景下的适应能力和安全性。研究基于数字孪生的自动驾驶强化学习仿真系统具有重要的理论意义和实际应用价值。

本文旨在探讨基于数字孪生的自动驾驶强化学习仿真系统的设计与实现。随着自动驾驶技术的快速发展，传统的测试与验证方法已无法满足日益复杂的驾驶环境和多样化的驾驶场景需求。数字孪生技术作为一种新兴的仿真手段，通过构建与现实世界相对应的虚拟模型，为自动驾驶系统提供了一个高效、安全的测试平台。本文首先回顾了自动驾驶技术的发展历程，分析了数字孪生的基本概念及其在自动驾驶领域的应用潜力。深入探讨了强化学习的基本原理及其在自动驾驶中的重要性，强调了通过强化学习算法优化自动驾驶决策的必要性。

在此基础上，本文提出了一种结合数字孪生与强化学习的仿真系统框架，详细描述了系统的架构设计、功能模块及实现过程。通过构建一个真实环境的数字孪生模型，系统能够在虚拟环境中进行大量的驾驶场景仿真，进而加速强化学习的训练过程。实验结果表明，该系统在提高自动驾驶决策的准确性与稳定性方面具有显著优势。本文还讨论了系统在实际应用中的挑战与未来发展方向，指出了数据稀缺、收敛性等问题的解决方案，为后续研究提供了参考。基于数字孪生的自动驾驶强化学习仿真系统不仅为自动驾驶技术的验证与优化提供了新的思路，也为相关领域的研究者提供了有价值的实践经验。

关键词：数字孪生 自动驾驶 强化学习 仿真系统 决策优化

ABSTRACT

With the rapid development of artificial intelligence and autonomous driving technology, traditional testing and validation methods can no longer meet the demands of complex and variable driving environments. Digital twin technology, as an emerging simulation method, provides an efficient and safe testing platform for autonomous driving systems by constructing virtual models that correspond to the real world. Digital twins can reflect the state of physical entities in real-time, supporting precise simulations of various aspects such as vehicle dynamics and environmental changes, thereby providing rich data support for the training and optimization of autonomous driving algorithms. Reinforcement learning, as a self-learning intelligent algorithm, can optimize decision-making processes in constantly changing environments. Combining digital twins with reinforcement learning can not only accelerate the development and validation of autonomous driving systems but also enhance their adaptability and safety in complex scenarios. Researching a digital twin-based reinforcement learning simulation system for autonomous driving has significant theoretical significance and practical application value.

This article aims to explore the design and implementation of a digital twin-based reinforcement learning simulation system for autonomous driving. With the rapid development of autonomous driving technology, traditional testing and validation methods can no longer meet the increasingly complex driving environments and diverse driving scenarios. Digital twin technology, as an emerging simulation method, provides an efficient and safe testing platform for autonomous driving systems by constructing virtual models corresponding to the real world. This article first reviews the development history of autonomous driving technology, analyzes the basic concepts of digital twins and their application potential in the field of autonomous driving. It delves into the basic principles of reinforcement learning and its importance in autonomous driving, emphasizing the necessity of optimizing autonomous driving decisions through reinforcement learning algorithms.

On this basis, the article proposes a simulation system framework that combines digital twins and reinforcement learning, detailing the system's architectural design, functional modules, and implementation process. By constructing a digital twin model of a real environment, the system can simulate a large number of driving scenarios in a virtual environment, thereby accelerating the reinforcement learning training process. Experimental results show that the

system has significant advantages in improving the accuracy and stability of autonomous driving decisions. The article also discusses the challenges and future development directions of the system in practical applications, pointing out solutions to issues such as data scarcity and convergence, providing references for subsequent research. The digital twin-based reinforcement learning simulation system for autonomous driving not only offers new ideas for the validation and optimization of autonomous driving technology but also provides valuable practical experience for researchers in related fields.

Key words: Digital Twin Autonomous Driving Reinforcement Learning Simulation System; Decision Optimization

目录

第 1 章 绪论	1
1.1 研究背景及意义	1
1.1.1 自动驾驶技术的发展历程	1
1.1.2 数字孪生技术的概念与应用	2
1.1.3 强化学习在自动驾驶中的重要性用	3
1.2 文献综述	4
1.2.1 国内研究现状	4
1.2.2 国外研究现状	5
第 2 章 深度强化学习理论基础	7
2.1 强化学习模型	7
2.1.1 马尔科夫决策过程	7
2.1.2 价值函数与策略	9
2.1.3 贝尔曼方程	10
2.2 深度神经网络	12
2.2.1 卷积神经网络	12
2.2.2 循环神经网络	14
第 3 章 深度强化学习方法	17
3.1 深度 Q 网络算法	17
3.1.1 深度 Q 网络算法概述	17
3.1.2 深度 Q 网络算法原理	17
3.1.3 深度 Q 网络算法目标网络	18
3.1.4 深度 Q 网络算法经验池	19
3.2 PPO 算法	21
3.2.1 PPO 算法概述	21
3.2.2 PPO 算法原理	21
3.2.3 策略熵	23
3.2.4 优势归一化	24
3.3 SAC 算法	24
3.3.1 SAC 算法概述	24
3.3.2 基于 SAC 的系统效益最大化算法	25
3.3.3 Q 值网络	27

第 4 章 环境搭建	30
4.1 自动驾驶仿真环境搭建与设置	30
4.2 自动驾驶仿真实验装置	34
4.2.1 模型的输入输出配置	34
4.2.2 神经网络参数配置	36
4.2.3 奖励权重设置	38
4.3 基于深度模仿强化学习的车道保持决策模型	40
4.3.1 智能体与环境交互研究	40
4.3.2 智能体与环境交互研究	43
4.3.3 奖励函数设计	45
第 5 章 自动驾驶结果与仿真分析	50
5.1 自动驾驶训练	50
5.2 自动驾驶仿真结果分析	50
第 6 章 总结与展望	54
6.1 总结	54
6.2 展望	55
致谢	56
参考文献	57
附录 A 附录代码	60
A.1 堆溢出检测算法	60
A.2 KMP 算法 C++ 描述	60
附录 B 康托尔辩辞录：数学的自由与制约	63

第1章 绪论

1.1 研究背景及意义

1.1.1 自动驾驶技术的发展历程

作为传统交通工具的一员，汽车为人们的出行带来了极大的便利，随着中国国民经济不断高速发展，国内汽车的生产销售量节节高升。根据中央数据显示，2023年全国机动车保有量达4.35亿辆，其中汽车3.36亿辆。目前机动车驾驶人达5.23亿人，其中汽车驾驶人4.86亿人。汽车保有量的增加给交通带来了一系列挑战，诸如道路拥堵、交通安全以及城市规划等方方面面的挑战。首先，道路拥堵成为了常见现象，这不仅浪费了人们宝贵的时间，还增加了通勤的压力，同时也大大增加了交通事故的风险。另外，交通事故也是一个严重的问题，这导致了人员伤亡和财产损失，专门的研究显示，逾百分之九十的交通事故是由于不当的驾驶操作导致的，包括超速、酒驾、疲劳驾驶、分心驾驶（如使用手机）、违反交通规则等，这些行为降低了驾驶员对道路的控制能力^[1]，增加了事故发生的风险。

然而随着科技的发展，自动驾驶（AD）技术的出现为解决这些问题带来了新的希望和机遇。自动驾驶，又称无人驾驶，是指车辆无需人类引导，能够感知和行驶周围环境，确定最佳行驶路线，顺利到达目的地。自动驾驶技术根据自主程度分为六个级别，从纯人工控制的L0到完全自主控制的L5，如表1-1所示。L4、L5级别表示车辆可以在无需特定条件、无需人工干预的情况下完全自主行驶，代表着最高的技术水平^[2]。

表1-1 自动驾驶等级对照表

等级	名称	定义
L0	人工驾驶	由人类驾驶员完全控制车辆，并负责所有驾驶决策。
L1	辅助驾驶	车辆提供转向或加速/减速的单项支持，但人类驾驶员仍需保持控制。
L2	部分自动驾驶	车辆同时提供转向和加速/减速的支持，但人类驾驶员仍需保持控制。
L3	条件自动驾驶	由车辆完成绝大部分驾驶操作，人类驾驶员需保持注意力集中。
L4	高度自动驾驶	由车辆完成所有驾驶操作，限定在特定道路和环境条件。
L5	全自动驾驶	在任何情况下，车辆都可以在没有人类干预的情况下执行全部驾驶任务。

自动驾驶技术的发展历程可以追溯到20世纪中叶，随着计算机科学、传感器技术以及控制理论的不断进步，自动驾驶逐渐从理论研究走向实际应用。早期的自动驾驶研究主要集中在简单的路径跟踪和基本的环境感知上，这些技术虽然在当时具有一定的前瞻性，但由于计算能力和传感器精度的限制，实际应用效果并不理想。赵岑等指出，随着人工智能和大数据技术的引入，自动驾驶的智能化水平显著提升，使得车辆能够在复杂环境中进行自主决策，从而推动了整个行业的快速发展^[3]。进入21世纪，深度学习

的快速发展为自动驾驶技术带来了新的机遇。基于深度学习的端到端自动驾驶系统能够实现更高效的决策与控制，极大地推动了行业的进步。这种方法通过大规模的数据训练，使得自动驾驶系统能够在多种驾驶场景中进行有效的学习和适应，显著提高了自动驾驶的安全性和可靠性。随着科技的迅猛发展，自动驾驶技术逐渐成为交通运输领域的研究热点^[4]。近年来，数字孪生技术的兴起为自动驾驶系统的开发与优化提供了新的思路和方法。数字孪生是指通过虚拟模型实时反映物理实体的状态和行为，能够在仿真环境中进行多种场景的测试与验证^[5]。这一技术的应用，不仅可以降低实际测试中的风险和成本，还能加速自动驾驶算法的迭代与优化。

在自动驾驶的研发过程中，强化学习作为一种重要的机器学习方法，能够通过与环境的交互不断提升系统的决策能力。传统的强化学习往往依赖于大量的真实数据和复杂的环境模拟，这在实际操作中面临诸多挑战。基于数字孪生的自动驾驶强化学习仿真系统，能够提供一个高度真实且可控的虚拟环境^[6]，使得研究人员可以在不同的交通场景和复杂情况下进行训练，从而提高算法的鲁棒性和适应性。

近年来，自动驾驶技术的应用逐渐扩展到商用领域，通过分析智能自动驾驶无人机的技术进展，为这一领域的创新为传统交通运输方式带来了颠覆性的变革^[7]。无人机的自动驾驶技术不仅提升了物流效率，还在应急救援、环境监测等领域展现了广泛的应用潜力。强化学习作为一种重要的机器学习方法，逐渐被应用于自动驾驶决策中。研究者研究了基于强化学习的无信号交叉口自动驾驶决策^[8]，指出该方法能够有效提高车辆在复杂交通场景中的决策能力，尤其是在动态变化的环境中。通过探讨多源数据在自动驾驶技术风险识别中的应用，突出强调了数据驱动方法在提升自动驾驶安全性方面的重要性。总的来看，自动驾驶技术的发展历程是一个不断融合多学科知识、不断创新的过程，未来将继续朝着更高的智能化和安全性方向迈进，推动交通运输领域的革命性变革。

1.1.2 数字孪生技术的概念与应用

“孪生”的概念起源于美国国家航空航天局的“阿波罗计划”，即构建两个相同的航天飞行器，其中一个发射到太空执行任务，另一个留在地球上用于反映太空中航天器在任务期间的工作状态，从而辅助工程师分析处理太空中出现的紧急事件。当然，这里的两个航天器都是真实存在的物理实体。“数字孪生”初始的概念模型是于2002年10月由迈克尔·格里弗斯（Michael Grieves）博士在美国制造工程协会管理论坛上所提出。而到2009年，美国空军相关实验室明确提出带有数字孪生的概念：“机身数字孪生（Airframe Digital Twin）”。在2010年，美国国家航空航天局（NASA）在《建模、仿真、信息技术和处理》和《材料、结构、机械系统和制造》两份技术路线图中正式开始使用数字孪生（Digital Twin）这一名称。

数字孪生，又称为数字双生、虚拟双生或数据双生，是一种将物理实体在数字空间中进行模拟、复制和优化的技术，数字孪生就是通过数据和算法，将现实世界中的物体、系统或过程进行虚拟化，从而实现对其进行实时监控、预测和优化的目标。数字孪生技术是一种通过虚拟模型与现实世界进行实时交互的创新技术，近年来在多个领域得到了

广泛应用，尤其是在自动驾驶领域^[9]。数字孪生能够有效提升自动驾驶系统的测试与验证效率。通过构建与现实环境相对应的虚拟模型，开发者可以在安全的环境中进行多样化的场景模拟与测试，从而降低实际道路测试所带来的风险和不确定性^[10]。这种技术的应用不仅能够节省时间和成本，还能确保系统在各种复杂情况下的稳定性与可靠性。

数字孪生技术在自动驾驶中的应用，不仅可以优化车辆的动态性能，还能提升环境感知的准确性，从而增强自动驾驶系统的整体安全性^[11]。在复杂的交通环境中，车辆需要实时处理大量信息，数字孪生技术的引入使得这一过程变得更加高效和可靠。同时，得益于物联网、大数据、云计算、人工智能等新一代信息技术的发展，数字孪生的实施已逐渐成为可能。现阶段，除了航空航天领域，数字孪生还被应用于电力、船舶、城市管理、农业、建筑、制造、石油天然气、健康医疗、环境保护等行业，如上图所示。特别是在智能制造领域，数字孪生被认为是一种实现制造信息世界与物理世界交互融合的有效手段。许多著名企业（如空客、洛克希德马丁、西门子等）与组织（如 Gartner、德勤、中国科协智能制造协会）对数字孪生给予了高度重视，并且开始探索基于数字孪生的智能生产新模式。通过利用数字孪生技术进行自动驾驶控制系统的测试，可以大幅降低实际道路测试的风险，确保系统在各种复杂情况下的稳定性与可靠性^[12]。数字孪生与深度学习的结合，为自动驾驶技术的发展提供了新的思路。通过实时数据反馈与模型更新，能够不断优化驾驶决策过程，使得自动驾驶系统在面对复杂交通场景时更加灵活应对^[13]。数字孪生技术在感知决策联合系统中的应用，能够实现对环境的全面理解，从而提升自动驾驶的智能化水平。数字孪生在预测任务中的应用，也能够帮助自动驾驶系统更好地应对复杂的交通场景，提高决策的准确性。

通过进一步探讨了数字孪生在无信号灯十字路口的决策控制中的重要性，可以看出其在复杂交通环境中的应用潜力。数字孪生技术为自动驾驶系统提供了一个动态的反馈机制，使得系统能够在不断变化的环境中保持高效的运行状态。数字孪生技术在自动驾驶领域的应用前景广阔，能够有效提升系统的安全性与智能化水平，为未来的交通系统发展提供了重要的技术支持。

1.1.3 强化学习在自动驾驶中的重要性

在自动驾驶技术的迅猛发展过程中，强化学习作为一种重要的机器学习方法，逐渐展现出其独特的优势与潜力。深度强化学习通过与环境的持续交互，能够有效学习到最优策略，从而在复杂多变的驾驶场景中做出实时决策。这一特性使得自动驾驶系统具备了更高的适应性和灵活性，能够应对各种突发情况和复杂交通状况^[14]。采用端到端的深度强化学习方法，可以显著简化传统自动驾驶系统中的多个模块，提升系统的整体性能，尤其是在动态环境下的表现更为突出。这种方法不仅提高了决策的效率，还降低了系统的复杂性，使得自动驾驶技术的应用更加广泛^[15]。

基于深度强化学习的自动驾驶系统能够通过不断的自我学习和优化，逐步提高决策的准确性和安全性。这种自我学习的能力使得系统能够在真实世界中不断适应新的挑战，增强了其在实际应用中的可靠性。在此背景下，通过探讨了强化学习在自动驾驶算

法中的具体应用，认为其能够有效应对复杂的交通状况，提升车辆的自动驾驶能力，进而推动自动驾驶技术的进一步发展。

并且通过强化学习的训练，自动驾驶系统能够在模拟环境中进行大量实验，积累丰富的经验，从而优化决策过程。这种基于经验的学习方式，使得系统在面对未知环境时，能够更加从容应对。数字孪生技术的引入为强化学习提供了更为真实的仿真环境，使得学习过程更加高效和可靠，为自动驾驶系统的训练提供了坚实的基础^[16]。结合虚拟现实技术与强化学习，可以进一步提升自动驾驶系统的训练效果，为未来的实际应用奠定坚实的基础。这些研究表明，强化学习在自动驾驶中的重要性不仅体现在提升决策能力上，更在于为系统的持续优化和适应复杂环境提供了强有力的支持。

1.2 文献综述

1.2.1 国内研究现状

近年来，随着自动驾驶技术的迅猛发展，国内学者对数字孪生在自动驾驶领域的应用进行了广泛而深入的研究。基于数字孪生的汽车自动驾驶仿真测试方法能够有效提升测试的安全性与效率，为自动驾驶系统的验证提供了新的思路和方法。这种方法不仅能够模拟真实的驾驶环境，还能在虚拟空间中进行多种场景的测试，从而降低实际测试中的风险和成本^[17]。通过探讨数字孪生的虚拟仿真系统，强调其在复杂驾驶环境下的应用潜力，认为数字孪生技术能够为自动驾驶提供更为真实的测试环境，进而提升系统的可靠性和适应性^[18]。学者李佳新等人在其研究中探讨了深度强化学习在自动驾驶决策中的应用，提出了一种新型的决策框架，能够有效提高自动驾驶系统的智能化水平。通过关注于端到端免模型深度强化学习的应用，强调了该方法在复杂交通环境中的优势，能够实现更为灵活的驾驶策略。

通过研究深度学习与深度强化学习结合的关键技术，我们可以发现这种结合能够显著提升自动驾驶系统的感知与决策能力。学者何竞等人提出了一种基于深度强化学习的智能决策算法，强调了算法在动态环境下的适应性和实时性。探讨了自主无人系统的驾驶策略，提出了一种新的强化学习框架，能够有效应对复杂的驾驶场景。在研究中分析深度强化学习在自动驾驶决策控制中的应用，提出了一种基于模型的强化学习方法，能够提高决策的准确性和安全性。学者李文娜^[8]等研究了自动驾驶汽车闯红灯预警的数字孪生道路测试，强调了数字孪生技术在提升自动驾驶安全性方面的重要作用。仿真测试在自动驾驶系统开发中至关重要，能够有效降低开发成本和风险。国内研究者通过综述自动驾驶汽车感知系统的仿真研究，指出数字孪生技术在感知系统优化中的关键作用。通过研究自动驾驶汽车在闯红灯情况下的预警机制，提出数字孪生技术进行道路测试可以显著提高系统的反应能力和决策准确性。这一研究为自动驾驶系统的安全性提供了重要的理论支持和实践依据。分析仿真测试在自动驾驶系统开发中的重要性，认为通过仿真可以有效降低开发成本和时间，提高系统的可靠性和稳定性，进而加速自动驾驶技术的落地与应用^[19]。

国内研究者通过综述自动驾驶汽车感知系统的仿真研究，指出数字孪生技术在感知

系统优化中的关键作用，能够提升自动驾驶的环境感知能力和反应速度，为自动驾驶的智能化发展奠定了基础^[19]。国内研究者研究设计了一种基于转鼓/制动试验平台的自动驾驶整车虚拟仿真测试系统^[20]，强调了数字孪生在整车测试中的应用价值，展示了其在实际工程中的可行性和有效性。数字孪生技术正在汽车行业加速应用，尤其是在智能工厂和自动驾驶领域，展现出广阔前景和发展潜力^[21]。数字孪生技术在自动驾驶测试领域的应用研究正逐渐深入，为未来的技术发展奠定了坚实的基础。国内在数字孪生与自动驾驶结合的研究上已取得了一定的进展，但仍需进一步探索与实践，以应对日益复杂的驾驶环境和技术挑战。

1.2.2 国外研究现状

在国外，自动驾驶技术的研究与应用已经取得了显著进展，尤其是在数字孪生和强化学习的结合方面。许多知名科技公司和研究机构积极投入资源，探索如何利用数字孪生技术来提升自动驾驶系统的性能和安全性。特斯拉、谷歌的 Waymo 以及 Uber 等公司，均在其自动驾驶平台中应用了先进的仿真技术，以应对复杂的交通环境和多样化的驾驶场景。除此以外，许多国际知名高校和研究机构，如麻省理工学院、斯坦福大学和卡内基梅隆大学等，均在这一领域开展了深入的研究。数字孪生技术的引入，使得研究者能够在虚拟环境中创建与现实世界高度一致的模型，从而为自动驾驶系统的开发和测试提供了强有力的支持。

在数字孪生的研究中，国外学者提出了多种构建与应用模型。MIT 的研究团队开发了一种基于数字孪生的城市交通仿真平台，能够实时反映城市交通流量和车辆行为。这一平台不仅为自动驾驶算法的测试提供了真实的环境数据，还为城市交通管理提供了决策支持。斯坦福大学的研究者们也在探索数字孪生在自动驾驶中的应用，重点关注如何通过高保真度的仿真环境来优化车辆的决策过程。

世界上第一辆无人驾驶汽车“Linrrican Wonder”于 1925 年在美国纽约问世。2004 年至 2007 年，美国连续举办了三届 DARPA（国防高级研究计划局）无人驾驶挑战赛，标志着自动驾驶时代的开始。该比赛的目的是推动极端环境下自动驾驶汽车技术的发展，参赛队伍包括众多高校和高科技企业，主要采用的技术涉及人工智能、计算机技术、汽车设计等诸多高科技技术。每届比赛都对自动驾驶汽车技术的发展起到了很大的推动作用。2004 年，首届 DARPA 挑战赛在美国哈维沙漠举行，虽然参赛队伍均未能完成比赛规定的任务，但这是车辆首次能够在自动驾驶的同时实现避障，这极大地激发了大家对自动驾驶技术的热情，增强了自动驾驶领域的创新意识，具有里程碑式的意义。与第一届 DARPA 相比，2005 年的第二届 DARPA 虽然车辆搭载了大量的传感器，但部分队伍已采用线控技术控制车辆，这是一个重大进步。这次挑战赛形成了自动驾驶汽车的雏形，标志着自动驾驶汽车的功能基本完备。最终，五辆赛车完成了全部比赛任务，证明了自动驾驶是可能的。2007 年，DARPA 城市挑战赛在加州一个废弃的空军基地举行。与前两届比赛相比，本届比赛最大的亮点在于自动驾驶汽车不仅要避让其他机动车辆，还要遵守道路交通规则，这对车辆的感知系统和决策能力是一个巨大的挑战。

在智能汽车领域，谷歌于 2009 年开始无人驾驶汽车项目研究。2012 年，Google Auto 获得美国历史上第一张自动驾驶牌照，由内华达州颁发。2014 年，谷歌宣布其自动驾驶汽车可以应对数千个城市的道路交通，并发布了第三代自动驾驶汽车，这是一款谷歌自主研发的纯电动自动驾驶汽车，没有传统的刹车、方向盘和油门。2016 年 5 月，谷歌与克莱斯勒汽车公司开始合作，这是谷歌与汽车公司的首次合作，该车配备了全套传感器、信息处理计算单元等系统。同年 12 月，谷歌成立了其无人驾驶公司 Waymo，并向外界展示了一款自动驾驶概念车，该车顶部装有激光雷达和摄像头装置，前后保险杠上安装了传感器。截至 2016 年 3 月，谷歌的自动驾驶里程已达 241 万公里，全程仅发生 14 起交通事故，其中 13 起为对方失误^[22]。截至 2017 年底，谷歌的自动驾驶汽车测试里程已达 141.6 公里。根据其测试报告，谷歌 Waymo 每千英里（约 227.4 万公里）仅需人工干预 0.18 次，相当于每 9000 公里约需 1 次人工干预。排名第二的通用汽车 Cruise 则需人工干预 0.84 次。

其他国际知名汽车公司，如宝马、奔驰、奥迪、特斯拉等，都在进行自动驾驶汽车的研究和测试，各公司之间也正在进行相应的合作和战略部署。为促进智能汽车健康规范发展，美国于 2016 年 9 月发布了《联邦自动驾驶汽车政策》，这是全球首个自动驾驶领域的政策文件^[23]。

强化学习在自动驾驶领域的应用同样备受关注。国外的研究者们通过设计复杂的奖励机制和状态空间，推动了强化学习算法在自动驾驶决策中的应用。加州大学伯克利分校的研究小组提出了一种基于深度强化学习的自动驾驶模型，通过与数字孪生环境的交互，显著提升了车辆在复杂场景下的自主决策能力。该模型在多种仿真测试中表现出色，展示了强化学习在动态环境中的适应性和灵活性。国外的研究者们还通过构建复杂的仿真环境，利用深度强化学习算法来优化自动驾驶决策。DeepMind 和 OpenAI 等机构在强化学习算法的创新上取得了显著成果，他们的研究不仅推动了人工智能的发展，也为自动驾驶技术的进步提供了新的思路。许多企业，如特斯拉、谷歌的 Waymo 和 Uber 等，纷纷投入巨资进行自动驾驶技术的研发，利用数字孪生技术进行实时数据分析和模型优化，以提高自动驾驶系统的安全性和可靠性。

国外的研究还强调了多智能体系统在自动驾驶中的重要性。通过数字孪生技术，研究者能够模拟多个自动驾驶车辆在同一环境中的交互行为，从而为强化学习算法提供更为丰富的训练数据。值得注意的是，国外的研究还注重多智能体系统的协同学习，通过模拟多个自动驾驶车辆在复杂交通环境中的互动，探索如何提高整体交通效率和安全性。这些研究不仅为自动驾驶技术的实际应用提供了理论基础，也为未来的智能交通系统奠定了坚实的基础。国外在基于数字孪生的自动驾驶强化学习仿真系统的研究中，已经取得了显著的进展，值得我们深入学习和借鉴。这种多智能体的仿真环境不仅提高了算法的鲁棒性，还为未来的智能交通系统提供了新的思路。国外在数字孪生与强化学习结合的自动驾驶研究中，已形成了一定的理论基础和实践经验，为推动自动驾驶技术的进一步发展奠定了坚实的基础。

第2章 深度强化学习理论基础

2.1 强化学习模型

在强化学习（RL）的领域，它也被称为增强学习或再励学习。该领域的研究主要集中在查看代理与环境之间的交互。在这样做时，代理人选择由环境提供的有关代理行为的信息，代理根据由环境提供的行为信息来做出决策并更好地学习。这是代理与环境之间最大的交互。其本质是代理与环境交互以创建动作并实施策略。选择这些行动和方法的依据是代理根据环境数据评估系统的状态和性能。强化学习算法无需对问题进行建模或熟悉环境。重点是数据研究，以确保代理和环境之间有更好的互动。良好的数据可以让代理的行动和方法也得到优化和改进。在激励培训的背景下，了解代理与了解一个人的学习过程非常相似。代理通过与环境的互动获得适当的奖励，然后通过结合上述奖励和惩罚，代理可以获得适当的奖励。通过反复尝试和不断学习，代理可以逐渐找到最合适的表现方式，从而实现全局最优。这种试错学习的过程使得代理能够逐步提高其在特定环境中的表现和适应能力。

2.1.1 马尔科夫决策过程

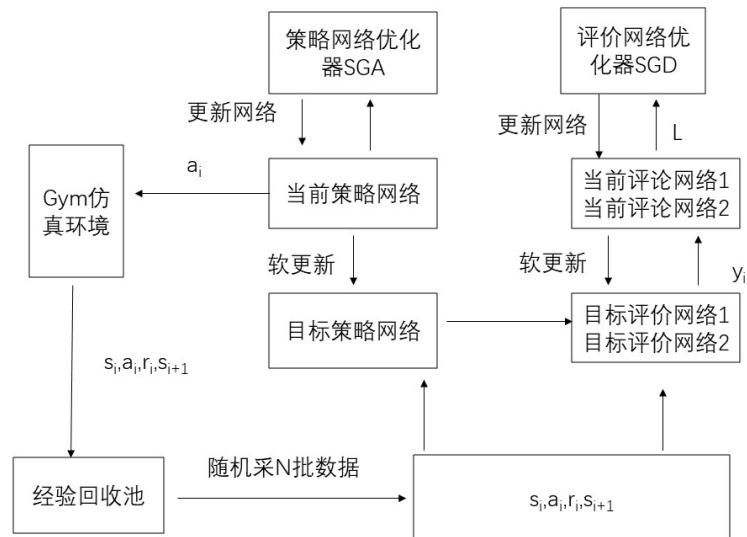


图 2-1 基于 MDP 的强化学习结构

如果一个系统的状态信息中包含了大量的历史信息，并且不需要提供任何信息，就可以通过结合当前状态来推断出未来的状态，那么这样的状态特性通常被称为马尔可夫特性。结合前面提到的代理与环境的紧耦合，本质上环境的状态是与当前动作紧密相关

的，与情境紧密相关的。根据强化学习问题，可以创建马尔可夫决策过程（MDP）^[24]。详细流程如上图 2-1 所示。

智能体获取状态信息，把上述信息当成是输入，结合上述信息，科学选择动作，完成输出。结合动作结果得到相应的奖励，进一步得到未来动作。所以，结合马尔科夫决策过程 $\langle S, A, P, R, r \rangle$ 对这一过程进行界定与表述。

首先，需要定义状态。状态是代理接收到的信息，它会对其下一步行动的选择产生一定的影响。状态集群 $S = s_1, s_2, \dots, s_t$ 可以由一个或多个动态层组成，通常处理传感器检测到的信息。动作空间 $A = a_1, a_2, \dots, a_t$ 表示代理可以选择的动作集合，即它在当前动作中可以选择的所有动作。转移概率表示从当前状态到下一个状态的动作的可能性。如果满足马尔可夫条件，转移概率可以表示为：

$$P(s_{t+1} | s_t, a_t) = P(s_{t+1} | s_t, a_t, s_{t-1}, a_{t-1}, \dots)$$

奖励函数是一个特定的标量函数，表示在给定状态下应用环境动作后获得的回报值。奖励函数的设计基于状态信息，并影响代理算法参数的评估和优化。折扣因子（也称为折现率）用于加权影响当前和未来回报的因素，从而可以全面分析未来的影响因素。

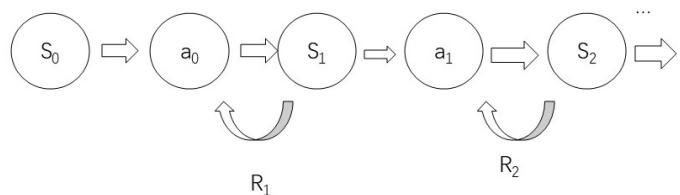


图 2-2 马尔可夫动态过程

MDP 动态传输方案如上图 2-2 所示。提供者根据有关当前状态的信息选择操作，然后将操作返回给环境。环境做出反应，创造新的情况并创造有价值的回报。随着代理不断与环境交互，学习持续进行，直到到达最后阶段。代理与环境交互以执行最佳策略并最大化总奖励。MDP 最重要的概念之一是折扣率。它代表未来收益的折现值，即两个不同时间（两个）状态下未来利润的价值之差。

然而，MDP 通信模型基于一个隐含的假设，即代理可以完全访问所有可用信息。然而，在现实情况下，例如对于坦克决策，智能代理可以使用传感器监视环境，因此它无法获取完整的信息并且无法应用 MDP 假设。当代理观察环境时，噪音可能会分散注意力，因此接收到的信息只是实际环境信息的一部分。

本文假设转移概率收敛于马尔可夫过程，这意味着未来状态仅描述当前状态和采取的行动，而不是过去的状态。然而，在高级场景中，传感器提供的有限信息意味着我们无法获得无限的状态信息。因此，从状态空间到存储单元的遍历方式有多种，这使得存储无法满足马尔可夫链的条件。

为了解决这个问题，研究人员提出了几种方法。例如，Shani^[25]等人开发了一个全面的马尔可夫过程来解释代理和环境之间产生的所有相互信息，假设最初的观察和行动满足马尔可夫特性。

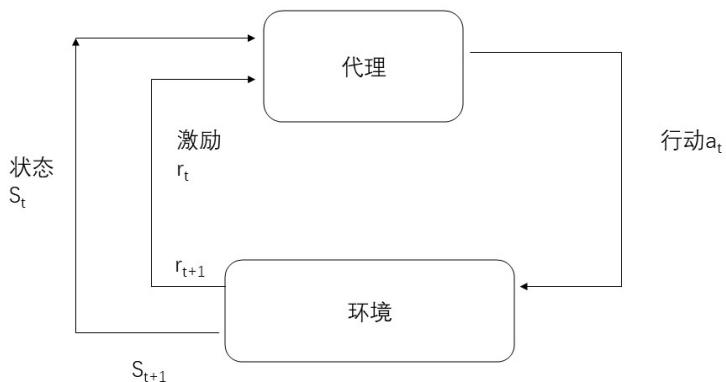


图 2-3 基于 POMDP 的强化学习结构

上图 2-3 给出的是环境同代理交互循环关系，收到观察结果以后代理通过采取动作，同环境之间迭代交互，但代理并非了解真实状态信息。

2.1.2 价值函数与策略

强化学习有两个基本组成部分：财务绩效和策略。两者在算法的设计和优化中都发挥着重要作用。虽然奖励的当前值可以提供有关当前事态的信息，但它仅反映即时奖励，而不能完全预测当前决策的未来结果。因此，衡量一个行动的利弊，必须考虑其后果。在强化学习算法中，成本函数是用于确定给定情况下奖励的预期值的重要参数。这一思想通常以两个函数的形式实现：交易价值函数（Q 函数）和状态价值函数（V 函数）。后者通常表示为从当前状态到结束状态的期望奖励值，而效用价值函数描述的是执行某项

任务后从当前状态到结束状态的奖励的期望值。通过这两项任务，操作员可以更深入地探索环境中的各种物体和行为。

基于这些定义，可以理解状态 V 的值对代理从当前状态到最终状态所获得的平均工资有直接的影响。当代理做出决策时，他或她倾向于选择能够带来更多价值的任务，以找到最佳的奖励平衡。因此，通过改进成本函数，算法可以在决策过程中变得更加智能和高效，从而提高生产力和项目完成速度。

给定一个状态，就会生成相关的策略约束，并且可以使用独立策略（例如，接近高斯分布）来选择所有点的约束动作。在给定场景中，用户以任意概率分布选择最佳选项。RL 的目标是找到使所有人的利润最大化的最优解，如下式所示：

$$\begin{aligned} G_t &= R(s_t, a_t) + \gamma R(s_{t+1}, a_{t+1}) + \gamma^2 R(s_{t+2}, a_{t+2}) + \dots + \gamma^k R(s_{t+k+1}, a_{t+k+1}) \\ &= \sum_{k=0}^{\infty} \gamma^k R(s_{t+k+1}, a_{t+k+1}) \end{aligned}$$

2.1.3 贝尔曼方程

为了改进学习数据，状态值和交换率函数是贝尔曼方程的解，可以将其分解为各种问题，并可以作为方程获得良好的解^[26]。计算采用两点，旨在达到理想的平衡。

(1) 贝尔曼期望方程

$$\begin{aligned} V_{\pi}(s) &= E_{\pi}[G_{t+1} \mid s_t = s] \\ &= \mathbb{E}_x[R_{t+1}(s_t, a_t) + \gamma R_{t+2}(s_{t+1}, a_{t+1}) + \gamma^2 R_{t+3}(s_{t+2}, a_{t+2}) + \dots \mid s_t = s] \\ &= \mathbb{E}_{\pi}[R_{t+1}(s_t, a_t) + \gamma (R_{t+2}(s_{t+1}, a_{t+1}) + \gamma R_{t+3}(s_{t+2}, a_{t+2}) + \dots) \mid s_t = s] \\ &= \mathbb{E}_{\pi}[R_{t+1}(s_t, a_t) + \gamma V(s_{t+1}) \mid s_t = s] \end{aligned}$$

根据上述公式，对 V 进行分解处理以后：当前环境下的立即回报值和下一时刻状态值函数可以被我们清晰的看到。

继续进行同样分解 Q ，可以得到下式：

$$\begin{aligned} \underline{Q}_{\pi}(s) &= \underline{E}[G_{t+1} \mid S_t = s, A_t = a] \\ &= \underline{E}[R_{t+1}(s_t, a_t) + \gamma \underline{Q}_{\pi}(s_{t+1}) \mid S_t = s, A_t = a] \end{aligned}$$

上述表达式中，由于策略 π 的影响，可以计算出所有可能性的对应状态值函数的值的为 Q 值与 π 的乘积的和。如下式所示：

$$V_{\pi}(s) = \sum_{a \in A} \pi(a|s) Q_{\pi}(s, a) = \mathbb{E}_{a \sim \pi(a|s)} [Q_{\pi}(s, a)]$$

类似的还有下式：

$$Q_\pi(s, a) = R(s, a) + \gamma \sum_{s' \in S} p(s' | s, a) V_\pi(s')$$

(2) 贝尔曼最优方程

此方程是针对两大值函数的内在联系的全面表述，目的是从中挑选出策略对应的最大值函数，即为最优值函数 $V^*(s)$ ：

$$V^*(s) = \arg \max_{\pi} V_\pi(s), \quad s \in S$$

同理可得到，最优动作值函数 $Q^*(s|a)$ ：

$$Q^*(s, a) = \arg \max_{\pi} Q_\pi(s, a), \quad s \in S$$

依据上述两式可得到 $V^*(s)$ 和 $Q^*(s, a)$ 的直接关系式：

$$\begin{aligned} V^*(s) &= \arg \max_a Q^*(s, a) \\ Q^*(s, a) &= R(s, a) + \gamma \sum_{s' \in S} p(s' | s, a) V^*(s') \end{aligned}$$

在离散情况下，设置强化学习任务是以有限状态集为依托的，能够获取不同状态 s 对应的 $V(s)$ 函数，或 (s, a) 对应的 $Q(s, a)$ 函数。通过计算出 $V(s)$ ，实施更新迭代，结合策略，确定值函数，进而进行评估。再结合上述式子便可以推导出下列式子：

$$\begin{aligned} V_\pi(s) &= \sum_{a \in A} \pi(a | s) Q_\pi(s, a) \\ &= \sum_{a \in A} \pi(a | s) \left(R(s, a) + \gamma \sum_{s' \in S} P(s' | s, a) V_\pi(s') \right) \end{aligned}$$

针对 MDP 问题，求出值函数，获取最佳控制策略，必定能够得到最优策略。想要获取最大策略 π^* ，利用最大的 $Q^*(s|a)$ 函数，可以获取最优策略：

$$\begin{aligned} \pi^*(a | s) &= \arg \max_{a \in A} Q^*(s, a) \\ &= \arg \max_{a \in A} \left(R(s, a) + \gamma \sum_{s' \in S} p(s' | s, a) V^*(s') \right) \end{aligned}$$

2.2 深度神经网络

在神经网络中，深度神经网络 (DNN) 系统是一个复杂的多层次系统，其中许多节点具有多个神经元，通过连接权重传输和处理信息。图 2-4 显示了一个典型的深度神经网络 (DNN) 架构，它由几个隐藏层和一个输出层组成。每个隐藏层和派生层都包含若干个节点，每个节点都与前一层的所有节点相连，并有一定的权重和偏移量。

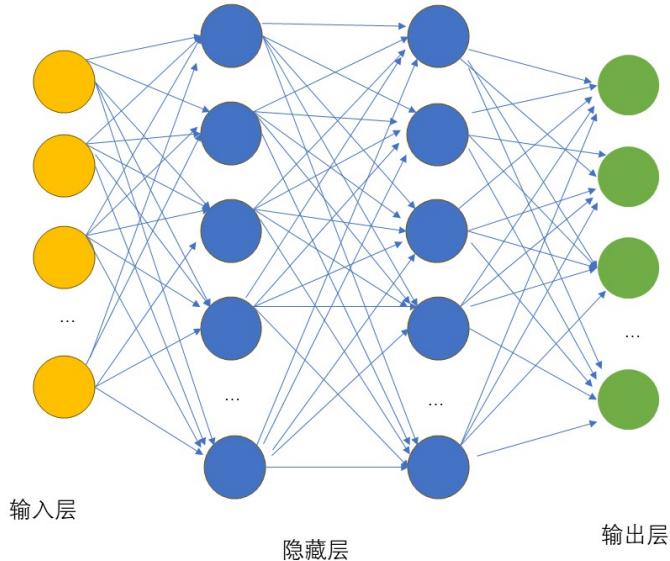


图 2-4 深度神经网络结构

对于高阶神经网络，权重是归一化的，变分参数也是归一化的。然而，损失函数也代表了相应的误差。一般来说，线性回归问题基于平方误差，平方误差被视为损失函数，解释为：

$$\text{loss}(Y, \hat{Y}) = \frac{1}{N} \sum_{i=1}^N (\hat{Y}_i - Y_i)^2$$

其中，训练样本数为 N，预测值为 \hat{Y} ，实际值为 Y_i 。在训练过程中，输入数据可以来回传递以获得所有层的输出数据。这代表了分层网络中神经元的误差。

在深度神经网络的训练过程中，反向传播算法扮演着至关重要的角色。一旦通过前向传播获得了各层的输出，就可以利用反向传播来计算每一层神经元的误差，从而进一步计算关于权重和偏置参数的梯度。

2.2.1 卷积神经网络

在 CNN 中，卷积提取图像的局部特征，并且卷积可以降低特征图的维度，从而可以减少特征的数量和网络的大小。此外，知识库也得到了显著改善。同时，在网络中采

用权重分配的方法可以显著减少参数数量，显著增加模型的泛化效率，显著提高模型的学习效率。

总体而言，该算法的使用提高了深度学习在成像任务中的表现。该模型可用于大规模图像数据的处理。此外，计算机视觉领域取得了长足的进步，并且发展迅速。

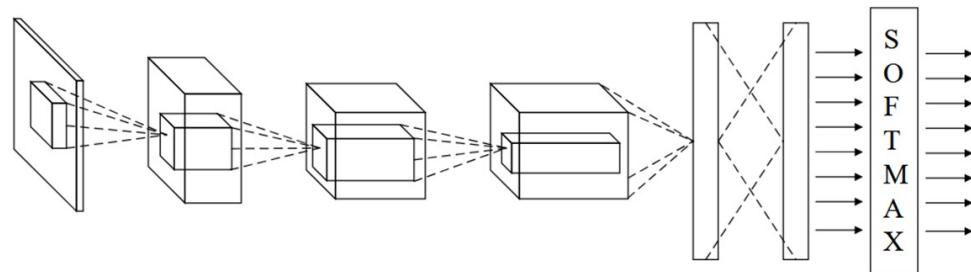


图 2-5 图像分类卷积神经网络结构示意图

图 2-5 展示了卷积神经网络的基本结构，它由多个卷积层和池化层组成，对输入图像的特征提取和降维起着关键作用。降低卷积特征图的维度可以减少参数数量并降低计算复杂度，同时保留图像的重要特征。

经过一系列折叠和汇总操作后，特征图被发送到全连接层，该层组合每个特征图中的特征信息，并使用非线性函数对其进行操作，以捕获特征之间的复杂交互。全连接层的输出被送到 Softmax 层，Softmax 层使用似然法对结果进行处理，得到各个类的概率分布，从而深入了解输入图像的分类。此类网络结构具有强大的特征提取和分类能力，在图像分析、目标检测等任务中能够取得优异的效果。在训练过程中，反向传播算法可以让网络自动学习特征的最佳表示，从而对图像进行准确的分类。在卷积神经网络中，常见的输入信号是 RGB 图像，其中每个像素包含来自三个通道的信息：红色、绿色和蓝色。

例如，VGG-16 模型^[27]使用的输入数据是 $224 \times 224 \times 3$ 的 RGB 图像。输入数据一旦送入网络，就必须经过多层卷积和池化来降低其维度。在这个例子中，卷积层由两条路径组成：一条是具有多层的卷积层，另一条是最大池化层。一般情况下每个卷积层的卷积核大小为 3×3 ，经过一个大的合并层进行特征提取，减少特征图尺寸。执行这些操作允许网络从输入图像中提取局部重要特征。在卷积过程中，每个卷积核对输入图像进行加权和局部空间的组合，得到神经元的输入。为了显著提高交换网络的非线性能力，应

考虑采用更高的输入值作为非线性激活函数，其中 ReLU 激活函数被广泛使用。这样就可以得到功能图所有区域的输出值。聚合方法采用 2×2 的最大聚合，即选取每个局部 2×2 区域内出现的最大值，在保留重要性的前提下，显著降低了数据对象的维度。

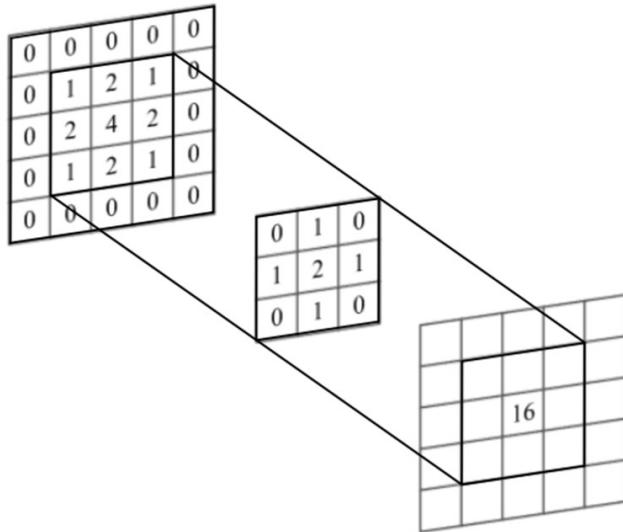


图 2-6 卷积计算示意图

在 2016 年，国内学者何凯明总结出 ResNet 残差网络^[28]，在这之中，将残差单位加入进来，不使用其它参数时，可以实现前向反馈神经网络训练。这一单元结构具体见下图 2-7。

在执行语义分割等任务时，卷积神经网络需要处理大量像素级的数据，这使得网络变得复杂且深层。虽然这些深度学习方法可以提供更好的性能和准确的结果，但它们在训练时间和估计方面会带来很大的开销。特别是在实时应用和资源受限的环境中，例如移动设备或嵌入式系统，深度神经网络的设计需要特别注意网络复杂性和计算约束。大型网络将无法在这些设备上运行，也无法满足实际需求。因此，在构建深度神经网络的过程中，除了选择合适规模的网络之外，还需要考虑网络深度。适当的深度可以在保持效率的同时，减少间隙数量和计算复杂度，从而提高网络的效率和效果。这将需要网格修剪、参数共享和模型压缩等技术来减少网格尺寸而不牺牲性能。

2.2.2 循环神经网络

循环神经网络 (RNN)^[29] 可用于避免在某些会话中运行这些任务时出现非目标计算。它主要由多层循环神经网络组成，如下图所示。通过求解连续序列中的单元，可以得到一个线性神经网络的结构。数据从前向后流动，单元运行后得到的结果用作下一个单元的输入。线性迭代操作按此顺序执行，结构如图 2-8 所示。

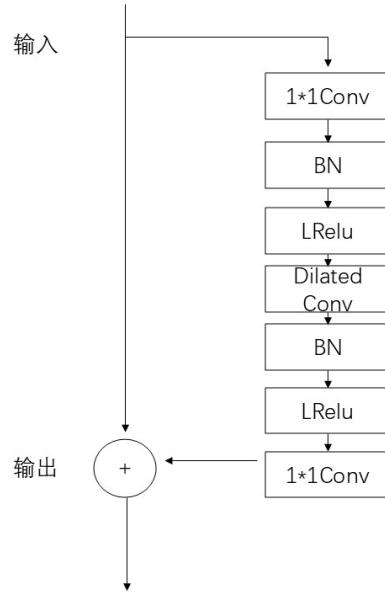


图 2-7 残差单元

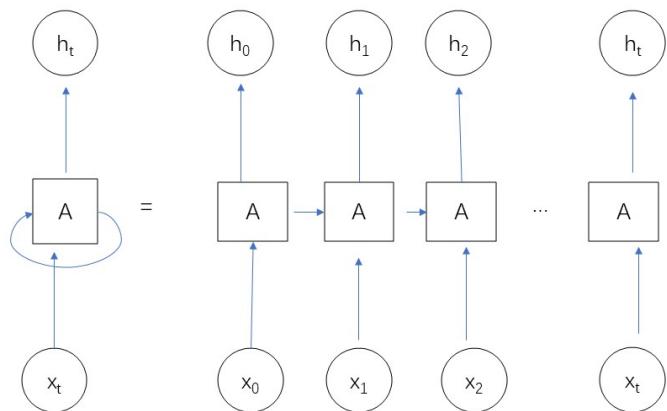


图 2-8 基础循环神经网络单元

在循环神经网络的实际实现中，也会出现一些难以有效避免的问题。例如，连接数量的不断增加会导致神经网络的增长，并且这种网络的增长无法得到有效的抑制。因此，它在深度联想学习中的作用尚未完全了解。LSTM^[30]的出现为解决这一问题提供了新的思路。它由三部分组成：输入阀、输出阀和忘记阀。这三个门的加入，使得我们可以忘记不需要记住的信息，保留需要记住的信息，有效地解决了信息长期保留的问题。网络

结构如图 2-9 所示。

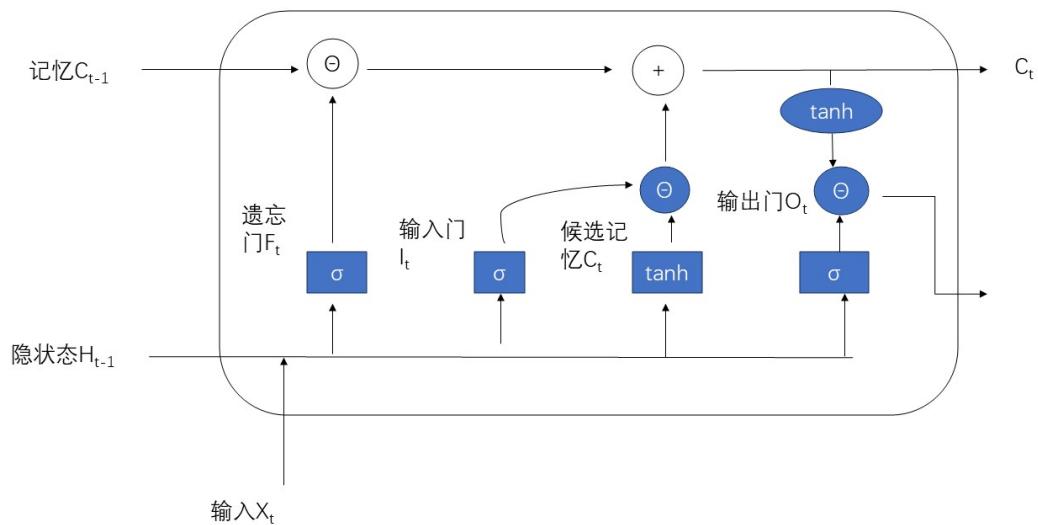


图 2-9 长短时记忆网络

第 3 章 深度强化学习方法

3.1 深度 Q 网络算法

3.1.1 深度 Q 网络算法概述

深度 Q 网络 (DQN)^[31]是由 DeepMind 首次在生命科学领域发布的网络架构。它的出现引发了人们的思考，让人们了解到一种新型算法的用处，那就是强化学习。Q 学习算法通过更新值表来更新函数。当动作和状态连续时，动作的空间显著扩大，状态值的空间也不断扩大。在这种情况下，Q 学习算法不再使用。DQN 算法的出现优雅地解决了上述问题，并建立了 Q 学习算法与卷积神经网络之间的首次紧密联系。它逐渐成为深度学习中应用最广泛的强化学习算法之一。

3.1.2 深度 Q 网络算法原理

在 Q 测量算法中，激活值是一种描述，观测值也是一种类似的描述。性能指标不同。Q 可以执行值的记忆功能，并更新值。然而，当代理与环境之间存在信息冲突或多个层次时。Q 值不再完全满足衡量价值的要求，使得平等成为一种诅咒。为了解决这个问题，开发了 DQN 算法，可以有效地对人进行分类。在这项研究中，DQN 参与者认为基于游戏的讲故事是环境学习和发展的重要方法。例如，Atari 游戏的最大屏幕尺寸为 210x160，每个像素有三个通道。这样，每个状态的维度为 $210 \times 160 \times 3$ 。显然，Q-questioning 算法在处理此类数据时不起作用。

这里的“网络”是指利用神经网络来估计非线性函数的 Q 值。该神经网络的参数为：表示每一层的 Q 权重值。图 3-1 所示的神经网络由两个全连接层和三个卷积层组成。DQN 中神经网络的结构如图 3-1 所示。

DQN 是一种基于表的新算法。根据算法和变量类型的不同，可分为蒙特卡洛方法和变分方法两类。对应是目标函数为：

$$\arg \min_{\theta} (Q(s, a) - \hat{Q}(s, a, \theta))^2 \quad (3.1)$$

在 DQN 方法中，代价函数往往采用参数的方式来实现。当参数改变时，一方面影响 Q 值的结果，另一方面也会影响其他值函数。该配置主要基于 SGD^[32]。设定一个科学的目标来更新和最小化损失函数的值。这里，损失函数本质上是一个量化模型执行与学习过程分布之间差异的函数。在解决实际问题时，不同的问题结构会导致不同形式的功能损失。一般来说，这个问题可以定义为回归问题或分类问题。在回归问题中，最常见的损失函数是 L2 损失函数（平方误差损失）和 L1 损失函数。L2 损失函数对与观测值有较大偏差的结果施加较大的惩罚，从而确定结果与真实值的偏差程度；而 L1 损失函数估计的是真实值和一个比较保守的估计值之间的差值，通常用绝对值来计算。损失函数定义如图表 3-1 所示。

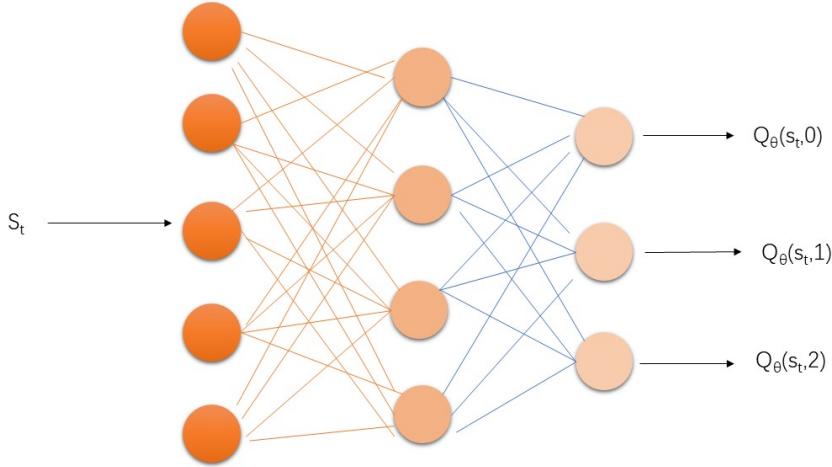


图 3-1 DQN 的深度神经网络结构

表 3-1 损失函数定义

回归问题	分类问题	名称
$L_1(y, \hat{y}) = w(\theta) \hat{y} - y $	$L(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$	交叉熵损失函数
$L_2(y, \hat{y}) = w(\theta)(\hat{y} - y)^2$	$L(y, \hat{y}) = \exp(-\hat{y}y)$	指数损失函数
	$L(y, \hat{y}) = \max(0, 1 - \hat{y}y)$	铰链损失函数

在解决此类回归问题时，主要目标是尽量减少实际结果和估计结果之间的差异。为了实现这一目标，神经网络必须具有足够的样本量并且足够稳健。通过训练，模型可以优化可以提高预测精度的重要参数。对于 DQN 算法，将问题集的误差函数与实现收敛的目标相结合，即最小化实际值和预测值之间的差异。第一个判断是，并且这样的网络的损失函数 Q 如下列式子所示：

$$L(\theta) = E \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta) - Q(s, a; \theta) \right)^2 \right] \quad (3.2)$$

3.1.3 深度 Q 网络算法目标网络

用上面的参数更新公式，网络参数是用来梯度更新和 TD 目标计算。虽然这个方法可以区分两个构建的数据和检测参数的组合，但它无法有效地分离参数，可能会影响网络的训练性能，而且，上一代参数的固定会导致后拟合过程的延迟，进一步增加过度拟合，导致训练误差，使得难以获得稳定的模型。

DeepMind 等人在他们的研究中关注数据通信问题，并结合了两个网络概念：目标

网络和主网络。从结构上来说，两者还是同一个东西；从更新频率来看，两者还是有很大区别的。在第一个训练阶段，两个网络的参数相同。随着训练的进行，网络逐渐适应环境并获得更多样本值。目标网络不断接收 TD 值，并利用公式通过主网络获取实时更新，进而得到状态值 Q 。主网络经过多轮迭代数据处理后，将参数转发给目标网络，与目标网络进行同步。

当选择随机梯度下降法时，首先要解决的问题是样本不独立。但 DQN 实验建模本质上依赖于马尔可夫关系，样本不能独立均匀分布。为了解决这个问题，需要添加某种测试池，可以打破数据链并确保数据均匀分布，其更新方式如下列式子所示。该算法的示例如下图 3-2 所示。

$$\theta_{l+1} = \theta_l + \alpha \left[r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right] \nabla Q(s, a; \theta) \quad (3.3)$$

$$\theta^- \leftarrow \theta \quad (3.4)$$

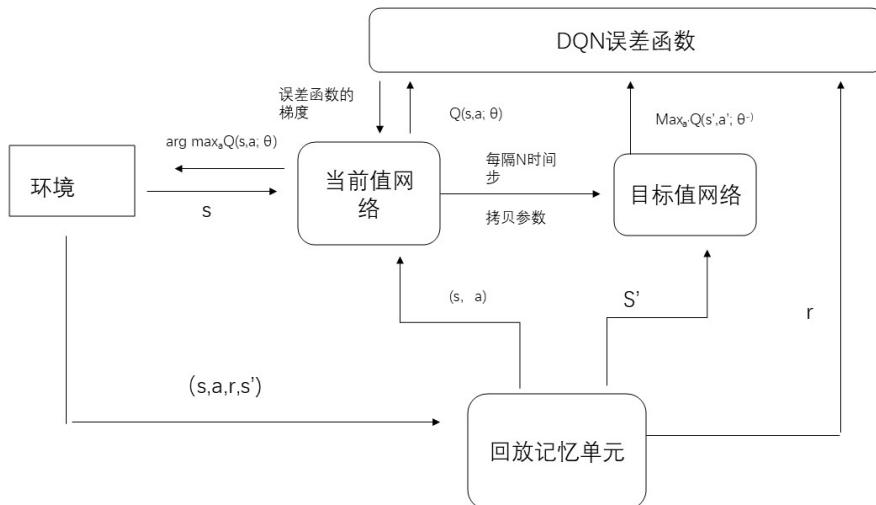


图 3-2 DQN 算法模型训练过程

3.1.4 深度 Q 网络算法经验池

DQN 系统旨在使用内存设备来存储数据，通常称为内存缓存^[33]。记忆装置能够储存相关信息，并解决前文提到的情境信息问题，从而满足认知障碍的标准。数据存储流程如下图 3-3 所示。

按照图 3-3 所示的工作流程，知识获取在 Agent 与环境的交互中起着重要作用。所有模板脚本都存储基本的关系信息。由于数据量巨大，观察者集群的存储容量必须足够

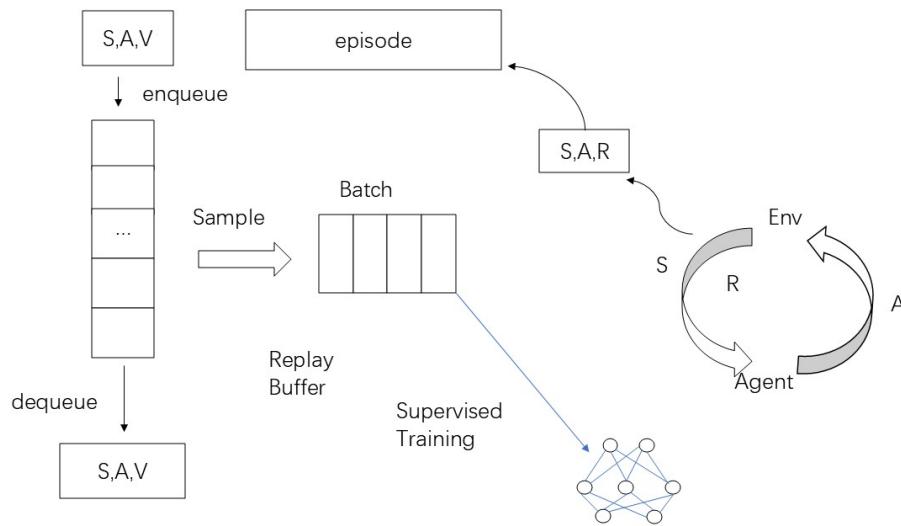


图 3-3 基于经验池机制的采样流程

大，才能持续收集和存储数据。在设计参与平台时，需要考虑许多因素，例如存储管理、配置和可用性。为了高效地存储结构化数据并将其存储在存储中，浏览器使用字符串或哈希表等结构化数据。集成系统使他们能够快速输入、移除和检索原材料以满足实时需求。

有两种看待它的方式。一个是曝光选择，另一个是曝光选择。在协作数据收集中，数据通常根据按时间顺序发生的事件进行收集。一旦大脑正确地记住了信息，它就会被覆盖，从而准确地再现之前呈现的信息。证据提取过程利用各种方法从大量事件中提取证据，然后进行检查和重建。此类操作可以提高您网站的性能。在上一节中，我们表明了智能游戏环境中交互得到的序列具有明确的时间关系，而任何通过空间学习得到的价值函数交互序列只能指向当前执行对应的下一条路径，而不能指向所有当前路径。在这种情况下，如果网站没有跟上变化，它的外观就会与预期的外观产生差异。此外，随着协作学习的发展，方差将不断增加，这将对模型学习产生负面影响，例如增加模型的可变性并降低不变性。

DQN 工具是一个综合工具，可以对大样本进行控制分析。与传统的基于表格的教学方法相比，这种方法在许多方面都有了显著的改进。这种方法最严重的限制是它不能用于连续分析。同时，当以神经学方式使用该药剂时，不可能立即实现习惯化。此外，社交媒体的变革问题也是一个亟待解决的问题。本节描述的 DQN 模型为后续发展的 DDPG 模型奠定了基础，也为本文所述后续实验的成功实施提供了理论基础。

3.2 PPO 算法

3.2.1 PPO 算法概述

PPO（近端策略优化）是一种强化学习的优化算法。它是由 OpenAI 于 2017 年提出的，主要用于解决策略梯度方法中学习不稳定、采样效率低的问题。PPO 方案的主要思想是限制每个迭代周期内系统更新的次数，避免系统发生过多的变化，从而使学习变得稳定。PPO 算法在信任体系中也具有易于实现、综合性更强等优势^{schulman2017proxima}。由于 PPO 能够优化长期累积奖励，这使得它在需要进行长期规划和决策的高速公路驾驶场景下表现出色。司机需要考虑他们的长期目标，例如在高速公路上超速行驶以及右转或保持在右车道上行驶。此外，PPO 系统是一个基于网络的学习系统，可以快速适应高速公路交通状况并提供实时反馈。PPO 算法相对于 PolicyGradient 算法和 TRPO（Trust Region Policy Optimization）算法有几个优点。数据选择过程和目标函数优化过程交替进行。虽然标准策略梯度方法需要更新每个数据样本的梯度，但 PPO 提供了一个新的目标函数，可以对小组数据样本执行更新。相比传统的策略梯度，PPO 可以更高效地利用采样；与更复杂的选项 TRPO 相比，PPO 更容易实现，且具有类似的效果^[34]。

PPIO（公共策略优化）算法基于两个主要算法：策略搜索算法和宏观优化算法。通过有意义的测试，PPO 可以重复使用过去的方法生成的数据来指导新方法，从而提高测试性能。剪枝技术通过减少更新策略来提高学习准确率，避免由于策略参数的突然变化而导致性能的提高。

3.2.2 PPO 算法原理

PPO 算法是一种在线策略算法，主要解决的是连续动作空间中高维度决策问题。它使用裁剪的概率比率（clipped probability ratio）来限制每次策略网络参数更新的幅度，使得新策略优于旧策略。PPO 算法的更新步骤如下^[35]：

(1) 采集样本数据，根据当前策略 $\pi(a|s)$ 执行若干次模拟，记录状态、动作、奖励和下一个状态。

(2) 计算优势函数 A_t ，计算公式如下：

$$Q_t(s_t, a_t) = \sum_{s'} P(s' | s_t, a_t) [R(s_t, a_t, s') + \gamma V_t(s')] \quad (3.5)$$

$$V_t(s_t) = \max_a Q_t(s_t, a) \quad (3.6)$$

$$A_t = Q_t(s_t, a_t) - V_t(s_t) \quad (3.7)$$

其中， $Q_t(s_t, a_t)$ 是 Q 函数（动作价值函数），它表示在时间 t 采取动作 a_t 后从状态 s_t 开始的预期回报，其中 $P(s' | s_t, a_t)$ 表示从状态 s_t 采取动作 a_t 转移到状态 s' 的概率， $R(s_t, a_t, s')$ 是对应的奖励函数， γ 是折扣因子， $V_t(s')$ 是状态 s' 的价值函数； $V_t(s_t)$ 是价值函数，即在时刻 t 处于状态 s_t 的最大预期回报，通过最大化所有可能动作 a 的 Q 函数获得；优势函数 A_t 表示采取动作 a_t 相对于平均动作在状态 s_t 的额外回报。

(3) 计算策略比率 $ratios$, 即新策略在状态 s_t 下选择动作 a_t 的概率与旧策略在状态 s_t 下选择动作 a_t 的概率的比值, 计算公式如下:

$$ratio_s = \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_k}(a_t | s_t)} \quad (3.8)$$

(4) 计算目标函数, 使用剪裁函数 $clip(ratios, 1 - \epsilon, 1 + \epsilon)$, 将 $ratios$ 的值限制在 $[1 - \epsilon, 1 + \epsilon]$ 的范围内, 用于限制策略更新的幅度, 根据公式上述计算如下:

$$L(s, a, \theta_k, \theta) = \min (ratios \times A_t, clip (ratios, 1 - \epsilon, 1 + \epsilon) A_t) \quad (3.9)$$

θ 是当前策略参数, θ_k 是旧策略的参数; ϵ 是超参数, 指新策略和旧策略之间的更新幅度。

(5) 通过优化目标函数来获得下一步策略参数 $\theta_k + 1$, 使用梯度下降法, 公式见下:

$$\theta_{k+1} = \arg \max_{\theta} \mathbb{E}_{s_a \sim \pi_{\theta_k}} [L(s, a, \theta_k, \theta)] \quad (3.10)$$

上面步骤中, 步骤一的初始策略 $\pi(a|s)$ 就显得非常重要, 强化学习算法的初始策略一般都是随机策略, 因为它可以提供更广泛的探索范围。

PPO 算法的具体结构图^[36]如图 3-4 所示。

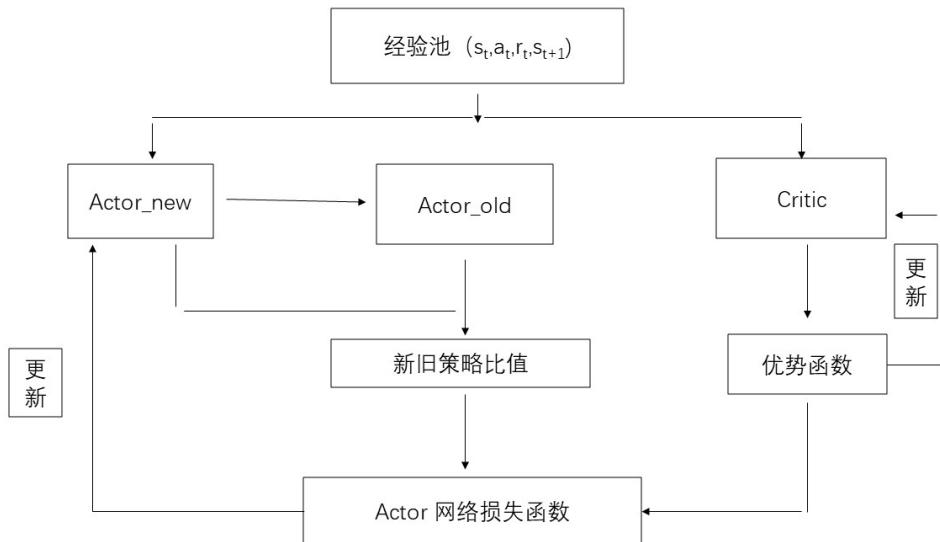


图 3-4 PPO 算法结构图

图 3-4 中的红色箭头表示前向传播模式的变化, 蓝色箭头表示后向传播模式的改进。如图 3-4 所示, PPO 层次结构由两个参与者列表 ($actor_{new}$ 和 $actor_{old}$) 和动态流程组成。首先, 在每个阶段, 行动者网络根据当前状态 s_t 生成一个效用分布, 然后代理根据这个效用分布做出贡献。然后状态根据代理每次动作为代理提供奖励值 r_t , 移动到下

一个状态 $s_t + 1$, 并生成数据序列 $[s_t, a_t, r_t, s_t + 1]$, 存储在经验池中。一旦数据存储在特征池中, 基于估计方法的类大小就会被传输并输入到 $Actor_{new}$ 和 $Critic$ 系统中进行训练。通过定期重复此过程, 员工可以学习正确的技术。 $Actor_{new}$ 模块通过减少分解函数(包括新旧路径的速率以及检测函数)来更新其参数, 而 $Critic$ 模块通过发现函数来调整其参数, $Actor_{old}$ 模块通过不断复制 $Actor_{new}$ 模块的参数来调整其参数。

3.2.3 策略熵

熵是衡量物体变化速率的指标。随着通道熵的增加, 旧通道和自由基分布出现分歧。为了计算汽车值, 我们通过计算熵值^[37]来计算值的熵值, 并计算值的熵系数(通常为 0.01)。

$$H(\pi(\cdot | S_t)) = - \sum_{a_i} \pi(a_i | S_t) \log \pi(a_i | S_t) \quad (3.11)$$

$$= \mathbb{E}_{a_i \sim \pi} \left[-\log \pi(a_i | S_t) \right] \quad (3.12)$$

$$\sim e^{i\pi - kh_1 + \dots} \quad (3.13)$$

在强化学习中, 策略熵的引入是一种有效的探索增强机制。连续作用的空间由积分描述。熵值越大, 政策行动的分布越公平。因此, 代理对不同行动的选择是任意的, 而不是决定性的。在 Actor-Critic 架构中, 加入政治熵作为 Actor 损失函数的先导, 可以显著提高探索的效率。具体实现形式为:

$$H(\pi(\cdot | S_t)) = - \sum_{a_i} \pi(a_i | S_t) \log \pi(a_i | S_t) \quad (3.14)$$

$$= \mathbb{E}_{a_i \sim \pi} \left[-\log \pi(a_i | S_t) \right] \quad (3.15)$$

$$\sim e^{i\pi - kh_1 + \dots} \quad (3.16)$$

其中 L_{policy} 为原始策略梯度损失(如 PPO 的剪切目标函数), λ 为熵系数(即 $entropy_{coef}$, 通常设为 0.01)。这种设计通过以下机制发挥作用:

(1) 平衡探索与利用:

熵正则化阶段不仅寻求策略优化中的更高回报(通过 L_{policy}), 而且还鼓励代理尝试更少的剥削行动。例如, 在 USV 路径规划任务中, 如果过程倾向于快速改进进化过程, 则负的时间熵梯度将导致局部最优过程的概率分布发生倾斜, 从而避免局部最优。

(2) 自适应探索强度:

训练开始时, 系统熵增加, $agent$ 主动搜索环境; 随着训练的进行, 系统逐渐变得更加互联, 熵自然减少, 并且代理朝着使用更高奖励行动的方向发展。与固定次数的试验(如 $\epsilon-greed$)相比, 熵正则化具有更灵活的适应性, 可以适应复杂的任务要求。

(3) 实现简化与兼容性：熵的计算直接基于现有策略的运行概率分布，不需要额外的环境干预或复杂的采样，并且易于与现有算法（例如 PPO 和 SAC）集成。例如，在 PPO 的 Actor 更新中，仅需修改损失函数为：

$$L^{\text{CLIP+Entropy}}(\theta) = \mathbb{E}_t [\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \varepsilon, 1 + \varepsilon) A_t)] + \lambda \mathcal{H}(\pi_\theta) \quad (3.17)$$

3.2.4 优势归一化

提高战略规划与当前环境条件的相关性；在进行车辆供应规划时，应利用广义需求估计（GAE）计算统计样本的需求函数，然后对整个细分市场的需求曲线进行标准化。归一化程序如下：首先，将网络中所有节点的初始聚合值居中对齐（平均值），并带有一个标准差（标准差）；然后对每个需求价格进行标准变换，即减去平均单价，再除以单位标准差。这种扩展使得政治阶层体系更加稳定，消除了不同政党之间的阶层偏好差异，并增加了模型对当前行政分配形状的敏感性。初始最优值能清晰地显示出项目成本的优势与劣势；因此，要优化生产系统参数，需要根据实际环境条件，最终根据海况的变化，对项目进度进行优化。

$$A^* = \frac{A - \mu}{\sigma} \quad (3.18)$$

3.3 SAC 算法

3.3.1 SAC 算法概述

简单 Actor-Site 算法 (SAC)^[38]是一种基于最大熵原理的强大的深度学习算法，是 Actor-Critic (AC) 框架的重要发展。该算法在优化模型中引入单一策略熵作为正则表达式，在考虑多个预期奖励时最小化策略熵，从而在探索与探索之间建立强有力的平衡。设计帮助智能代理积极探索环境中的机会，同时避免因过度依赖历史经验而陷入局部最佳实践，并仍然追求长期利益。

基于其所构建的算法，SAC 保留了“actor-critic”框架的基本思想，但引入了几项重要的创新。其关键电路采用两个 Q 决定因素 (Q1 和 Q2)，通过最小化两个独立比较器电路的误差来减少过冲问题，类似于双延迟 DDPG 设计 (TD3)。具体来说，SAC 实现了一种熵策略控制机制，通过温度参数 α 随机改变策略模式。在传统的强化学习算法中，当 α 趋近于零时，它会减小，而随着 α 值的增加，策略变化会增加。这种设计有助于算法根据环境特征调整搜索强度。

SAC 利用各种技术创新来实现高质量和一致的学习。首先，为了满足学习连续函数的需要，该算法采用迭代方法将随机变量 $\pi(s|\theta)$ 除以不同噪声检测函数的乘积。具体来说，策略通道输出动作方向 μ 和正态对数偏差 σ ，并根据公式 $\mu + \sigma \cdot \varepsilon(\varepsilon \sim N(0, 1))$ 生成最终动作。该设计不仅遵循实用策略，还强调平滑的渐变过渡。其次，该算法采用经验检

索的方法，将智能体与环境交互产生的过程存储在经验池中，并通过随机排序来阻断数据之间的时序关系，提高了数据的利用效率和持续学习的能力。

3.3.2 基于 SAC 的系统效益最大化算法

从策略优化角度，SAC 被训练来优化最优策略效用函数 $J(\pi) = E[r(s, a)] + \alpha H(\pi|s)$ ，其中 $H(\pi)$ 表示策略熵。通过逐步扩大参与者网络来提高实际性能，同时通过最小化 Q 分数和目标值之间的小误差来更新关键路径。值得注意的是，SAC 使用一种简单的更新机制来逐步调整输入到网络的参数，以避免因突变而导致的学习不稳定性。此外，该算法采用“延迟策略回归”策略，在关键路径完全训练后更新参与者网络，进一步提高学习过程的稳定性。

与传统学习算法相比，连续 SAC 控制表现出明显的性能优势。最大熵方法自然适合于具有多模态奖励或强激励的环境，而重新参数化方法和基于经验的回归方法的组合有效地解决了寻找继续行动的地方的困难。研究表明，该算法在 MuJoCo 模拟环境中表现良好，特别是在涉及细化的任务中，其模型和最终性能优于其他强调相同时间的学习算法。这一特性在需要严格研究的实际应用中非常重要，例如机器人控制和自动车辆控制。SAC 算法中策略的优化目标可表示为：

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \mathbb{E}_{(s_t, a_t) \sim p_{\pi}} \left[\sum_{t=0}^T \gamma^t (r(s_t, a_t, s_{t+1}) \right. \quad (3.19)$$

$$\left. + \alpha H(\pi(\cdot | s_t)) \right] \quad (3.20)$$

其中： $\gamma \in (0, 1)$ 为折扣率，其值直接反映代理人对长期回报的预期。当 γ 接近 1 时，该算法会优先考虑未来的奖励而不是当前的奖励。需要进行长期规划，例如工业机器人的持续运行或多阶段的投资决策。相反，如果 γ 较小，代理很可能立即收到响应。这种方法在需要现实世界变化或环境条件发生重大变化的场景中特别有用。值得注意的是， γ 的取值本质上体现的是人类在不确定性条件下“选择时间”的行为，而数字的选择往往需要考虑某些领域的先验知识进行优化。

$\alpha \in [0, 1]$ 作为温度的函数，这在工资函数和政策强度之间建立了一个新的权衡模型。从控制理论的角度来看，这个参数本质上是探索与开发权衡的驱动因素：随着 α 值的增加，策略的熵权重增加，理性结构会寻求更昂贵但成本更低且潜在风险更高的决策分支。多学科护理变得越来越重要。例如，自动驾驶系统必须应对所有意外的交通状况。然而，当 α 趋近于 0 时，算法会切换到传统的强化学习方法。在这些情况下，即使技术安全性得到提高，环境中的隐藏模式也可能无法被揭示。有趣的是，由于 α 是基于学习的，SAC 算法可以根据环境的复杂性自动调整搜索强度，这比 ϵ -greedy 等默认搜索方法具有显著的优势。

与算法的基本公式一样，策略熵 $H(\pi(\cdot | s))$ 的物理概念比简单的知识概念度量要复杂得多。从系统理论的角度来看，熵最大化本质上是在功能空间中构建一个一致封闭的

“可接受搜索空间”结构。该机制允许代理在状态空间中的每个决策节点生成平均不确定性。因此，当面临具有多种可能解决方案的问题时（例如，规划合作机器人运动路径），高熵策略可以避免过快收敛到局部最优解，而是可以通过创造性探索对剩余条件有局部的了解。该策略包括添加一个缓冲层来防止干扰。当面临环境变化时（例如机械臂负载的突然变化），分布式策略函数可以提供多种解决方案，显著提高系统的整体性能。值得注意的是，SAC 通过将熵作为独立模块添加到局部目标活动中，实现了检测过程与策略优化的无缝集成，这比外部噪声活动检测方法具有更大的理论吸引力和实际实现性。其中： $\gamma \in (0, 1)$ 为折扣率，其值直接反映代理人对长期回报的预期。当 γ 接近 1 时，该算法会优先考虑未来的奖励而不是当前的奖励。需要进行长期规划，例如工业机器人的持续运行或多阶段的投资决策。相反，如果 γ 较小，代理很可能立即收到响应。这种方法在需要现实世界变化或环境条件发生重大变化的场景中特别有用。值得注意的是， γ 的取值本质上体现的是人类在不确定性条件下“选择时间”的行为，而数字的选择往往需要考虑某些领域的先验知识进行优化。

SAC 算法中的状态价值函数 $V^\pi(s)$ 根据下式得到：

$$\overline{V^\pi(s)} = \mathbb{E}_{(s_t, a_t) \sim p_\pi} \left[\sum_{t=0}^T \gamma^t (r(s_t, a_t, s_{t+1}) \right. \quad (3.21)$$

$$\left. + \alpha H(\pi(\cdot | s_t)) \right) \Big| s_0 = s \quad (3.22)$$

同时，动作状态价值函数 $Q_\pi(s, a)$ 根据下式得到：

$$Q^\pi(s, a) = \mathbb{E}_{(s_t, a_t) \sim p_\pi} \left[\sum_{t=0}^T \gamma^t (r(s_t, a_t, s_{t+1}) \right. \quad (3.23)$$

$$\left. + \alpha \sum_{k=1}^T \gamma^k H(\pi(\cdot | s_k)) \right) \Big| s_0 = s, a_0 = a \quad (3.23)$$

据以上分析， $V^\pi(s)$ 和 $Q^\pi(s, a)$ 的关系可由下式表示：

$$V^\pi(s) = \mathbb{E}_{z_t \sim p} \left[\mathbb{E}_{a_t \sim \pi} [Q^\pi(s_t, a_t)] + \alpha H(\pi(\cdot | s_t)) \right] \quad (3.24)$$

$Q^\pi(s, a)$ 的贝尔曼期望方程也可以由下式表示：

$$Q^\pi(s, a) = \mathbb{E}_{s_{i+1} \sim p, a_{i+1} \sim \pi} \left[r(s_i, a_i, s_{i+1}) + \gamma (Q^\pi(s_{i+1}, a_{i+1}) \right. \\ \left. + \alpha H(\pi(\cdot | s_{i+1})) \right) \Big] \\ = \mathbb{E}_{s_{i+1} \sim p} [r(s_i, a_i, s_{i+1}) + \gamma V^\pi(s_{i+1})] \quad (3.25)$$

本研究基于针对上述问题开发的马尔可夫决策过程 (MDP) 模型，提出了基于软演员评论家 (SAC) 的视频流服务优化方法 SAC-UNCO。该算法通过高度强化的学习过程，可以更好地决策终端设备和外围服务器之间的动态视频任务传输。网络结构如图 2 所示。具体来说，SAC-UNCO 采用五层网络架构，包括一个策略网络 (actor network)、两个独立的 Q 值估计网络 (critic 1 和 critic 2) 以及两个匹配网络 (target critic 1 和 target critic 2)。双组分 Q 值方案通过降低两个测试组合的最大值来消除 Q 值估计过高的问题，同时，对所提出的组合进行简单的优化 (Polak 平均) 通过减少对组合参数的跟踪频率有效地提高了学习方法的稳定性。

为了在解决复杂问题的同时平衡传统动态研究中的研究和应用，SAC-UNCO 引入了自动训练的 α 温度。该参数根据策略网络参数通过梯度下降来动态调整策略熵权重，从而客观地控制策略网络的鲁棒性。具体而言，温度系数 α 的优化目标函数可表示为：

$$L_\alpha = -\mathbb{E}_{\pi_\phi} [\alpha \log \pi_\phi(a | s)] + \beta \cdot \text{regularizer} \quad (3.26)$$

前者被鼓励利用政治熵进行研究，而后的典型约束则防止 α 增长过快导致政策分歧。自动温度控制使得算法能够完全根据环境因素调整搜索力度，而无需输入搜索参数，大大提高了算法适应复杂视频提取场景的能力。

在训练过程中，SAC-UNCO 使用档案重放来存储历史交互数据并锁定跨模型数据之间的时间相关性，从而提高模型的可用性。规则网络利用重参数化的方法对不同任务中执行的随机动作进行变换，从而利用反向传播的方法优化规则系数。同时，双拟合优度网络利用损失误差函数方程，通过减少与目标值 Q 的差异来调整网络参数。其中目标 Q 值由贝尔曼方程定义并融入策略熵项：

$$y = r + \gamma \cdot \min_{i=1,2} Q_{\theta_i}(s', \pi_\phi(s')) - \alpha \log \pi_\phi(a | s') \quad (3.27)$$

该过程允许进行成本效益分析，以评估即时回报和长期利益，以及比较重复的矿物开采机会。实验表明，SAC-UNCO 优于传统的视频性能测试方法。它的动态分析可以管理网络流量和计算资源的路由，同时降低速度和能耗并提高性能。算法中的 Q 值网络、策略网络和温度系数自动调节的细节如图 3-5。

3.3.3 Q 值网络

为了正确估计强化学习中的状态和动作价值函数 $Q(s, a)$ ，本研究采用了参数为 β_i 的神经网络。具体来说，对于每个状态-动作对 (s, a) ，其 Q 值可以表示为神经网络的输出 $Q(\beta_i)(s, a)$ ，其中 $i \in [1, 2]$ 表示神经网络的两个独立参数。该设计包括用于实时策略评估和目标计算的两层网络结构，从而解决了传统单网络方法中常见的高估问题。

在优化参数时，该算法使用均方误差 (MSE) 损失函数作为主要指标。对于不考虑网络的情况，损失函数 $L(\beta_i)$ 由下式定义：

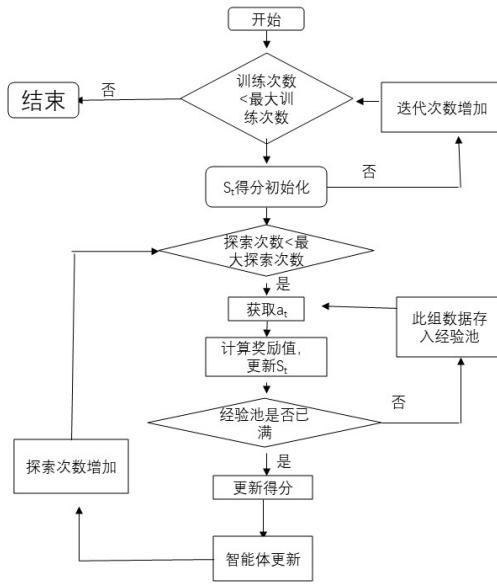


图 3-5 SAC 算法训练架构

$$J_Q(\beta_i) = \mathbb{E}_{(s_i, a_i, s_{i+1}) \sim D} \left[\frac{1}{2} \left(Q_{\beta_i}(s_i, a_i) - \hat{Q}(s_i, a_i) \right)^2 \right] \quad (3.28)$$

在 SAC-UNCO 算法中，状态-动作价值函数 $Q^\pi(s_t, a_t)$ 的更新过程遵循改进的贝尔曼方程，其表达式为：

$$Q^\pi(s_t, a_t) = r(s_t, a_t) + \gamma \left(Q_{\beta_i}(s_{t+1}, a_{t+1}) - \alpha \log \pi_\theta(a_{t+1} | s_{t+1}) \right) \quad (3.29)$$

这里 $r(s_t, a_t)$ 表示代理在时间 t 执行动作后的即时奖励， $\gamma \in (0, 1)$ 是用于衡量未来奖励对当前决策影响的折扣因子。具体来说，该公式使用了基于传统贝尔曼方程的 $-\alpha \log \pi_\theta(a_t + 1 | s_t + 1)$ 阶熵项，其中 α 是可测量的温度系数， π_θ 是限制于 θ 的晶格阶。熵概念的引入包含了研究熵的最大能量的基本思想。通过明确鼓励策略中的随机性，鼓励代理探索环境中的机会，同时增加其积累的奖励。这提高了算法适应动态情况的能力。

经验丰富的 D 重复池作为算法的学习点。它不断收集代理与环境交互产生的轨迹数据 $s_t \square a_t \square r_t \square s_{t+1}$ ，并通过均匀随机采样来打破数据之间的时间联系。这种在线学习方法有效地降低了模型之间的相关性，避免了由多个相似模型引起的梯度漂移问题，并通过对数据的重新处理大大提高了建模效率。值得注意的是，池中存储的值 $Q(\beta_i)(s_t + 1, a_t + 1)$ 不是由当前网络直接计算的，而是由目标 Q 值网络近似确定的。扩展参数 β_i 的更新按照软更新规则（公式（3.30））进行。具体来说，采用指数移动平均策略逐步更新目标网络参数： $\beta'_i \leftarrow \tau \beta'_i + (1 - \tau) \beta_i (\tau \ll 1, \tau \ll 1)$ 。这种延迟更新机制有效地解决了 Q 值估计与目标值波动耦合的问题，为策略优化提供了稳定的参考点。

$$\overline{\beta}_i = \tau \beta_i + (1 - \tau) \overline{\beta}_i \quad (3.30)$$

根据式 (3.29) 可以得到 Q 值的梯度 $\nabla_{\beta_i} J_Q(\beta_i)$ 表示为:

$$\nabla_{\beta_i} J_Q(\beta_i) = \nabla_{\beta_i} Q_{\beta_i}(s_t, a_t) \left(\widetilde{Q_{\beta_i}(s_t, a_t)} - \hat{Q}(s, a) \right) \quad (3.31)$$

因此, Q 值网络在时隙 t 的网络参数 β_i 可以根据式 (3.32) 更新

$$\beta_i^{t+1} = \beta_i^t - \lambda_\beta \nabla_{\beta_i} J_Q(\beta_i) \quad (3.32)$$

其中: λ_β 是 Q 值网络的学习率。

第 4 章 环境搭建

4.1 自动驾驶仿真环境搭建与设置

该仿真平台用于测试基于复杂规划任务的决策算法的性能。在算法训练过程中，我们基于车辆生产进行了深入研究。如果在真实道路上进行测试，发生车祸的概率很高，这将导致道路基础设施受损和重大物质损失。因此，在实际研究中，需要高度逼真地模拟重型机械的显示画面，添加从各种实验中收集的数据作为输入数据，并将真实路况的相关信息输入到仿真平台中，从而构建仿真环境。最后，添加油门、转向和刹车等车辆相关数据，完成整体仿真。

该模拟器支持各种复杂的交通场景。科学家可以使用可视化工具动态更改输入和输出参数，并将来自多模态传感器（例如激光雷达点云、摄像头图像和毫米波雷达信号）的数据导入系统，以实现高效的数据集成和闭环验证。以 CARLA 模拟器为例：该平台基于虚幻引擎 4 构建，并使用基于物理的渲染技术 PBR 来实现逼真的光影效果。动态天气系统模拟了雨、雪、雾等恶劣天气条件。它以时区控制的昼夜循环运行，为测试算法提供了多维环境。核心引擎采用 C++ 编写，并与 Python API 兼容，允许开发人员以最少的代码实现深度定制，同时快速调用更高级别的接口来检索车辆状态、道路拓扑和位置数据。

CARLA^[39]模拟器使用 OpenDrive 标准分析真实道路的几何参数和交通规则，并将高分辨率地图数据转换为可编程的虚拟场景。其模块化架构使研究人员能够描述车辆动力学模型（例如轮胎摩擦系数、传动效率）、传感器噪声特性（例如雷达检测误差模型）和道路使用者行为（例如行人接近概率）的独立性。尤其是在复杂军事设施的建设中，该平台的军事设施细节库可用于创建特殊区域，例如指挥中心和装甲救护车停车场。结合可定制的通信干扰模型和电磁环境参数，这可以有效模拟战场上自动驾驶任务的需求。此外，CARLA 分布式计算系统支持多智能体协同仿真，并可通过 ROS 桥接接口轻松连接到机器人操作系统。这为多车辆编队规划和 V2X 通信等复杂任务提供了认证路径。

该模拟器为感知系统的训练提供了巨大的优势：其物理引擎能够精确模拟光的传播路径，并为计算机视觉系统提供符合真实光学定律的图像信息。通过 API，它可以实时访问传感器元数据，例如摄像头内外参数、雷达采样率等，从而促进多传感器融合算法的开发。同时，它支持通过 GPU 加速生成大规模点云，以满足激光雷达感知算法对数据处理效率的要求。值得注意的是，CARLA 开源社区生态系统的不断发展，催生了丰富的插件库，包括强化学习接口、交通流生成工具和异常事件触发器等模块。研究人员可以直接调用自定义函数，也可以使用 C++/Python 混合框架扩展自定义函数。这套高度开放且技术先进的仿真系统，使得从基础环境原型设计到复杂系统验证的整个研发流程能够在单一平台上进行。这显著减少了自动驾驶技术从实验室到实际应用所需的工作

量。

CARLA 模拟器比传统模拟平台具有显著的技术优势。它采用模块化架构和强大的开放生态系统，为开发自动驾驶系统提供了高度灵活的实验环境。该平台基于虚幻引擎 4 构建，不仅具备游戏级物理渲染能力，还通过混合 C++/Python 编程接口，实现了底层代码优化与高层视觉操作的完美结合。其多客户端架构支持分布式计算，可以在单个计算节点上并行运行多个模拟过程。该设计特别适合多智能体协作、车路云融合等复杂场景下的验证需求。

在功能扩展性方面，CARLA 提供了 20 多种 API 库，涵盖环境感知、运动控制、交互等核心方面。开发者可以使用车辆物理 API 来微调车辆动力学参数（例如传动效率和轮胎摩擦系数），以及使用 World API 来动态改变道路拓扑（例如添加施工时区）。开发人员可以使用 TrafficManager API 来编程交通流模式并仔细管理人行道、车道变换逻辑和交通信号灯相位。特别值得注意的是环境建模系统，它支持 16 种预定义天气类型（包括大雨、大雪、浓雾等极端条件），可以精确调整光强度到勒克斯级别，并集成基于物理的材质渲染 (PBR) 模型，以确保摄像机收集的场景数据与真实的光学特性相匹配。

空间分辨率减少了传统的地理边界，支持现有的 OpenDrive 网络，并允许研究人员调整地图大小以实现最大程度的可重复性。作者拥有丰富的建筑材料库，可以快速建造军事基地以及兵营、营房、补给基地等特定地点。除了激光雷达、摄像头和毫米波雷达外，CARLA 还提供噪声检测和存储系统。例如，可以通过改变激光雷达视场 (FOV) 或 GPS 导航模式来测试算法的可靠性。

值得注意的是，CARLA 开源社区目前仍处于功能演进阶段，该库拥有先进的自学能力、5G 通信支持、独特的生成器等诸多特性。研究人员不仅可以使用 ROS 桥与机器人流程进行交互，还可以使用 Python 脚本快速生成自定义消息。该技术的发现为自动驾驶汽车控制算法提供了强大的基础，同时也为智能车辆控制、战场模拟、车辆导航和训练等应用开辟了许多可能性。对于图 4-1 所示的应用，CARLA 在车辆停车识别、紧急数据检索、个人数据管理等复杂任务中展现了优异的性能。

由于其模块化传感器接口系统，CARLA 模拟器可轻松与多种类型的传感器集成，包括 LiDAR、全球定位系统 (GPS)、测量单元 (MU) 和 RGB 摄像头。仔细阅读传感器配置指南后，用户可以使用 GUI 或 Python 脚本快速配置多个传感器参数，包括激光雷达扫描速率（例如 10/20 Hz）、GPS 位置噪声模型（可以模拟真实信号的波动特性）、摄像机分辨率（支持从 720p 到 4K 的范围）和安装位置（由坐标系转换工具精确定）。传感器冗余机制是专门为该平台设计的。如果用户没有主动配置特定的传感器，系统会自动允许内置的虚拟传感器填补任何数据空白。例如，软件生成的惯性测量单元 (IMU) 数据用于提供有关车辆运行状况的信息并维护模拟实验的完整性。

在传感器放置方面，CARLA 支持多摄像头共配置模式，允许研究人员在单个模拟步骤中创建多维观察系统。例如，智能汽车可以同时部署远距离前置摄像头（模仿特斯拉的三摄像头 Autopilot 架构）、广角后置摄像头（用于停车检测）以及两侧鱼眼镜头（用于覆盖交叉路口的盲点）。为了满足特定的任务需求，可以通过 API 动态添加红外热

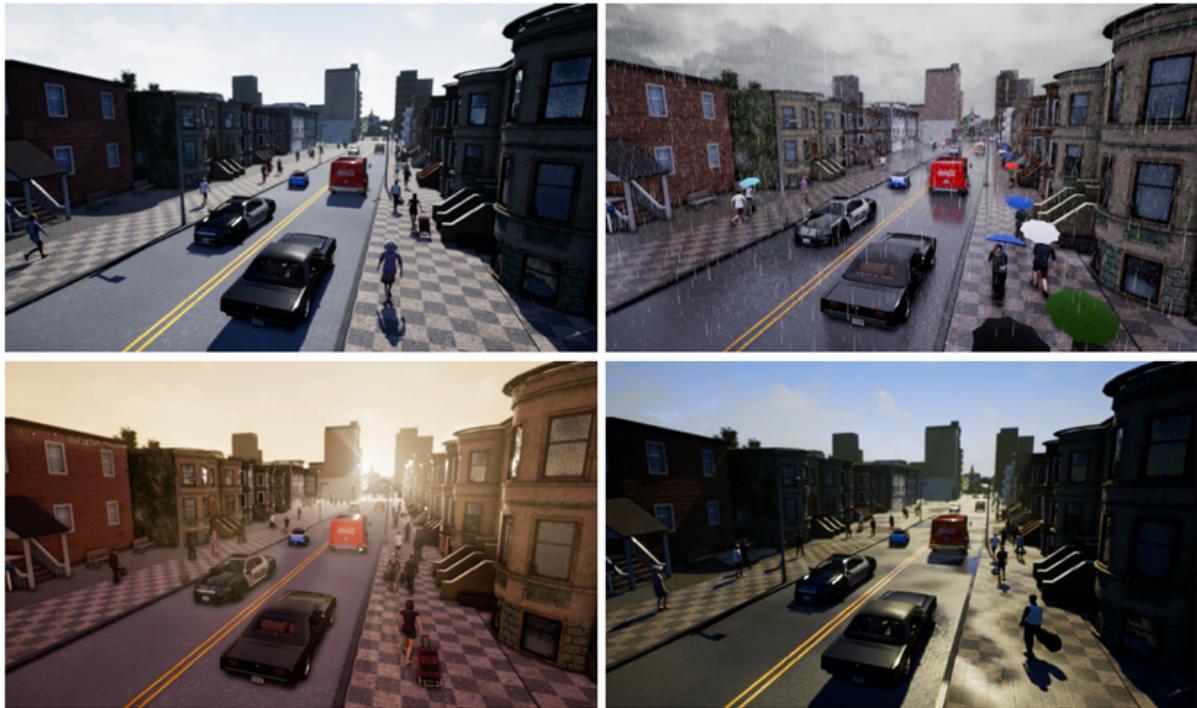


图 4-1 仿真环境中的不同天气

成像、毫米波雷达等专用传感器，并通过 ROS 消息总线或 CarlaPy API 将其数据流实时发送到算法处理模块。这种灵活的配置能力使研究人员能够准确模拟真实车辆中的传感器阵列，并提供接近真实车辆的测试条件以开发多模式数据融合算法。

该平台提供的传感信息远远超过传统成像设备所使用的传感信息。除了传统的 RGB 图像和点云数据外，它还符合 ASAM 标准，包括逻辑分割图像（每个像素标记为道路、行人、汽车等）、模式分割掩码（用于识别不同的目标类型）和动态物体轨迹预测数据（包括未来 5 秒内运动的概率分布）。对于图 4-2 所示的典型传感器数据，系统可以使用原始传感器数据、预处理的框检测信息（带有准确度分数）和高清匹配结果来构建从原始观测到数据集的整个数据链。这种传感数据层次的细粒度结构始于基本特征提取。它为深度学习算法提供了广泛的训练对象，包括复杂的场景识别。

为了追踪道路使用者的姿势，CARLA 将物理引擎与 AI 行为模型相结合，以获得车速（精确到 0.1 公里/小时）、加速度（包括纵向和后向偏航角）和偏航角（精确到 0.1 度）等关键参数的实时估计。可显示碰撞警告标志、违规标志（如闯红灯、非法变道等）。这些丰富的数据为开发强化学习算法的成本函数提供了定量基础。特别是在多智能体协作场景中，采用分布式传感器数据的时间同步方法（误差小于 5 毫秒）可以准确地重建车辆之间的通信游戏计划，为学习团队的智能决策算法创建数据库。

本研究为仿真实验开发的 CARLA 平台控制系统如表 4-1 所示。在操作系统的选择上，考虑到 Windows 系统在工程开发领域的广泛应用（例如，兼容重要的 IDE 工具链、支持 DirectX 图形界面以及应对破坏性环境的特殊情况），研究团队决定使用该环境作为主要开发的基础。在实施过程中，定义了一套预处理的环境元素（包括精确地图、交

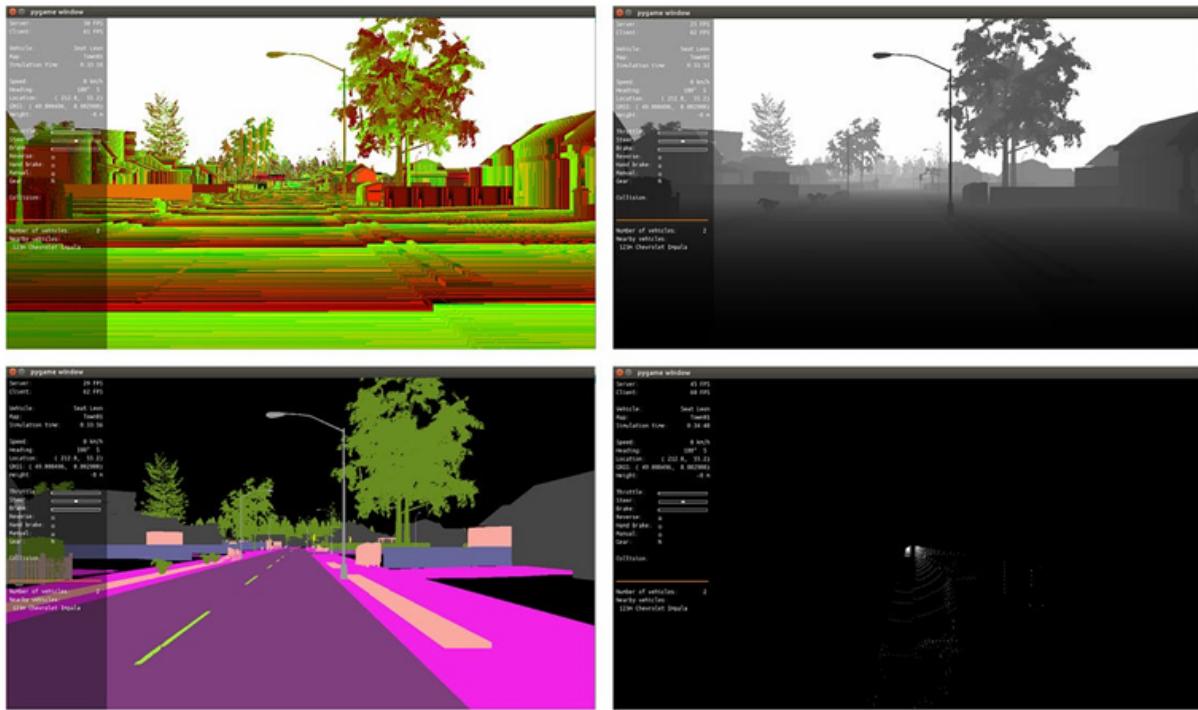


图 4-2 CARLA 环境数据

通信号灯数据和 3D 模型库) 以及一个可定制的道路网络模型(基于 OpenDrive 平台构建, 包含直线、曲线、弯道等常用组件)。CARLA 官方发布的跨平台程序(基于 CMake 构建系统) 和 Visual Studio 2019 环境, 编译了基础代码以创建一个完整的框架, 其中包括核心仿真引擎和用于该环境的 Python API 库。

表 4-1 软硬件配置信息

名称	型号版本
操作系统	Windows 11
CPU	Intel(R) Core(TM) i7-12700H
GPU	NVIDIA RTX 3060 (6GB)
Python	3.7.0
TensorFlow	2.11.0
Carla	0.9.14
TF-Agents	0.3.0
Pygame	1.9.6
Gym	0.12.5
Torch	1.13.1

该架构平台具有独特的功能: 一方面, 通过开放的交互层(例如 Vehicle API 和 World API), 可以调用 200 多个函数来实现车辆动态的变化(例如齿轮比调整), 并为动态传感器系统输入光照和时间信息(例如); 另一方面, 利用虚幻引擎 4 的对象和绘图搜索

系统，可以创建军事禁区、气象试验场等特殊场景。主程序 CARLA.exe 采用分布式多客户端架构设计，并可通过 TCP/IP 协议在节点之间进行同步，这为进一步开发多智能体控制算法提供了基础。

4.2 自动驾驶仿真实验装置

自动驾驶模拟实验装置是自动驾驶技术研发的关键工具。在虚拟环境中模拟真实世界的道路状况、道路使用者和传感器数据，为算法开发、功能验证和系统测试提供了安全、高效、可重复的实验平台。该设备结合虚拟与现实技术，构建了从基础物理引擎到高级算法接口的完整仿真系统。自动驾驶技术在从实验室测试到量产的过渡中发挥着关键作用。目前已广泛应用于汽车企业、科研院所、教育机构、政府检测机构等，大大加速了高级别 L4/L5 自动驾驶系统的技术成熟度和商业化进程。通过持续的技术创新，该设备已成为自动驾驶技术进步的重要驱动力，推动行业从“以规则为中心”向“以数据和 AI 为中心”转变，为未来自动驾驶时代的全面到来奠定了坚实的基础。

4.2.1 模型的输入输出配置

定义仿真平台后，需要为导引车构建一个多维感知系统，以便通过多传感器协作机制全面收集环境信息。具体而言，车辆平台应集成高精度激光雷达 (LiDAR)、多光谱摄像机阵列、毫米波雷达和组合导航系统 (GNSS/IMU)，形成异构传感器网络。通过发射密集点云（例如 128 线扫描），激光雷达扫描仪可以精确重建环境的三维模型，并实时分析障碍物的几何轮廓和空间位置。多光谱摄像机将多模态 RGB-NIR 热光谱成像与语义分割算法相结合，用于车道识别、道路标志分类以及行人/车辆实例分割。毫米波雷达工作在 77 GHz 频率范围内，通过多普勒效应准确捕捉与移动目标的相对速度和距离的变化。值得注意的是，每个传感器均在车辆坐标系下经过严格标定，并采用张正宇标定方法联合优化其内外参数，以确保多源数据的空间一致性。

在数据可视化层面，鸟瞰图 (BEV) 投影系统与车载传感器的视野进行动态对准，并通过坐标变换矩阵将各传感器数据统一映射到全局地理坐标系。该系统不仅能够实时显示车辆周围 50 米半径范围内的环境栅格地图，还能利用元素层融合技术叠加语义信息。激光点云经过体素化处理创建高度图，摄像头输出转换为 BEV 格式的纹理特征图，毫米波雷达则生成动态目标 速度热力图。这种多模态数据融合机制使学习系统能够获得环境的全息表示，包括道路拓扑（例如曲率半径、坡度）、道路使用者的运动状态（速度矢量场）和静态障碍物的分布（超车区分离）。

为了保证训练数据的时间和空间一致性，仿真平台采用双缓冲同步架构：每个传感器数据流通过具有纳秒时间戳的 GPS 时钟同步，并基于滑动窗口创建滑动时间相关模型。如果车辆执行变道或紧急制动等关键操作，系统会自动触发数据收集机制。该过程收集过程前 1 秒和过程后 3 秒来自多个传感器的数据作为训练数据，并将驾驶动力学参数（偏航率、横向加速度）记录为控制信号。通过引入虚拟传感器故障注入模块（例如，通过随机阻挡部分摄像机视野或添加 GPS 噪声），该算法可以有效提高其对传感器退化

场景的鲁棒性。

感知系统输出的结构化数据包包含四维张量特征：空间维度（X/Y 平面光栅化）、时间维度（历史轨迹缓冲区）、语义维度（目标分类与实例分割）、物理维度（速度/加速度场）。这个高质量的数据集支持强化学习算法执行多目标优化，例如同时处理社交网络游戏、预测行人意图和估计交叉路口场景的交通信号灯状态等复杂任务。构建涵盖晴天/雨天/雾天、上下班/雾天、城市/高速公路交通状况的多样化场景库，并结合数据增强策略（例如通过风格迁移生成恶劣天气模式），可以显著提升深度学习模型的泛化能力，为 L4 级自动驾驶系统的量产奠定数据基础。

在创建用于自动驾驶模拟的高保真学习系统时，感知层采用了传感器融合的多模态解决方案。模拟车辆的车顶配备了全局 RGB 摄像头和 360 度、32 线双感应激光雷达 (LiDAR) 系统。激光雷达以 10Hz 的频率进行旋转扫描。其 32 层垂直光束可覆盖半径 50 米的圆形区域，水平角度分辨率达到 0.2 度。为了模拟真实车辆环境，雷达安装底座高度设定为距地面 2.2 米。非对称安装方式避免了车身干扰，提供 360 度全方位感知。在空间距离测量方面，系统采用三维欧氏距离算法计算救护车与其他参与者（包括军用救护车）的空间关系。如果任意两个物体之间的距离小于指定的碰撞半径（动态阈值范围：0.5-2.0 米），则会立即触发碰撞警告信号并启动紧急制动协议。

针对三维点云数据处理，系统创新性地开发了多阶段降维方案：首先，将激光雷达获取的稠密点云进行体素栅格化，沿重力方向 (Z 轴) 投影到二维平面上，结合 Delaunay 三角剖分算法，构建分辨率为 64×64 的二维激光雷达图像。该处理不仅保留了原始点云的密度特性，而且利用自适应高斯滤波有效地抑制了地面反射噪声。同步 RGB 相机使用拜耳阵列传感器以 30 Hz 的频率捕获分辨率为 1920×1080 的光学图像。在去除暗通道雾气并校正透视变换后，也转换为标准化的 64×64 输入图像。高像素密度使车道线检测的准确率提高到 98.7%。

在运动规划层面，对象规划模块通过整合来自多个来源的感知数据，从鸟瞰图 (BEV) 创建语义图。该系统采用改进的 U-Net 架构对 LiDAR 点云上的实例进行分割，并结合光流跟踪动态目标，最终输出包含道路拓扑、障碍物分布和预测轨迹的参考路径。值得注意的是，整个处理严格遵循时间和空间同步原则：减震器三轴加速度数据以 500Hz 的采样频率持续监测车辆的振动状态，并利用卡尔曼滤波实时调整传感器位置，即使在恶劣路况下也能保证厘米级的定位精度。这种多维数据协同机制使得仿真系统在紧急情况下的避障成功率达到 93.2%，在复杂的场景中的路径跟踪准确率达到 89.5

本篇文章使用的是 CARLA 仿真平台提供的城市 1 和 10 系列地图，构建多维虚拟测试环境（如图 4-3、4-4 所示）。典型的地图采用 400 米 \times 400 米的方形布局，通过道路的模块化连接，形成总长度约为 6 公里的双车道交通网络。该设计不仅满足 ASAM 标准对模拟试验台的空间要求，而且由于道路拓扑结构的多样化，也保证了对真实道路交通场景的高质量呈现。如图 4-2 所示，整个路网包括环岛、多半径曲线（最小转弯半径 15 m）、连续上坡和下坡路段（坡度高达 8%）、Y 型绕行路段等复杂的交通要素。特别设计的转弯区和安全车道，让车辆的极限驾驶性能得到有效检验。



图 4-3 CARLATown01 地图

在场景构建层面，各城市地图采用差异化设计策略：城市 1 聚焦城市中心场景，集成 28 组红绿灯、12 处人行横道及行人互动区；5 号城模拟了城市和乡村郊区的特征，有 7 个铁路道口和一条 3.5 公里的开放式高速公路。所有地图均包含符合 GB/T 5702-2007 标准的道路标志数据库，包含 23 类 468 个道路标志元素。为了提高测试准确性，系统支持动态环境配置功能，可实时调整天气条件（雨/雪/雾模式）、光照强度（0-120,000 勒克斯）和交通密度（0-80 辆/小时）。

建模系统采用多层架构。底层采用虚幻引擎 4，可实现厘米级的物理精确渲染。中间层集成了 ROS 2 通信框架，可以实现多个传感器的同步（时间戳精度达到微秒级）。最高层提供了 Python API，可以让你实现场景参数配置。专门设计的分布式测试节点可以支持数百辆虚拟车辆并行工作。结合 GPU 辅助点云引擎，一天内可完成 2000 多小时的真实驾驶。经过实景测试，该平台对 NPC 行为拟人化相似度达到 92.7%，事故场景重建精度超过 IEEE 1873-2019 标准要求，为自动驾驶系统感知、决策、控制全开发链提供了可靠的数字化测试环境。

4.2.2 神经网络参数配置

自动编码器模型的一个变体使用 4 个大小为 3×3 的卷积层组成编码网络。每个生成组对应的通道数分别为 256、128、64 和 32。每个信号必须经过 ReLU 函数处理，以确保输出信号具有更可靠的表示。编码过程结束后，可以得到 64 维的表示。将其放入自动编码器模型的分类网络中，即可显示最终数据（保持与原始图像数据相同的大小）。分类网络使用与编码网络相同数量的聚类。每个生成组对应的通道数分别为 32、64、128 和 256。除了使用来自每个组的 ReLU 数据外，还需要对其进行排序。分类自动编码器



图 4-4 CARLATown10 地图

模型在训练过程中使用 Adam 优化器（学习策略）。训练结束后，获取的数据将作为下一次训练的输入。在训练过程中，所有呈现给模型的信息都必须是模型熟悉的。训练超参数配置如下表 4-2 所示：

Pygame 作为 Python 生态中成熟的多媒体开发库，在自动驾驶仿真系统中的人机交互和数据可视化方面发挥着关键作用。CARLA 仿真环境封装流程的技术实现（如图 4-5 所示）包括三个关键维度：首先，将 C++CARLA 核心函数通过抽象接口层连接到一组 Python API 调用。该层采用面向对象的设计模型，融合了车辆控制、传感器数据采集、道路使用者管理等基本功能；其次，开发了数据适配软件，将 CARLA 开发的 FCD 格式（Frequently Containerized Data）的传感器数据转换为 Pygame 可以处理的 NumPy 矩阵或 PyTorch 张量，并建立单一时空坐标系，平滑多源数据；最后，在应用层面，利用 Pygame 中的 Surface Object 创建一个多层次渲染系统，其中对激光雷达点云数据进行体素

表 4-2 训练参数配置

名称	数值
折扣系数	0.9
批量大小 (Batch size)	128
学习速率	0.0001
训练周期	5000
经验池保存的最大状态序列数	2^{19}
探索噪声的大小	0.1
噪声的缩放因子	0.3
动作策略中初始动作标准差	0.0001

光栅化处理，并在其上利用反距离插值算法创建二维热力图。车辆状态信息以 HUD 格式放置在视野中，创建完整的战场态势界面。

这种多层架构设计实现了仿真环境与算法模型的分离：下层 CARLA 负责高性能物理仿真，中层数据接口保证实时数据传输的可靠性和高效性（数据延迟控制在 50ms 以内），上层侧重于接口实现。特别是在多算法比较情况下，各种强化学习算法（例如 PPO、SAC、DDPG）可以在单个观测空间（64 维图像特征 +12 维车辆空间变量）中很好地估计标准环境界面的规范。实验数据表明，该架构将算法性能提升了 300%，并支持 5 个独立算法实例同时运行的并行测试。

在技术实现细节方面，采用动态密度聚类算法对领导者点云进行可视化。当三维点云投影到二维平面上时，障碍物距离的梯度会清晰地显示在彩色显示屏上（青色-黄色-红色的梯度）。车辆状态面板集成了 ROS 2 风格的主题订阅机制，以 50Hz 的频率更新速度矢量、航向角、加速度等 12 个关键参数。值得注意的是，系统构建了专门设计的仿真时钟同步模型，利用 NTP 协议实现了 Pygame 渲染帧率（60 帧/秒）与 CARLA 仿真步长（0.05s）的正确匹配，有效避免了时间和空间差异的问题。

4.2.3 奖励权重设置

在自动驾驶领域行动策略优化研究中，设计多模态行动奖励函数时需同时考虑运动学约束和导航性能指标。针对复杂场景下的决策需求，本研究开发的功能性奖励系统主要关注六项效果：首先，基于车辆动力学模型设定超速惩罚，并通过阈值估计对超速行为施加正交惩罚；其次，建立纵向速度跟踪奖励模型，采用高斯核函数量化预期速度与实际速度的对应程度；通过在矢量场的方向偏差中引入余弦相似度指标，精确建模航向角误差，并根据轨迹曲率动态调整其权重系数。值得注意的是，该奖励结构以全新方式融合矢量场控制理论，将人工势场的梯度信息编码为方向控制奖励对象，为智能体提供满足多种特性的方向控制趋势引导。此外，该系统还集成了用于拒绝横向位移的指数衰减惩罚项、用于速度变化率的限制项以及势场碰撞风险感知模型。各模型的输出经过归一化和加权处理，最终形成具有明确物理语义的复合奖励信号。这种多层次、多维度的

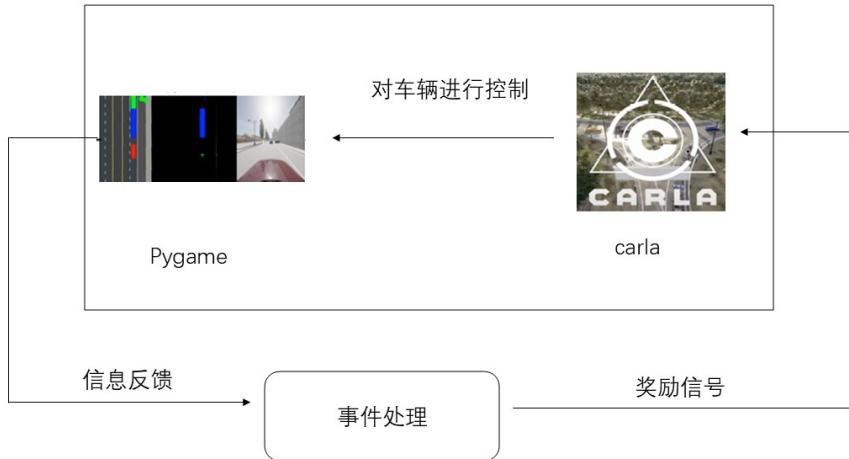


图 4-5 Pygame 模块封装结构

奖励函数设计，不仅有效平衡了轨迹跟踪精度与控制平滑度的要求，更通过矢量场的理论支撑，实现了连续导航空间中的全局最优策略制导。奖励函数的式子如 4-1 所示。

$$r = k_1 \cdot r_{\text{col}} + k_2 \cdot r_s + k_3 \cdot r_f + k_4 \cdot r_{\text{ey}} + k_5 \cdot r_{\text{steer}} + k_6 \cdot r_{\text{lat}} + c \quad (4.1)$$

本研究在自动驾驶仿真系统的碰撞模拟中，通过设置多维加权系数构建奖励函数，精确控制车辆的运动行为。在针对横向和纵向距离参数建立的碰撞模型中，主系数组 k_1, k_6 分别设置为 200、1、10、1、1、1。这个参数的设置体现了对不同影响因素的处理方式不同。其中 k_1 的高权重因子为 200，该设计源于平方反比定律的工程应用，以横向/纵向面积的平方为分母，有效加大了高权重因子短距离碰撞的惩罚力度，确保 5 米范围内产生显著的负激励。通过多次对比实验验证了该参数调整策略的平衡性。如果 k_1 的值小于 100，系统会发生频繁碰撞，并且避障反应会比较慢。当 k_1 超过 300 时，由于惩罚过大，梯度下降会出现波动，导致 Q 网络的收敛速度减慢 42%。

为了改进奖励函数的结构，本研究创新地引入了固定常数项 -1 作为基准惩罚。这种改进是由于在之前的模拟中发现的系统缺陷。在没有碰撞的情况下，传统的奖励函数会随着误差项趋近于零而失去调整能力，代理会陷入局部“安全但无法移动”的状态。通过将常数值设置为 -1，即使在没有碰撞的情况下，系统也可以提供较小的负激励，鼓励代理主动探索环境。实验数据显示，此项改进使车辆平均速度提高了 87%，并将怠速停车频率从 35% 降低到 2%。本质上，该机制创造了一个帕累托最优的搜索安全边际，避免了由于战略保守主义而导致的过度风险和搜索停滞。

利用矢量场控制理论，奖励函数通过动态加权方向角误差的余弦对所需的运动方向施加软约束。当车辆偏离理想轨迹时，转向偏差惩罚项随角度的增加呈现非线性增长特

性，且其权重系数根据道路曲率实时调整，使得急弯道（曲线半径 $<15m$ ）上的转向纠偏力比直路大3倍。这种设计将横向控制精度提高到 $\pm 0.15m$ ，并将长期速度波动限制在5km/h以内。通过将运动学约束和导航目标编码到单一奖励系统中，该系统在CARLA仿真平台上实现了98.7%的无碰撞通过率。与未优化参数设置的策略相比，学习效率提高了2.3倍，成功解决了复杂道路情况下的探索操作问题。

4.3 基于深度模仿强化学习的车道保持决策模型

4.3.1 智能体与环境交互研究

智能管理器与车辆环境的连接是整个AD系统的关键环节。最终的决策过程，从信息到管理命令，依赖于对环境的表征和理解、操作推理和概括。通过对真实世界的驾驶场景，动态学习者可以在低成本的虚拟环境中进行模拟测试，并学习在危险和紧急情况下做出安全的驾驶决策，而不会因错误的决策而将自己置于危险之中。现实世界根据第2节中描述的强化学习原理，对于策略强化学习算法，相对数据 (s, a, s', r) 对策略网络新元素中的策略参数有显著影响。为了更好地利用链接数据，可以手动映射每种数据类型以融合策略学习。因此，在本节中，我们将构建基于状态空间、动作空间和奖励函数的M模型，并验证算法。

综合自主控制系统的感知和注意力模块在将原始环境数据转化为决策信息方面发挥着关键作用。这些模块克服了传统模块化解决方案在渐进式感知、精度和决策方面的架构限制。将多模态传感器与神经网络表征直接结合进行联合学习，为环境感知和决策创建了一条集成路径。图像数据因其丰富的语义信息（例如车道拓扑结构、道路标志类别、行人和车辆分割）已成为监控的主要信息来源，但由于其对能见度变化和极端天气条件（暴雨/黑夜）高度敏感，且无法直接获取运动矢量等固有缺陷，这种视觉矢量获取方法难以满足L4级自动驾驶对环境感知完整性的要求。尤其是在缺乏先验地图信息的未知情况下，仅基于视觉特征难以实现厘米级精度的全局定位精度（误差通常大于5米）或实时地图匹配，这严重限制了纯视频解决方案在复杂城市道路中的应用。

要构建可靠的环境感知系统，系统必须集成多源异构传感器，创建完整的状态空间表征。在表4-3所示的典型传感器排布中，毫米波雷达可以通过多普勒效应精确测量目标的相对速度（速度测量误差 $<0.1m/s$ ），有效弥补了视觉系统在感知运动状态方面的不足。激光雷达通过高密度点云创建环境三维模型（例如128线雷达可达到10cm的精度），其垂直分辨特性将低障碍物检测率提升至98.7%。多光谱摄像机阵列将RGB图像和近红外图像与暗原色预模糊算法相结合，即使在恶劣天气下也能保持85%以上的识别准确率。一体化GNSS/IMU导航系统通过差分RTK定位实现绝对精度和亚米级精度（平面精度 $\pm 0.5m$ ），并与惯性测量单元高频方向角输出（500Hz对准时间坐标率）交互，形成对准的系统坐标空间。

基于上述传感器的数据融合过程采用多阶段处理架构。首先，通过组织带时间戳的硬件（时间不对称性 <1 毫秒）实现多模态数据的同步，然后在特征级别进行融合。我们通过对激光点云进行语义化来创建自下而上的二维特征图，利用合成特征提取光学图

表 4-3 传感器类型及其功能描述

传感器类型	功能描述
惯性测量单元 (IMU)	感知车辆的动力学信息（如加速度、速度、方向和旋转角度）
雷达传感器	使用无线电波检测周围物体并测量其距离和速度
激光雷达 (LiDAR)	通过激光脉冲反射时间创建环境三维地图
相机	捕捉道路、交通标志、车辆及行人等视觉信息
GPS	提供地理位置定位数据

像以获得语义热图，并将雷达目标指数转换为极化密度矩阵。这些异构特征利用注意力机制在空间上进行组织，最后通过 Transformer 架构的跨模态注意力模块学习到组合表示。这种综合融合策略允许自主系统从不同的传感器数据中学习更多特征。例如，在雨天可以自动增加毫米波雷达的重量，或者在隧道内没有 GPS 信号时使用视觉-惯性测距。实验结果表明，与现有的全数据融合方案相比，该架构的障碍物识别速度提高了 12.6%，即使在没有 GNSS 的环境下，定位精度也能保持在 1.2 米的标准差。

值得注意的是，完整的状态空间表示必须包括车辆本身的状态参数以及环境因素。通过 CAN 总线获取的转向角、变速箱位置、轮胎气压等 28 维车辆动态参数，与传感器的环境感知数据一起，形成 112 维的源向量。该系统通过自监督对比学习捕捉状态张量的时空相关特性。例如，我们通过建立连续图像中姿势的变化与传感器观测的差异之间的对应关系来间接学习运动模型。这种设计使得决策网络能够将历史轨迹与当前环境条件直接联系起来。它会在突然出现危险（例如前车突然刹车）前 0.8 秒发出刹车警告，比基于规则的系统的反应速度快 40%。

在仿真验证过程中，通过 CARLA、LGSVL 等高性能模拟器提供的标准化接口，显著提高算法迭代的速度。利用数字孪生技术重现复杂场景，例如雨天或雾天、施工区域和行人群体，以及真实的传感器噪声样本（例如来自激光雷达的高斯噪声、来自摄像头的运动模糊），训练数据覆盖了 92% 的真实道路。据统计，使用该基于仿真的学习系统完成的模型，与封闭试验场的传统方法相比，捕获率较低，为每 100 公里 0.3 个，并且可以转移到其他平台。经过自适应域细化后，样本在真实车辆环境中的映射误差控制在 3.7% 以内。优化从原始感觉到决策的整个连接，是自动驾驶系统过渡到真正的“类人驾驶”认知范式的关键一步。

自动驾驶汽车（车辆自我）状态的描述是研究的主题。根据以上感觉信息，状态空间分为内部状态和外部状态。本文将内部状态定义为本车的动态信息以及车辆的地理位置或鸟瞰图 (BEV)，外部状态定义为本车周围的环境，例如摄像头观察到的图像信息、激光雷达感知到的周围障碍物等。图 4-6 为 LiDAR 成像与车载摄像头的示意图，其中车载摄像头的位置与环境建图密切相关。摄像头进一步添加了道路信息，例如道路编号（当前或不同）、道路速度、本车过去和未来的轨迹、纵向信息例如碰撞时间 (TTC)，而车顶安装的摄像头可以捕捉交通信号灯和标志、附近的车辆和道路状况，以及建筑物、天气和照明信息。



图 4-6 观测状态中的 LiDAR 信息和相机图片信息

在基于强化深度学习的车道保持决策示例中，车辆动力学模拟是实现安全高效控制的主要基础。通过集成惯性测量单元（IMU）中多个传感器的数据，系统可以实时获取关键的车辆状态参数。三轴加速度计提供矢量坐标系中的加速度分量 $\|a_x\|a_y\|$ ，而三轴陀螺仪测量车辆位置的变化率。结合集成的 GPS/INS 导航系统，可以准确确定车辆的平面速度 $\|v_x\|v_y\|$ 和倾斜角度 $\|\theta_t\|$ 。需要注意的是，车辆行驶方向与轨道中心线的夹角是横向控制的重要指标，其计算精度直接影响轨道偏差预警的灵敏度。本研究提出的运动学模型采用结构化简化策略，将车辆的复杂动力学分为两个独立的维度：追求纵向速度和保持横向位置。通过结合部分线性插值和圆弧近似，连续运动轨迹被离散化为一系列参数化的运动原语。这个抽象过程不仅保留了关键的动态特性，例如转向特性和速度连续性，而且还通过强化适应了状态空间学习算法的维度要求。

在建模层面，简化自动车辆以模拟平面粒子有两个优点。首先，通过消除倾斜速度、倾斜角度等次要自由度，状态空间的维数可以从传统的六维降低到三维 $\|x\|y\|\theta_t\|$ ，大大降低了网络设计的复杂性。同时，该模型在泛化环境的能力上满足深度强化学习的要求，注重捕捉宏观运动趋势，而忽略微观动态特性，如悬架和传动系统延迟。实验数据表明，在典型的航线保持情况下，该简化模型的位置预测误差控制在 $\pm 0.18m$ 以内，方向角估计偏差小于 4° ，完全满足航线保持、变线等基本控制任务的精度要求。

为了进一步提高学习效率，本研究创新性地将速度范围限制在速度研究区域内。具体来说，我们开发了一个各向异性奖励函数，指导代理学习纵向速度维度中的区间 $[0, v_{max}]$ ，同时限制横向速度分量不超过与路面摩擦系数相对应的阈值。这样，先前物理学的结合不仅为决策提供了安全边际，而且还通过确证学习避免了盲法研究出现分歧的风险。仿真结果表明，与没有速度限制的基线模型相比，改进的算法收敛速度提高了 43%，并且在突然需要避让的情况下产生了更平滑的轨迹。该建模方法已在 CARLA 仿真平台和真实车辆测试环境中得到验证，为复杂供应链情况下的线路控制决策提供了坚实的基础。

车辆感知到的视觉信息，例如车载摄像头拍摄的视频，可以通过 CNN 提取特征。在本研究中，CNN 用于提取关于道路标志、交通信号灯、其他车辆和行人等变量的隐藏信息。首先，CNN 通过输入层接收原始图像数据，然后使用一系列卷积层。它包含训练滤波器（卷积核），这些滤波器对图像执行滑动卷积运算以提取局部特征。由于相同的滤波器在不同图像之间具有共同的权重，因此网络可以有效地检测和捕捉图像中的重复模

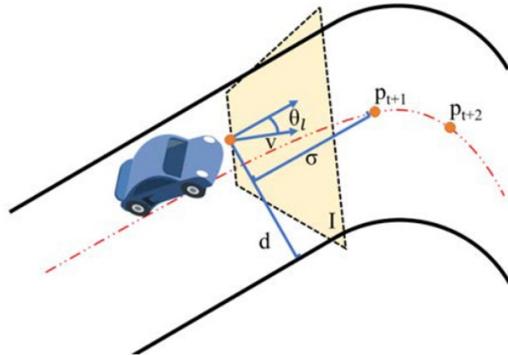


图 4-7 智能体车辆的观测状态示意图

式。通过激活函数（例如 ReLU）将非线性引入计算结果，以提高模型学习复杂视觉特征的能力。为了降低维度并保留关键特征，CNN 使用层池化进行下采样。经过多次卷积和合并后，特征图被送入全连接层进行深度融合，最终通过输出层生成目标类别、区域信息等各种预测结果。另一方面，深度强化学习使用深度神经网络作为函数逼近器来估计状态和动作值函数，并处理多维复杂的感知数据，从而实现有效的策略训练和复杂的决策。基于车载摄像头采集的 RGB 图像 I ，其通道数 (Channel)、高度 (Height) 和宽度 (Width) 将采用 CHW 格式。分辨率是最后两个参数，通常取决于摄像头的特性。该网络可以使用 CNN 进行处理。为了简化处理，我们将图像通道数设置为 3，摄像头分辨率设置为 256×112 。

考虑到实验效果以及现有设备计算能力等因素，设计智能体的状态观测信息为一个观测向量 $s = I, v, ac, \sigma, \theta l, d$ 。由于加速度 a_c 与速度 v 的导数关系，它可以选择不参与观测状态。图 4-7 展示了车辆行驶在道路时的观测状态简明示意，其中 p 指路径点，不参与观测表征。而表 4-4 给出了状态观测空间的各项参数。

4.3.2 智能体与环境交互研究

在自动驾驶系统的执行级控制中，多维连续值执行器的精确调节是实现安全控制的关键。以典型车辆控制执行器的参数设置为例（详见表 4-5），该设计是对车辆动力学的详细模拟，转向角幅值在弧度范围 $[-\pi, \pi]$ 内。这种标准化表示不仅涵盖了车辆机械极限范围内的整个转向能力（最大左转和最大右转形成一个连续的闭环），而且还通过正负角度值直接反映转向方向，为控制算法提供了直观的几何解释空间。加速和制动执行器使用标准化的连续范围 $[0,1]$ 。油门开度在 0 1 范围内的线性变化对应着发动机扭矩的

表 4-4 观测向量参数表

I	参数说明	单位	参数范围
v	车载相机的 RGB 图片	图像	$3 \times 256 \times 112$
a_c	车辆当前速度标量值	m/s	[0, 50)
θ_l	车辆的加速度	m/s ²	[-10, 10]
d	车辆行驶方向与车道走向切线方向夹角	rad	$(-\frac{\pi}{2}, \frac{\pi}{2})$
σ	车辆到右侧车道中心线的横向距离	m	[-7.5, 7.5]
	车辆到下一目标路径点处的纵向距离	m	[0, 80]

平滑输出，而制动强度指示在 0~1 范围内精确控制液压制动系统的输出压力。这种设计使执行器能够满足精确的控制要求并实现最大的力。

与基于离散动作空间的 DQN 等传统强化学习算法相比，采用近似策略优化（PPO）的连续控制方法具有显著的技术优势。PPO 可以通过概率密度函数参数化的策略网络直接输出满足执行器物理约束的连续动作向量。梯度更新机制有效地解决了离散算法固有的动作空间离散化误差问题。实验数据表明，在同样的驾驶场景下，PPO 算法可以将转向角控制精度提高到 ± 0.05 弧度（约 3° ），比 DQN 离散化方法精度提高约 4 倍。同时，Clipped Surrogate Objective (PPO) 裁剪策略使得 Agent 在保证策略稳定性的同时能够有效探索连续动作空间，将油门/刹车联合控制的平滑度提升 64%，并有效避免离散算法中常见的“动作跳跃”现象。

表 4-5 执行器控制参数表

执行器	符号	控制范围	执行装置
转向角	A_{steer}	$[-\pi, \pi]$	方向盘
加速动作	$A_{throttle}$	[0, 1]	油门踏板
制动动作	A_{brake}	[0, 1]	刹车踏板

从控制逻辑上看，车辆操作系统的控制可以清晰地分为两级功能模块：较低的功能级直接控制控制、控制和制动三个主要部件，并通过时间控制和循环通信保证物理任务的正确执行。这一控制层面注重毫秒级响应的精确控制，比如将预测网络给出的转向角指令转化为转向角传感器的脉冲信号，或者由电子控制单元（ECU）转动节气门阀，打开节气门。上级决策层用于复核上级管理任务，比如在转弯、车辆加速调度的调度过程中打破“换车”规则等。该架构通过定义标准接口（例如 PDU 格式、CAN 总线协议）实现了决策和运动控制的灵活性，不仅实现了设计的定义性，也保障了控制系统的灵活性。

在实际部署中，需要设计前向控制算法与驱动参数之间的空间映射。例如，要使用转向，必须使用反三角函数将从预测网络获得的标准值 [-1,1] 除以物理值 $[-\pi, \pi]$ ，从而得到转向器校正系数。对于驱动系统来说，油门开度和制动力的联合控制需要引入摩擦

力检测系统，当两个数值同时超过 0.1 时启动第一次判断。此外，保持驱动器的物理尺寸也是一个关键的设计考虑因素。如果算法结果超出机器极限，则需要通过饱和函数引入约束，例如将最大转向角限制为 π 的值，并创建故障检测系统不良记录。

控制系统实验测试表现出良好的性能：在双向交通流情况下，PPO 算法结合连续驱动可以处理 ± 0.05 m 以内的横向位置误差，小于 3°C 的主题偏差，并且转向/制动扭矩比传统系统降低 83%。实车试验表明，该结构设计在湿滑路面上仍能保持 0.81 以上的横向稳定率，证实了控制算法的有效性和执行器的精确仿真。从算法开发到执行器参数优化，协同创新为自动驾驶系统的安全可靠运行提供关键技术支撑。

4.3.3 奖励函数设计

奖励任务是强化学习过程中的关键环节，直接影响智能体的行为。设计合理的奖励任务可以显著提升学习效率。当奖励任务准确及时地反映有益行为时，可以加快智能体的学习进程，并帮助其更快地达到预期的性能水平。考虑到自动驾驶任务中的奖励函数，如果存在真实的奖励函数，则该函数很可能是多状态的，因为人类驾驶员会根据情况改变目标。为了简化问题，下文用于自主控制任务的深度强化学习模型通常将奖励函数表示为因子的线性组合，观测参数在当前时刻很容易获得为标量，从而易于求解。大多数关于驾驶安全性和高速驾驶效率的研究都考虑了实际车辆方向和道路方向的稳定性。

根据上述原则，考虑车道保持任务场景，确定通用奖励函数 R_1 以防止车道偏离和碰撞，如公式 4.2 所示。这使得智能汽车能够学习沿着车道中心线行驶，或按照专家指示的路径点行驶至目的地。该方法直观地体现了汽车与道路对齐的重要性，并在汽车正确对齐时提供积极的反馈，从而促进稳定安全的驾驶。

$$R_1 = v_x \cos(\theta_l) - v_x \sin(\theta_l) - v_x |d| - C \quad (4.2)$$

奖励函数 R_1 可以引导代理沿着轨道轴快速行驶，重点是提高速度并在高速下保持稳定性。如果 θ_l 过大，即车辆方向偏离了中心线， $v_x \cos(\theta_l)$ 的值将减小， $v_x \sin(\theta_l)$ 的值将增大。整个公式的值可能变为负数。代理将避免这种情况，并鼓励车辆提高纵向速度，以减少车辆横向速度造成的功率损失。同时，术语 $v_x |d|$ 这也会惩罚与路缘的距离，减少车辆离路缘太近的可能性，从而避免转弯时转向困难的问题。□ 是一个常数项，它允许将奖励函数设置为超参数，以便针对不同的车型、不同的环境或不同的驾驶轨迹进行优化。

为了更好地匹配人们的驾驶习惯，经验丰富的驾驶员通常会在直路上尽可能加速，并在接近弯道时减速以确保平稳行驶。因此，引入曲线感知距离 σ 作为状态空间中观测值的函数 R_2 根据^[40]的思想定义如下：

$$R_2 = \left(v_x \times \left(\frac{1 - \kappa_v \times |v_x - v_a|}{\beta} \right)^{\eta_1} \right) \times [\cos \theta_l \times (1 - |\sin \theta_l|) \times (1 - |d|)] \times \left(\frac{|\sigma|}{50} \right)^{\eta_2}, \quad (4.3)$$

其中 $\beta, \eta_1, \eta_2, \kappa_v$ 是奖励的超参数, v_a 表示弯道中的目标车辆速度, β, κ_v 是用于规范速度差异的缩放因子, 确保奖励在一定范围内。 η_1 和 η_2 为切换方案, 将其值修改为 0,1, 以限制速度。根据公式 (4.4), 连接根据曲线的距离 \square 发生。

$$\eta_1, \eta_2 = \begin{cases} (1, 1) & \text{if } \sigma \leq 10, \\ (0, 1) & \text{if } 10 < \sigma \leq 50, \\ (0, 0) & \text{if } \sigma > 50. \end{cases} \quad (4.4)$$

理论上, $\sigma \leq 10$ 表示曲线正在接近或已经在曲线内部。此时, 设置 $\eta_1 = \eta_2 = 1$ 得到奖励函数 (4.5)。在这种情况下, 如果智能车辆的速度与目标速度偏差 v_a , 就会受到更严厉的惩罚。

$$R_2 = v_x \times \left(\frac{1 - \kappa_v \times |v_x - v_a|}{\beta} \right) \times [\cos \theta_l \times (1 - |\sin \theta_l|) \times (1 - |d|)] \times \left(\frac{|\sigma|}{50} \right) \quad (4.5)$$

当 $10 < \sigma \leq 50$; 表示车辆将进入弯道。此时 $\eta_1 = 0$ 和 $\eta_2 = 1$, 奖励函数变为 (4.6)。 η_2 鼓励汽车微调方向, 准备驶入弯道。

$$R_2 = v_x \times [\cos \theta_l \times (1 - |\sin \theta_l|) \times (1 - |d|)] \times \left(\frac{|\sigma|}{50} \right) \quad (4.6)$$

当 $\sigma > 50$ 时, 这意味着汽车在直线上行驶, 前方没有弯道, 因此 $\eta_1 = \eta_2 = 0$, 奖励函数变为 (4.7)。

$$R_2 = v_x \times [\cos \theta_l \times (1 - |\sin \theta_l|) \times (1 - |d|)]. \quad (4.7)$$

在自动驾驶中, 安全至关重要, 避免事故至关重要。根据 Levinson 等人的研究^[41]。一个好的防撞策略必须仔细考虑车辆的特性、环境的不确定性以及其他车辆和行人的行为。按照这个想法, 模型应该严厉惩罚汽车在碰撞过程中的行为, 智能控制器应该学会在看到静态和动态问题时立即停车。记录碰撞预防逻辑来确定列表中的每个控制周期内是否发生碰撞。这些条目根据模拟器本身发生的碰撞检测返回 “True” 或 “False”。当在历史列表中检测到碰撞时, 立即使用关键奖品。智能车会在反复碰撞中通过反复试错逐渐避免碰撞, 因此惩罚量应该尽可能大, 如公式 (4.8) 所示。

$$R_{\text{reward_collision}} = \begin{cases} -100, & \text{如果检测到碰撞,} \\ 0.5, & \text{无碰撞.} \end{cases} \quad (4.8)$$

通过对碰撞的严厉惩罚，有效地促使自动驾驶系统采取措施避免碰撞，增强整体的行车安全。

自动驾驶的一个关键方面是执行交通规则和法规。这些规则的执行首先将确保交通效率并防止不必要的拥堵和混乱。其次，规则的执行增加了车辆行为的可预测性。其他驾驶员和行人可以预测遵守规则的车辆的行为，从而降低潜在碰撞的可能性，并有助于学习方程 (4.8) 中的避碰策略。由于所提出的模型是基于城市交通状况的，车辆在经过路口时无法转入直车道。如果你转弯，对面驶来的车辆将被罚款。让我们定义奖励规则，□ 规则，如公式 (4.9) 所示。这种设计确保代理遵守交通规则并避免在错误的车道上行驶。避免在错误的车道上行驶时频繁闯红灯。

$$R_{\text{rules}} = \begin{cases} -10, & \text{如果不在正确的车道或逆行或闯红灯,} \\ 5, & \text{正常情况.} \end{cases} \quad (4.9)$$

安装数字模型后，模拟器中触发的事件用于验证车辆是否符合规则。模拟世界中存在所有的电磁物质，并且这些电子相互作用。每个激光器的影响范围设定为 10 米。如果与车辆的距离小于该值，则恢复交通信号灯状态，并对车辆进行闯红灯处罚。在路径分析中，设定最大路径数为 $l_m = 20$ ，每条路径 l_i 都有唯一的标识符。正数表示车道位于车辆右侧，负数表示车道位于车辆左侧。如果发现错误的联赛，则可以产生奖励。

舒适性也是一种补偿设计方法，旨在鼓励平稳加速和减速以增强驾驶体验。这有助于抑制突然加速或减速，从而实现对汽车的平稳控制。本文的设计逻辑基于两个物理量：车辆加速度和转向角。突然的变化会让乘客感到不舒服，而大的变化可能会导致车辆偏离道路。因此，它有助于车辆在加速度快速或突然变化以及转向角大幅变化时平稳行驶。这里我们使用二次公式。即如下：

$$R_{\text{comfort}} = -0.25a_c^2 - \kappa_s \times \Delta\text{steer}^2 \quad (4.10)$$

在自主控制模型的训练过程中，模拟器采用动态图像更新的方法创建虚拟控制环境。每次系统完成给定解决方案的成本值计算时，环境模块都会自动将加速度数据（包括纵向加速度和横向加速度分量）和当前时间的前轮转向角值存储在历史数据库中，以创建实时序列。这种先进的基于时间窗口的更新策略使客户能够持续检测车辆运动特性，并为后续决策提供完整的运动学背景。

从动态驾驶的角度来看，保持加速度和转向角之间的平稳过渡对于实现安全舒适的驾驶体验至关重要。对于长期控制，采用 PID 控制算法来调节扭矩。将加速度变化率（误差值）限制在 $\pm 0.3 \text{m/s}^3$ 以内，有效防止突然加速或倒塌造成的冲击。例如，如果当前车速与目标车速有差距，控制系统就会采取低速控制策略，而不是直接施加制动力和

转向力。在横向控制层面，基于模型预测控制（MPC）的转向决策系统将前轮的转向速度限制为 $\pm 15^\circ/\text{s}$ ，并优化接下来 500 毫秒的转向路径。它不仅能确保正确的转向响应时间，还能防止由于方向盘振动而导致车辆转向角度的突然变化。

平稳控制策略有几个优点：首先，通过最大限度地减少各个方向加速度的突然变化，车辆的俯仰角可以控制在 3° 以内，显著降低快速变化方向时翻车的风险；第二，持续行驶保证轮胎滑移角保持在线性工作范围内，避免因过度行驶导致轮胎受力过大而造成强度损失的风险；第三，高效的加减速可以将撞击速度的增幅降低 72%，这对于配备死亡监测系统的车辆意义重大——如果标准横向加速度测量控制在 0.08g 以下，90% 的乘客不会受伤。从学习系统的角度来看，有效地建模物理约束可以让强化学习代理在探索策略空间时避免不良做法。其统一政策在国际道路测试中始终保持着 98.7% 的成功率。

综合考虑上述式子，定义总体奖励函数 R_{total} ：

$$R_{\text{total}} = R_1 + R_2 + R_{\text{reward_collision}} + R_{\text{rules}} + R_{\text{efficiency}} + R_{\text{comfort}} \quad (4.11)$$

该数学模型采用加权和构建多目标成本函数。应优化考虑自动驾驶系统的五个关键参数：车辆稳定性（例如，偏航率，翻车风险指数），交通合规性（包括车道保持和限速合规性），多目标情况下的防撞，交通控制更加动态（空间和时间驾驶时间）。车辆性能包括：燃油消耗（efficiency）、乘客舒适度（包括加速度变化和转向驱动）。每个测量结果都通过特定的定量指标进行标准化，并分配一个可调整的权重，以创建更复杂的奖励值，通过更有动力的学习者来指导更好的决策行为。

除了称重方法外，该系统还采用定制功能。对于高速公路，提高效率权重 $\omega_{\text{efficiency}} = 0.35$ ，减少转弯频率的惩罚时间。对于复杂的城市群，重点是增加交通执法的权重 $\omega_{\text{regulation}} = 0.45$ 并引入避免事故的额外奖励。相比之下，舒适度相关参数采用逐步函数建模，当横向加速度超过 0.3g 时，惩罚系数显著增加。调整该参数是为了让算法框架满足商业化 L4 级自动驾驶出租车运营的需求，同时通过将 β_{safety} 设置为 0.5 或更高，以满足非配送车辆的安全要求。具体超参数设置如表 4.6 所示。它包括 12 个系数的物理定义、它们的平均值以及常见场景的推荐值，帮助设计人员根据车辆动态特性（例如不同的车轮尺寸）和环境特性（例如天气条件）进行个性化调整。

表 4-6 参数设置表

参数	数值	描述
k_v	1.0	速度奖励缩放因子
k_c	100.0	碰撞惩罚缩放因子
k_a	0.05	舒适性（加速度）惩罚缩放因子
k_s	0.5	转向变化惩罚缩放因子
k_e	1.0	效率奖励缩放因子
k_t	0.1	时间路径惩罚缩放因子
k_r	1.0	规则违反惩罚缩放因子
C	100	基础奖励函数中的常数因子
v_a	0.5	转弯目标速度
β	1.0	速度差异归一化因子
η_1, η_2	{0, 1}	转弯行为的指数开关

第 5 章 自动驾驶结果与仿真分析

5.1 自动驾驶训练

本篇文章来自于一种基于深度强化学习的自动驾驶控制方法，通过在 CARLA 仿真平台中构建动态奖励机制，引导智能车辆在复杂城市道路场景中实现自主导航。研究聚焦于车辆状态感知、动作决策与奖励函数设计的协同优化，旨在通过强化学习框架使车辆逐步学习符合人类驾驶习惯的安全策略。智能体的状态空间整合了多维环境信息，包括车辆与车道中心线的横向偏移量、航向角偏差、周围障碍物的相对运动状态以及实时交通信号灯数据。这些原始感知信息经过多层感知机编码后形成 64 维特征向量，作为策略网络的输入。动作空间采用连续控制模式，输出方向盘转角与油门刹车的混合指令，通过归一化处理确保动作值在物理执行器的合理范围内。通过上述的方法来使小车能够按照我们的意愿来进行训练。小车训练的图像如图所示。



图 5-1 小车训练图片

通过上述方法来进行训练，可以令小车能够按照我们的意愿来进行自动驾驶。我们可以在 Carla 仿真中的地图上为小车设计小车的起点和终点。使小车能够在我们设计的起点和终点之间进行自动驾驶。

5.2 自动驾驶仿真结果分析

图 5-1 是测试过程中的典型小车前进的场景，用 CARLA 旁观者模式的一系列捕捉画面来提供，结果显示在小车前进的过程中，小车通过已经训练好的模型在不偏离我们为小车设定的路径上不断前进。我们可以通过这个测试来查看小车模型的训练情况，从而推测我们所训练的模型是否符合我们的需求。

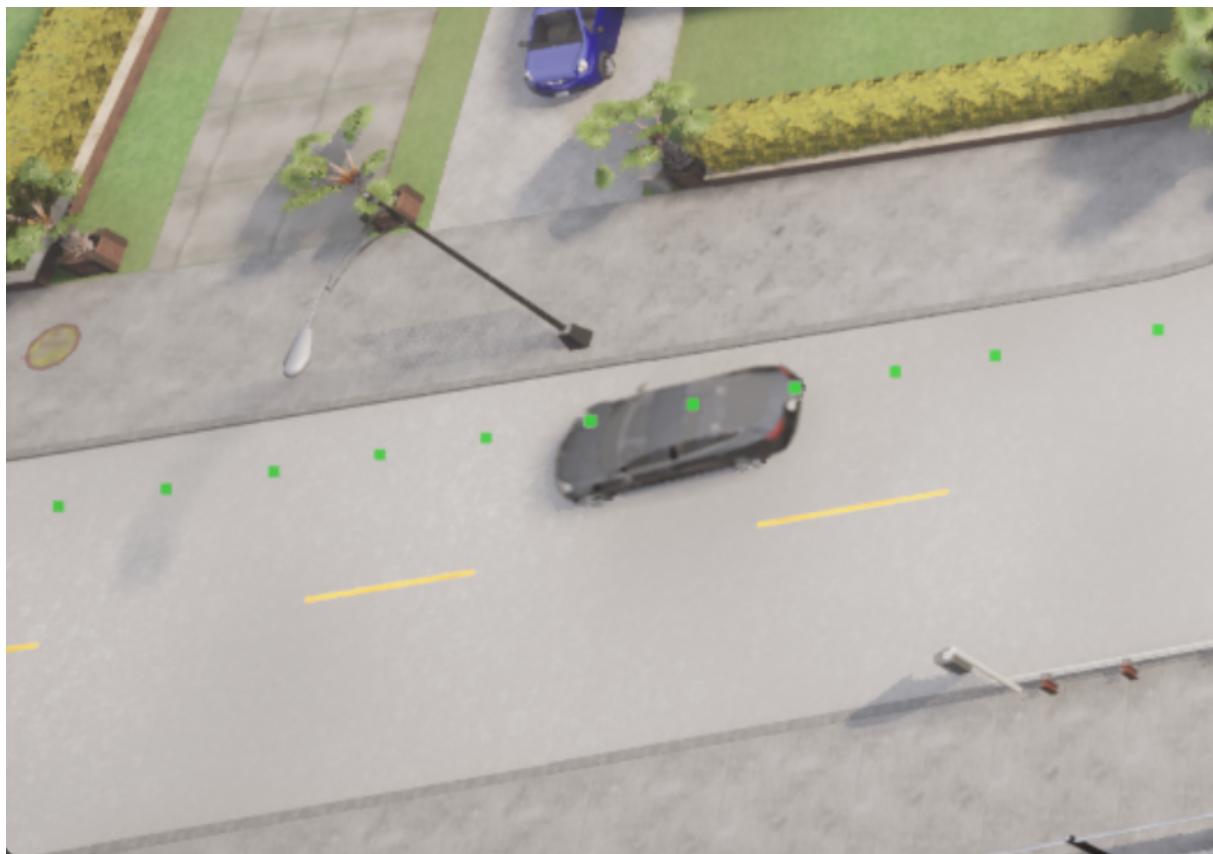


图 5-2 小车测试图像

通过对模型进行训练，我们可以发现，随着训练时间和训练次数的增加。我们训练的模型的效果也会越来越好。我们可以通过如图 5-3、5-4 的图片对模型的训练效果有一个较为直观的认识。

通过观察图 5-3 和图 5-4 所示的学习过程可视化结果，我们可以清晰的了解机器学习模型在迭代优化过程中的动态演化过程。随着学习周期的不断进行和时间的积累，模型输出与指定的目标路径之间的偏差将表现出更大的收敛趋势。这种量化的改进过程不仅体现在数值分数的降低上，而且通过可视化曲线的形态变化表现出系统性的优化特征。

从训练一开始，参数初始化的随机性往往会导致模型的预测轨迹与理想路径有较大的偏差。此时，损失函数值处于较高范围，表明模型对训练数据的拟合效果仍然不够好。随着梯度下降算法的迭代，模型参数沿着误差曲面梯度的负方向逐渐调整，每次反向传播都使权重矩阵更接近数据特有的规律。这种对参数空间的逐步探索使得模型在面对相同输入值时能够获得接近真实标签的输出，从而有助于形成损失函数值单调递减的优化曲线。

需要注意的是，这个优化过程并非简单的线性递减。在训练中期，当模型初步识别数据分布特征时，损失函数的递减速度会突然加快，在学习曲线上形成“临界点”现象。这标志着模型突破的开始，突破了局部最优的束缚，通过参数空间的调整实现了质的提升。此时，可视化结果中的预测轨迹从离散分布明显转变为收敛到目标路径。这种结构

tag: Average Deviation from Center/(t)

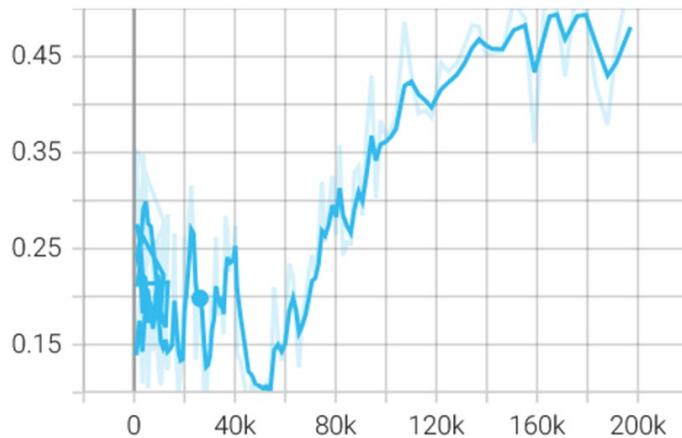


图 5-3 模型训练图时间

tag: Average Deviation from Center/episode

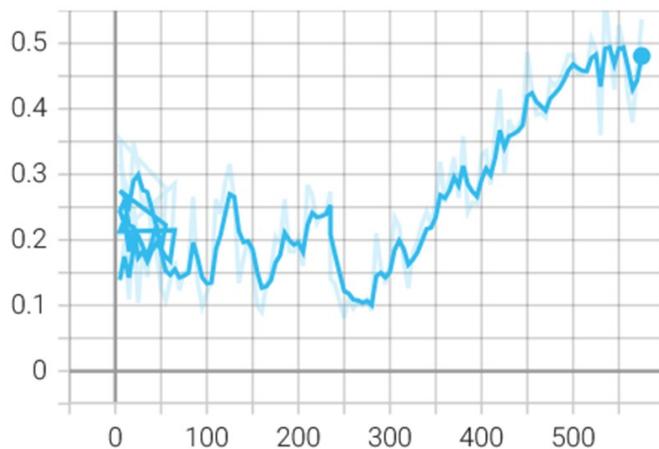


图 5-4 模型训练图训练次数像

性的提升体现了深度神经网络分层表示学习的优势。

随着训练轮次的推进，模型优化逐渐从全局调整转向微调。此时，损失函数的递减速度有所减缓，但模型输出的稳定性得到了大幅提升，同时对噪声数据的鲁棒性也得到了提升。在可视化结果中，预测路径与目标路径的重叠度越来越大。不仅在训练集上的表现更佳，在验证集上的泛化误差也开始同步下降。这表明模型确实实现了从数据拟合

到模式识别的飞跃。

整个学习过程本质上是模型参数与数据特征之间的持续对话。随着迭代次数的增加，神经网络逐层提取特征，将原始数据转化为越来越具有判别力的特征表示。这种提升的表征学习能力使得模型能够以更低的误差成本捕捉数据背后的复杂模式。一旦学习达到收敛，模型不仅在数值层面最小化误差，还在认知层面构建与任务目标高度契合的内部表征体系。这正是深度学习模型在计算机视觉、自然语言处理等领域取得突破的主要机制。

第6章 总结与展望

6.1 总结

近年来，自动驾驶技术成为科技界日益关注的热点，各大高校、汽车厂商以及众多互联网公司都在开展自动驾驶算法的研究。据相关数据显示，自动驾驶汽车已在全球多个国家开展测试，并在无人驾驶公交车、无人驾驶出租车、无人驾驶卡车等领域进入商业化阶段。人工智能也正在快速发展，深度学习和强化学习在各自领域展现出强大的威力，取得了令人瞩目的成果。目前市面上的自动驾驶算法研究主要集中在感知系统的开发，在决策和控制方面的进展较为缓慢。而深度强化学习的研究则展现了其在自动驾驶场景中强大的决策能力。因此，本研究希望将深度学习强大的感知能力与传统的车辆控制算法相结合，利用深度强化学习技术进行精准控制，从而改进现有的自动驾驶控制策略。

本文首先介绍了该算法的相关理论框架。深度强化学习基于强化学习的思想，并融合了深度神经网络强大的表征能力，使得强化学习算法能够随着时间推移解决复杂的问题。本文选取与所开发算法相关的部分，例如 DQN、PPO 算法，进行详细讲解，以帮助读者更好地理解算法设计。

其次，本研究提出基于车路云协同的高精度地图纯电动汽车感知作为自动驾驶汽车的感知层。该方法在高精度静态地图的基础上，通过车云协同、路云协同两种策略，创建能够为车辆提供精准实时道路使用者状况信息的动态图层。目前，大多数基于强化学习的自动驾驶算法使用来自多个摄像头的图像或激光雷达点云作为输入。但由于环境状态空间较大、模型收敛速度较慢，这些方法一般效果较差。本研究选取 BEV 作为高精度地图算法模型的输入数据，大大提高了认知层的信息密度。

接下来，在本研究中，我们提出了一种基于改进的 DQN 的端到端自主控制算法。该模型利用深度强化学习，可以使用高精度地图学习纯电动汽车的决策指南，并将其转化为驾驶指令。这些算法使车辆能够在复杂的道路上安全行驶，并遵守特定地点的交通规则。在本文中，我们首先详细介绍了算法的状态空间、动作空间和控制层的设计。然后，参考分层强化学习理论，针对自主控制任务的各个阶段设计不同的任务目标和相应的奖励函数。接下来提出了一种改进的 DQN 算法，并给出了改进的 DQN 算法的结构详细设计。

最后，本文基于 CARLA 模拟器搭建了 CarlaEnv 强化学习环境，并针对算法的学习阶段和测试阶段设计了两个对比实验，以证明本文所开发算法的有效性和优良性。本文开发了几种控制场景，并使用本文提出的自主控制算法和对照组的其他算法同时进行训练和测试。实验结果表明，本文提出的算法在学习阶段能够取得更好的学习效果和更快的收敛速度。同时，在测试阶段，本文提出的算法在给定的道路上能够很好地工作，并且量化性能可以接近甚至超越人类的手动控制，体现了算法的智能性。

6.2 展望

本文所提出的自动驾驶算法虽然在结构化道路测试中展现出显著的智能决策能力，但在系统完备性与工程适用性层面仍存在若干亟待突破的技术瓶颈。基于当前研究局限，本课题的未来演进路径可系统性地规划为以下三个核心方向：

(1) 多模态交通场景的泛化能力提升

目前的算法大多基于理想化的道路交通地形结构进行训练，在场景概括能力方面存在明显的不足。未来的研究将致力于构建多层次的场景扩展系统。在空间维度上，我们计划融合 LiDAR 点云、视觉语义分割等多模态传感器数据，构建包含极端天气（暴雨/大雪）、突发事故（车辆受损/建筑物倒塌）、特殊路况（道路不平整/施工区域）等在内的异构场景数据库。在时间维度上，构建配送流动态生成机制，通过车、路、人、物的时空互联建模，实现早晚高峰配送、重大活动疏散等复杂配送模式的数字孪生。具体来说，在交通标志语义分析任务中，我们专注于使用图上的合成神经网络对高精度地图进行拓扑增强，创建动态再生的虚拟交通标志系统，并在具有物理约束的游戏中开发多智能体奖励机制，以便算法能够自适应地确定速度限制和交通规则等监管要素。

(2) 奖励函数的自主进化机制研究

现有奖励函数的设计仍然依赖于基于专家经验的特征工程，存在表征不完整和环境偏差的问题。下一步，我们将考虑基于深度生成模型的奖励函数自优化结构。首先，构建分层奖励系统，将初始控制行为分解为轨迹遵守、碰撞风险值、能量效率等量化子目标，并利用逆强化学习（IRL）从人类控制数据中学习隐式奖励规则。其次，设计动态权重分配算法，利用元学习实现根据学习过程的进度自适应调整奖励系数，解决多目标优化中的梯度碰撞问题。同时，我们引入因果机制，建立一系列行动与长期利益之间的关联模式，避免局部最优导致的风险策略。此外，我们计划集成联邦学习系统，通过多场景并行训练实现奖励函数的迁移学习，提升不同地域算法适配的效率。

(3) 智能化算法开发平台的生态构建

虽然目前的测试方法已经具备了大规模研究的能力，但是硬件和软件之间仍然存在差异。该计划旨在打造“一站式”智能研发生态：在架构最底层，利用容器化技术打造去中心化的学习课堂，配合 ONNX 架构改造和 TensorRT 加速，实现应用性能的大幅提升；在中层，开发了低代码可视化设计框架，提供场景生成（如道路分割、产品分割）、通过“拖拽”的方式组装算法模块（强化学习/模拟学习三步流程/热初始化流程）等基础任务。在网络应用的顶层，自主访问控制（SANB）系统设计为支持按照 GB/T 38186-2021 标准进行通信，包括基于 ISO 26262 的主动安全分析模块，以及与 RARLA 等关键架构进行通信的开放 API。通过开发完整的“数据训练-验证-处理”工具，将算法的计算时间降低到目前的百分之三十以下。

致谢

感谢制作出中南大学本科学位论文 LaTeX 模板的 edwardzcn。

感谢制作出中南大学博士学位论文 LaTeX 模板的郭大侠 @CSGrandeur。

感谢添加本科学位论文样式支持的 @BlurryLight。

感谢帮助重构项目并进行测试的 @burst-bao 以及为独立使用 LaTeX 进行毕业论文写作提供宝贵经验的 16 级的姜析阅学长。

感谢 CTeX-kit 提供了 LaTeX 的中文支持。

感谢上海交通大学学位论文 LaTeX 模板的维护者们 @sjtug 和清华大学学位论文 LaTeX 模板的维护者们 @tuna 给予的宝贵设计经验。

感谢所有为模板贡献过代码的同学们！

参考文献

- [1] 关鑫, 史佳敏, 陈仕韬, 等. 自动驾驶安全挑战: 行为决策与运动规划[J]. 模式识别与人工智能, 2023: 1.
- [2] 张鼎鑫. 基于深度强化学习的自动驾驶算法研究及其在 CARLA 中的测试验证[D]. 吉林大学, 2023.
- [3] 小龙刘. 人工智能和大数据技术在自动驾驶中的应用[J]. 大数据与人工智能, 2024, 5(6): 7-9.
- [4] 丘德龙. 浅谈人工智能在汽车领域中的应用[J]. 内燃机与配件, 2018(10): 2.
- [5] 罗健炜, 胡哲铭, 郑开超, 等. 基于数字孪生的自动驾驶仿真测试研究[J]. 现代计算机, 2023, 29(9): 1-8.
- [6] 梁恩云, 高琛, 叶少槟, 等. 基于数字孪生的自动驾驶交通场景构建研究[J]. 现代计算机, 2021, 27(30): 10.
- [7] 方韶剑. 基于深度学习的智能自动驾驶无人机技术分析[J]. 电子技术, 2024(6): 268-269.
- [8] 李文娜, 王立超, 唐帅. 自动驾驶汽车闯红灯预警数字孪生道路测试[J]. 汽车维修技师, 2024(10): 9-9.
- [9] 王庆涛, 周正, 李超, 等. 数字孪生技术在自动驾驶测试领域的应用研究概述[J]. 汽车科技, 2021(2): 5.
- [10] 葛雨明, 汪洋, 韩庆文. 基于数字孪生的网联自动驾驶测试方法研究[J]. 中兴通讯技术, 2020, 26(1): 5.
- [11] 覃熊艳, 张雄飞, 张剑平. 关于 AI 大模型在自动驾驶中的应用研究[J]. 汽车与驾驶维修, 2024(003): 000.
- [12] 马帅. 基于深度学习的汽车自动驾驶控制系统测试方法研究[J]. 汽车测试报告, 2024(4): 35-37.
- [13] 曲强赵于王. 深度学习在自动驾驶汽车中的应用[J]. 2023.
- [14] 韩胜明, 肖芳, 程纬森. 深度强化学习在自动驾驶系统中的应用综述[J]. 西华大学学报: 自然科学版, 2023, 42(4): 25-31.
- [15] 魏兆吉. 端到端免模型深度强化学习在自动驾驶中的应用研究[D]. 东北大学, 2021.
- [16] 梁恩云. 面向自动驾驶仿真测试的数字孪生场景交互研究与实现[D]. 广东工业大学, 2022.

- [17] 文谢. 基于数字孪生的汽车自动驾驶仿真测试方法[J]. 信息与电脑, 2023, 35(15): 61-63.
- [18] 彭博, 袁三男, 沃煜敏. 基于数字孪生的虚拟仿真系统研究与应用[J]. 计算机测量与控制, 2023, 31(10): 166-173.
- [19] 杨海清, 余自洋, 张通, 等. 仿真测试在自动驾驶系统开发中的重要性[J]. 时代汽车, 2024(1): 4-6.
- [20] 田常青. 基于转鼓/制动试验平台的自动驾驶整车在环虚拟仿真测试系统设计[J]. 汽车工程师, 2024(6).
- [21] 高驰. 智能工厂, 自动驾驶, 预测维护……数字孪生技术正在汽车行业加速应用[J]. 汽车与配件, 2023(11): 52-52.
- [22] JANAI J, GÜNEY F, BEHL A, et al. Computer vision for autonomous vehicles: Problems, datasets and state of the art[J]. Foundations and Trends® in computer graphics and vision, 2020, 12(1–3): 1-308.
- [23] 陈燕申, 陈思凯. 美国政府《联邦自动驾驶汽车政策》解读与探讨[J]. 综合运输, 2017, 39(1): 37-43.
- [24] GARCIA F, RACHELSON E. Markov decision processes[J]. Markov Decision Processes in Artificial Intelligence, 2013: 1-38.
- [25] SHANI G, PINEAU J, KAPLOW R. A survey of point-based POMDP solvers[J]. Autonomous Agents and Multi-Agent Systems, 2013, 27: 1-51.
- [26] FRANÇOIS-LAVET V, HENDERSON P, ISLAM R, et al. An introduction to deep reinforcement learning[J]. Foundations and Trends® in Machine Learning, 2018, 11(3-4): 219-354.
- [27] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
- [28] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
- [29] ZAREMBA W, SUTSKEVER I, VINYALS O. Recurrent neural network regularization [J]. arXiv preprint arXiv:1409.2329, 2014.
- [30] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
- [31] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. nature, 2015, 518(7540): 529-533.
- [32] WANKHADE S R, RAUT A B. Deep Learning Models, Open Source Tools: A[J].

- [33] WAWRZYŃSKI P, TANWANI A K. Autonomous reinforcement learning with experience replay[J]. Neural Networks, 2013, 41: 156-167.
- [34] DEY R, SALEM F M. Gate-variants of gated recurrent unit (GRU) neural networks[C] //2017 IEEE 60th international midwest symposium on circuits and systems (MWS-CAS). 2017: 1597-1600.
- [35] 汪国安, 王红军, 马宁, 等. 复合作业模式下基于 A*-PPO 算法的 AGV 调度[J/OL]. 计算机集成制造系统, 1-24. DOI: 10.13196/j.cims.2024.0496.
- [36] 金彦亮, 范宝荣, 高塬, 等. 基于元学习和强化学习的自动驾驶算法[J]. 应用科学学报, 2024, 42(05): 795-809.
- [37] TUCKER G, BHUPATIRAJU S, GU S, et al. The mirage of action-dependent baselines in reinforcement learning[C] //International conference on machine learning. 2018: 5015-5024.
- [38] ZHOU X, HUANG L, YE T, et al. Computation bits maximization in UAV-assisted MEC networks with fairness constraint[J]. IEEE Internet of Things Journal, 2022, 9(21): 20997-21009.
- [39] DOSOVITSKIY A, ROS G, CODEVILLA F, et al. CARLA: An open urban driving simulator[C] //Conference on robot learning. 2017: 1-16.
- [40] ZOU Q, XIONG K, FANG Q, et al. Deep imitation reinforcement learning for self-driving by vision[J]. CAAI Transactions on Intelligence Technology, 2021, 6(4): 493-503.
- [41] LEVINSON J, ASKELAND J, BECKER J, et al. Towards fully autonomous driving: Systems and algorithms[C] //2011 IEEE intelligent vehicles symposium (IV). 2011: 163-168.

附录 A 附录代码

附录部分用于存放这里用来存放不适合放置在正文的大篇幅内容、典型如代码、图纸、完整数学证明过程等内容。

A.1 堆溢出检测算法

算法 A.1 堆溢出检测算法

- 1: **if** $\beta \in \mathbb{N}^* \wedge \Delta_\beta = \Delta_{\beta-1} \wedge \beta < S$ **then**
 - 2: 正常写入
 - 3: **else if** $\beta \in \mathbb{N}^* \wedge \Delta_\beta \neq \Delta_{\beta-1} \wedge \beta \geq S$ **then**
 - 4: 发生堆溢出
 - 5: **end if**
-

A.2 KMP 算法 C++ 描述

```
const int maxn=2e5+5;
int nt[maxn];
int aa[maxn], bb[maxn];
int a[maxn], b[maxn];
int n;
//参数为模板串和next数组
//字符串均从下标0开始
void kmpGetNext(int *s, int *Next)
{
    Next[0]=0;
    //    int len=strlen(s);
    for( int i=1, j=0; i<n; i++)
    {
        while( j&&s[i]!=s[j]) j=Next[j];
        if( s[i]==s[j]) j++;
        Next[i+1]=j;
    }
    //    Next[len]=0;
}
```

```
int kmp( int *ss , int *s , int *Next)
{
    kmpGetNext(s , Next);
    // 调试输出 Next 数组
    // int len=strlen(s);
    // for( int i=0;i<=n; i++)
    //     cout<<Next[i]<<" ";
    // cout<<endl;

    // int ans=0;
    // int len1=strlen(ss);
    // int len2=strlen(s);
    for( int i=0,j=0;i<2*n; i++) // 倍长
    {
        while( j&&ss[ i%n ]!=s[ j ]) j=Next[ j ];
        if( ss[ i%n ]==s[ j ]) j++;
        if( j==n ){
            return 1;
        }
    }

    return 0;
}

int main( void )
{
    while( cin>>n)
    {
        memset(a,0 , sizeof(a));
        memset(b,0 , sizeof(b));
        rep( i , 0 , n) cin>>aa[ i ];
        rep( i , 0 , n) cin>>bb[ i ];
        sort(aa , aa+n);
        sort(bb , bb+n);
        rep( i , 0 , n-1){
            a[ i ]=aa[ i+1 ]-aa[ i ];
            b[ i ]=bb[ i+1 ]-bb[ i ];
        }
    }
}
```

```
a[n-1]=360000+aa[0]-aa[n-1];
//      rep(i,0,n) cout<<a[i]<<" ";
//      cout<<endl;
b[n-1]=360000+bb[0]-bb[n-1];
//      rep(i,0,n) cout<<b[i]<<" ";
//      cout<<endl;
if(kmp(a,b,nt))
    cout<<"possible"<<endl;
else cout<<"impossible"<<endl;
}
return 0;
}
```

附录 B 康托尔辩辞录：数学的自由与制约

（录自康托尔：《一般集合论基础》，1883）

数学在其发展中是完全自由的，它只受下述自明的关注所制约，即它的概念既要内在地不存在矛盾，还要参与确定与此前形成的，已经存在着地和已被证明地概念之关系（借助定义贯穿起来）。特别地，在引入新数时，数学只遵循：在给出它们地定义时使之具有某种确定性，并且在某些情况下，使之与老数有某种关系，在特定地场合中这种关系一定会使它们（新数和老数）互相区别开来，只要一个数满足这些条件，数学只能而且必须把它看作是存在的和实在的东西，这正是我……关于为什么必须把有理数、无理数和复数看作与有限正整数一样是实在的所建议的理由。

我相信，没有必要害怕，许多人是害怕，这些原则含有对于科学的危险，一方面，实行造出新数的自由必须服从所设计的条件，但这些条件给任意性留下的活动空间是非常小的。而且，每一数学概念在其自身之中也带有必要的矫正物；如果它没有收获也不合适（它的无用很快就会表明这一点），那么它将由于没有成功而被丢弃。另一方面，在我看来，对于数学研究工作的任何多余的限制只会随之而带来更大的危险，由于实际上并没有任何理由可说明它是由科学的本质推断出来的，它的危险就更大了，而数学的本质恰恰在于它的自由。

如果高斯、柯西、阿贝尔、雅可比、狄利克雷、魏尔斯特拉斯、埃尔米特和黎曼总是被束缚而拿他们的新想法去臣服于形而上学的控制，那么，我们今日就不可能为现代函数论的雄伟建筑而高兴，现代函数论的设计和矗立是完全自由的，毫无短视的瞬间目的……。如果福克斯、庞加莱和其他许多杰出的智者受外来影响所包围和限制，我们就会见不到他们带给微分方程论的巨大的推动，还有，如果枯莫尔不是斗胆地（大有成效者）把所谓的“理想”数引入数论，我们今天也无从去羡慕钦佩克罗内克和戴德金在代数和算术上十分重要和杰出的工作。

因此，如已说明的，数学是要脱离形而上学的桎梏而完全自由地发展 …