東南大學

硕士学位论文

基于强化学习的自动驾驶仿真原型设计

专 业 名 称: 交通运输工程

研究生姓名: 史云阳

导 师 姓 名: 刘志远 教授

戴一凡 副研究员

PROTOTYPE DESIGN OF AUTONOMOUS VEHICLE SIMULATION BASED ON REINFORCEMENT LEARNING

A Thesis Submitted to

Southeast University

For the Academic Degree of Master of Engineering

BY

SHI YUN YANG

Supervised by

Prof. Zhiyuan Liu & Yifan Dai

School of Transportation

Southeast University

2021/05/06

摘 要

随着智能网联、5G 等技术水平的不断提高以及生产成本的逐渐下降，自动驾驶将迎来一个全面发展的时期。未来很长一段时间内，人类驾驶车辆将与自动驾驶车辆、辅助驾驶车辆共同行驶于城市路网中，形成具有新型特征的混合交通流。因此，针对混合交通流的建模与仿真，自动驾驶虚拟仿真测试的相关研究必将成为智能交通领域的热点问题。然而，现有交通仿真软件均存在缺乏自动驾驶车辆模型的问题，自动驾驶仿真软件则无法提供真实的交通流模型。为了解决上述问题，本论文结合了人工智能领域前沿的深度强化学习算法，基于交通仿真软件 SUMO 进行二次开发，构建自动驾驶车辆微观控制与宏观路径规划模型，同时针对 SUMO 与自动驾驶仿真软件 Carla 的联合仿真测试展开了相关研究。

现有仿真路网搭建工具（或软件）存在精度不高、效率低下、易产生路网错误等问题，论文针对上述问题提出了一种融合多源地图数据的智能化路网优化方法，包括融合 SHP 与 OSM 数据的车道数修改算法与考虑路网拓扑关系的新型仿真路网道路 ID 修改算法。在此基础上，结合活动出行链理论，完成了基于活动的多模式出行输入。在实现仿真路网的智能化搭建与优化的基础上还原了更加真实的交通场景。

其次，论文分析了深度强化学习算法的原理以及经典的车辆跟驰、换道模型的模型机理。在此基础上，将交通领域知识融合到了自动驾驶车辆微观控制的强化学习问题定义中，结合了深度强化学习 PPO 算法，完成了模型的搭建。并结合了 SUMO 仿真场景进行模型训练，结果显示自动驾驶模型能够更安全、快速、高效的完成驾驶任务。

然后，在对经典的路径规划问题深入分析的基础上，论文构建了基于深度强化学习的自动驾驶宏观路径规划模型，结合了深度强化学习的 DQN 算法与定制化的 SUMO 仿真场景完成了模型搭建与训练，结果显示自动驾驶模型在获取起终点信息后能够根据道路交通的变化选择最优路径，更快地到达终点。

最后，论文将自动驾驶微观控制模型与宏观路径选择模型实现了一体化整合，并选取了苏州古城区域作为案例城市进行测试，结果表明该模型具有高泛化性，

可以广泛应用于众多交通仿真场景。在此基础上，分析了 SUMO 与 Carla 的联合仿真原理，实现了案例城市的实时协同自动驾驶联合仿真，完成了全链条的自动驾驶虚拟仿真测试。

关键词：强化学习；自动驾驶；微观行为控制；路径选择；虚拟仿真测试

Abstract

With the continuous improvement of intelligent network, 5G technology and the gradual decline of production costs, autonomous vehicle will usher in a period of comprehensive development. For a long time in the future, human driving vehicles, autonomous vehicles and auxiliary driving vehicles will coexist in the urban road network, forming a new pattern of mixed traffic flow. Therefore, for the related research of the modeling and simulation of mixed traffic flow, autonomous vehicle virtual simulation test will become a hot issue in the field of intelligent transportation. However, the existing traffic simulation software is lack of autonomous vehicle model, and autonomous vehicle simulation software can not restore the real traffic flow scene. In order to solve the above problems, this paper combines the advanced deep reinforcement learning algorithm in the field of artificial intelligence, carries out the secondary development based on the traffic simulation software SUMO, constructs the micro control and macro path planning model of autonomous vehicle, and carries out the related research on the co-simulation test of SUMO and Carla.

Firstly, aiming at the problems of low accuracy, low efficiency and road network errors in the existing simulation road network construction, this paper proposes an intelligent road network optimization method based on multi-source map data, including lane number modification algorithm based on SHP and OSM data and a novel road ID modification algorithm based on road network topology. On this basis, combined with the theory of activity travel chain, the multi-mode travel input based on activity is completed. On the basis of realizing the intelligent construction and optimization of the simulation road network, a more real traffic scene is restored.

Secondly, the paper analyzes the principle of deep reinforcement learning algorithm and the model mechanism of classic car following and lane changing model. On this basis, the traffic domain knowledge is integrated into the definition of reinforcement learning problem for micro control of autonomous vehicles, and the model is built by combining with deep reinforcement learning PPO algorithm.

Combined with SUMO simulation scene, the results show that the autonomous model can complete the driving task more safely, quickly and efficiently.

Then, based on the in-depth analysis of the classic path planning problems, the paper constructs the macro path planning model of autonomous based on deep reinforcement learning, and completes the model construction and training by combining the DQN algorithm of deep reinforcement learning and customized SUMO simulation scene. The results show that the autonomous vehicle macro control model can select the optimal path according to the change of road traffic after obtaining the information of the starting and ending points, and reach the destination faster.

Finally, the paper integrates the micro control model of autonomous with the macro route selection model, and the ancient city of Suzhou is selected as a case city to test. The results show that the model has high generalization and can be widely used in many traffic simulation scenarios. On this basis, the co-simulation principle of SUMO and Carla is studied, the real-time co-simulation of the case city is realized, and the virtual simulation test of the whole chain is completed.

Key words: Reinforcement learning; Autonomous vehicle; Micro behavior control; Path planning; Virtual simulation test

目录

第一章 绪论

1.1 研究背景和意义

随着近些年中国综合国力与经济实力的不断增强，社会活动与城市人口日益频繁与密集，城市交通问题也日益显著[1]。2020 年，全国机动车保有量达 3.72 亿辆，全国 70 个城市的汽车保有量超过 100 万辆[2]。汽车保有量逐年增高、车辆利用率不高、部分人群的交通意识低下等因素导致了城市交通拥堵、交通安全等问题。根据中国公安部与国家统计局的相关数据，我国近些年的年交通事故量均大于 20 万起，仍处于较高的水平，这不仅造成巨大的经济财产损失，更是威胁到了民众的出行安全。同时根据 2018 年国内对于道路交通事故主要原因比例分布的分析数据显示，86%的道路交通事故由机动车违法造成[3]。根据美国交通部数据显示，近 94%的致命车祸都是由人类驾驶员的失误造成的，人类驾驶员由于疲劳驾驶、酒驾、操作不规范、路况判断失误等造成交通事故成为传统出行方式的一大痛点。

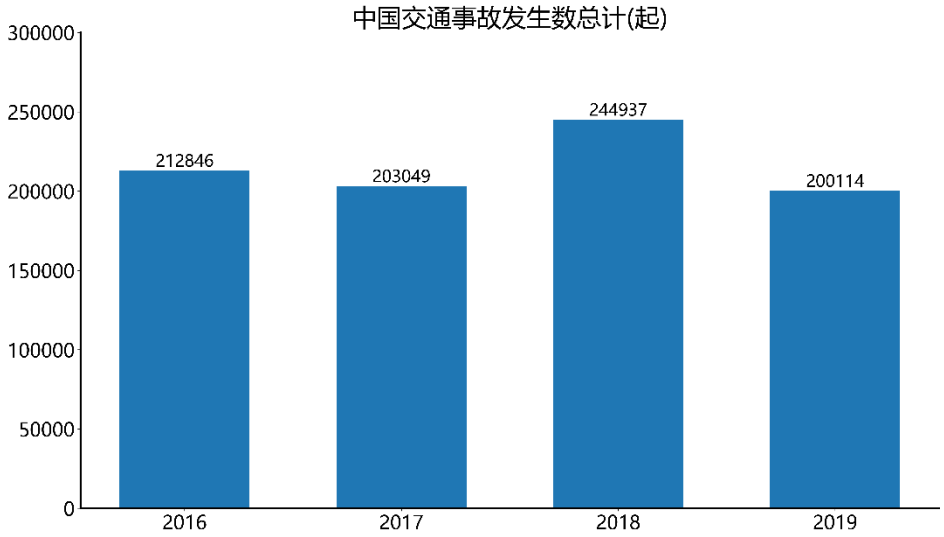


图 1-1 2016-2019 中国交通事故发生数总计图

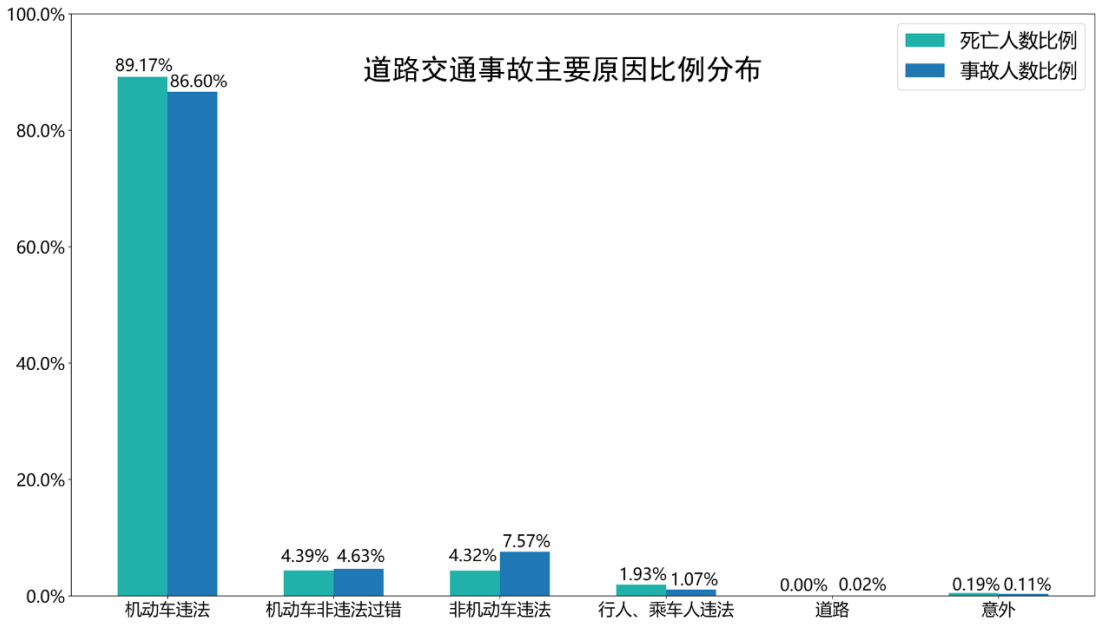


图 1-2 2018 年道路交通事故主要原因比例分布图

随着车联网、5G 等信息技术的快速发展，自动驾驶相关的技术成为了近些年汽车领域与交通领域研究的重点与热点，许多国家发布了关于自动驾驶技术的相关政策，并纳入国家顶层规划。自动驾驶被认为会是改变传统交通与车辆领域、提高道路交通通行效率、提高道路交通安全的新兴技术。2018 年 1 月，国家发改委发布《智能汽车创新发展战略》[4]，提出到 2020 年，智能汽车新车占比达到 50%，2025 年中国标准智能汽车的技术创新、产业生态、路网设施、法规标准、产品监管和信息安全体系全面形成，到 2035 年，中国标准智能汽车享誉全球，率先建成智能汽车强国，全民共享“安全、高效、绿色、文明”的智能汽车社会。2018 年 12 月，工信部发布《车联网（智能网联汽车）产业发展行动计划》[5]，提出到 2020 年，实现车联网（智能网联汽车）产业跨行业融合取得突破，具备高级别自动驾驶功能的智能网联汽车实现特定场景规模应用。自动驾驶车辆依托于先进的传感器设备与人工智能等算法，实现无需人类驾驶员操控的机器自主驾驶。具备自动驾驶功能的车辆作为一种新兴车型具备降低人工成本、减少燃油消耗、提高道路交通安全水平等特点。车辆自主驾驶能够避免疲劳驾驶、酒后驾驶等人为因素所引起的交通事故[6]。另一方面，通过感知与车联网等技术，能够使自动驾驶车辆提前预知道路交通状况，从而做出相应的反应来规避风险。

国内外众多公司已经在自动驾驶相关业务领域研发多年，国内的百度公司2014 年开始启动“百度自动驾驶汽车”，同时在 2017 年 4 月发布了一项名为“Apollo（阿波罗）”的开源自动驾驶仿真软件平台新计划，在 2020 年 9 月，百度 Apollo 正式宣布在北京开放自动驾驶出租车服务 Apollo Go。除了百度以外，国内一些其他的公司也在自动驾驶车辆领域开展了深入的研发工作，例如华为、腾讯、Momenta、小马智行等。国外方面，以美国为代表，德国、英国、日本等国家均在自动驾驶领域开展了深入的研发工作。谷歌公司早在 2009 年就公开宣布开发无人驾驶汽车，并于 2016 年成立了独立的自动驾驶研发公司 Waymo，其发展定位在于提供系列的人工智能与智慧出行相关服务。2019 年，Waymo 收购了专注于研究模仿学习在自动驾驶中应用的 Latent Logic 公司，标志着其认为机器学习相关人工智能算法是突破自动驾驶技术的关键。

在自动驾驶汽车被大众接受为一种新的交通方式之前，它必须被证明至少和人类驾驶车辆一样安全[7]。近些年，自动驾驶车辆相关的事故频发，车辆的驾驶安全性成为了热点。2018 年 3 月在美国亚利桑那州发生了全球首例自动驾驶车辆在公共路面撞人致死的交通事故，Uber 的一辆自动驾驶汽车与一名正在过马路的行人相撞，导致行人死亡。特斯拉作为自动驾驶领域的黑马，也因为车辆安全问题被大众质疑，根据特斯拉 2020 年第三季度的报告，在其启动自动驾驶系统的驾驶过程中，每行驶 459 万英里（约合 738 万公里）会发生一起事故。

Nidhi Karla 指出自动驾驶车辆需要至少数亿公里里程以上的测试才能够保证车辆驾驶的安全性，在这种情况下，即使每天二十四小时不断地进行实际道路测试，也需要数十年的时间才能完成一辆自动驾驶车辆的测试实验。因此，自动驾驶技术的发展亟需一种新的方法来替代完全使用实际道路进行测试的方法，虚拟仿真测试则是当今比较合适且流行的方法之一。仿真具备可加速、成本低等优点，同时仿真环境具有模拟车辆动力学性能以及高精度还原现实世界的能力。对于自动驾驶车辆的发展而言，采用计算机仿真模拟的手段来对于自动驾驶车辆算法进行安全性的测试未来将成为该领域的一种规范与标准，虚拟仿真测试与真实物理测试将构成相互结合的有机整体，两者缺一不可[8]。

然而现有的仿真软件在自动驾驶车辆模型方面都较为欠缺，如何在仿真中构建自动驾驶车辆的模型成为了现今的重点与难点。强化学习算法作为新兴的人工智能算法在游戏、机器人等领域发展火热，越来越多的研究人员将强化学习算法应用于交通领域。强化学习算法由于其具备良好的解决连续决策问题的能力，近些年也被应用于自动驾驶模型中。Cathy Wu[9]团队结合交通仿真软件 SUMO 与深度强化学习提出了命名为 FLOW 的应用框架，但其定义的自动驾驶模型较为简单，且仅适用于如环形道路、匝道等特殊场景，具有较大的局限性。同时，对于自动驾驶仿真软件而言，虽然具备高性能的车辆动力学模型与三维模型输入的能力，但其对于交通流模型的刻画比较欠缺，无法满足还原真实交通流从而达到自动驾驶算法测试的能力。

综上所述，虽然自动驾驶技术被认为是一种改变未来交通格局的新兴技术，但是在目前仍处于发展阶段，亟需结合仿真测试的方法对于自动驾驶算法进行优化。然而传统的交通仿真软件缺乏对于自动驾驶车辆模型的定义，基于强化学习的方法在这方面被验证了相关的可能性。

因此，本论文拟基于已有交通仿真软件进行二次开发，在考虑传统车辆动力学与交通流特征的基础上，结合强化学习算法建立高泛化性的自动驾驶车辆模型。同时，探索自动驾驶仿真软件与交通仿真软件结合的可能性，为自动驾驶算法虚拟仿真测试提供真实的交通流场景。

1.2 国内外研究现状

1.2.1 交通仿真研究现状

交通仿真主要通过运用计算机数字模拟的方法来反映复杂的道路交通现象，同时基于不同的设计方案为交通规划、交通管理与控制提供评价与验证方法，是深入进行交通研究的重要工具。回顾整个交通仿真行业的发展，按照发展的时间周期与刻画对象可以大致将交通仿真分为三大类。第一大类为宏观交通仿真软件，该类仿真软件以交通流为研究单元，仅考虑车流在拓扑路网上的分配问题,主要利用交通四阶段等方法，计算交通小区之间的交通量产生、吸引、分布，其典型

代表为 VISUM、TransCAD、CUBE、EMME 等。第二类为中观交通仿真软件，该类软件对交通流的描述以若干辆车构成的队列为单元，能够描述队列在道路节点的流入流出行为，对路段车辆的换道行为也可以进行较为细致的描述。中观仿真模型其内核是动态交通模型结构，可用于探究路网交通系统运行状况的时空演变机理。动态交通模型结构主要由动态 OD 矩阵（Origin Destination Matrix）获取及动态交通分配两部分组成。应用广泛的中观交通仿真模型主要有 DynusT（Dynamic Urban Systems in Transportation）及 TRANSIMS（Transportation Analysis and Simulation System）等[10]。第三类为微观交通仿真软件，该类仿真软件以车辆、行人等智能体为研究对象，研究不同交通主体之间的相互影响关系及对于交通网络的影响。融合宏微观一体化、多模式交通、基于智能体的交通仿真技术，可以与大数据、人工智能、车联网等新兴技术紧密结合，近年来逐渐成为交通仿真领域的主流。目前广泛使用的微观交通仿真模型有 SUMO、VISSIM、Q-PARAMICS、TransModeler、AIMSUN 等。

交通仿真的核心与基础是对于交通模型的刻画与描述。在宏观层面，“四阶段预测法”是交通需求预测所应用的经典方法，其主要核心包括交通发生与吸引（第一阶段）、交通分布（第二阶段）、交通方式划分（第三阶段）、交通分配（第四阶段）。20 世纪 70 年代以来，基于四阶段理论体系的软件得以应用，代表性的有美国的 TRANPLAN、TransCAD、CUBE，英国的 TRIPS、加拿大的 EMME/2 及我国东南大学王炜老师团队开发的 TranStar 等[11]。微观层面的交通仿真核心是交通流理论与相关模型。交通流理论是指运用数学与物理的相关方法研究车流、人流等不同交通参与物在不同交通设施内的运动特征及规律，其研究与分析揭示了交通参与物系统层面的动力学与统计学特征。微观交通流模型主要包括跟驰模型（如刺激-反应类模型、安全间距类模型、优化速度类模型等）、换道模型（如 Gipps 模型、离散选择类模型、MOBIL 模型等）、元胞类模型（单车道、多车道模型、元胞模型等）、行人交通流模型（如社会力模型等）等[12]。

另一方面，随着城市多模式交通（如私家车、公交、地铁、自行车）的不断发展，基于活动与出行链的行为模型在交通仿真中逐步应用。活动模型的理论基础来源于瑞典地理学家 Hägerstrand[13]等人，其团队提出了人类行为的相关时空

约束界限，同时对于参与活动的可选择范围进行了约束限制。这类约束包括活动耦合、活动权限和活动能力。Chapin[14]提出了参与活动是满足人类本能需求的动机理论，同时研究发现影响活动参与方式的因素主要有家庭成员、义务和社会责任等影响因素。M.Fried[15]等人研究提出，个体在城市空间中不断地调整和适应自身的活动行为，行为适应过程的本质就是降低当前的期望需求、机会资源与约束条件之间的不平衡。在目前主流的基于活动的模型构建方法中主要采用效用最大化函数来优化个体对活动出行计划的选择。在时间和费用的约束条件下，个体选择众多活动出行计划中效用值最大的一个，因此，个人偏好决定了在时空限定条件下不同的个体如何选择活动。

表 1-1 交通仿真软件对比

仿真软件	MATSim[17]	SUMO	VISSIM	DTALite[18]	AnyLogic[19]
语言	JAVA	C++/Python	VB/VC 等	C++/python	JAVA
主要模型	智能体、基于活动模型	跟驰模型、换道模型	跟驰模型、换道模型	中观智能体模型	智能体模型
层次	微观	微观	微观	中观	微观
开源	是	是	否	是	否
局限性	接口复杂、难以二次开发、不具备自动驾驶模型	微观跟驰换道模型需要一定程度优化、不具备自动驾驶模型	大量基于随机选择和概率分布、无法二次开发	颗粒度不足、无法二次开发、不具备自动驾驶模型	不具备交通模型、不具备自动驾驶模型
多模式仿真	支持	支持	支持	不支持	一般
大范围路网	是	是	否	否	是
计算效率	高	高	高	低	一般

Simulation of Urban Mobility（SUMO）[16]是一款由德国航空航天中心交通运输研究所开发的开源的、多模式微观交通仿真软件。该软件支持大规模复杂路网的输入及公交、行人、地铁等交通模式，还具有强大的 TRACI Python 接口，能够灵活地与现有人工智能相关的 Python 库（如 Tensorflow，OpenAI）等结合，具有良好的二次开发基础与条件。在现如今人工智能与大数据技术快速发展的时代，SUMO 以其微观、开源、接口丰富、支持大范围交通网络、具有良好的开发

者社区等特点，在交通规划、信号控制、多模式出行、自动驾驶等领域具有广泛应用。

综上所述，现有的交通仿真大部分不具备自动驾驶仿真的功能，但是对于交通模型的刻画大部分较为准确。微观交通仿真软件 SUMO 具备开源、支持多模式交通、二次开发可拓展性高等特点，本论文拟基于 SUMO 进行二次开发，针对城市多模式交通、自动驾驶车辆进行模型构建。

1.2.2 自动驾驶研究现状

自动驾驶汽车是指通过搭载先进的车载控制器、数据处理器、传感器等装置，借助 5G、V2X 等现代移动通信与网络技术实现，从而获取道路、交通参与物实时共享信息，并具备在复杂道路行驶环境下的传感感知、决策规划、控制执行等功能的系统。一个完整的自动驾驶系统基于车载传感设备与环境感知算法对周围道路交通环境进行精准感知，并基于感知信息通过车载中心设备自主地控制车辆的转向和速度，使车辆能够安全、可靠地行驶，并达到预定目的地[8]。

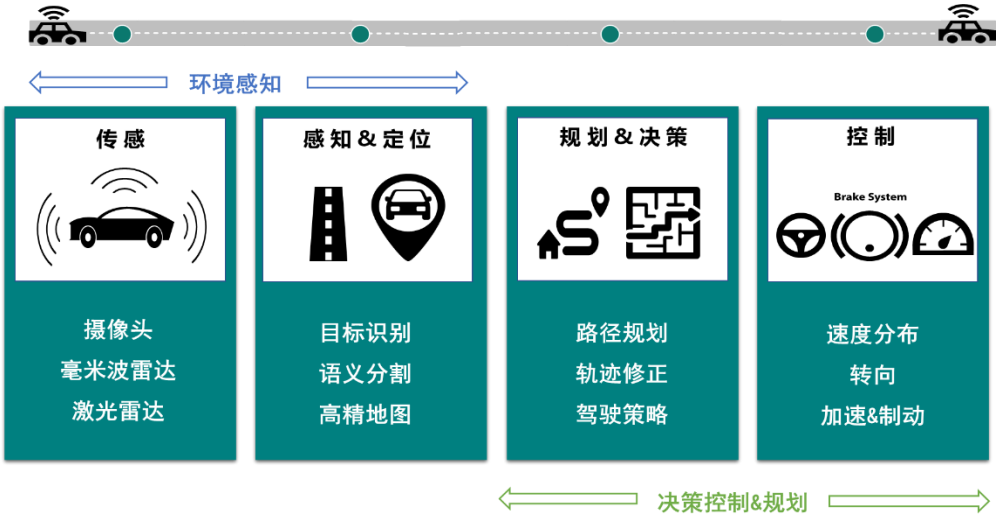


图 1-3 自动驾驶车辆核心技术体系图

环境感知与识别：

随着人工智能技术和 5G 技术的出现与革新，国内李国红[20]等研究人员经过相关调研发现 2015 年后全球自动驾驶技术专利申请人数呈井喷式增长。在自动驾驶关键技术的感知、决策、控制三大模块中，环境感知技术逐渐成为国内外自

动驾驶行业专利申请的重点，我国环境感知专利申请数量居全球首位。实现精准的环境感知是自动驾驶车辆正常行驶的重要保障。HATA[21]提出一种使用道路特征检测的方法对车辆进行定位，将车辆感知系统获取的车辆形态信息与已有网格图相匹配，对车辆进行较为精准的定位。来飞、黄超群和胡博[22]等人提出除了环境感知，驾驶员对车辆的信任程度也在一定程度影响自动驾驶车辆的运行水平，自动驾驶车辆未来的发展应该要攻克特殊场景和地理环境限制的难题。汤立波[23]，付长军[24]认为车联网产业呈现不断融合发展的趋势，丰富的车内外信息能够促使车辆向智能化和自动化发展，车联网将是助力自动驾驶的关键技术。

决策与规划：

在自动驾驶决策与规划层面，路径规划是其中的关键问题之一。杨文彦[25]结合了社会力模型和不同行人的异质性，对人车混行路口的行人轨迹进行预测，来为自动驾驶汽车的决策与路径规划提供依据。韩小健[26]考虑了自动驾驶客车与周围障碍物的实际尺寸，设计了一种改进 RRT 算法来解决结构化道路环境下的自动驾驶客车路径规划问题。吴伟[27]等人提出了自动驾驶环境下的交叉口自动驾驶路径规划模型，研究认为自动驾驶环境下交叉口不需要信号控制，可以针对单个车辆进行路径优化与建模。Wang[28]建立了一个基于支持向量机（SVM）的自动驾驶车辆决策模型。Rehder[29]等人使用贝叶斯网络（BN）来预测自动驾驶车辆实行车道变换的概率。对于自动驾驶决策规划问题，机器学习的算法应用越来越广泛，但是很少有研究人员将传统的规划算法与机器学习算法进行结合与优化。

车辆行为控制：

自动驾驶车辆的行为控制问题可以分为横向控制和纵向控制[30]。横向控制主要指让车辆通过合理的转向运动控制来沿着已经规划好的、合理的全局路径行驶。同时，以自动驾驶车辆与全局路径之间的横向距离和行驶方向的偏差最小化为目标，通过调整转向输入来提高车辆运动的平滑性与稳定性。纵向控制则是车辆根据道路线性与特性，在满足运动学约束与安全车距等的前提下，通过控制油门和制动系统来使车辆保持期望的速度和加速度。目前的大多数自动驾驶控制研究都只关注了其中的一种控制从而简化了自动驾驶车辆的控制问题[31]。

Moriarty[32]等人在早期的时候结合监督学习和强化学习，开发了一种公路环境下的车道选择策略模型。其实验结果表明，装备有学习模型控制器的车辆总是设法将速度保持在接近期望的速度，从而减少了车道换道的次数。Rausch[33]等人也基于监督学习来实现自动驾驶的转向控制。Eraqi[34]等人将时间上下文信息引入深度学习的自动驾驶转向模型，基于视觉和动态时间依赖性同时结合 C-LSTM 模型来训练自动驾驶车辆。在自动驾驶车辆的纵向控制应用中，机器学习算法应用广泛，例如自适应巡航控制设计（ACC）模型[35]。ACC 模型可描述为复杂非线性系统的最优跟踪控制问题[36]。Huang[37]等人提出了一种基于参数批处理的 Actor-critic 强化学习算法，实现基于 Actor-critic 算法的自动驾驶车辆的纵向控制。同时设计了一个多目标奖励函数来奖励算法的追踪精度和行驶平顺性。该方法通过在各种驾驶环境（如平坦、光滑、倾斜等）下的现场试验得到了验证。

自动驾驶仿真测试：

为了保证自动驾驶算法实现落地应用，需要进行高效的测试，除了现实物理场景的自动驾驶测试之外，还衍生出软、硬件在环仿真测试，驾驶员在环仿真测试，车辆在环仿真测试。张珊[38]等提出了一套虚拟测试方法，在 CarMaker 中复现 CIDAS 数据库中的一例交通事故，转化驾驶员在环的仿真模拟的真实状态，验证了驾驶员在环测试对交通事故场景的应对能力。但是该实验也展示出其对模型依赖性较大的缺点，模型的修改和表现受驾驶者经验影响较大。葛雨明[39]等人提出数字孪生是自动驾驶车辆仿真的新兴测试方法。通过有限环境下的车联网条件，对物理环境进行复杂道路场景映射。从而实现快速仿真场景，但是难以验证其真实情况。目前这种方法仍处于起步阶段，并且强依赖于网络与数据，且在信息传输与环境映射等方面仍有巨大的发展空间。汤辉[40]等人基于虚拟驾驶场景，结合虚拟背景交通因素和驾驶模拟器对自动驾驶车辆进行检测。其实验结果表明模拟驾驶器控制的背景车可以很好地模拟现实中的驾驶行为，同时具备进行多车协同测试的可能性。Suh J[41]等人总结了现代汽车工程师使用航空驾驶模拟器的原型与机理并搭建了闭环仿真平台，使用该平台进行自动驾驶汽车仿真测试，不仅提升了测试效率，还实现了交通参与者与交通要素的实时交互，为应对更高级别的自动驾驶测试提供了方向。周干[42]等人总结得出早期的自动驾驶车辆测试

主要是从车辆的动力学角度进行仿真测试，同时尝试应用 CarSim 进行自动驾驶算法测试。当 ADAS 系统出现后，辅助 ADAS 功能进行验证的仿真软件开始大量出现，如 Prescan，Carmaker，Carcraft 等[43]。Carmaker 是一款集成车辆动力学、ADAS 系统的自动驾驶仿真软件，Carcraft 具备使用真实世界的驾驶回放数据进行测试的功能从而验证自动驾驶算法的可靠性，同时也支持基于软件的虚拟场景重构。同时，也有相关研究人员开始尝试使用多款软件结合进行测试，Duy Son [44]提出了结合 Amesim 和 Prescan 的联合仿真测试框架。另一方面，也有许多研究人员使用典型的交通仿真软件如 CORSIM、NETSIM、OREAU、SUMO、VISSIM 以及 PRAMICS 等进行自动驾驶车辆算法仿真虚拟测试[45–48]。

目前众多的研究人员基于不同的软件进行自动驾驶虚拟仿真测试平台的开发，但对于一个完整、成熟的自动驾驶仿真平台而言，其需要具备以下功能：能够接入自动驾驶感知和决策控制系统、具备还原静态场景、实现动态仿真、传感器仿真、车辆动力学仿真、并行加速计算等功能[8]，如图所示。自动驾驶算法与仿真平台相互结合才能形成闭环，达到持续迭代和优化的状态。因此，基于已有的仿真软件开发一款满足自动驾驶虚拟仿真测试需求的仿真平台与框架对于自动驾驶算法测试与发展非常重要。本论文拟结合二次开发的微观交通仿真软件 SUMO 与自动驾驶仿真软件 Carla，实现实时协同联合仿真，从而满足自动驾驶虚拟仿真测试需求。

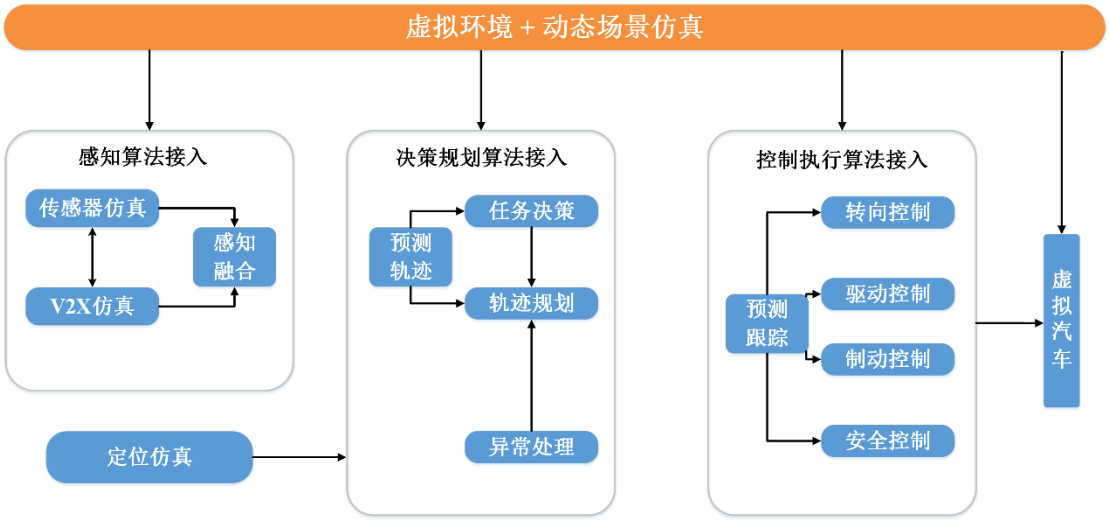


图 1-4 自动驾驶虚拟仿真测试平台框架图

1.2.3 强化学习研究现状

强化学习算法作为一类机器学习算法，不同于监督学习与无监督学习，凭借着独特的试错学习、反馈机制近些年成为人工智能领域的热门。2016 年基于数百万人类围棋专家的棋谱进行强化学习训练的人工智能程序 AlphaGo 以 4:1 总比分战胜了围棋世界冠军，2017 年 Deepmind 团队又创新性地让其自由、随意地在棋盘上进行自我博弈并不断自我学习，舍弃了所有人类的棋谱数据。经过三天训练的新版本 AlphaGo Zero 以 100:0 的比分战胜了 AlphaGo，取得了压倒性的胜利，这也引发了众多研究人员对于强化学习算法的强烈关注。

强化学习在交通领域中应用广泛，Abdulhai 和 Kattan[49]早在 2003 年就分析了强化学习算法在交通工程和交通控制领域的潜力，同时提出了使用 Q-learning 的方法来创建智能交通系统。目前强化学习在交通领域最广泛与流行的应用是信号控制问题。如何优化一个城市区域的交通信号控制方案，使给定交叉口的车辆总等待时间最小化是交通信号控制的关键问题。近年来，许多研究人员通过使用基于强化学习的多智能体理论来解决信号控制优化问题，并出现了诸多研究成果。Abdoos[50]考虑了多个交叉口的环境，并结合多智能体的 Q-learning 算法，以减少交叉口平均等待时长和提高网络通行流量为目标的同时，防止了网络出现过饱和的情况。Khamis 和 Gomaa[51]综合考虑最小化出行等待时间、最大交叉口流量、中等车速下最小燃料消耗，在保证干线道路绿波的条件下，建立强化学习目标函数及其约束。同时使用 GLD 交通模拟器及智能驾驶员模型（IDM）模型对道路动态特性（交通需求、天气条件）进行模拟，构建多智能体分布式交通信号控制器，比传统单目标控制器效果有明显优化。Zhao[52]将信号灯分组控制加入了问题定义中，提出了一种模糊控制的多智能体分组信号灯实时仿真方法。Jin 和 Ma[53]在使用信号灯分组控制基础上加入了随机最优控制策略，将每个信号灯组建模为一个智能体，在交叉口动态生成相位序列。区域交通信号控制可以看作是一个复杂的多智能体连续时间调度问题，其涉及到协同的、不协同的、自动控制的智能体之间的博弈。Clempner 和 Poznyak[54]利用连续时间马尔科夫博弈，通过计算纳什平衡点，结合两阶段迭代法，分别计算出平衡点的初步近似值与前一步的调整值，并通过一个三叉路口的算例，验证了该方法的有效性。Mckenna 和 White[55]

基于 SUMO 交通仿真环境建立加拿大渥太华市的仿真交通模型，提出了一种自适应信号灯控制系统算法，其对交叉口平均等待时长的优化效果明显优于固定信号控制器。

另一方面，随着深度强化学习的不断发展，将深度强化学习算法应用于自动驾驶问题的解决是强化学习在交通领域的另一大应用。随着 David Sliver 及其团队提出了人工智能的终极目标是：人工智能=深度学习（Deep Learning）+强化学习（Reinforcement learning）=深度强化学习后，深度学习与强化学习的结合越来越引起研究人员的重视。2016 年，Sallab[56]等人基于深度强化学习方法在开源赛车模拟器（The open racing car simulator，TORCS）上实现了自动驾驶汽车的车道保持控制，并对比了离散空间的 DQN 方法和连续动作空间的 DDAC 方法，证明了 DDAC 算法能够获得更好的车辆控制效果以及更为平滑的运动轨迹。Chae[57]等人利用 DQN 算法对于智能体进行训练，并尝试让其学习处理行人横穿马路的场景，从而实现车辆的自主制动控制。Zong[58]等人为了实现自动驾驶智能体的自主避障，采用 DDPG 算法对于车辆智能体进行训练，通过学习加速度和转向控制来达到避障目标，并在 TORCS 环境中进行了测试。Shalev-Shwartz[59]等人将长短期记忆网络（Long short term memory networks，LSTM）算法与强化学习算法相结合，使用游戏环境作为训练环境，解决了自动驾驶车辆的纵向控制以及汇入环岛的控制问题。2018 年，Maximilian[60]等人将强化学习的 A3C 算法结合到自动驾驶车辆中，其研究基于 CNN 对于车载摄像头获取的图像数据进行处理，采取端到端的方式训练自动驾驶车辆，并在仿真环境中取得了较好的效果。2019 年，英国的自动驾驶公司 Wayve 的 Alex Kendall[61]等人提出了一种基于 DDPG 算法的自动驾驶框架，该框架仅使用单目图像作为输入进行车道线跟踪学习，并在现实环境中进行了实车测试。

综上所述，强化学习算法已经被证明了其在解决连续决策问题上的优势，同时在交通领域已有广泛的应用。但是目前基于强化学习的模型还存在比较大的局限性。本论文拟结合交通工程领域知识，针对自动驾驶车辆进行强化学习模型构建，同时基于微观仿真软件 SUMO 进行二次研发。

1.3 研究内容与技术路线

1.3.1 研究内容

本论文的主要研究内容分为以下四个部分。

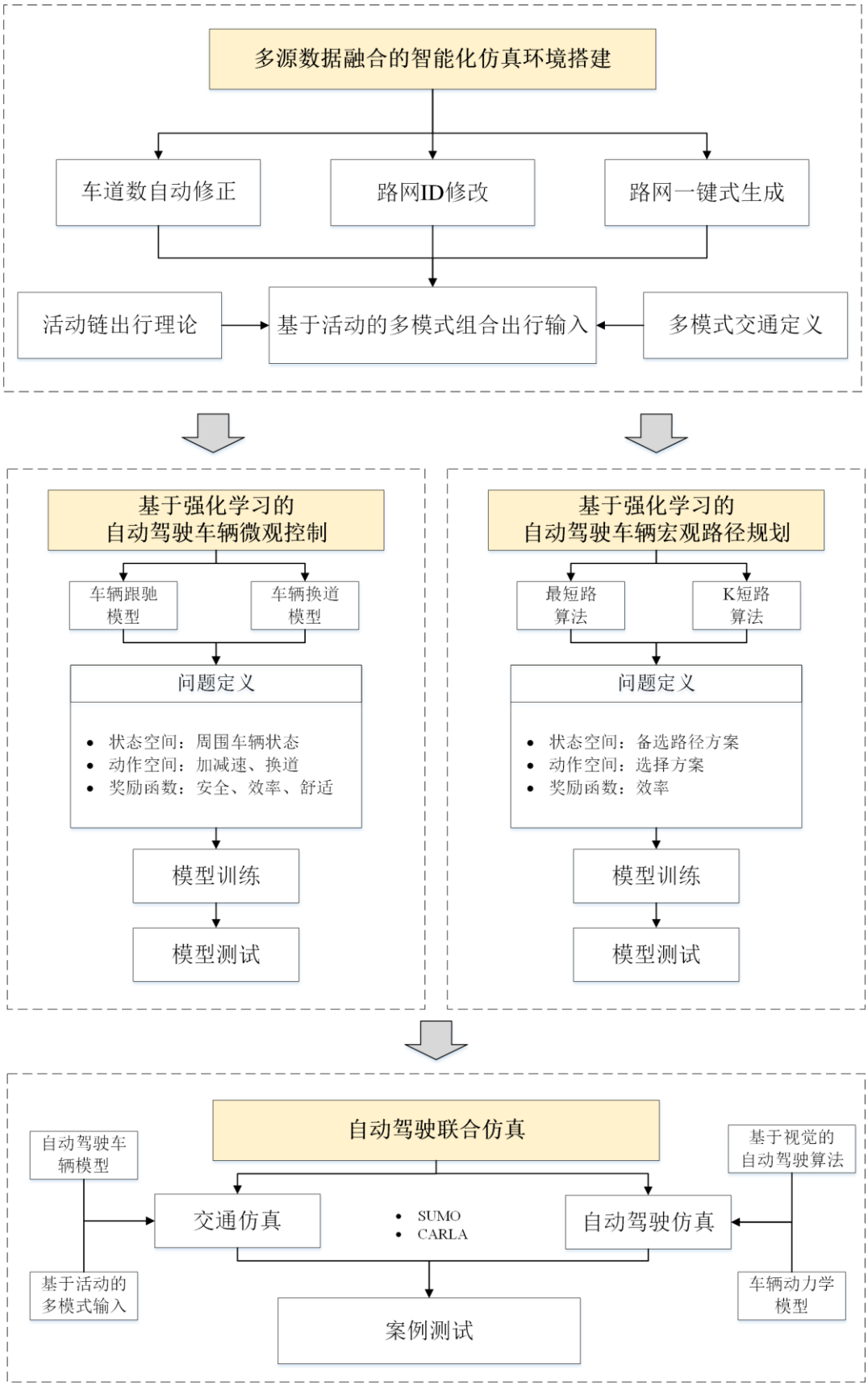
（1）基于交通仿真软件 SUMO，考虑不同的输入数据需求，研究基于 GIS, Open street map 不同输入形式的自动化大规模复杂路网封装方法，包括车道数修正、道路 ID 修改，使其能够满足精细化智能交通仿真的需求（例：智能体能够根据路网状态的变化进行反馈，采取相关动作）大幅提高交通仿真基础路网建模的效率与灵活性，同时基于活动出行理论构建基于仿真的多模式活动组合出行输入。

（2）分析传统的车辆微观控制模型与理论，将自动驾驶车辆的控制与决策分为两部分：宏观路径选择与微观行为控制。考虑自动驾驶车辆行驶过程中与周围车辆的相对状态信息，基于强化学习的 DQN 算法，实现自动驾驶车辆的加减速控制与换道决策。研究与 SUMO 的 Traci 接口交互方法，实现基于 SUMO 仿真的自动驾驶车辆微观控制。

（3）研究传统的宏观路径选择算法原理，考虑自动驾驶车辆路径选择的特征，结合 K 短路算法，构建基于强化学习的自动驾驶车辆宏观路径选择模型。基于 SUMO 的 Traci 接口交互逻辑，研究自动驾驶车辆宏观路径选择模型与微观控制模型的一体化整合方法，实现基于 SUMO 仿真的自动驾驶车辆一体化控制。

（4）基于 SUMO 与 Carla 的 Traci 与 Python API 接口，分析两款仿真软件的通讯逻辑与核心模型的特征与机理，研究两款仿真软件实现实时、协同仿真的技术与方法。基于视觉的自动驾驶算法嵌入仿真软件 Carla，研究基于联合仿真的自动驾驶测试方法，并选取典型城市区域进行实例验证。

1.3.2 技术路线



1.4 本章小结

本章确定了研究背景和研究意义，综述了交通仿真、自动驾驶、强化学习三个方面的研究成果。在总结目前研究不足的基础上，提出了本论文的研究目的，拟定了主要研究内容，确定了研究技术路线。

第二章 多源数据融合的仿真环境智能化搭建

现有微观交通仿真软件如 Vissim 等都需要进行非常复杂的手动路网搭建工作，且效率低下，精度有限。而路网作为仿真的基础，模型的精准度直接影响了仿真的结果。本论文拟基于开源交通仿真软件 SUMO，同时考虑不同的输入数据需求，研究基于 GIS, Open street map, xml 等不同输入形式的自动化大规模复杂路网封装方法具有十分重要的意义，同时使其能够满足精细化智能交通仿真的需求（例：智能体能够根据精细化的路网信息进行采集与分析，采取相关动作），能够大幅提高交通仿真基础路网建模的效率与灵活性，减少人工手动搭建路网的大量成本。

截至 SUMO 目前发布的最新版本 1.8.0，虽然支持使用官方提供的 Netconvert 插件将 OSM，SHP，Opendrive 等多源格式路网转换为 SUMO 的投影坐标系格式路网文件，从而实现快速路网搭建，但是各类形式的输入自成体系，无法互相补充与校正，同时存在道路、交叉口 ID 随意命名，混乱等问题，不利于进行仿真分析与智能体训练环境的二次封装及开发。同时，OSM 格式路网作为 SUMO 官方推荐的地图数据源存在中国城市道路车道数大量缺失与错误的问题，其会导致仿真结果与现实差距过大。而 SHP 格式路网在使用 SUMO 官方的 Netconvert 插件转换时会出现道路断开，扭曲等问题。因此，本论文考虑采用信息完整的 SHP 文件对于 OSM 数据进行修正与补全，基于新提出的道路 ID 命名规则实现路网的智能化搭建。完成路网搭建后，基于 SUMO 的多模式定义规则完成传统公交、地铁等多模式的出行方式构建，并结合 SAGA[62]（一款基于 SUMO 的开源项目包）实现基于活动的多模式组合出行。

2.1 SUMO 路网属性介绍

目前，SUMO 中的路网结构主要有路段（edge），节点（junction），转向关系（connection），车道（lane）这四个基本组成部分。这四个基本组成部分均包含丰富的属性信息。

2.1.1 路段

在 SUMO 中，路段（edge）主要包括道路名称（ID），路段起点（from），路段终点（to），道路重要程度（priority），功能（function），道路属性（type），各属性具体含义如表 2-1 所示。

表 2-1 路段属性描述表

属性名称	类型	备注
ID	字符串	路段名称，具有唯一性
from	字符串	路段的起点
to	字符串	路段的终点
priority	整数	表明道路的重要程度
function	Normal，internal 等值	表示路段功能，默认为 normal
type	字符串	定义道路属性

其中，function 表示路段的功能有 normal，connector，internal，crossing，walkingarea 等，normal 表示正常路段，如城市道路或高速公路的某一路段。internal 表示交叉口内的路段，internal 类型的路段为节点内部道路，主要是为了定义车辆的等待位置等信息。crossing 表示交叉口的人行横道，walkingarea 表示行人通行区。其中 type 用来定义道路的属性，如 highway. motorway，highway.secondary 等，用来确定道路限速，车道数量，运行通行车辆种类等信息。

2.1.2 节点

在 SUMO 中，节点（junction）表示为交叉口或路段的连接处，主要有节点名称（ID），横坐标（x），纵坐标（y），交叉口连接车道（intLanes），交叉口内部车道（incLanes）等。同时，由于节点内的道路具有不同的优先级，其通过基于数字索引的规则来区分不同车道的优先级别。同时对于交叉口内部道路，其预先定义的一些规则，如从左转的车辆具有优先通行权、右转车辆待需要等待直行行人等，同时也支持用户自定义内部交叉口道路，各属性含义如表 2-2 所示。

表 2-2 节点属性描述表

属性名称	类型	备注
ID	字符串	节点名称，具有唯一性
x	浮点数	节点的横坐标
y	浮点数	节点的纵坐标
intLanes	列表	交叉口连接的车道
Inclanes	列表	交叉口内部车道

2.1.3 转向关系

转向关系（connection）定义了对于某个交叉口，进口车道与出口车道的连接情况。其基本的属性有如下几个，进口路段（from），出口路段（to），进口车道（fromLane），出口车道（toLane），通道（via），信号灯（tl），连接方向（dir）。各个属性的含义如表 2-3 所示。

表 2-3 转向关系属性描述表

属性名称	类型	备注
from	字符串	连接的进口路段
to	字符串	连接的出口路段
fromLane	索引	进口车道的索引
toLane	索引	出口车道的索引
via	字符串	交叉口内部车道（internal 型）
dir	枚举类型	连接方向
tl	字符串	控制该点的交通灯 ID

其中，dir 有如下的选项：

s 代表直行，t 代表转向，l 代表左转，r 代表右转，L 代表部分左转，R 代表部分右转，invalid 代表无方向。转向关系决定了车辆在道路与道路的连接之前能够有哪些可选的车道，如高速公路的加速路段则需要基于汇入匝道与主线道路的转向关系进行相应的调整，来达到与现实相符的汇入场景。

2.1.4 车道

车道（lane）是包含在路段内的路网结构，其主要包括如下几个属性，车道名称（ID），索引（index），允许通行车辆类型（allow），禁止通行车辆类型（disallow），限行速度（speed），车道长度（length），车道宽度（width），几何位置（shape）。各个属性的含义如表 2-4 所示。

表 2-4 车道属性描述表

属性名称	类型	备注
id	字符串	车道名称，具有唯一性
index	整数	车道的编号
allow	列表	允许通行的车辆种类
disallow	列表	禁止通行的车辆类型
speed	浮点型	最大限速
length	浮点型	车道的长度
width	浮点型	车道宽度
shape	浮点型	车道中线坐标

综上所述，SUMO 可通过定义路段，节点，转向关系和车道这 4 个基本元素内的属性来进行路网的快速搭建。但仍存在许多问题，如 1、一些车道级别的转向关系与实际情况不符 2、OSM 数据缺少车道数信息，只能通过道路类型来进行推测 3、虽然支持多种类型电子地图的导入，但各类输入形式无法相互补充校正 4、道路与节点的 ID 混乱，不利于进行智能体训练环境的二次封装。针对以上的问题，本研究通过将 SHP 和 OSM 两类电子地图的融合，实现基于多源输入地图数据的智能化 SUMO 路网搭建工作。

2.2 多源数据融合的仿真路网优化

SUMO 的路网文件描述了地图上交通相关的部分以及仿真中车辆行驶的道路和交叉路口。从宏观上看，SUMO 的网络是一个有向图。道路交叉的节点在路网中被称为“junctions”，也代表了交叉口。“edges”代表了一段道路，其包含了车

道“lanes”。在交叉口中存在“internal edges”，车辆在交叉口的行驶需要遵循其规则。“connections”表示车道与车道的连接情况，如左转，直行等。具体来说，SUMO实现一次仿真相关的文件格式与含义如表所示。

表 2-5 SUMO 基础仿真配置文件表

文件格式名	含义	功能与内容
*.edge.xml	路段信息文件	描述路段信息
*.node.xml	节点信息文件	描述节点信息
*.net.xml	整体路网信息文件	描述路网整体信息（可替代以上两种文件）
*.rou.xml	车辆输入文件	描述车辆类型、起终点、经过路段、出发时间等
*.add.xml	额外信息文件	描述如公交站点信息、poi 信息、停车场信息等
*.sumocfg	配置文件	描述此次仿真的时间步长、输入、输出文件等

2.2.1 仿真路网车道数校正

OpenStreetMap（以下简称 OSM）是一款开源的、面向全球的世界地图。OSM作为高质量的地图数据被广泛的应用于各项交通研究中，同时也被 SUMO 官方作为推荐的仿真路网搭建的数据来源。但是由于中国的 OSM 数据存在信息更新频率较为缓慢与滞后等问题，其质量相较于国外存在较大的差异，特别是车道数的相关信息在中国许多城市中缺失严重。因此，将 OSM 数据作为仿真路网的输入数据仍需要进行优化与调整。

基于上述思路，本论文基于 OSM 数据的结构与属性构建面向不同道路属性的 SUMO 路网关系表，并使用包含准确车道数信息的 SHP 文件对于 OSM 格式路网进行补全，来弥补 OSM 数据车道数不准确的问题，最后结合 Netconvert 插件完成基础 SUMO 仿真路网的搭建。

表 2-6 OSM 路网等级与类型对应表

OSM 路网等级字段	对应的城市道路
motorway	高速公路
trunk	快速路
primary	主干道
secondary	次干道
tertiary	城市次要车行道路立交、匝道
motorway_link	高速公路立交，匝道
trunk_link	立交，匝道，桥上引道，机场进站快速路，国道改道
primary_link	城市主要车行道路立交，城市主要车行道路匝道
secondary_link	城市次要车行道路立交、匝道
tertiary_link	支路匝道、三级公路匝道

基于表 2-6 中的 OSM 路网与城市道路等级字段，制定与 SUMO 路网文件匹配的考虑中国道路特征的道路类型属性对应表格，部分数据对应关系如下：

<type id="highway.primary" numLanes="3" speed="13.89" priority="12" oneway="false" disallow="rail rail_urban rail_electric rail_fast tram ship"/>

<type id="highway.secondary" numLanes="2" speed="13.89" priority="11" oneway="false" disallow="rail rail_urban rail_electric rail_fast tram ship"/>

<type id="highway.footway" numLanes="1" speed="2.78" priority="1" oneway="true" width="2" allow="pedestrian"/>

基于上述对应关系表，完成基础的 SUMO 仿真路网搭建。但对于拥有完备车道数量信息的 SHP 文件情况下，采用车道数修正的方法对于 OSM 数据进行补全修正，主要步骤如下：

步骤一：SHP 文件中的坐标系一般不是标准的 WGS84 坐标系，其中的经纬度等信息均经过一定的坐标偏移，如 GCJ02 坐标系，百度坐标系等。因此，需要先对 SHP 文件的坐标系进行转换，使其坐标系与 OSM 进行匹配。

步骤二：将 SHP 转换为 XML 格式，并将道路进行分割，同时提取车道数信息，将其分割成为小段道路，取小段道路的中点，基于最近邻匹配算法寻找距离该点最近的道路，将 SHP 与 OSM 进行匹配。

步骤三：使用 Netconvert 生成 SUMO 支持的路网 net.xml 文件，将匹配的车道数不符的道路进行替换，并将其写入新的路网文件进行替换。

表 2-7 融合 SHP 与 OSM 的车道数修改算法流程

	初始文件转化
1	读取 SHP 文件，将 SHP 格式转为 shp.osm 格式（xml 形式）
2	基于 OpenStreetmap 下载同区域路网，通过 Netconvert 转换成 sumo 支持的 net.xml 格式路网
	文件解析
3	解析 shp.osm 文件，根据<nd ref>属性对 SHP 文件道路分割，形成多个小路段
4	提取 SHP 文件车道数信息和节点经纬度信息，与小路段建立对应关系
	最近邻匹配
5	提取小段道路的起终点经纬度，计算道路中点
6	输入道路中点，半径 r，基于 sumolib（SUMO 的 Traci 层 Python 库）内 getNeighboringEdges（）函数对道路进行匹配
7	获取所匹配到的道路，并按照与原道路中点的距离进行排序
8	提取距离最近道路，建立 SHP 道路与 OSM 道路的匹配关系
9	建立数据表，记录 SHP 道路车道数与对应 OSM 道路的车道数
10	查找两车道数不同的 OSM 道路，以 SHP 文件道路车道数为基准，进行修正
	SUMO 路网文件的修正
11	将需要修改车道数的道路写为 xml 数据格式，包括道路 id，起终点，优先权，车道数等信息，保存为更新后的 edges.xml 文件
12	在 Netconvert 中输入原路网文件和更新的 edges.xml 文件，输出修改车道数后路网文件

通过上述算法，即可完成基于 SHP 准确车道数信息的 SUMO 路网搭建工作，弥补了 OSM 数据缺少车道数的缺点，为后续路网搭建工作提供良好的基础。

2.2.2 仿真路网道路 ID 优化

SUMO 的路网文件以 XML 的形式进行存储，默认 SUMO 中对于道路（EGDE）的 ID 命名如下：

<edge id="838996685" from="7829422133" to="566899952" priority="11" type="highway.secondary">

<lane id="838996685_0" index="0" allow="pedestrian" speed="13.89" length="7.92" width="2.00" shape="3145.19,4156.31,3153.05,4157.29"/>

<lane id="838996685_1" index="1" allow="bicycle" speed="13.89" length="7.92" width="1.00" shape="3145.01,4157.80,3152.87,4158.78"/>

</edge>

默认的道路命名规则采用随机数对于道路 ID 进行命名，无法体现道路与道路、道路与交叉口间的关系，难以从节点、路段等进行对于路网拓扑结构进行判断。另一方面，对于特定路段的仿真，道路的随机、混乱命名会耗费使用者大量额外的时间去校正道路正确性。

因此，本论文提出了一种考虑路网拓扑关系的新型 SUMO 路网道路 ID 命名规则方法，替换了原有的 SUMO 路网道路 ID 混乱、随意的命名规则，同时继承 2.2.1 节更新的道路文件进行道路 ID 修改，完成路网的一体化优化构建，且该算法具有高度适用性，能够应用于几乎所有基于 OSM 数据与 Netconvert 插件自动生成的路网文件。

新路网道路 ID 命名规则：道路起点 ID@道路终点 ID#0（对于两个节点之间含有多条道路，则以道路起点 ID@道路终点 ID#0，道路起点 ID@道路终点 ID#1 的方法进行命名）。

根据 2.1 节介绍的 SUMO 路网属性，路段、交叉口、车道、转向关系各个属性间存在相互依赖关系，因此需要结合路网各属性间的关系来进行道路 ID 的更新与替换，避免出现 ID 不匹配等问题。具体的基于 SUMO 路网文件的路网道路 ID 修改方法见表 2-8 所示。

表 2-8 考虑路网拓扑关系的新型路网道路 ID 修改算法流程

	解析 xml 文档:
1	读文件，将 xml 文档加载到内存
2	解析 edges 标签下元素，获得 edges 表及 lanes 表
3	解析 junctions 标签下元素，获得 junction 表及 request 表，将 junctions 表中 inclanes 及 intlanes 分别拆解为 edge 及 laneindex 两列
4	解析 tlLogic 标签下元素，获得 tlLogic 表及 phase 表
5	解析 connections 标签下元素，将 via 列拆解为 viaedge 及 vialane
	确定 edge id 替换规则:
6	提取 edges 数据表中，所有 function 非 internal 的 edge，读取原始 edge id, from (junction id), to (junction id) 值
7	记录 Edge id 到 edge from @ to 的对应规则
8	计算 from @ to 出现次数，在 edge from@to 后标注 #[出现次数]
	替换 edge id
9	根据 edge id 替换规则，替换 edges 表中 edge id；lanes 表中 edge id；junctions 表中 inclanes edge id, intlanes edge id；connections 表中 from edge/to edge ID, connections 表中 viaedge id；
	写入新的 xml 文档
10	合并数据表中拆解的列
11	将数据表内容按照 xml 层级写入新的 net 文档

总的来说，以新 ID 命名规则命名的路网具有以下优势：

1、通过直接判断@字符前后节点 ID 是否互换即可判断是否为对向车道。

2、通过 ID 即可判断赋予车辆路径是否连通（如赋予车辆的路径为：“A@B”，“B@C”，“C@D”则表示该路径连通）。

3、便于为基于强化学习的智能体车辆进行路径定义（具体见 4.3 节）。

基于以上算法即可完成融合 SHP 与 OSM 数据的智能化路网搭建，在解决了仅使用 OSM 数据作为输入存在车道数不准确、路网 ID 命名混乱的问题基础上，为基于强化学习的智能体车辆搭建工作提供了良好的路网环境支持。

2.3 基于活动的多模式出行仿真输入

现今大部分的交通仿真更关注区域交通问题的优化与解决，很少对于个人细节到基于活动出行的行为进行深入地刻画。如比较主流的微观交通仿真 Vissim 等便不具备实现上述的功能，但对于一个现实城市区域级别的仿真而言，单个人的出行并不是执行一次就完成或直接在仿真路网中消失，每个人的出行都以链条形式存在，这对于更精准的还原交通场景具有重要意义。因此，本论文结合 SUMO 考虑通过定义公交、地铁等多模式出行方法，结合使用 SUMO 官方提出的 SAGA 插件来构建基于活动多模式组合出行输入来提供更真实、细致的交通场景。

2.3.1 多模式出行方式定义

由于大部分的公共交通具有固定的行驶线路（例如：传统公交、地铁等），因此在 SUMO 中可以通过预先确定公交站点的位置，并让公交在这些位置停留一段预先给定的时间来模拟公交的运行。其中，SUMO 中针对公交车辆与站点的具体定义行驶如下：

```
<busStop id="busstop1" lane="2/1to1/1_0" startPos="20" endPos="40" lines="100 101 102"/>
<vType id="BUS" accel="2.6" decel="4.5" sigma="0" length="12" minGap="3" maxSpeed="70" color="1,1,0" guiShape="bus"/>
<vehicle id="0" type="BUS" depart="0" color="1,1,0">
        <route edges="2/0to2/1 2/1to1/1 1/1to1/2 1/2to0/2 0/2to0/1"/>
        <stop busStop="busstop1" duration="20"/>
</vehicle>
```

上述定义了一个 ID 为“busstop1”的公交站点，ID 为“BUS”的公交车型，ID 为“0”的公交车，同时该车在仿真开始即出发，根据预先定义的路径，在 ID 为“busstop1”的公交站点停留 20 秒。根据以上定义形式，可参考定义地铁、小汽车等出行，为 SUMO 构建交通流量的输入。

2.3.2 基于活动的组合出行

在人们进行出行活动的时候，通常倾向于根据自己进行活动的位置（例如：工作、运动、学校等）执行每天的出行。因此，一个城市的交通出行应该是出行者们在一天中所进行的一系列出行的集合，同时这些出行也是基于他们所进行的活动的序列。在日常生活中，每个人通常会构建一个完整的日常出行计划，这个计划是由简单的出行连续组合而成。例如，一个人可以开车从家到学校，送完孩子后再开车去上班，同时将车停在上班地点的停车位。中午时，可以选择步行到公交站，乘坐公交车到离午宴餐馆较近的公交站点，再步行前往餐厅。类似的步骤也可以用在回办公室的路上，结束工作后再开车回家。由此可见，完整的出行计划由多个活动组成，每个活动都与位置、开始时间和持续时间相关。连接每项活动的往返行程的复杂性可能会有所不同，这取决于出行者当时的需求与起终点位置之间存在的出行计划[63]。

目前已有大量的研究人员针对基于活动的城市出行模式进行了建模研究[64,65]，研究表明，基于活动的出行模型更好地代表了个人的出行决策[66]，与传统的集计需求模型相比，提高了所产生的城市交通出行的真实性。

SAGA 是一款在 SUMO 基础上的基于活动的多模式出行场景生成器。SAGA 基于 OSM 文件提取构建多模式场景所需的数据，并按一定的步骤与顺序生成构建仿真场景所需的配置文件（例如，停车区域、建筑信息等）。同时，针对 OSM 数据在还原基于活动的交通场景时的缺失问题，SAGA 提出一套合理的缺失信息填补规则。本论文拟以更新后的路网文件作为输入，结合 SAGA 生成器构建基于活动的多模式组合出行输入，具体的执行步骤如下。

步骤一：基于 2.1 与 2.2 小节完成多源数据融合的路网文件生成。

步骤二：结合 Polyconvert 插件（SUMO 官方发布的生成 POI 信息插件）生成提取相关的环境几何特征。

步骤三：基于步骤一的路网文件及从 OSM 提取的公共交通数据，生成公交的公共交通时刻表与公交出行文件。

步骤四：（生成交通分析区：TAZ）：基于 OSM 数据提取不同区域边界，根据不同区域现有的建筑物、基础设施等的数量进行加权，估计区域的潜在吸引力，

同时根据允许车辆通过的最近街道和允许行人通过的最近路径计算出每栋建筑的两个可能入口。

步骤五：基于加权 TAZ 生成一个 24 小时的通用 OD 矩阵（amitran 格式）。

步骤六：生成默认的基本出行模式配置文件（Json 格式）。

其中，针对步骤六的出行模式配置文件，SAGA 主要定义了三类活动：家庭（H），主要活动（P）和次要活动（S）。三种方式中所需的所有参数都由具有给定平均值和标准差的高斯分布来定义。每一个活动链必须以家庭活动作为开始和结束，并且需要包含至少一个主要活动。次要活动可以任意插入到出行链中，但不可能有两个连续的次要活动。

另一方面，对于选择次要活动的决策过程，基于 Guido[67]提出的理论，认为在仅有部分出行信息情况下，人们通过更倾向于优化他们的日常生活品质，因此次要活动倾向于在离家近或者去主要活动（例如：工作）的路上。

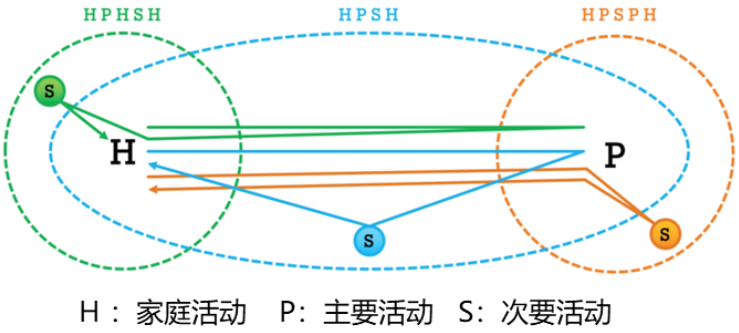


图 2-1 次要活动选择原理图

因此，次要活动在出行链中的位置决定了其相对于家庭活动和主要活动的相对位置。如图所示，在出行链 H→P→H→S→H（绿色）中，选择 S 的位置在以 H 为中心的圆的区域内。类似地，在出行链 H→P→S→P→H（橙色）中，S 的位置在以 P 为中心的具有参数化半径的圆内。最后，在出行链 H→P→S→H（蓝色）中，S 的位置选择在椭圆覆盖的区域内，椭圆的焦点分别在 H 和 P 的位置。根据这一规则，可以为所有可能的序列计算 S 的位置，这些序列以 H 开始和结束，至少有一个 P，且没有两个连续的 S。

基于以上出行活动选择的机理，结合插件 SAGA，则可完成基于 SUMO 的多模式组合出行输入，构建贴近现实的交通场景，为自动驾驶算法测试提供更精准的交通场景。

2.4 本章小结

本章首先介绍了 SUMO 中所定义路网的属性与结构，重点介绍了结合 SHP 与 OSM 地图数据作为输入的智能化车道数修改、考虑路网拓扑关系的路网 ID 修改的智能化路网搭建工作。最后基于活动出行链理论定义不同的出行模型与选择概率，结合 SUMO 的多模式出行定义规则，实现基于活动的多模式组合出行仿真输入。

第三章 基于强化学习的自动驾驶车辆微观控制

由于交通是一个复杂、动态的系统，因此对于车辆微观行为的刻画非常复杂与困难。虽然存在许多监督学习类的算法如模仿学习等能够结合人类的专家数据集进行训练，也被应用于自动驾驶问题的解决[68–71]，但是这类学习任务需要大量的专家轨迹数据集，在复杂、动态多变的交通环境中采集这类数据无疑需要非常高的成本且难度很大。此外，这类数据需要人工进行标记的同时也无法涵盖所有的现实场景。而强化学习算法则在处理决策与控制问题方面显示出了非常巨大的潜力，智能体通过不断与环境交互，试错的方式学习，这样就不需要对每个样本进行人工标记，大量地节省了成本。同时，基于强化学习的智能体能够与环境交互构建许多监督学习无法完成得场景。

但是，将强化学习应用到复杂的交通应用领域存在着巨大的挑战，特别是对于涉及在动态变化的环境中与其他车辆进行广泛交互的自动驾驶任务。目前许多研究人员针对自动驾驶车辆在混合交通流场景下的建模仍采用的是自适应巡航控制模型（ACC），该模型根据车辆速度和驾驶员舒适性来调整车辆行为。类似地，协同加速控制模型（CACC）使用控制理论来同时优化几个相邻车辆的性能[72]，例如最小化一个车辆的燃料消耗。也有部分研究人员使用监督学习或者模仿学习的方法来构建自动驾驶模型。然而，此类方法实质上对自动驾驶算法和人类驾驶员表现的最佳性做出了隐含的假设，某种意义上可以理解为最佳的人类驾驶员特性即为此模型的最佳上限。但是，自动驾驶车辆存在许多潜在可能性，甚至表现会超过人类驾驶员的上限。因此，本论文在充分研究车辆驾驶的微观特征基础上，结合自动驾驶车辆的微观控制特征，采用强化学习的方法在微观交通仿真软件 SUMO 中构建自动驾驶车辆微观控制模型。

3.1 强化学习算法

强化学习算法的本质其实是参考了人类的学习过程，当我们去思考学习的本质时，首先想到的可能是我们通过与环境的互动来进行学习。比如当一个婴儿在玩耍、挥舞手臂或四处张望时，他没有明确的老师，但他却与周围环境有直接的

运动与联系。实际的动作与行为会使这种联系产生大量关于因果关系的信息，包括执行某些行为的后果，以及为了实现目标应该做出哪些行为。在人类的学习过程中，我们通常都会非常敏锐地意识到环境对我们所做出行为的反应，并尝试通过改变我们的行为来影响所发生的事情，在不断地交互中学习几乎是所有学习方式与智力理论的基础。强化学习则是基于以上思想的方法，替代了传统利用公式或模型定义人类学习方式的方法，是一种智能体在环境中探索、交互、学习并评估各种学习方法的有效性的计算方法。另一方面，强化学习的一大特点是智能体并没有被告知应该采取何种行动，而是必须通过不断地探索、尝试来发现哪些动作或者动作的组合能带来最大化的利益与回报。有些情况下，当前采取的行动可能不仅影响眼前的奖励还会影响到远期的回报。因此，试错搜索与延迟奖励这两点是强化学习最重要的区别特征[73]。

3.1.1 基本概念

智能体（agent）：任何有独立思想并且可以与所处环境进行交互的实体。在交通场景下，智能体可以是行人，车辆，信号灯等。

状态（state） s：当前时刻智能体对周围环境的感知。所有时刻的感知集合构成了状态空间 S。

动作（action） a：智能体在当前状态下采取的行动。在当前状态下能采取的所有动作构成了动作空间 A。

策略（policy） π：智能体当前时刻状态下应该采取哪个动作的控制准则。在数学上的含义为：使用概率密度函数表示智能体在每个状态下采取各个动作的概率。

奖励（reward） R：在智能体采取动作后，环境对智能体的反馈效果。奖励 R 可以为正反馈或负反馈。

回报（return） U：智能体从当前时刻至行动结束所能获得的累积奖励 R 之和。

状态转移（state transition） P：当智能体采取动作后，由当前状态转移到下一个状态的过程。状态转移过程大多数具有随机性，该随机性来源于环境。

3.1.2 马尔可夫决策过程

强化学习应用于问题对象需要决定采取何种行动来最大化某些奖励的情况，这种需要控制智能体行为决策的问题往往也被归类为控制问题。当智能体位于环境中，在时间 t 处于某个状态 S_t，在该状态下，采取一个动作 a_t，该动作以某种方式影响环境，导致智能体的状态在该时间步长 $(t+1)$ 中转变为 S_{t+1}。在这种状态下，智能体还会获得一个奖励，称为 R_{t+1}，如图 3-1 所示。

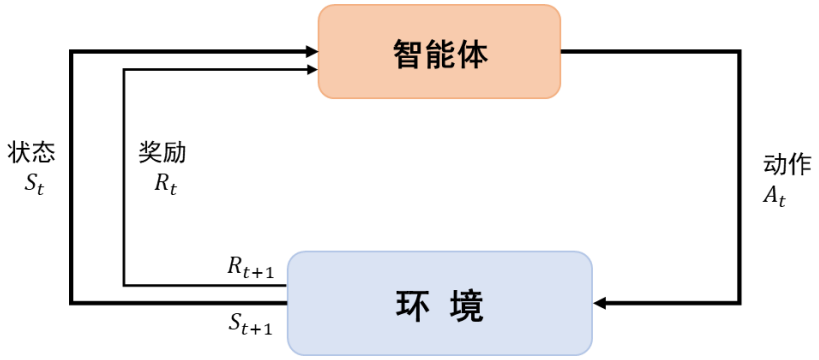


图 3-1 马尔可夫决策过程图

该类决策控制问题中的这种智能体与环境的相互作用用数学方式即可表述为马尔可夫决策过程（MDP）。MDP 可以被描述为一个 5 元组 (S, A, P, R, γ)，其中 S 表示为状态空间（state space）、A 表示为动作空间（action space）、P 表示为状态转移概率、R 表示为回报奖励（reward）、γ 表示为折扣因子（discounted factor）。

给定一个马尔科夫决策过程，当智能体位于某个状态 s 时，智能体的任务是选择一个可以获得最大累积奖励的动作 a，该动作被认为是该状态下的最佳动作。另一方面，最大累积奖励意味着当前状态所选择的动作并不是给予最大即时奖励的动作，而是最终使智能体在其状态结束时获得的最高奖励的一系列动作。策略 π：$S \to A$，即将所有状态与动作相对应，定义在每个状态下要采取的动作。因此，智能体的任务是在所有可能的状态中找到所有最优动作的集合，这一策略也被称为最优策略。

在强化学习中，任意t时刻的状态S_t下执行策略π中的动作a_t都存在一个对应的奖励R_t，由于强化学习所研究的问题具有马尔可夫性，因此系统的整体回报U_t与当前时刻的奖励R_t和未来时刻的奖励R_{t+n}有关，所以存在等式：

$$U^t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^n R_{t+n} \tag{3-1}$$

式中，γ表示折减因子。其取值一般小于 1，可以理解为当下的奖励与反馈是比较重要的，随着时间的推移，当前状态下的奖励的影响会逐渐减小。

3.1.3 基于价值的强化学习方法

在强化学习问题中，为了促使智能体学习并获得最优的策略，需要使得智能体获得评估其所学策略好坏的能力。如在给定的状态下，为智能体每次选择的动作给予相应的分数或价值，如果智能体未来所获得的累计奖励值越高，那么其对应的动作价值则越大，这也被归类为基于价值的（Value Based）强化学习方法。

$$Q_\pi(s_t, a_t) = E[U_t \,|\, S_t = s_t, A_t = a_t] \tag{3-2}$$

其中，$Q_\pi(s_t, a_t)$的值则是用来描述状态-动作对(s_t, a_t)的价值。其中，$Q(s,a)$则是该强化学习问题中的动作价值函数。通过寻找t时刻所有策略π的动作价值函数的最大值，可以得到最优的策略π的动作价值函数$Q^*(s_t, a_t)$。

$$Q^*(s_t, a_t) = \max Q_\pi(s_t, a_t) \tag{3-3}$$

同时，取最优策略π中的动作集A的最大值，即可获取每一次的最优动作a^*。

$$a^* = \arg\max Q^*(s,a) \tag{3-4}$$

在基于价值的强化学习模型中，逼近最优的策略π的动作价值函数$Q^*(s_t, a_t)$是其主要的目标。基于价值的强化学习模型应用广泛，比如 Q-learning、SARSA、DQN 等。但当某个问题的动作空间是连续动作空间的时候，如机械臂的连续控制问题。则需要对动作空间离散化，但基于此的高维度动作空间会使得整个求解过程变得异常困难，因此，基于价值的强化学习模型并不适用于解决连续动作空间的问题。

Q-learning 是应用最广泛的无模型强化学习算法，其通过表的形式来存储状态，动作值与Q值的转换。表中的每个值代表一个状态-动作对，所有的Q值更新规则如下:

$$Q(s,a) \leftarrow Q(s,a)+\alpha[r(s,a)+\gamma \max_a Q(s',a)-Q(s,a)] \tag{3-5}$$

其中，$\alpha \in [0,1]$是学习率。对于$\alpha=1$的情况，则表示所有的先验知识全部丢失。Q-learning 是一种离线策略的算法，这意味着模型会使Q值收敛到Q^*，即最优策略。离线策略算法并不需要遵循特定的策略来更新Q值，这也使得其可以从历史数据中学习，而不必以任何特定的方式选择动作。相反，SARSA 是一种在线策略学习算法，其更新Q值的规则如下：

$$Q(s,a) \leftarrow Q(s,a)+\alpha[r(s,a)+\gamma Q(s',a')-Q(s,a)] \tag{3-6}$$

如上述更新规则所示，SARSA 这样的在线策略算法要求控制器能够按为其找到Q值的策略来执行动作选择。这通常会使得收敛速度加快，但也因此允许从历史数据中学习或找到最佳策略。总结来说，这两种算法的区别是，在 Q-learning 中，Q值是使用新状态下的最大Q值更新的，而在 SARSA 中，是使用新状态下选择的动作的Q值进行更新。

3.1.4 基于策略的强化学习方法

另外一种方法被称为基于策略（Policy Based）的强化学习方法。该类方法不对智能体选择的策略进行评估，取而代之的是直接优化策略，使得智能体能够朝着累计奖励期望提高的目标进行优化与改进。智能体依靠策略函数来对于策略进行优化，策略函数的输入为当前t时刻的状态S_t，输出为所有动作的概率值。依据策略函数得到的概率值对所有动作随机抽样后，确定在状态S_t下进行的动作a_t。

$$V_\pi(s_t)=E_A[Q_\pi(s_t,A)] \tag{3-7}$$

状态价值函数$V_\pi(s_t)$只与当前策略π和状态S_t有关。因此，状态价值函数可以用来评价当前状态下不同策略的价值。当动作空间为离散的情况，状态价值函数$V_\pi(s_t)$可以写成：

$$V_\pi(s_t)=\sum_a \pi(a|s_t)\cdot Q_\pi(s_t,a) \tag{3-8}$$

如果是连续的动作空间，则可以使用积分的形式将其替代。同时，基于策略的学习方法核心即将策略参数化，将累计奖励与反馈的期望作为目标函数：

$$J(\theta)=E[U_t|\pi_\theta] \tag{3-9}$$

因此，策略学习将简化为求取函数$J(\theta)$的最大值，同时策略梯度（Policy Gradient）的方法是比较主流的方法来更新策略网络求解函数$J(\theta)$的最大值。

3.1.5 价值-策略结合的强化学习方法

基于价值与基于策略的学习方法各有优劣，基于策略的学习方法通过参数化策略的方法能够处理复杂的高纬度状态空间问题，但是，该类方法容易导致策略梯度的方差过大，同时容易陷入到局部最优。因此，出现了将策略网络与价值网络同时训练更新的方法，被称为策略价值结合的强化学习方法（Actor Critic Method）。其目的为使智能体通过策略网络做出的动作价值越来越高的同时，使得价值网络对动作价值的评价越来越精准。此时，可以把策略网络当作行动者（actor），价值网络当作裁判（critic）。价值网络会对智能体通过策略网络做出的动作进行评价，帮助更新策略网络参数，使其目标函数$J(\theta)$的值更大。该类方法通常采用神经网络来估计价值函数，损失函数通常包含两个部分：

$$Loss_T=\{Loss_v,Loss_p\} \tag{3-10}$$

其中，$Loss_T$为总损失函数，$Loss_v$为价值网络的近似误差，$Loss_p$为累计反馈期望的损失。比较流行的该类算法主要有 PPO、DDPG、A3C 等，其中 PPO 算法在游戏、机器人等领域表现出了出色的性能。

3.2 车辆行为控制模型

对于车辆的微观行为刻画是微观交通仿真软件的核心，其中包括跟驰（Car-following）与换道行为（lane-changing），这些行为通过跟驰模型与换道模型进行刻画。

3.2.1 跟驰模型

跟驰理论是指在无法超车的单一车道场景下，车辆列队行驶时后车跟随前车行驶的一种交通流状态，其中，经典的跟驰模型有以下几种：

刺激-反应（GM）模型[74]**：**

刺激-反应模型作为最早被提出的经典跟驰模型，是现今许多改进跟驰模型的原型。刺激-反应模型将前车对主车驾驶人的作用表示为一种刺激，主车驾驶员的感知能力则为敏感系数，驾驶员的反应则表示为两者作用的乘积。最早的GM 模型由 Chandler 和 Herman 等提出，经过改善与验证，其最终形式如下:

$$a_n\left(t+T\right)=\frac{\lambda \upsilon_n^m\left(t+T\right)}{\left[x_{n-1}\left(t\right)-x_n\left(t\right)\right]^l}\left[\upsilon_{n-1}\left(t\right)-\upsilon_n\left(t\right)\right] \tag{3-11}$$

式中：$a_n\left(t+T\right)$为第n辆车在$t+T$时刻的加速度；$x_n\left(t\right)$和$\upsilon_n\left(t\right)$分别为第n辆车在t时刻的位移和速度；λ为灵敏度系数；T为反应时间；m,l均为待标定系数。

安全距离（CA）模型[75]**：**

安全距离模型，也被称为防碰撞模型，该模型是应用比较广泛的跟驰模型之一，与刺激-反应类模型不同的是，此类模型基于基本的牛顿定律等，将安全距离加入到模型之中，主要强调的是主车要与前车保持一个指定的安全距离来避免发生相撞。由于其物理意义比较清晰明确，此类模型被广泛应用于交通仿真软件中，如英国的 SISTM 模型，欧盟的 SPEACS 模型，美国的 INTRAS 和 CARSIM 模型等。

安全距离模型最早是由 Kometani 和 Sasaki 在 1959 年提出来的，通过前车和跟驰车的速度来得出安全跟驰距离，跟驰车辆在安全距离内后车可以根据前车的

状态（突然减速刹车等）将自身车速下降到一个安全水平内，进而避免碰撞，但由于该模型的设定条件较为宽松，导致其与实际情况并不相符。由于该模型中前导车与跟驰车辆的速度对安全距离影响较大，因此该模型较多应用于低速交通流场景。其基本形式为：

$$x_{n-1}(t)-x_N(t)=a_1\upsilon_{N-1}^2(t)+\beta_1\upsilon_N^2(t+T)+\beta\upsilon_n(t+T)+b_0 \tag{3-12}$$

式中：α_1，β_1，β 和 b_0 属于待测参数。

智能驾驶（IDM）模型[76]**：**

智能驾驶模型（Intelligent Driver Model），由 Treiber，Helbing 等人提出，该模型包含了自由状态下的车辆加速趋势，同时考虑与前导车碰撞而减速的趋势与意图。该模型同时考虑了期望速度和期望车头间距，其中期望车头间距是由速度、速度差、最小间距、最大加速度和期望车头时距等共同决定，是多种社会力（包含驱动力和阻力）作用下产生的车辆行驶加速度，但是该模型中并没有考虑驾驶者的反应时间。IDM 模型在车辆处在加减速行驶状态时，能使过程更为平滑合理，更符合驾驶者的实际情况和驾驶习惯，相对其他基于物理规律定义的模型来说更能准确地反应驾驶员的驾驶特征。其表示形式为：

$$a_n(t)=\omega[1-(\frac{v_n(t)}{v_0})^{\delta}-(\frac{s^*(v_n(t),\Delta v(t))}{x_{n-1}(t)-x_n(t)-l_c})^2] \tag{3-13}$$

$$S^*(v_n(t),\Delta v(t))=S_0+S_1\sqrt{\frac{v_n(t)}{v_0}}+Tv_n(t)+\frac{v_n(t)\Delta v(t)}{2\sqrt{wd}} \tag{3-14}$$

式中：$a_n(t)$ 为第 n 辆车 t 时刻的加速度，$x_n(t)$ 为第 n 辆车 t 时刻的位置，$v_n(t)$ 为第 n 辆车 t 时刻的速度，$\Delta v(t)$ 为本车与前车之间的速度差，s^* 为当前状态下驾驶员的期望保持间距，ω 为车辆的起步加速度，d 为舒适减速度，δ 为加速度指数，S_0 为静止安全距离参数，S_1 为速度相关的安全距离参数，T 为安全车头间距。

3.2.2 换道模型

换道模型的本质则是通过车辆横向的作用来描述驾驶特征与任务。同时，许多研究人员认为换道行为对于道路交通安全产生了负面影响。换道行为通常被分为自由型换道与强制型换道。自由型换道的主要目的是驾驶员为了达到自己的期望速度或者更换目前的驾驶环境，而强制性换道则是为了达到计划的目的地，比如前方需要左拐，驾驶员提前向左换道来达到自己的目的[77]。

Gipps[78]模型是一个经典的换道行为模型，该模型中对于驾驶员的换道行为的刻画主要通过两个因素来影响，一个是保持本身的期望车速，另一个是保持在预期需要执行换道动作的正确车道上。但是由于该模型中的驾驶员行为是确定的，所以无法展现出不同驾驶员之间的差异。其主要的表达如下：

$$v_n(t+T)=b_nT+\{b_n^2T^2-b_n[2(x_{n-1}(t)-s_{n-1}-x_n(t))-v_n(t)T-\frac{v_{n-1}(t)^2}{\hat{b}}]\}^{1/2} \quad (3\text{-}15)$$

其中，$v_n(t+T)$ 是车辆 n 相对于前车 $n-1$ 在 $t+T$ 时刻的最大安全速度，b_n 是第 n 辆车的驾驶员准备采取的最严重的刹车措施，$\hat{b}$ 是第 n 辆车的驾驶员对于 $b_{n\text{-}1}$ 的估计，T 是更新速度和位置的时间步长，$x_n(t)$ 是第 n 辆车 t 时刻的位置，s_{n-1} 是第 $n-1$ 辆车的有效长度。

SUMO 中的换道模型则主要采用的是 LC2013 模型[79]，相较于其他微观换道模型相比，该模型明确区分了四种不同的换道动机，战略性换道、合作性换道、战术性换道和规则性换道。在每一个仿真步中，该模型执行以下步骤：

步骤 1：计算最优的下一步的目标车道（称为最优车道）；

步骤 2：在假设停留在当前车道上与采用来自前一步骤的换道策略相关的速度请求相结合的情况下计算安全速度；

步骤 3：基于模型计算换道请求行为（向左变道、向右变道、停留在原车道）；

步骤 4：执行换道动机或计算下一仿真步长的速度请求（包括提前规划多个仿真步长），是否改变速度取决于换道请求的紧急程度。

同时，SUMO 的换道模型支持用户针对其 10 余种参数进行调整来进行定制化处理。

3.3 问题定义

传统的人工驾驶员不合理的驾驶行为会导致车辆能源消耗的加大，造成资源浪费与环境污染的加重，同时，不正确与不合理的车道换道行为可能是导致交通事故和交通堵塞的主要原因。对于理想化的自动驾驶车辆而言，其相对于人类驾驶员更加安全，能够保持更小、稳定的安全距离，执行更平稳、高效的驾驶行为。由于自动驾驶车辆的控制问题具有高维度、状态和动作空间连续、非线性等特点，在充分研究传统跟驰，换道模型的车辆行为与特性的基础上，针对自动驾驶车辆微观行为的特点，对自动驾驶车辆的微观控制行为模型进行定义。

基于上一小节对于经典的车辆跟驰与换道模型的研究，本论文考虑自动驾驶车辆的微观控制模型应在一定程度上保留人类驾驶特征的基础上进行优化。针对跟驰行为，理想化的单车自动驾驶车辆应该在尽可能与前车保持最小安全距离的情况下实现跟驰。同时，针对前车突然刹车、减速等反应，相较于人类驾驶员应该具备更快的反应时间。对于自动驾驶车辆换道行为，综合考虑强制换道与自由换道两个方面，目的是为了更快速、更合理、更高效的到达目标位置。当车辆接受到明确的宏观路径规划结果后，自动驾驶车辆基于周围车辆与车辆本身的状态来决定在什么时刻以及如何进行换道行为。一旦模型计算出结果，则基于 SUMO 来产生相应的控制命令来执行，具备以上跟驰与换道特征的自动驾驶车辆，主要具备以下特点：在避免与周围车辆碰撞的前提下，实现高效率、平稳、舒适的驾驶操作。同时，本论文提出的基于强化学习的微观控制驾驶行为模型具有高泛化性，能够适用于 SUMO 中所有的道路情况。

3.4 模型构建

本论文提出一种全新的基于深度强化学习 PPO（Proximal Policy Optimization）近端策略优化算法[80]的理想化单车自动驾驶微观控制行为模型，整个模型的系统架构主要分为两部分：基于强化学习的模型与仿真模拟环境。基于 SUMO 的 TRACI 接口，自动驾驶车辆智能体获取相关的环境信息作为输入到强化学习模

型（模型需要提前不断训练与优化）中，模型输出相应的动作返回到仿真环境中，执行改变仿真环境中车辆的动作行为。

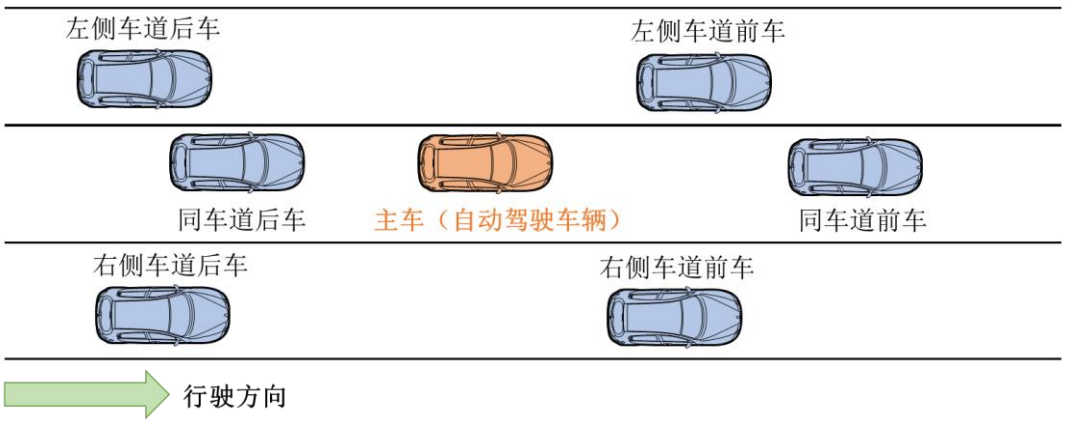


图 3-2 自动驾驶车辆微观控制场景图

如图所示，当车辆执行换道等决策时会受到本车和其他车辆之间相互作用的影响，因此，本论文考虑自动驾驶车辆（主车）周围 6 辆车与主车的状态信息，构建强化学习模型，包括主车的同车道前车，同车道后车，左侧车道前车，左侧车道后车，右侧车道前车与右侧车道后车。状态空间总共由 22 个变量组成，分别是主车的 4 个状态变量，包括纵向位置、横向位置、纵向加速度与纵向速度，以及周围车辆的 18 个状态变量（每辆车包含 3 个状态信息，包括车辆纵向速度、纵向加速度与主车的横向距离）。对于动作空间，定义为主车的纵向加速度、车辆向左侧车道变道、车辆向右侧车道变道以及车辆保持现有车道。为了使模型达到预期效果，本论文设计了一种多因素影响的奖励函数，考虑自动驾驶车辆行驶的安全性、效率与舒适性。考虑舒适性的奖励函数可以表达为：

$$R_c(t) = -\alpha \cdot \mathbb{I}(a_{lc} \neq 1) + \beta \cdot a_{acc}^2 \tag{3-16}$$

其中，α, β 是相应的权重系数，a_{lc} 为换道动作，向左侧车道换道表示为 0，保持当前车道表示为 1，向右侧车道表示为 2。引入该奖励的目的是避免突然的加速与减速导致车内人员感到不适。考虑效率的奖励函数可以表示为：

$$R_e(t) = -\xi \cdot \left| V_y - V_{desired} \right| \tag{3-17}$$

其中，ξ是相应的权重系数，V_y为车辆的纵向速度，$V_{desired}$为期望车速，默认设置为道路限速值。设置该奖励函数的目的是设法让自动驾驶车辆在不超过速度限制的情况下尽快移动到目标车道。考虑安全性的奖励函数可以表示为：

$$R_s(t)=\begin{cases}-1000 & if\ \ collision\\ 0 & else\end{cases} \tag{3-18}$$

如果车辆发生碰撞，则给予非常大的负奖励。综上，整体的奖励可以表示为以上三种奖励的整合：

$$R(t)=R_c(t)+R_e(t)+R_s(t) \tag{3-19}$$

综上，自动驾驶车辆的目标与任务是在最短的时间内（即最大平均速度）到达其目标终点，同时遵守速度限制并避免与交通碰撞。这里需要注意的是，对于自动驾驶车辆控制来说，该模型的前提是车辆已经获得起点与终点之间的宏观行驶路径，该模型在遵守该路径的前提下，完成车道级别的微观控制。

为了提高强化学习模型搭建的效率，本论文采用目前比较火热的开源强化学习框架：Tensorforce[81]。目前已有许多研究机构与组织提出了强化学习的框架与基准算法，比如 OpenAI Baselines，Rllib，Tensorforce 等。Tensorforce 是一个开源的深度强化学习框架，构建在 Google 的 TensorFlow 框架之上。本论文采用近端策略优化 PPO 算法进行算法更新，该算法的更新规则如下：

算法 3.1 PPO，Actor-Critic 类型

for iteration=1, 2, … do

for actor=1, 2, …, N do

Run policy $\pi_{\theta_{old}}$ in environment for T timesteps

Compute advantage estimate $\hat{A}_1,...,\hat{A}_T$,

end for

Optimize surrogate L wrt θ, with K epochs and minibatch size $M \le NT$

$\theta_{old} \leftarrow \theta$

end for

同时，本论文将动作掩码（Action masking）的方法融合到强化学习算法之中，其主要的思路是模型在进行最大值运算之前对输出的一部分Q值进行屏蔽。这样做的直接影响是当采用取最大值来选择最佳动作时，只需要考虑与动作子集相关联的Q值。当给定一个状态，可以限制（或屏蔽）智能体不需要探索或从其结果中学习的任何一组动作。例如，在变道问题中，如果主车在最左边的车道上，那么向左行驶将导致离开高速公路。因此，我们可以在与向左边车道运动的动作相关联的Q值上加上一个掩码，使得智能体在这种状态下永远不会被选择，即将交通领域相关的先验知识直接结合到学习过程中，这也意味着不需要为不同的场景设置许多的负面奖励，从而简化了奖励功能。另一方面，由于智能体不用探索这些状态，这也使得整个学习过程本身变得更快、更有效，智能体最终学习的是必要Q值的实际空间的子集。

3.5 模型训练

为了提高模型的训练效率，搭建了如同所示的仿真训练环境。

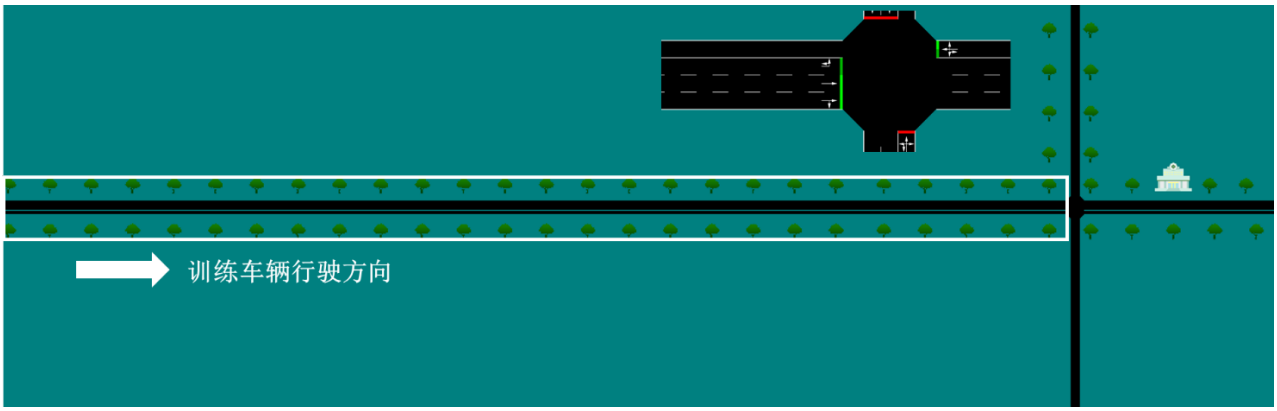


图 3-3 微观控制模型仿真训练场景图

仿真环境如图 所示，智能体车辆由西向东行驶，道路长度为 1km，三车道，交叉口出口道方向为左转，直行，右转，使得训练场景能够基本涵盖微观控制的情况。同时，道路的最小和最大速度限制为$120(km/h)$，所有车辆都必须遵守，对于智能体车辆，定义最大加速度为$3\,m/s^2$，最小加速度为$-3\,m/s^2$。

在 PPO 算法的超参数方面，本论文选择 Adam 进行优化，比较重要的参数设置如下:学习率为 0.015，折扣系数为 0.999，批大小（mini-batch size）为 128，探索率（exploration）为 0.001，训练 2500 回合。

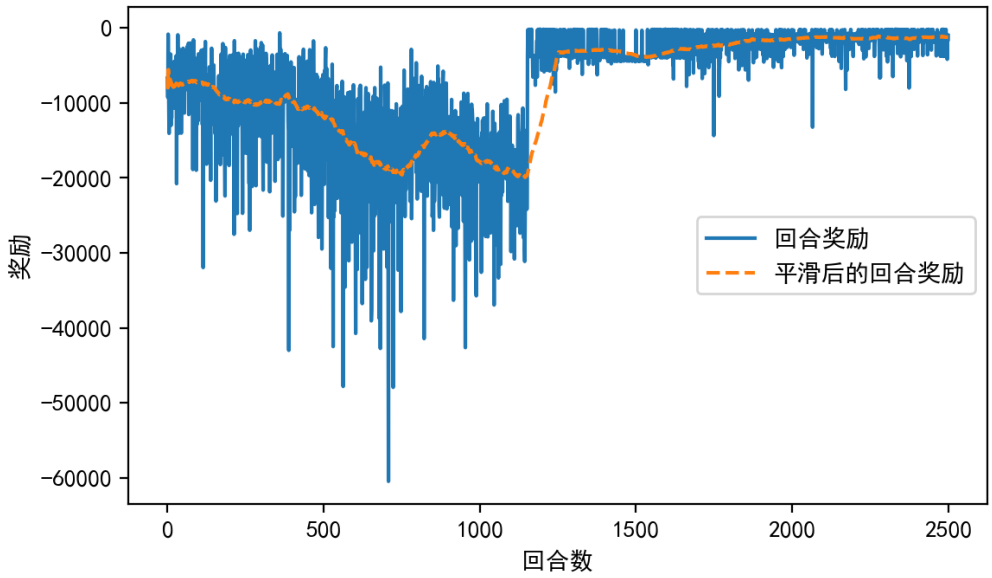


图 3-4 累积奖励学习曲线示意图

由上述奖励曲线图可以看出，该模型在 1200 轮左右便能够收敛，通过 SUMO 的 GUI 画面观察，该模型控制的自动驾驶车辆能够在确定目标车道后，根据周围车辆的状态寻找到最优的行驶行为到达终点。

3.6 本章小结

本章首先总结了强化学习算法的基本概念与原理，同时介绍了一些比较流行的强化学习算法。分析了交通领域中经典的车辆跟驰与换道模型的模型机理，基于此提出自动驾驶车辆微观控制行为的假设。在此基础上，结合强化学习的 PPO 算法构建了自动驾驶车辆微观控制模型，并结合了 SUMO 仿真进行了训练与模型效果测试。

第四章 基于强化学习的自动驾驶车辆宏观路径规划

自动驾驶车辆的控制由微观控制与宏观路径规划两大模块组成，如图 4-1 所示。对于自动驾驶车辆的规划问题，目前大部分的自动驾驶算法公司往往都采用导航路线作为规划路线，而传统的地图导航软件在实际商业运用中大都基于传统的路径规划算法进行了调整与优化。实际应用中由于用户的请求量巨大，往往会使用历史出行方案或基于图论的最短路来进行路径推荐。但由于道路交通的动态性与复杂性，需要综合考虑路况信息、道路环境变化等其他因素，传统的最短路算法往往不一定是最优路径。因此，基于历史出行数据的行程时间预测方法则被提出应用到地图软件的路径推荐算法中来为乘客提供多项选择。

而对于交通仿真而言，大部分仿真软件仅仅使用最短路算法实现车辆的路径规划，忽略了仿真中道路交通的动态变化。另一方面，历史的出行数据由于获取较为困难且难以在仿真中直接应用。因此，本论文结合强化学习算法提出了基于强化学习的动态路径规划方法来实现自动驾驶车辆在仿真中的宏观路径规划。

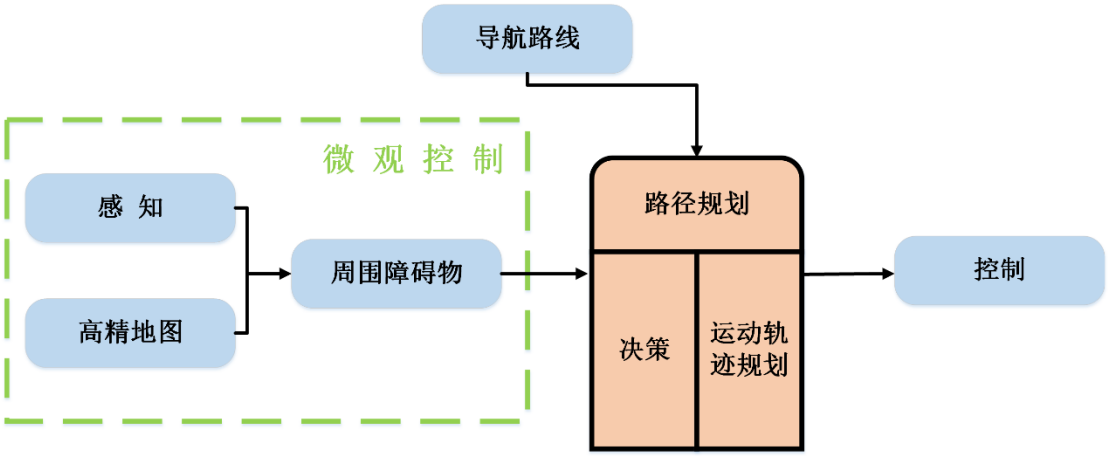


图 4-1 自动驾驶车辆路径规划与微观控制逻辑图

4.1 路径规划问题

对于交通仿真模型而言，车辆的宏观路径规划一直都是需要解决的关键问题，大部分交通仿真软件通过搜索 OD 之间的最短路来完成车辆的宏观路径规划。在车辆路径规划领域中，往往都是通过寻找起点与终点之间的最短路来解决问题。

计算网络中两个点之间的最短路，最短路算法也成为了经典的算法之一。例如 Dijsktra 算法、Floyd 算法、A*算法等都为该领域经典的算法，同时由于不同的算法具有不同的特点，需要结合不同的应用场景来选择合适的算法。

4.1.1 最短路问题

Dijkstra[82]算法是最早应用求解最短路问题的算法之一，属于一种贪心算法，无方向性的进行搜索，每一步都选择当前的最优解并生成新的状态，一直到达目标状态为止，因此能够保证解为全局最优解。但同时由于其算法的逻辑与特性也导致了该算法的计算量非常大，当网络节点数不断增大时计算量会成为瓶颈之一。因此，A*算法作为一种启发式算法被提出来解决 Dijkstra 算法在大网络中计算量“爆炸”的情况。相比于 Dijkstra 算法的盲目搜索，A*算法的路径搜索更具有目的性，其在搜索过程中建立启发式搜索规则来衡量实时搜索位置和目标位置的距离关系，使搜索方向优先朝向目标点所处位置的方向，最终达到提高搜索效率的效果，然而并不能保证能够得到最优解，但更适用于网络节点数量巨大的情形。Dijkstra 算法步骤：

步骤 1：将起点距离设置为 0，除起点以外的所有点距离设置为无穷大。

步骤 2：设置所有点，包括起点，设置为非访问节点。

步骤 3：将当前距离最小的非访问节点设置为当前节点“C”。

步骤 4：对于当前节点的每个相邻节点“N”：将当前距离“C”加上连接节点“C”与节点“N”的边的权重。如果小于当前与节点“N”的距离，就将其设置为新的当前距离“N”。

步骤 5：将当前节点“C”标记为已访问节点。

步骤 6：从步骤 3 开始重复上述步骤，直到到达目的地。

4.1.2 K 短路问题

在现实生活的路径规划应用中，除了需要求解最短路之外，往往还需要知道其他的备选路径。例如：当高德、百度地图用户在使用软件进行导航时，通常会推荐多条道路，包括用时最短路线、红绿灯数最少路线等，用户会根据当前的需

求选择自己需要的线路。K 短路算法则是用来求解多条满足条件的路径方案的算法，K 最短路径问题实质上属于求解最短路算法的扩展与变形。1959 年，Hoffman 和 Pavley[83]第一次提出 K-最短路径问题，此后受众多研究者们广泛关注,目前已有多种不同的 K 短路问题的求解算法。Yen[84]提出的 K 短路算法可以理解两部分，首先基于经典的最短路算法如 Dijkstra 算法求解出第一条最短路径 $P_{[1]}$，之后在此基础上依次推算出其他 $k-1$ 条最短路径。在求解 $P_{[i+1]}$ 时，将 $P_{[i]}$ 上除了终止节点外的所有节点都视为偏离节点，并计算每个偏离节点到终止节点的最短路径，再与之前的 $P_{[i]}$ 起始节点到偏离节点的路径拼接，构成候选路径，进而求得最短偏离路径。该算法流程如图：

算法 4.1 Yen 的 K 短路算法

Compute Γ_t^*

$p_1 \leftarrow$ shortest path from s to t， $k \leftarrow 1$， $X \leftarrow \{p_k\}$， $\Gamma_k \leftarrow \{p_k\}$

While $k < K$ and $X \neq \varnothing$

Do begin

$X \leftarrow X - \{p_k\}$

$v_k \leftarrow$ deviation node of p_k

for each node $v \in p_{v_t t}^k$

Do begin

if $A(v) - A_{\Gamma_k}(v) \neq \varnothing$

Then begin

Compute the arc (v, x) such that $c_{vx} + c(p_{xt}^*)$ is minimized over $A(v) - A_{\Gamma_k}(v)$

$q \leftarrow p_{sv}^k \Diamond \{v, (v, x), x\} \Diamond p_{xt}^k$

$X \leftarrow X \bigcup \{q\}$

$q_{vt} \leftarrow \{v, (v, x), x\} \Diamond p_{xt}^*$

$\Gamma_k \leftarrow \Gamma_k \bigcup \{q_{vt}\}$

end

end

$k \leftarrow k+1$

$p_k \leftarrow$ shortest path in X

end

4.2 问题描述

当仿真中的自动驾驶车辆具备了第三章所述的微观控制功能后，需要结合宏观路径规划算法来完成规划层面的整体控制。该问题可以被定义为：当主车获得起点与终点信息时能够根据道路交通的变化动态地调整行驶路径，最终选择成本最小的最优路径到达终点。

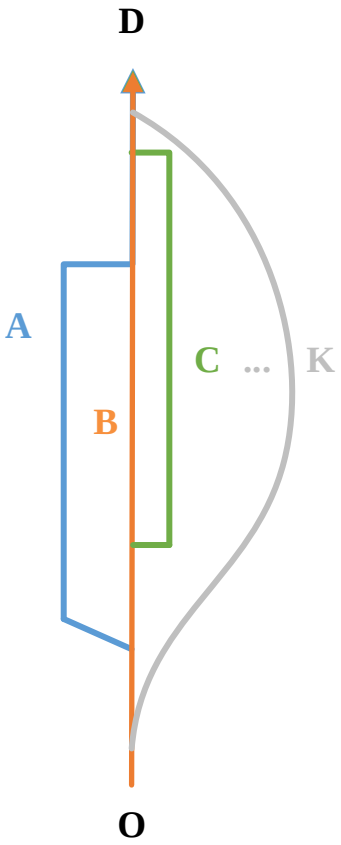


图 4-2 K 条备选路径示意图

当车辆从起点出发时刻起需要及时返回多条备选最优路径方案，为了避免大规模路网下计算效率过低导致返回信息不及时的问题，选择基于 K 短路算法构建 K 条最优备选路径，称为 K 条初始路径方案。由于 K 短路算法的特性，K 条最短路会存在较多的重复路段，如图 4-2 所示，同时为了避免出现仿真中车辆在交叉口来不及做决策的情况，选择不同的初始方案重合路段的前一个路段作为强化学习智能体的决策路段。同时，车辆智能体选择决策时刻 K 条初始方案中最优的一条，不断重复直至到达终点。综上，本强化学习问题描述为智能体车辆加入仿真环境后，确定终点位置的同时返回 K 条最短路径组合方案，智能体在每条路段选择当前情况最优的初始方案并直至完成一次出行。

4.3 模型构建

仿真中定义车辆或者一股车流的出行文件需要定义为基于路段（edge）ID 的连续序列，无法通过起终点为节点编号（junction）来定义车辆出行，但 K 短路算法等都是基于节点（node）信息进行检索，因此需要解决仿真中车辆起点与终点和邻近的交叉口匹配的问题。该问题通过路网拓扑结构来进行逻辑判断比较复杂，但是通过之前修改路网 ID 的规则来进行则非常便利。例如['O@1#0','2@3#0',...,'5@D#0']这条路径，仅需取该列表第一个元素中@之后的数和最后一个元素中@之前的数就能作为 K 短路算法的输入节点编号。

另一方面，对于不同的路径，一般通过行程时间来判断哪条路径更优。由于信号灯从时间上将交通量进行了分离，仿真中很容易出现一段道路上车辆都在等待红灯，在计算该时间步路段的行程时间（路段长度除以路段车辆平均速度）时会出现问题，因此本论文采用每条路径的剩余路径长度、剩余路径车辆长度与剩余路径车辆数来对于不同路径进行评估。

对于该自动驾驶车辆的宏观路径选择强化学习问题，本论文结合深度强化学习 DQN 算法实现，并进行如下定义。状态空间定义为 K 条路径的剩余路径总长度、剩余路径总车辆数、剩余路径总车辆长度，如下式：

$$s_t = (\mu_1^1, \mu_1^2, \ldots, \mu_1^k, \mu_2^1, \mu_2^2, \ldots \mu_2^k, \mu_3^1, \mu_3^2, \ldots, \mu_3^k) \tag{4-1}$$

其中，μ_1^1表示为备选路径中的第 1 条路径的剩余路径总长度，μ_2^3表示为备选路径中的第 3 条路径的剩余路径总车辆数，μ_3^k 表示为备选路径中的第k条路径的剩余路径总车辆长度。

动作空间即为初始K条最短路径集合，定义为：

$$P=\{p_1,p_2,\ldots,p_k\} \tag{4-2}$$

同时决策点在K条路径重复路段的前一个路段。由于本问题定义的目标为寻找到最优的路径以最快时间到达终点，因此奖励函数仅考虑效率，定义为：

$$r_t=-t \tag{4-3}$$

基于 DQN 的详细算法流程如下：

算法 4.2 自动驾驶车辆宏观路径规划 DQN 算法

Initialize replay memory $\Re$.

Initialize action-value function Q.

For $i\in\{1,2,...,n_{episode}\}$ do

initialize K shortest path set $P=\{p_1,p_2,...,p_k\}$ according to distance.

initialize state vector $s_t=(\mu_1^1,\mu_1^2,\ldots,\mu_1^k,\mu_2^1,\mu_2^2,\ldots\mu_2^k,\mu_3^1,\mu_3^2,\ldots,\mu_3^k)$.

For $t\in\{1,2,...,T\}$ do

Select action a_t~ExperienceReplay$(Q(s,a;\theta|\varepsilon)$.

Perform action a_t, observe the remained path length、remained number of vehicles and remained length of vehicles $\prod_{i=1,2,3}\mu_i$ and the next state s_{t+1}.

Obtain reward r_t

Append (s_t, a_t, r_t, s_{t+1}) to $\Re$.

Sample random minibatch of memories (s_j, a_j, r_j, s_{j+1}) from $\Re$.

If terminal:

$y_j = r_j$.

Else:

$y_j = r_j + \gamma \max_a Q(s_{t+1}, a; \theta)$.

End if

Update weights $\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{s,a}\left[\left(y - Q(s,a;\theta)\right)\nabla_\theta Q(s,a;\theta)\right]$.

End for

End for

其中，经验回放（ExperienceReplay）是 DQN 算法的一个特殊设计。DQN 构造了一个经验集合$\Re$，对于每轮仿真采样得到的数据，将其存入该集合内，在更新神经网络参数时，并不直接使用刚刚采样得到的数据，而是从$\Re$中随机选取一些样本进行反向传播。相比于直接使用每一个时间步的采样数据进行训练，经验回放的优势包括：（1）每一步的数据都可以在模型参数更新过程中被反复利用，数据利用效率更高；（2）强化学习通常解决的是多步决策问题，例如对于一个 5 步决策问题，需要找到一个有序的动作序列$\{a_1, a_2, ..., a_5\}$，最大化这 5 个动作带来的总回报。在多步决策问题中，前一步的状态和动作与后一步的状态和动作存在关联，每一步的动作都会影响下一步的状态，此时，仿真采样得到的数据之间关联性较大，而使用经验回放可以使采样数据更接近于独立同分布，使模型更容易收敛[85]。

与上一章相同，强化学习智能体从动作空间中选择动作时为随机选择，但是许多情况存在一些显然不合理的动作，这些动作一定程度上可以避免。比如车辆不能选择与当前路段没有交集的初始路径等，因此本章所提出的宏观路径选择也采用了动作掩码的方法，将车辆在做决策时把无法选择的路径备选方法的动作概率设为 0，避免了智能体车辆需要先学习如何不选择错误的动作，大大缩短智能体车辆的训练时间，提高训练效率。

4.4 模型训练

本论文采用 Adam 优化器基于训练数据迭代地更新与优化神经网络权重模型，Adam 模型是由 Kingma 和 Lei Ba[86]在 2014 年所提出的，并广泛被研究者们使用。Adam 更新规则如下：

步骤 1：对于 t 时间步的梯度进行计算：

$$g_t = \nabla_\theta J(\theta_{t-1}) \tag{4-4}$$

计算梯度的指数移动平均数，同时综合考虑时间步的梯度动量。系数 β_1 为指数衰减率，控制权重分配（动量与当前梯度），通常取接近于 1 的值。

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{4-5}$$

步骤 2：计算梯度平方的指数移动平均数，系数 β_2 为指数衰减率，控制之前的梯度平方的影响情况。

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{4-6}$$

步骤 3：由于 m_0 初始值为 0，会导致 m_t 在训练初期阶段偏向于 0。因此，需要对梯度均值 m_t 进行偏差纠正，降低偏差对训练初期的影响。

$$\hat{m}_t = m_t / (1-\beta_1^t) \tag{4-7}$$

步骤 4：与 m_0 类似，因为 v_0 初始化值为 0 同样会导致在训练初始阶段 v_t 偏向于 0，需要对其进行纠正。

$$\hat{v}_t = v_t / (1-\beta_2^t) \tag{4-8}$$

步骤 5：更新参数，其中学习率默认值一般为10^{-3}，ε取值一般为10^{-8}，为了避免除数变为 0 的情况。

由下列表达式可以看出，对更新的步长计算，能够从梯度均值与梯度平方两个角度进行自适应地调节，而不是直接由当前梯度所决定。

$$\theta_t = \theta_{t-1} - \alpha * \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon) \tag{4-9}$$

基于以上优化规则与强化学习模型，相关算法的配置参数如下：

初始化参数 K 取 10，表示预先计算 10 条备选最短路径，同时构建这样一个神经网络，如图 4-3 所示，输入为状态s和动作a，按照上述定义，输入神经元为 31 个（包含 30 维状态向量和 1 维动作向量）神经网络的输出则为价值函数值$Q(s,a)$，双层中间层，每层具有 64 个神经元，训练回合 20000 轮。

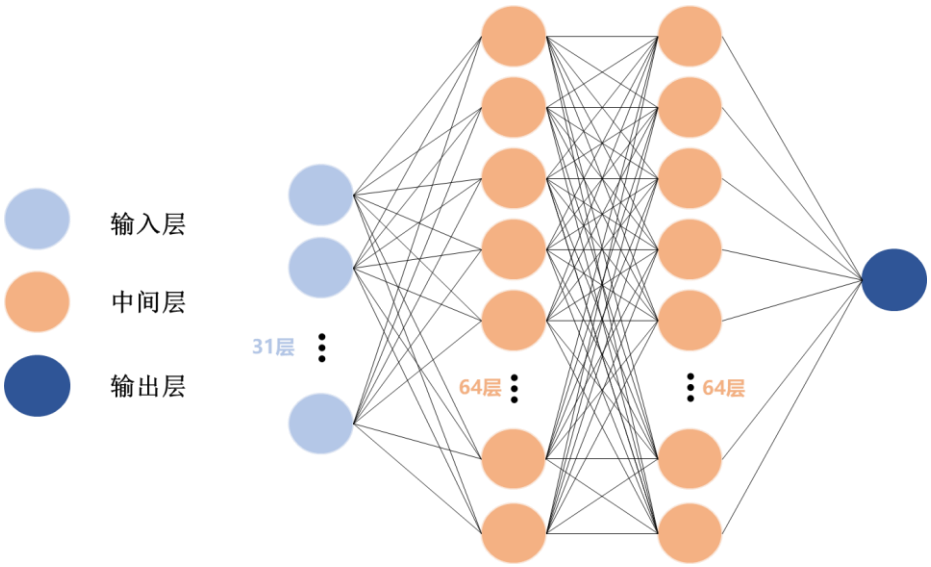


图 4-3 神经网络结构图

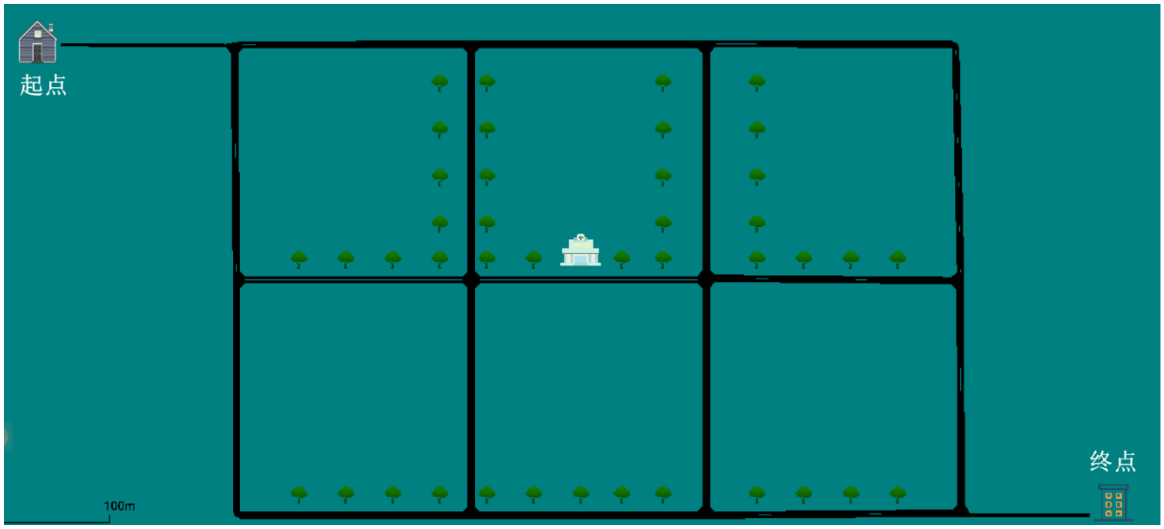


图 4-4 宏观路径选择模型仿真训练场景示意图

为了提高模型训练效率，构建如图 4-4 所示仿真训练场景，智能体车辆从路网起点出发选择路径到达终点，为了增加场景随机性，使得车辆出发时间在一定范围内随机波动。

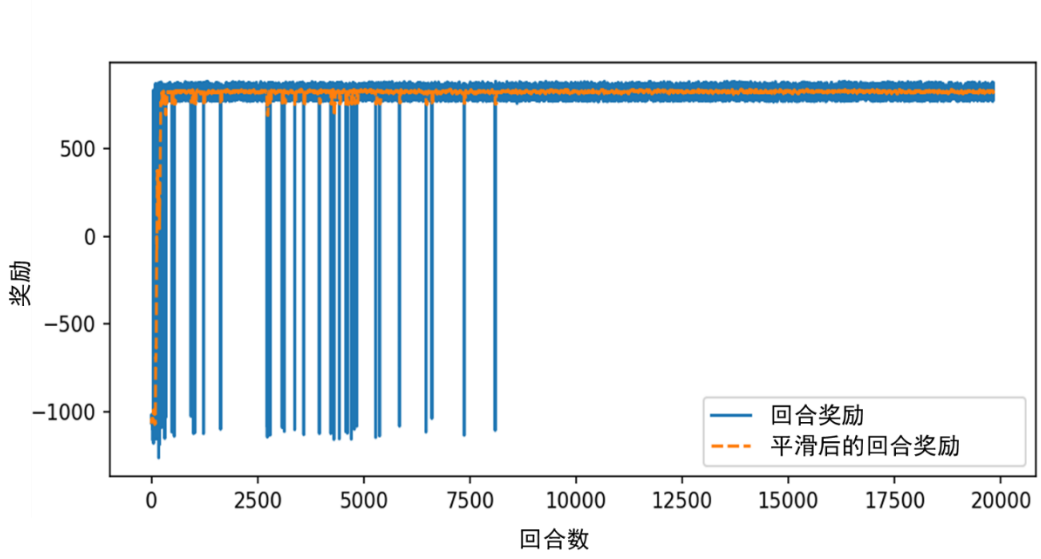


图 4-4 累积奖励学习曲线示意图

根据图 4-4 所示，模型在 8000 轮左右基本收敛，同时通过 SUMO 的 GUI 界面对于模型进行测试，在仿真随机种子 200 的情况下，车辆在仿真第 426 秒进入路网，使用默认的路径选择模型，车辆在 724 秒到达终点，同时选择出发时刻的最短路的车辆在 662 秒到达终点，训练好的强化学习模型通过提前改变策略，在 561 秒到达终点，同时也是这个问题下的最优值，证明模型的训练效果良好。

4.5 本章小结

本章首先对于传统的路径规划问题进行了介绍，并提出了自动驾驶车辆进行宏观路径选择的特征假设。在此基础上，详细介绍了基于深度强化学习 DQN 算法的自动驾驶车辆宏观路径选择算法的步骤与原理，最后基于 SUMO 的仿真场景进行了测试与验证。

第五章 案例分析

5.1 研究区域

本论文选择苏州古城作为研究区域，实现基于真实城市路网情况下的仿真测试。苏州古城路网呈棋盘状，主次干道呈现“三纵七横”的特点，其中横向主干道为干将路，横向次干道有十梓街，十全街，竹辉路，景德路，东中市，桃花坞大街等道路，纵向主干道为人民路，学士街，临顿路等道路，此外，苏州古城区支路较密集且多为双车道道路。苏州古城区域目前共有苏州地铁 1 号线，苏州地铁 4 号线两条地铁线路，其中苏州地铁 1 号线为东西走向，主要有相门、临顿路、乐桥和养育巷 4 个地铁站。苏州 4 号线为南北走向，主要有南门、乐桥、察院场和北寺塔 4 个地铁站。

表 5-1 苏州古城区域公共交通统计

名称	数量
公交线路数量（条）	24
公交站点数量（个）	108
地铁站点数量（个）	4
地铁线路（条）	2

5.2 仿真结果分析

仿真输入数据包括：苏州古城区域的公共交通信息（站点经纬度、公交线路运营数据）、苏州古城区域的 OSM 路网数据、SHP 路网数据。在爬取苏州古城的 OSM 数据后，经过分析发现由于古城区域的道路存在许多辅路，且 OSM 的更新不够及时，存在车道数缺失严重的问题，导致直接使用 OSM 数据构建仿真路网会使得路网与实际不符。因此，本论文拟采用具有完备车道数信息的 SHP 文件结合融合 SHP 与 OSM 的车道数修改算法与考虑路网拓扑关系的新型路网道

路 ID 修改算法进行苏州古城仿真路网搭建工作。包含苏州古城原始 SHP 文件的信息如果 5-1 所示，其中 p_lanes 代表正向车道数，n_lanes 代表反向车道数。

序号	SmUserID	name_chn	fnode	tnode	road_class	p_lanes	n_lanes	direction	forwardroa
1	0	胥门外大街	10,320	7,549	47,000	1	1	1	5,905,073,455,...

图 5-1 苏州古城原始 SHP 文件包含字段示例图

结合第二章提出的融合多源数据的智能化仿真路网搭建方法，完成苏州古城仿真路网的两类数据匹配、车道数修改、道路 ID 修改等工作，并结合 Netconvert 插件完成 SUMO 路网文件构建。如图 5-2 所示，左边为 OSM 官方网站中苏州古城区域示意图，右边即为通过路网修改与优化搭建方法后的 SUMO 仿真路网示意图，除了仿真道路构建之外，也将区域的 POI 信息进行爬取与展示，同时以不同的颜色进行区分，如图 5-2 右所示。

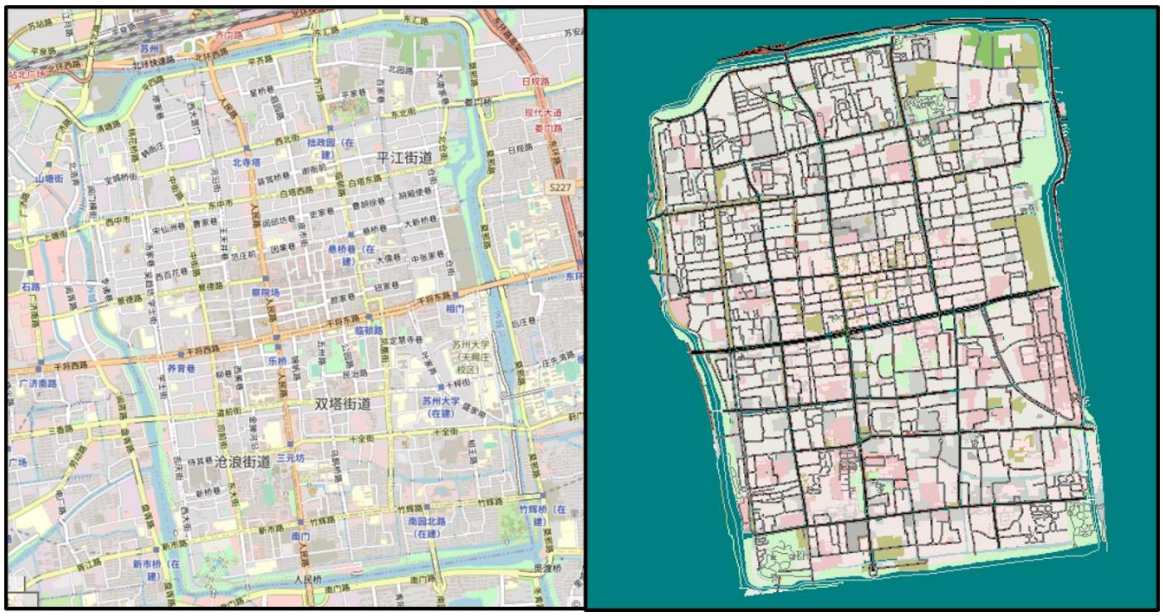


图 5-2 苏州古城路网示意图（左：OSM 地图 右：SUMO 仿真路网）

基于搭建好的 SUMO 古城仿真路网，结合 SAGA 与第二章的相关算法步骤进行基于活动的多模式交通出行输入。首先基于获得的公共交通数据构建苏州古城的公共交通出行方式，包括地铁、传统公交等。其次基于 OSM 数据中的行政边界将古城路网分为 8 个交通小区，同时输入 10000 个人口数据进行测试，整体多模式交通仿真效果如图 5-3 所示。实现苏州古城区域的多模式组合出行，包括传统公交、非机动车、步行、地铁的出行模式。同时支持在仿真界面显示每个人的出行计划，单个行人的 ID 连续，在完成整个链条出行后再消失。单个行人在进行活动期间，如工作，则在对应的 POI 区域停留，结束后返回路网进行出行。

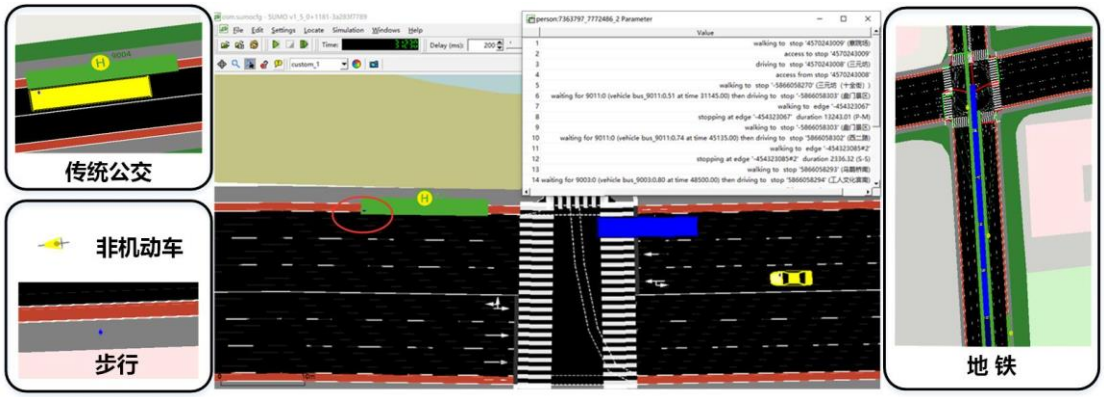


图 5-3 苏州古城多模式组合出行图

在完成苏州古城的仿真路网与仿真输入基础上，需要对第三章基于强化学习的微观控制模型与第四章的强化学习自动驾驶车辆宏观路径选择模型进行一体化整合。由宏观路径选择模型在获得具体的起点与终点信息后，不断动态调整并计算得到宏观路径。在获得的宏观路径基础上，由强化学习微观控制模型返回智能体车辆在仿真路网中的具体加减速与换道动作，完成一体化的仿真流程。随机选取多对 OD 对，使用一体化整合后的自动驾驶车辆微观控制模型进行测试，同时与 SUMO 仿真默认的跟驰与换道模型车辆进行对比，具体的对比结果如表：

表 5-2 自动驾驶模型与默认仿真模型出行方案行程时间对比

序号	仿真默认模型行程时间（秒）	自动驾驶模型行程时间（秒）	差值（%）
1	507	478	5.7
2	600	549	8.5
3	1020	617	39.5
4	384	372	3.1
5	884	798	9.7
6	1294	925	28.5
7	925	812	12.2
8	679	529	22
9	782	615	21.3
10	1127	865	23.3

在苏州古城区域随机选取 10 个 OD 对，根据上表所示，自动驾驶车辆模型相对于传统模型而言，行程时间均有显著缩短。同时，基于 GUI 可以观察到车辆选择路径合理，并无碰撞、危险驾驶等行为发生，模型表现良好。

运行仿真平台的设备参数如下：处理器为 intel(R)Core(TM)i7-10750H CPU @2.60GHz 2.59 GHZ、内存 16G，GeForce RTX 2060 显卡，windows 操作系统，python（3.7）。

5.3 自动驾驶联合仿真

基于 SUMO 的自动驾驶车辆模型的构建实现了交通仿真中的 L5 级别理想化单车自动驾驶功能，但由于交通仿真软件大部分均缺少三维信息输入与三维仿真环境，因此，仍不能满足全链条的自动驾驶算法测试需求。本论文基于此需求，探索了交通仿真软件 SUMO 与自动驾驶仿真 Carla 的实时协同联合仿真方法。

5.3.1 仿真软件 Carla 介绍

自动驾驶仿真软件 Carla[87]是由西班牙巴塞罗那自治大学计算机视觉中心指导开发的开源仿真模拟器，目前被广泛用于自动驾驶系统的开发、训练与验证。Carla 依托于虚幻引擎（Unreal Engine）进行开发，使用服务器和多客户端的架构，支持高渲染的三维模型。Carla 提供许多开源的数字资产（包括建筑及车辆等）以及由这些资产搭建的简单测试场景。另一方面，Carla 支持使用由三维道路搭建软件 Roadrunner 制作输入路网和配套的高精地图。Carla 也支持包括激光雷达、多摄像头等多类传感器模型仿真，能够自由地调节环境的光照和天气，这弥补了现有交通仿真软件不具备传感器模型、无法支持三维信息的输入、车辆物理碰撞模型较为简单等缺陷。由于 Carla 提供了一整套高度灵活的 Python API 接口，可以便捷的实现对仿真中的车辆，信号灯等元素的控制，因此，也为实现其与交通仿真相结合的自动驾驶-交通联合仿真提供了可能性。

Carla 作为一款自动驾驶仿真软件具有交通仿真缺乏的各类传感器模型以及对于车辆的高精度控制功能，但由于其依托于高质量的渲染模型来保证仿真的高精度与视觉的真实感（包括建筑物、植被、交通标志、车辆、行人等）。因此，

其对于计算机配置等要求很高，无法做到大规模路网的仿真，且仅具备简单的交通控制模型。综上所述，若 Carla 与交通仿真能够结合，实现联合仿真，则能够弥补各自的不足与缺陷，很大程度上提高自动驾驶算法仿真测试的效果，并为自动驾驶仿真测试提供新的思路。

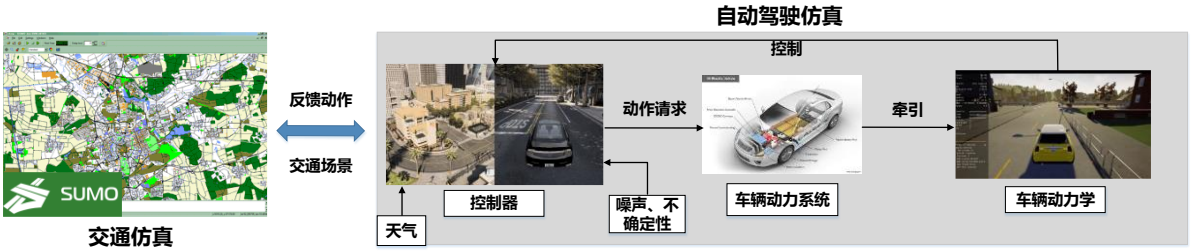


图 5-4 联合仿真体系框架图

5.3.2 3D 路网搭建

在实现联合仿真之前，需要搭建 Carla 必需的三维仿真环境，而 Roadrunner 则是 Carla 官网最推荐的三维道路编辑软件。Roadrunner（Design 3D scenes for automated driving simulation）是一款交互式道路编辑器，可用于针对自动驾驶系统仿真和测试设计三维场景，同时支持自由地创建区域特定的道路标志、以及设置和配置交叉路口处的交通信号配时、相位和行车路径等。本论文使用 Roadrunner 作为基础路网编辑工具构建苏州古城区域的三维仿真环境。

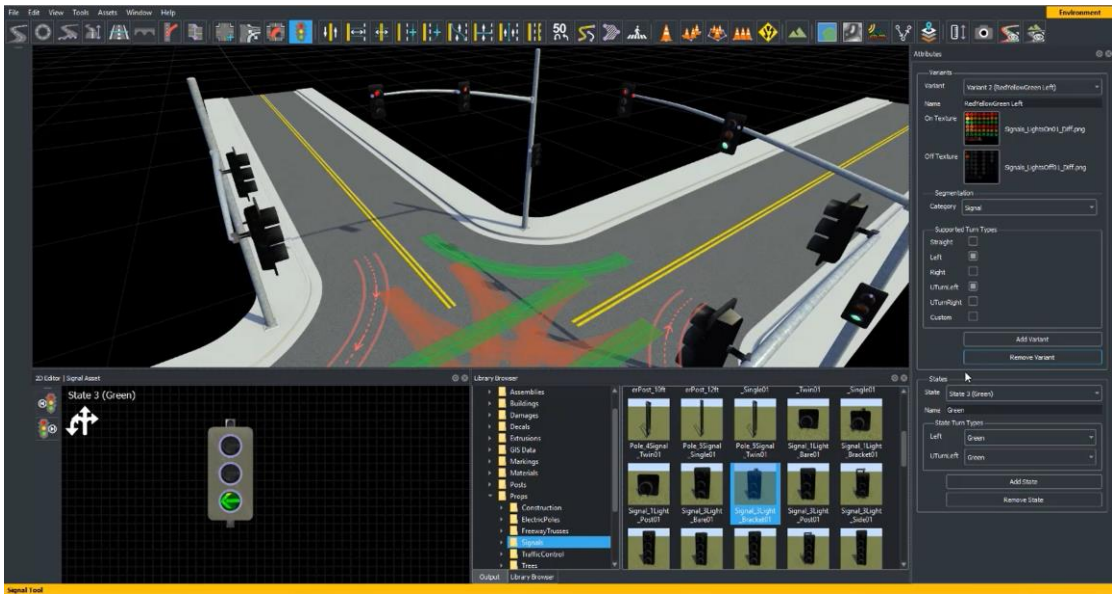


图 5-5 Roadrunner 软件使用示意图

基于该软件搭建三维仿真地图主要分为以下几个步骤：

步骤 1：以 OSM 文件作为底图导入 Roadrunner 中，并调整相关坐标系，使其地图中点为原点；

步骤 2：基于已有的地图模型进行优化，包括标志牌、信号灯等，同时结合已有三维模型库对三维环境进行搭建；

步骤 3：导出.fbx 文件（3D 模型文件）以及.xodr 文件（高精地图格式文件）

基于以上步骤即可完成基础的三维地图的搭建，为 Carla 仿真提供基础路网文件。同时，该软件支持手动交互式的在路网中增加树木、河流、道路标志标牌等元素，为自动驾驶车辆算法测试提供更真实的三维场景。

5.4 联合仿真

5.4.1 基础算法逻辑

实现 Carla 与 SUMO 的联合仿真主要通过 SUMO 的 TRACI 接口与 Carla 的 Python API 接口进行整合，主要分为以下几个步骤：

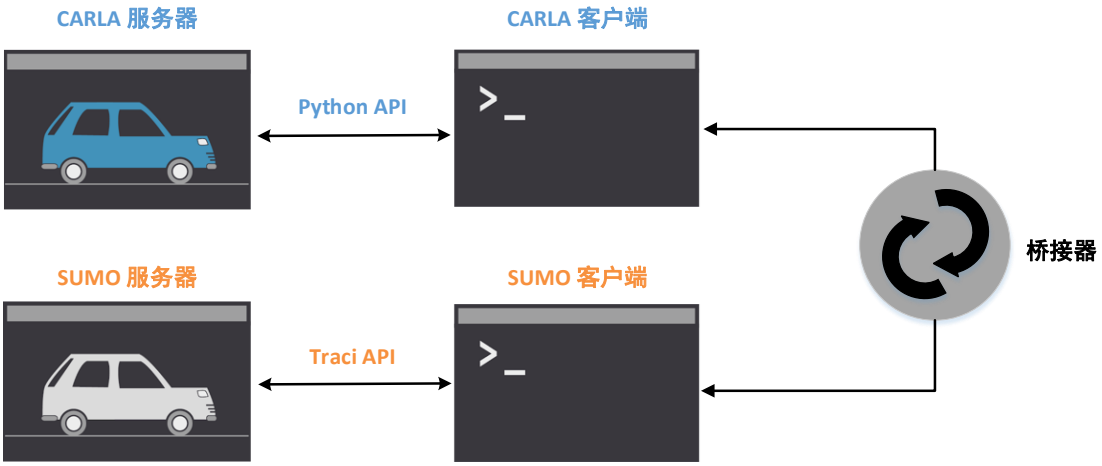


图 5-6 联合仿真桥接原理图

步骤 1：在 SUMO 端基于 Carla 的 blueprints 库预定义好不同的类，包括车辆的属性信息（vtypes）集成，如 ID，车辆类型，颜色，长度，宽度，高度，位置，方向角等；

步骤 2：SUMO 端读取配置文件与路网属性信息（.net.xml）；

步骤 3：Carla 端配置仿真客户端参数；

步骤 4：Carla 端读取配置文件与路网属性信息（.xodr）；

步骤 5：将两类信息进行匹配，如车辆类型等；

步骤 6：运行 run_synchronization.py（预编码实现两款软件相互通讯），同时对于同样类型的车辆，两款软件共享状态信息。

基于以上的算法逻辑与操作，则可实现 SUMO 与 Carla 的实时协同联合仿真，同时支持生成由 SUMO 控制或由 Carla 控制的两大类车辆。

5.4.2 联合仿真测试

基于上一节的联合仿真控制算法逻辑则可实现苏州古城区域的联合仿真，在此基础上，为了保证自动驾驶算法测试的完成性，尝试加入基于视觉控制的自动驾驶车辆（该模型以三维的视觉图形作为输入，并基于预定义的碰撞等规则实现车辆的控制），完成交通仿真软件 SUMO 提供真实的交通流场景（基于活动的多模式组合出行输入），自动驾驶仿真软件 Carla 提供真实的基于三维信息输入的自动驾驶车辆模型，完成了自动驾驶算法虚拟仿真测试全流程测试。

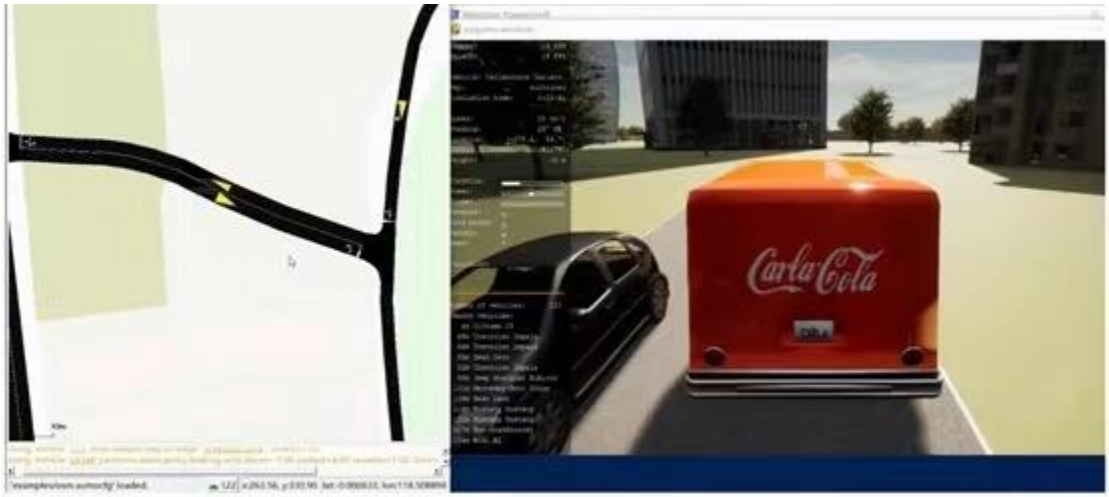


图 5-7 基于苏州古城路网的联合仿真测试图

实现 SUMO 与 Carla 的实时联合仿真不仅打通了交通仿真与车辆仿真领域的壁垒，同时为自动驾驶虚拟仿真测试提供了新的测试可能性。交通仿真软件虽然对于交通流的规律刻画的更加精细与准确，但大部分仅包含车辆、道路等的二维信息，许多场景下具有很大的局限性。而车辆领域内如自动驾驶仿真软件大多具备优秀的车辆动力学模型，同时支持强大的三维渲染场景，能够更加逼真的模拟

现实世界，但是往往忽略了对交通流规律的精准刻画与不同驾驶员的行为特征建模。因此，实现两类仿真软件的实时协同仿真能够进行优势互补，完成了自动驾驶虚拟仿真测试的全流程，具有重要的意义。

第六章 结论与展望

6.1 主要研究成果

仿真是解决交通问题的关键手段与方法之一。随着自动驾驶技术的发展，对于自动驾驶仿真测试、人类驾驶与自动驾驶混行交通流相关问题的研究成为了迫切的需求。然而，目前没有一款仿真软件能够同时提供真实的交通流场景与自动驾驶模型，为了解决上述问题，本论文基于深度强化学习的方法，基于交通仿真软件 SUMO 进行二次开发，构建自动驾驶行为决策模型，同时与自动驾驶仿真软件 Carla 进行协同联合仿真，完成全流程的自动驾驶虚拟仿真测试。论文的主要研究成果如下：

（1） 融合多源数据的仿真环境智能化搭建方法

论文通过剖析 SUMO 目前搭建仿真路网的缺陷与不足，提出了融合 SHP 与 OSM 的车道数补全方法以及考虑路网拓扑关系的新型路网道路 ID 修改方法。在此基础上，结合开源项目 SAGA，考虑了人类出行活动的连续性，完成了基于优化仿真路网的城市多模式活动组合出行，同时在苏州古城区域完成了测试与验证。

（2） 基于深度强化学习的自动驾驶车辆微观控制模型

论文通过对于经典的车辆跟驰、换道模型的模型机理分析，提出了自动驾驶车辆的跟驰与换道行为特征假设，考虑了自动驾驶车辆周围 6 辆车的状态信息，并设计了一种考虑效率、安全与舒适性的多维度奖励函数，基于深度强化学习 PPO 算法完成了模型的构建与训练，结果表明该模型能够按照奖励目标完成驾驶任务。

（3） 基于深度强化学习的自动驾驶宏观路径选择模型

论文在分析传统的路径规划问题基础上，提出了基于 K 条最短路的自动驾驶动态路径选择模型，解决了全局路径搜索存在维度爆炸等的问题，并结合深度强化学习 DQN 算法完成了模型的构建与训练，结果表明该模型能够选择最优路径，更快速、高效的到达终点。

（4） 基于深度强化学习的两阶段自动驾驶仿真模型

论文在完成基于强化学习的微观控制与宏观路径选择模型的基础上，完成了两个模型的一体化整合，在车辆进入路段上时执行路径选择动作，在路段中间行驶时执行微观跟驰与换道动作。同时在苏州古城路网中进行了测试，与传统仿真模型相比，自动驾驶车辆能够寻找到行程时间更短的路径并执行更合理的微观跟驰与换道动作。

（5） 结合 SUMO 与 Carla 的联合仿真测试方法

论文完成了交通仿真软件 SUMO 与自动驾驶仿真软件 Carla 的协同联合仿真，SUMO 提供真实的基于活动的多模式交通流场景，Carla 提供基于视觉的初级自动驾驶车辆算法，并在苏州古城的案例区域内进行了测试，完成了联合仿真的初探。

6.2 论文创新点

论文的主要创新点如下：

（1）提出了一种多源地图数据融合的 SUMO 仿真路网智能化搭建方法，解决了现有仿真路网搭建存在路网车道缺失、精度不高的问题。本论文提出的融合 OSM 与 GIS 的车道数优化与考虑网络拓扑结构的道路 ID 修改方法为 SUMO 仿真路网搭建提供了新的高精度、高便捷性的方法。

（2）建立了一种高泛化性的基于深度强化学习的自动驾驶微观控制与宏观路径选择相结合的仿真模型，突破了现有大部分交通仿真软件不具备自动驾驶模型的瓶颈。目前交通仿真软件无法准确地模拟自动驾驶车辆的驾驶行为，论文创新的提出了一个两阶段的深度学习自动驾驶车辆模型，第一阶段模型完成宏观路径选择，第二阶段模型完成自动驾驶车辆的微观跟驰、换道行为，且该模型具有高泛化性，能够迁移至任意仿真路网中进行测试，为混合交通流研究提供了基础。

（3）提出了一种以基于活动的交通出行作为输入的自动驾驶联合仿真测试方法，为自动驾驶算法虚拟仿真测试提供了新思路。目前自动驾驶仿真测试多以片段式场景测试为主，尚未有相关研究以真实还原的交通场景进行自动驾驶仿真测试，同时联合仿真测试方法能够弥补各自软件的不足，论文提出的基于 SUMO 与 Carla 的联合仿真测试方法结合了基于活动的交通出行输入，并在苏州古城的

真实路网中完成了测试，为后续接入真实车辆自动驾驶算法提供了真实的仿真场景。

6.3 未来研究展望

本文针对基于强化学习的自动驾驶车辆仿真与基于 SUMO 和 Carla 的自动驾驶联合仿真测试进行了相关研究，但受限于作者的理论知识水平和研究时间，在部分研究中尚有待进一步完善的地方，主要包括以下几点：

（1）本论文主要的研究对象为 L5 级别的单车自动驾驶车辆，没有考虑智能网联环境对于自动驾驶车辆的行为建模影响。在实现 5G、车联网等情况下，自动驾驶车辆可以更快速的获取范围更广的道路交通信息，很大程度上会影响车辆的宏观路径选择等行为（如，提前预知到前方发生事故，从而提前改变行驶路径），未来需结合智能网联对于自动驾驶车辆行为与决策的影响进行进一步的研究。

（2）自动驾驶车辆与人类驾驶车辆比较大的区别是自动驾驶车辆可以被政府等部门操控，未来很长一段时间会是人类驾驶与自动驾驶车辆混行的情况，目前论文中没有考虑自动驾驶车辆之间的相互合作，但对于政府部门而言，更关注区域或者城市的交通运行状况，考虑自动驾驶车辆之间的合作与博弈可以是新的探索。

（3）针对自动驾驶与交通仿真软件的联合仿真测试，本论文仅做了初步的测试与展示。在完成联合仿真的体系与流程后，研究不同自动驾驶车辆渗透率下的混合交通流特征对于交通管理部门管理未来混合交通流场景来说具有重要的意义。

致谢

韶华易逝，流水浮年。行文至此，意味着砥砺耕耘数百个日日夜夜的硕士研究生阶段即将划上句号。在此，我由衷的向三年内给予我关心、支持和帮助的老师、亲人、同学和朋友们表达感谢。

感谢我的导师刘志远教授。记得第一次和刘老师见面是在 2018 年的大年二十九，短短半个小时的交流便点燃了我心中科研的火种。在我硕士期间进行科研学习的每一阶段，刘老师都给予了非常大的帮助与指导。刘老师一丝不苟的科研态度，勇于打破常规的科研思维，严于律己的工作态度、严谨治学、潜心育人的工匠精神深深地影响着我，也让我产生了继续攻读博士，继续潜心科研的思想。在此，向刘老师致以衷心的感谢和深深的敬意。

感谢东南大学交通学院的王炜教授、刘攀教授、过秀成教授、程琳教授、李志斌教授、沈永俊教授、于斌教授、王晨副教授和蒙纳士大学的 Graham Currie 教授、Inhi Kim 老师、马振良老师在课程学习上的指导；感谢付晓副教授、陈景旭副教授、顾子渊副教授、张文波老师、童蔚苹老师在我研究生生涯中提供的帮助和指导，感谢我的校外指导老师戴一凡博士以及清华大学苏州汽车研究院的同事们在实习生活中对我的培养。

感谢 TLab 的每一位组员，TLab 是一支团结、有温度、有战斗力的团队，三年的时间让我学习、收获很多。感谢 TLab 的陈新元师兄、黄迪师兄、程启秀师兄、黄凯师兄、邢吉平师兄、陈为杰师兄、刘洋师兄、贾若师兄、刘少韦华师兄、顾宇师兄、沈培琳师姐、张奇师兄、夏严师兄、席殷飞师兄、李喆康师兄、俞俊师兄、王路濛师姐、袁诗琳师姐、袁钟琪师姐、宋雨嘉师姐、杨丽师姐、殷如阳师姐、张媛师姐、张凯师兄对我科研学习和项目实践方面的帮助和指导。感谢同届好友吕呈、王泽文、吴鑫骅、袁钰、黄江彦、安秦鹤、杨晨、余冠一的帮助，我们一起成长，相互帮助，共同进步！也感谢 TLab 的各位师弟师妹们，一直以来对我的帮助和支持。

最后，特别感谢挚爱的父母与家人们，漫漫求学路离不开家人们一直对我的默默支持、信任，衷心感谢你们的理解与支持。

论文完稿之际，体会到了做学问亦同做人，只有经过春种、夏耕、秋收、冬藏才能得以沉淀升华。

不忘初心，止于至善。这三年的点点滴滴都将成为我一生的财富来陪伴我谱写人生新的篇章！

参考文献

[1] Yuan Y . Application of Intelligent Technology in Urban Traffic Congestion[C]// 2020 International Conference on Computer Engineering and Application (ICCEA). 2020.

[2] 中华人民共和国统计局. 中国统计年鉴[M]. 中国统计局, 2020.

[3] 中华人民共和国统计局. 中国统计年鉴[M]. 中国统计出版社, 2018.

[4] 国家发展改革委. 智能汽车创新发展战略[R], 2018.

[5] 工业和信息化部. 车联网（智能网联汽车）产业发展行动计划[R], 2018.

[6] 中国电动汽车百人会研究部. 自动驾驶应用场景与商业化路径（2020）[R].

[7] Medrano-Berumen C, Akbas M I. Abstract Simulation Scenario Generation for Autonomous Vehicle Verification[C]//2019 SoutheastCon. Huntsville, AL, USA: IEEE, 2019: 1–6[2021–04–30].

[8] 当家移动绿色互联网技术集团有限公司（51VR）. 中国自动驾驶仿真技术研究报告（2019）[R], 2019.

[9] Wu C, Kreidieh A, Parvate K, 等. Flow: A Modular Learning Framework for Autonomy in Traffic[J/OL]. ArXiv:1710.05465 [Cs], 2020[2021–05–01]. http://arxiv.org/abs/1710.05465.

[10] 单肖年, 陈小鸿. 交通仿真模型融合微观车辆排放模型研究综述[J]. 交通运输工程与信息学报, 2021: 1–18.

[11] 王炜, 陈学武. 交通规划（第二版）[M]. 北京: 人民交通出版社, 2017.

[12] 王昊. 交通流理论及应用（第一版）[M]. 北京: 人民交通出版社, 2020.

[13] Hägerstrand T. Reflections on “What about people in regional science?”[J]. Papers in Regional Science - PAP REG SCI, 2005, 66: 1–6. DOI:10.1111/j.1435-5597.1989.tb01166.x.

[14] Courgeau D. Chapin F., Stuart Jr. — Human activity patterns in the city. Things people do in time and in space[J]. Population, 1976.

[15] Fried M, Havens J, Thall M. TRAVEL BEHAVIOR-A SYNTHESIZED THEORY[J]. Motor Vehicles, 1977.

[16] Lopez P A, Behrisch M, Bieker-Walz L, et al. Microscopic Traffic Simulation using SUMO[C]//The 21st IEEE International Conference on Intelligent Transportation Systems.

[17] Balmer M, Meister K, Rieser M, et al. Agent-based simulation of travel demand: structure and computational performance of MATSim-T[J]. 2008.

[18] Zhou X, Taylor J, Pratico F. DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration[J]. Cogent Engineering, 2014, 1(1): 961345.

[19] Merkuryeva G , Bolshakovs V . Vehicle Schedule Simulation with AnyLogic[C]// International Conference on Computer Modelling & Simulation. IEEE, 2010.

[20] 李国红, 张超, 姜磊. 全球自动驾驶技术专利发展态势分析[J]. 中国电信业, 2020(09): 70–75.

[21]Hata A Y, Wolf D F. Feature Detection for Vehicle Localization in Urban Environments Using a Multilayer LIDAR[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(2): 420–429.
[22]来飞, 黄超群, 胡博. 智能汽车自动驾驶技术的发展与挑战[J]. 西南大学学报(自然科学版), 2019, 41(08): 124–133.
[23]汤立波, 康陈. 车联网产业融合发展趋势[J]. 电信科学, 2019, 35(11): 96–100.
[24]付长军, 李斌, 乔宏章. 车联网产业发展现状研究[J]. 无线电通信技术, 2018, 44(04): 323–327.
[25]杨文彦. 基于社会力的自动驾驶汽车行人轨迹预测模型[J]. 公路交通科技, 2020, 37(8): 127–135.
[26]韩小健, 赵伟强, 陈立军,等. 基于区域采样随机树的客车局部路径规划算法[J]. 吉林大学学报(工学版), 2019, v.49;No.205(05):49-61.
[27]吴伟, 刘洋, 刘威,等. 自动驾驶环境下交叉口车辆路径规划与最优控制模型[J]. 自动化学报, 2020, 46(09): 1971–1985.
[28]Wang X, Fan J, Liu N. A Novel Decision-Making Algorithm of Autonomous Vehicle Based on Improved SVM[C]//Proceedings of the 2020 International Conference on Aviation Safety and Information Technology. Weihai City China: ACM, 2020: 643–648.
[29]Rehder T, Muenst W, Louis L, et al. Learning lane change intentions through lane contentedness estimation from demonstrated driving[C]//IEEE International Conference on Intelligent Transportation Systems.
[30]Rajamani R, Tan H S. Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons[J]. IEEE Transactions on Control Systems Technology, 2000, 8(4): P.695-708.
[31]Ritvik Sharma. A Survey of Reinforcement Learning Techniques[J/OL]. 2019. Unpublished, 2019[2021–05–01]. http://rgdoi.net/10.13140/RG.2.2.26529.15204. DOI:10.13140/RG.2.2.26529.15204.
[32]Moriarty D E, Handley S, Langley P. Learning Distributed Strategies for Traffic Control[J]. MIT Press, 1998.
[33]Rausch V, Hansen A, Solowjow E, et al. Learning a deep neural net policy for end-to-end control of autonomous vehicles[C]//2017 American Control Conference (ACC). . DOI:10.23919/ACC.2017.7963716.
[34]Eraqi H M, Moustafa M N, Honer J. End-to-End Deep Learning for Steering Autonomous Vehicles Considering Temporal Dependencies[J]. 31st Conference on Neural Information Processing Systems (NIPS 2017), MLITS workshop, 2017.
[35]Vahidi A, Eskandarian A. Research advances in intelligent collision avoidance and adaptive cruise control[J]. IEEE Transactions on Intelligent Transportation Systems, 2004, 4(3): 143–153.
[36]Moon S, Moon I, Yi K. Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance[J]. Control Engineering Practice, 2009, 17(4): 442–455.

[37]Huang Z, Xu X, He H, et al. Parameterized Batch Reinforcement Learning for Longitudinal Control of Autonomous Land Vehicles[J]. Systems, Man, and Cybernetics: Systems, IEEE Transactions on, 2017.
[38]张珊, 王蕾, 郭魁元,等. 基于交通事故的自动驾驶虚拟测试方法研究[J]. 中国汽车, 2020, No.338(05): 36–41.
[39]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, 026(001): 25–29.
[40]汤辉, 王立, 李志斌. 驾驶模拟器在自动驾驶系统中的应用研究[J]. 汽车文摘, 2020, No.528(01): 32–35.
[41]Suh J, Yi K, Jung J, et al. Design and evaluation of a model predictive vehicle control algorithm for automated driving using a vehicle traffic simulator[J]. Control Engineering Practice, 2016, 51: 92–107.
[42]周干, 张嵩, 罗悦齐. 自动驾驶汽车仿真测试与评价方法进展[J]. 汽车文摘, 2019, 519(04): 48–51.
[43]张微, 李鑫慧, 吴学易,等. 自动驾驶仿真技术研究现状[J]. 汽车电器, 2019(8): 13–15.
[44]Tong D S , Bhave A , Auweraer H . Simulation-Based Testing Framework for Autonomous Driving Development[C]// IEEE 2019 International Conference on Mechatronics. IEEE, 2019.
[45]雍加望, 冯能莲, 陈宁. 自动驾驶汽车硬件在环仿真实验平台研发[J]. 实验技术与管理, 2021, 38(02): 127-131+135.
[46]宋皓晨, 吴鼎新. 基于 Vissim 的车联网及自动驾驶车辆交通仿真研究[J]. 物流工程与管理, 2018, 40(09): 57–59.
[47]谭安祖, 余曼, 翁文勇,等. 基于开源仿真器的蜂窝 V2X 的性能分析[J]. 电子技术应用, 2020, 46(06): 93–96.
[48]D D J, A R. Simulation and Performance Analysis of CIT College Campus Network for Realistic Traffic Scenarios using NETSIM[J]. Procedia Computer Science, 2020, 171.
[49]Abdulhai B, Kattan L. Reinforcement Learning: Introduction to Theory and Potential for Transport Applications[J]. Canadian Journal of Civil Engineering, 2003, 30(6): 981–991. DOI:10.1139/l03-014.
[50]Abdoos M, Mozayani N, Bazzan A. Holonic multi-agent system for traffic signals control[J]. Engineering Applications of Artificial Intelligence, 2013, 26(5–6): 1575–1587.
[51]Khamis M. Adaptive Multi-Objective Reinforcement Learning For Traffic Signal Control Based on Cooperative Multi-Agent Framework[J]. 2013.
[52]Zhao D, Dai Y, Zhen Z. Computational Intelligence in Urban Traffic Signal Control: A Survey[J]. IEEE Transactions on Systems Man & Cybernetics Part C, 2012, 42(4): 485–494.
[53]Jin J, Ma X. A group-based traffic signal control with adaptive learning ability[J]. Engineering Applications of Artificial Intelligence, 2017, 65: 282–293.

[54] Clempner J B, Poznyak A S. Modeling the multi-traffic signal-control synchronization: A Markov chains game theory approach[J]. Engineering Applications of Artificial Intelligence, 2015, 43: 147–156.
[55] Mckenney D, White T. Distributed and adaptive traffic signal control within a realistic traffic simulation[J]. Engineering Applications of Artificial Intelligence, 2013, 26(1): 574–583.
[56] Sallab A E, Abdou M, Perot E, et al. End-to-End Deep Reinforcement Learning for Lane Keeping Assist[J]. 2016.
[57] Chae H, Kang C M, Kim B D, et al. Autonomous Braking System via Deep Reinforcement Learning[J]. IEEE, 2017.
[58] Xiaopeng, Zong, Guoyan, et al. Obstacle Avoidance for Self-Driving Vehicle with Reinforcement Learning[J]. SAE International Journal of Passenger Cars - Electronic and Electrical Systems, 2018.
[59] Shalev-Shwartz S, Ben-Zrihem N, Cohen A, et al. Long-term Planning by Short-term Prediction[J]. 2016.
[60] Treder M , Lauermann J L , Eter N . Automated detection of exudative age-related macular degeneration in spectral domain optical coherence tomography using deep learning[J]. Graefe\"s Archive for Clinical and Experimental Ophthalmology, 2017.
[61] Kendall A, Hawke J, Janz D, et al. Learning to Drive in a Day[C]//2019 International Conference on Robotics and Automation (ICRA).
[62] Codecà L, Erdmann J, Cahill V, et al. SAGA: An Activity-based Multi-modal Mobility Scenario Generator for SUMO[C].
[63] Codeca L , Frank R , Engel T . Luxembourg SUMO Traffic (LuST) Scenario: 24 Hours of Mobility for Vehicular Networking Research[C]// Vehicular Networking Conference. IEEE, 2016.
[64] Toader B, Cantelmo G, Popescu M, et al. Using Passive Data Collection Methods to Learn Complex Mobility Patterns: An Exploratory Analysis[C]//International Conference on Intelligent Transportation Systems.
[65] Yeung S, Aziz H, Madria S. Activity-Based Shared Mobility Model for Smart Transportation[C]//2019 20th IEEE International Conference on Mobile Data Management (MDM).
[66] Bowman J L, Ben-Akiva M E. Activity-based disaggregate travel demand model system with activity schedules[J]. Transportation Research Part A Policy & Practice, 2001, 35(1): 1–28.
[67] Guido C, Francesco V. Incorporating activity duration and scheduling utility into equilibrium-based Dynamic Traffic Assignment[J]. Transportation Research Part B Methodological, 2018: S019126151730084X-.
[68] Bhattacharyya R, Wulfe B, Phillips D, et al. Modeling Human Driving Behavior through Generative Adversarial Imitation Learning[J]. 2020.
[69] Pomerleau D. ALVINN : An autonomous land vehicle in a neural network[J]. Advances in Neural Information Processing Systems, 1989, 1.

[70]Koopman P, Wagner M. Challenges in Autonomous Vehicle Testing and Validation[J]. SAE International journal of transportation safety, 2016, 4: 15–24.
[71]Codevilla F, Santana E, López A M, et al. Exploring the Limitations of Behavior Cloning for Autonomous Driving[J]. IEEE, 2019.
[72]秦严严, 王昊, 王炜,等. 混有 CACC 车辆和 ACC 车辆的异质交通流基本图模型[J]. 中国公路学报, 2017, 30(010): 127–136.
[73]Sutton R, Barto A. Reinforcement Learning: An Introduction[M]. Reinforcement Learning :An Introduction, 1998.
[74]Chandler R E, Montroll H. Traffic Dynamics: Studies in Car Following[J]. Operations Research, 1958, 6(2): 165–184.
[75]Kometani E, Sasaki T. On the stability of traffic flow[J]. j.opns.res.japan, 1958.
[76]Treiber M, Hennecke A, Helbing D. Congested Traffic States in Empirical Observations and Microscopic Simulations[J]. Physical Review E, 2000, 62: 1805–1824.
[77]Zheng Z. Recent Developments and Research Needs in Modeling Lane Changing[J]. Transportation Research Part B: Methodological, 2014, 60: 16–32. DOI:10.1016/j.trb.2013.11.009.
[78]Gipps P G. A Model for the Structure of Lane-Changing Decisions[J]. Transportation Research Part B: Methodological, 1986, 20(5): 403–414. DOI:10.1016/0191-2615(86)90012-3.
[79]Erdmann J. SUMO’s Lane-Changing Model[M/OL]. BEHRISCH M, WEBER M, //Modeling Mobility with Open Data. Cham: Springer International Publishing, 2015: 105–123[2021–05–01]. http://link.springer.com/10.1007/978-3-319-15024-6_7. DOI:10.1007/978-3-319-15024-6_7.
[80]Schulman J , Wolski F , Dhariwal P , et al. Proximal Policy Optimization Algorithms[J]. 2017.
[81]Kuhnle A, Schaarschmidt M, Fricke K. Tensorforce: a TensorFlow library for applied reinforcement learning[M/OL]. https://github.com/tensorforce/tensorforce.
[82]Dijkstra E W. A note on two problems in connexion with graphs[J]. Numerische mathematik, 1959, 1(1): 269–271.
[83]Hoffman W, Pavley R. A Method for the Solution of the N Th Best Path Problem[J]. J. ACM, 1959, 6(4): 506–514. DOI:10.1145/320998.321004.
[84]Yen J Y. Finding the K Shortest Loopless Paths in a Network[J]. Management Science, 1971, 17(11,): 712–716.
[85]Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. Computer Science, 2013.
[86]Kingma D, Ba J. Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.
[87]Dosovitskiy A, Ros G, Codevilla F, et al. Carla: An Open Urban Driving Simulator[C]//Proceedings of the 1st Annual Conference on Robot Learning.