基于分层强化学习的自动驾驶决策控制算法

李伟东[1],马草原[1],史 浩[2],曹 衡[2]

（1.大连理工大学 汽车工程学院,辽宁 大连 116024;2.华电煤业集团数智技术有限公司,北京 102488）

摘 要:针对强化学习模型在自动驾驶任务中存在收敛速度慢和应用场景单一的问题,提出了一种两层的强化学习框架来替代传统的决策层与控制层。决策层将驾驶行为分为车道保持、左变道和右变道,决策层选择对应的行为后,通过改变控制层输入的方式完成该行为。然后,结合强化学习和在线专家提出了一种训练控制层的新方法RL_COE。最后,在Carla中搭建了高速公路仿真环境对本文算法进行验证,并与强化学习基线算法进行比较,结果表明:该方法大大提高了算法的收敛速度和稳定性,可以更好地完成驾驶任务。

关键词:车辆工程;自动驾驶;分层强化学习;在线专家;Carla

中图分类号:U495 **文献标志码**:A **文章编号**:1671-5497(2025)05-1798-08

DOI:10.13229/j.cnki.jdxbgxb.20230891

An automatic driving decision control algorithm based on hierarchical reinforcement learning

LI Wei-dong[1],MA Cao-yuan[1],SHI Hao[2],CAO Heng[2]

(1. *School of Automotive Engineering, Dalian University of Technology, Dalian* 116024, *China*; 2. *Huadian Coal Industry Group Digital Intelligence Technology Co., Ltd., Beijing* 102488, *China*)

Abstract: To address the issues of slow convergence and limited applicability of reinforcement learning models in automatic driving tasks, a two-tiered reinforcement learning framework is proposed as a substitute for the decision and control layers. Within this framework, the decision layer categorizes driving behaviors into lane keeping, left lane change, and right lane change. Subsequently, after the decision layer selects the appropriate behavior, execution is achieved by modifying the input to the control layer. Then, in combination with reinforcement learning and online experts, a new method RL_COE is proposed to train the control layer. Finally, the proposed algorithm is verified in the highway simulation environment based on Carla and compared with the baseline reinforcement learning algorithm. The results show that this method significantly improves the convergence and stability of the algorithm, and can better perform the driving task.

Key words: vehicle engineering; automatic driving; hierarchical reinforcement learning; online experts; Carla

---

收稿日期:2023-08-22.

基金项目:辽宁省科技创新重大专项项目(ZX20220560);辽宁省科技计划项目(ZX20220771);华电煤业集团数智技术有限公司项目(HDSZ/WZ2023/JSKF/246).

作者简介:李伟东(1975-),男,副教授,博士.研究方向:智能车辆,人工智能,汽车轻量化.
E-mail: liweidong@dlut.edu.cn

0　引　言

无论是在学术界还是工业界，自动驾驶都是当下的研究热点之一[1]。自动驾驶可以缓解交通拥堵、减少交通事故的发生以及减少温室气体的排放[2]。自动驾驶任务可以分解为感知、决策、控制3个子任务，其中决策模块是分析感知结果并为控制模块提供规划信息的重要桥梁。目前的主流决策模型可以分为基于学习和基于规则两大类[3]。

基于学习的方法又可以分为模仿学习和强化学习。模仿学习通过神经网络拟合大量驾驶数据以接近人类驾驶轨迹。但是，模仿学习存在误差累计问题，车辆常常会进入数据没有覆盖的危险场景，导致神经网络作出错误的决策[4]。

强化学习不仅有神经网络的强大拟合能力，而且具备解决序列化决策问题的能力以及在交互中学习的特点[5]。近年来有许多研究人员将其应用于自动驾驶任务中。Kendall等[6]将前视摄像头图片数据作为输入，使用DDPG算法在仿真环境中训练了一个车道保持模型，并将其迁移到了真实世界中；Wurman等[7]在Gran Turismo赛车游戏中，使用结合多步奖励的SAC算法在混合场景下训练，击败了世界上最优秀的电子竞技玩家；李文礼等[8]采用五次多项式拟合高速公路换道路径，根据横向位置偏差和横摆角误差建立了奖励函数，使用DDPG算法进行训练，算法性能超越了传统MPC算法。但是传统强化学习算法存在许多无用探索，且没有利用先验知识，导致算法学习效率低，模型收敛慢[9]。

为了解决这个问题，研究人员提出了许多优化方法。例如Zhu等[10]在强化学习训练过程中加入了监督器，用于约束并修正危险试错动作，将模型与基于MPC的ACC算法进行了比较，实验结果表明：该模型具有安全、高效、舒适的速度控制能力，并且优于人类驾驶员；Huang等[11]使用提前收集好的专家数据，在训练过程中，通过规范算法策略与专家策略之间的差异，使专家策略指导算法的学习，在仿真环境中的测试结果表明算法性能超过了人类。然而强化学习在处理复杂问题时还是存在维度灾难问题，导致模型训练困难。

针对上述问题，本文提出了一种两层的强化学习框架分别代替自动驾驶任务中的决策层和控制层，决策层使用强化学习方法来学习宏观行为决策，使用改变状态空间的方法使控制层完成相应行为，以此降低强化学习求解问题的复杂度，提高算法的收敛速度。在控制层的训练过程中使用在线专家和强化学习相结合的方法，充分利用了先验经验，有效解决了强化学习在自动驾驶任务中的“冷启动”问题。最后在基于Carla搭建的高速公路仿真环境中，验证了算法的性能。

1　算法框架

1.1　深度Q网络

深度Q网络（Deep Q-network，DQN）是一种强化学习算法，旨在通过深度学习方法来解决马尔科夫决策过程的问题[12]。DQN算法通常适用于具有离散动作空间的强化学习问题，其目标是拟合动作价值函数（Q函数），用于衡量在给定状态下采取不同动作的长期回报。该函数可以表示为：

$$Q_\pi(s_t,a_t)=\mathop{\mathrm{E}}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t,s_{t+1})\right] \tag{1}$$

式中：γ为折扣因子，是小于1的超参数，保证该函数有界；s_t和a_t分别为某一时刻的状态及该时刻所采取的动作；s_{t+1}为下一时刻的状态；π和τ分别为当前的策略和使用该策略产生的交互数据；R为奖励函数，一般为人为设计，衡量该状态下所采取动作的好坏程度。

DQN算法使用时序差分方法，最小化Q值和目标Q值之间的均方误差来学习Q网络的参数，这可以用以下损失函数来表示：

$$L_Q=\mathop{\mathrm{E}}_{(s_t,a_t,r_t,s_{t+1},d_t)\in D}[(Q(s_t,a_t)-r_t-\gamma\max_a\hat{Q}(s_{t+1},a))^2] \tag{2}$$

式中：D为经验回放池，储存在之前探索中的经验，包括状态s_t、动作a_t、奖励r_t、下一时刻状态s_{t+1}以及回合结束标志d_t，训练过程中，这些经验被随机抽样以用于网络更新；$\hat{Q}$为目标网络，训练开始时其参数和Q网络相同，在训练过程中，使用软更新的方式更新目标网络；$\max_a\hat{Q}(s_{t+1},a)$表示使用目标网络寻找下一时刻的最大Q值。

目标网络可以稳定训练过程，降低数据的相关性，并提高算法的收敛性。

1.2　柔性演员-评论家算法

柔性演员-评论家算法（Soft actor-critic，SAC）使用演员评论家框架，同时将策略的信息

熵加入奖励中以鼓励探索[13]。目的为实现输出动作的均值收敛到最优,同时在面对环境扰动时,也能使智能体迅速回归正常状态,算法的目的为寻找最优策略:

$$\pi^{*}=\arg\max_{\pi}\mathrm{E}_{\tau\sim\pi}\sum_{t=0}^{\infty}[\gamma^{t}R(s_t,a_t,s_{t+1})+\alpha\gamma^{t}H(\pi(\cdot|s_t))] \tag{3}$$

式中:α为权衡系数,表示对探索的重视程度;$\pi(\cdot|s_t)$为策略函数,其输出为动作a_t的均值μ和方差σ;$H(\pi(\cdot|s_t))$为策略的信息熵。

考虑策略信息熵的动作价值函数可以表示为:

$$Q_{\pi}(s_t,a_t)=\mathrm{E}_{\tau\sim\pi}[\sum_{t=0}^{\infty}\gamma^{t}R(s_t,a_t,s_{t+1})+\alpha\sum_{t=1}^{\infty}\gamma^{t}H(\pi(\cdot|s_t))] \tag{4}$$

SAC算法中的3个函数均使用神经网络拟合,分别为策略网络$\pi(s_t)$和动作价值网络$Q(s_t,a_t)$以及目标动作价值网络$\hat{Q}_{\pi}(s_t,a_t)$。与DQN类似,目标网络采用软更新的方式更新参数,动作价值网络使用时序差分方法进行更新,其损失函数可以表示为:

$$L_Q=\mathrm{E}_{(s_t,a_t,r_t,s_{t+1},d_t)\in D}[(Q(s_t,a_t)-y(r_t,s_{t+1},d_t))^2] \tag{5}$$

$$y(r_t,s_{t+1},d_t)=r_t+\gamma(1-d_t)[\hat{Q}(s_{t+1},a'_{t+1})-\alpha\log\pi(a'_{t+1}|s_{t+1})] \tag{6}$$

式中:$a'_{t+1}=\pi(s_{t+1})$表示对下个时刻动作的估计。

策略网络通过最大化Q网络估计的未来奖励值进行更新,其损失函数可以表示为:

$$L_{\pi}=\mathrm{E}_{(s_t,a_t,r_t,s_{t+1},d_t)\in D}[Q(s_t,\tilde{a}_t)-\alpha\log\pi(\tilde{a}_t|s_t)] \tag{7}$$

式中:$\tilde{a}_t=\pi(s_t)$,这是由于回放池中的数据在策略网络更新后不再符合当前策略网络的分布,因此,需要对动作进行重新采样。

1.3 结合在线专家的强化学习方法

传统强化学习方法存在收敛慢和不稳定的问题,一些研究人员通过先使用模仿学习预训练,然后再使用强化学习训练来解决该问题[14]。但是,人类在示范时会选择较好的动作,因此,示范数据不可能覆盖所有的边界情况,这会造成动作价值函数的估计过于乐观[15]。

鉴于上述问题,本文提出了一种将在线专家和强化学习相结合的方法,该方法不需要人类的示范数据,并且容易与无模型强化学习算法相结合,本文将其称为RL_COE(Reinforcement learning combined with online experts)方法。

经验回放使智能体可以利用过往的经验来更新策略,极大提高了数据效率。交互经验的形式为$<s_t,a_t,r_t,s_{t+1},d_t>$,为了配合本文算法,将其修改为$<s_t,a_t,a_t^{e},r_t,s_{t+1},d_t>$。智能体与环境交互时,根据当前状态$s_t$和策略网络得到动作$a_t=\pi(s_t)$,同时在线专家得到专家动作$a_t^{e}=\pi^{e}(s_t)$,环境执行动作$a_t$,并计算出下一时刻的状态$s_{t+1}$,同时根据奖励函数给出奖励$r_t$。

在线专家可以是根据经验设计的简单、粗糙的策略,因为汽车控制领域已经发展得较为成熟,本文采用了PID控制器作为在线专家。策略网络的更新过程中,将拟合专家策略的损失项添加到损失函数中,结合式(7)可以得到SAC_COE策略函数损失函数:

$$L_{\pi}=\mathrm{E}_{(s_t,a_t,a_t^{e},r_t,s_{t+1},d_t)\in D}[Q(s_t,\tilde{a}_t)-\alpha\log\pi(\tilde{a}_t|s_t)+(\tilde{a}_t-a_t^{e})^2] \tag{8}$$

在训练初期,策略网络有较大的随机性,智能体在选择动作时有很大的概率会采取不安全的动作,在线专家可以为策略网络提供大致的更新方向,从而有效解决强化学习的"冷启动"问题,并解决专家数据覆盖不了边界场景的问题。随着训练的进行,策略网络的损失逐渐增大,强化学习中的损失项开始发挥主要作用,智能体逐渐完善自身策略,并在探索中逐步提升性能。

1.4 分层强化学习框架

传统强化学习算法通常只能针对特定驾驶场景进行设计,使用传统的强化学习算法综合学习多个驾驶任务往往会产生维数灾难[16]。为了解决该问题,本文基于一种分层强化学习模型H_DQN[17],并对其做出了改进以适应自动驾驶任务。改进后的模型架构如图1所示,模型主要包括决策层π_{b}和控制层π_{c}。

决策层负责选择是否变道,采用DQN算法,其输出为车道保持、左变道和右变道3种子行为,因此,其动作空间被定义为三维的独热编码向量。控制层负责跟踪目标轨迹以及与前车保持安全距离,使用SAC_COE算法,其输出为控制刹车油门及方向盘的二维的连续变量$[a_{\mathrm{a}},a_{\mathrm{s}}]$,$a_{\mathrm{a}}$和$a_{\mathrm{s}}$的范围都为$[-1,1]$。$a_{\mathrm{a}}$为正值表示使用加速踏板,负值表示使用刹车。$a_{\mathrm{s}}$为正值表示方向盘向左旋转,负值表示方向盘向右旋转。

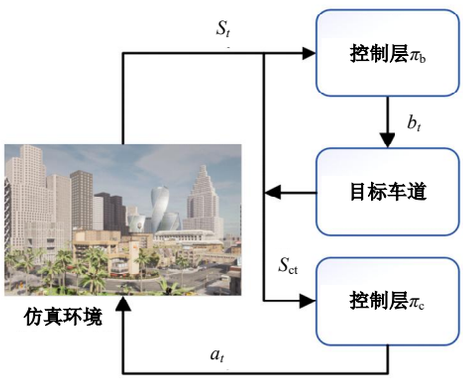


图 1　算法框架示意图

Fig. 1　Schematic diagram of algorithm framework

控制层和决策层以不同的时间步运行，控制层的执行频率和仿真频率相同，控制层每运行一定步数后，决策层运行一次。决策层根据环境状态s_t输出变道指令$b_t=\pi_b(s_t)$，当决策层不执行或输出车道保持时，目标车道不变，决策层输出向左或向右变道会改变目标车道。然后环境状态s_t结合目标车道上的轨迹点信息，生成控制层状态s_{ct}，而控制层会跟踪目标轨迹点，从而完成决策层输出的变道指令，在模型中体现为输出动作$a_t=\pi_c(s_{ct})$并作用到环境中。

如果两层策略同时训练，可能会导致上层策略不停地选择同一个行为使算法退化为单层模型[18]。因此，本文先训练控制层，然后冻结训练好的控制层参数并与决策层结合整体训练。

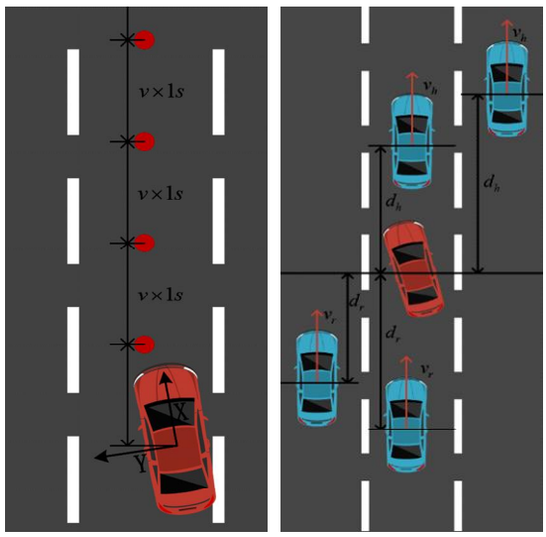
2　实验设计

2.1　状态空间

为了更加贴合实际场景，使用仿真环境的真值加入噪声扰动作为实验所用状态量，噪声服从区间$[-0.05v_{true},0.05v_{true}]$上的均匀分布，其中$v_{true}$为仿真环境真值。环境状态主要包括周围车辆的信息以及自车信息，控制层状态还包括目标轨迹点信息。

本文采用了目标车道中心的5个等间距航点作为目标轨迹点信息，如图2(a)所示。这5个点表示在目标车道中心未来5 s内的期望轨迹，并将每个点的2D绝对坐标转换为车辆坐标系下的2D坐标。为了描述车辆质心与车道中心的偏移程度，目标轨迹点信息还包括目标车道中心在车辆坐标系下的横向距离e。

本文将每条车道上的车辆信息表示为：$[d_r/d_d、v_r/v_t、d_h/d_d、v_h/v_t]$，如图2(b)所示。其中，



(a)目标轨迹点　　(b)周围车辆

图 2　状态空间示意图

Fig. 2　Schematic diagram of state space

d_d为对车辆的最大检测距离，取100 m；d_h和d_r分别为自车质点与前方、后方车辆在车道中心的投影距离；v_h和v_r为前后车的速度；v_t为目标速度。

当车辆前方检测距离内没有其他车辆时，令$d_h=d_d$，$v_h=v_t$。当车辆后方检测距离内没有其他车辆时，令$d_r=d_d$，$v_r=0$。这样的设计可以表示一个安全的状态。使用$[-1,-1,-1-1]$来表示该车道不存在。整体的周围车辆信息包括自车所在车道和左右两条车道上的车辆。

自车信息包括自车的速度v，是否发生碰撞的标志位c，如果发生碰撞$c=1$，其他情况下为0。

2.2　奖励函数

奖励函数在强化学习中起着至关重要的作用，本文所设计的框架中控制层和决策层所负责的任务不同，因此，分别为其设置不同的奖励函数。

根据控制层所负责跟车与轨迹跟踪任务，控制层的奖励函数如式(9)所示：

$$R_c=r_{vc}+0.3r_e+0.4r_d+0.3r_{steer}+r_c \tag{9}$$

式中：r_{vc}为车速奖励，鼓励自身车速v在前方检测距离内无车的情况下接近目标车速v_t，在有车的情况下接近前车车速v_h，在状态空间中，前方无车情况下$v_h=v_t$，所以该奖励项可以表示为：

$$r_{vc}=1-\frac{|v-v_h|}{v_h} \tag{10}$$

r_e为偏离车道中心惩罚，保证车辆在道路中心行驶：

$$r_e=-\frac{|e|}{3.5} \tag{11}$$

r_d为自车与前车的距离惩罚,保证自车与前车保持一定的安全距离:

$$r_d=\begin{cases}\dfrac{d_h}{d_d}-1, d_h\leqslant 0.5d_d\\ 0, \text{else}\end{cases} \tag{12}$$

r_{steer}为舒适性惩罚,其目的为防止方向盘抖动,以及防止过大的方向盘转角:

$$r_{steer}=-a_s{}^2 \tag{13}$$

r_c为碰撞惩罚,保证行驶安全性:

$$r_c=-10c \tag{14}$$

因为本文采用先训练控制层然后整体训练的方式,决策层只需要考虑通行效率和变道时机,故其奖励函数可以设计得简单一些:

$$R_b=r_{vb}+r_c \tag{15}$$

式中:r_{vb}为通行效率奖励项,鼓励车辆按照目标车速v_t行驶,在前方车辆车速较慢的情况下合理、安全的变道:

$$r_v=1-\frac{|v-v_t|}{v_t} \tag{16}$$

2.3 仿真环境

使用自动驾驶仿真平台进行算法验证可以降低成本,并且在算法迭代过程中更容易验证算法的性能。Carla仿真平台提供了多样化的地图和车辆模型,提供了灵活的Python接口,允许用户对车辆、交通、天气等参数进行设置,而且集成了优秀的车辆动力学模型[19]。

为验证本文算法的性能,使用Carla仿真平台搭建了高速公路仿真环境,用于算法的训练和验证。主车车型为Carla内置的奔驰E级轿车,目标速度设定为80 km/h,在路径上随机生成20个由仿真平台控制的npc车辆来模拟交通环境,npc车辆的速度在60~100 km/h随机选择。仿真平台以10 Hz的频率运行,决策层每5 s运行一次,每一回合的仿真时长为100 s。选择Carla内置地图town 05中的一段三车道环形道路进行训练,在town 04的四车道道路中进行验证,地图和路线如图3所示。

实验使用Python 3.7和Calar 0.9.13来搭建仿真平台,使用Pytorch 1.13.0训练神经网络。实验所用的计算机配备英伟达RTX3060显卡,和英特尔12代i7处理器。

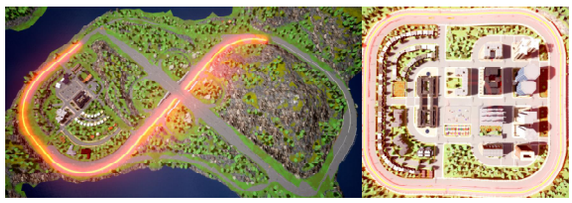


图3 仿真环境地图

Fig. 3 Schematic diagram of state space

3 实验结果

3.1 控制层训练

为验证结合在线专家对强化学习算法性能的提升效果,使用SAC_COE和DDPG_COE在town 05地图中训练500个回合,并与对应的基线算法进行比较。在控制层的训练过程中使用简单的策略进行随机变道,变道策略和决策层的运行频率一致。变道策略为:若旁车道在检测范围内没有车辆,而且本车道前方车辆速度低于目标速度,则以50%的概率选择变道。

几种算法均采用了相同的网络结构和超参数,算法超参数如表1所示,网络的隐藏层均为[64×128×128×64×16],在训练开始前对所有网络的参数进行了归一化处理。4个模型在训练过程中的累计奖励曲线如图4所示。

实验结果表明:结合了在线专家的两种算法在20回合左右就收敛到了较好的效果,相比之下,SAC算法在300回合左右才开始收敛,DDPG算法一直没有收敛到很好的效果。说明在线专家在训练前期起到了良好的引导作用,极大加快了算法的收敛速度。此外,SAC_COE算法的稳定性明显优于DDPG_COE,这是由于DDPG依赖于外部噪声,而SAC算法输出的为动作为高斯分布,抗干扰能力更强。

使用训练好的策略网络在town 04中进行100回合的测试。测试结果如表2所示,可以看出

表1 算法超参数

Table 1 Algorithm hyperparameters

超参数	值
折扣因子	9.9×10^{-1}
目标熵	-2
策略网络学习率	2.5×10^{-4}
价值网络学习率	5×10^{-4}
权衡系数学习率	5×10^{-4}
软更新率	1×10^{-2}
经验回放池容量	5×10^{6}
单次训练样本数量	512

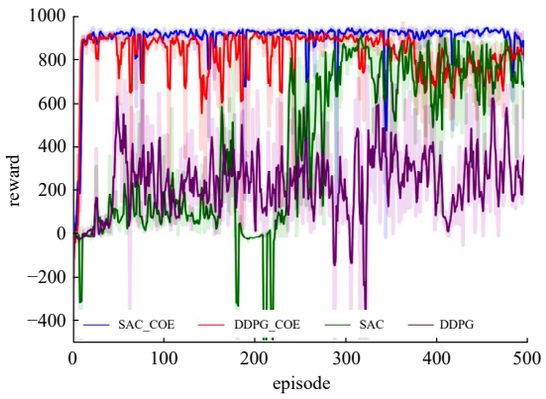


图 4　控制层训练过程

Fig. 4　Training process of control layer

表 2　控制层测试结果

Table 2　Control layer test results

算法	成功率/%	平均累计奖励	累计奖励标准差
SAC_COE	100	928.2	13.3
DDPG_COE	100	908.4	23.6
SAC	95	806.6	155.7
DDPG	23	532.1	197.4
PID	98	857.8	32.1

SAC_COE 和 DDPG_COE 无论是在平均累计奖励还是稳定性上都超过了 PID 算法。其中 SAC_COE 有最高的平均累计奖励和最好的稳定性。而 SAC 和 DDPG 两种基线算法均表现一般。说明本文算法有着良好的泛化性、稳定性，且对专家策略起到了很好的优化效果。

随后将 npc 车辆的数目设置为 0，并将变道策略关闭，在 town 04 地图中测试了算法的车道保持性能。图 5 展示了几种算法在车道保持场景中偏离车道中心距离的结果，表 3 为测试结果的统计值。

实验结果表明：虽然 3 种算法在弯道处均存在横向偏差，但是 SAC_COE 在进入弯道时有更

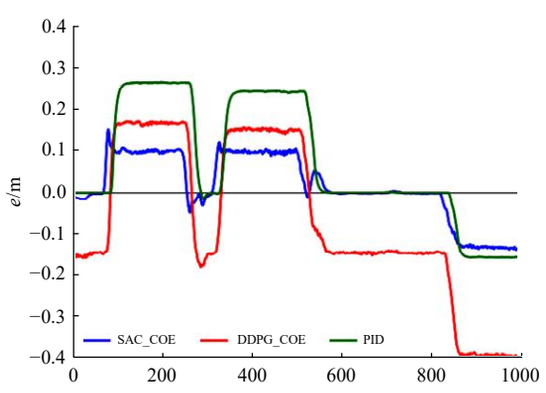


图 5　车道保持横向偏差

Fig. 5　Lateral deviation of lane maintenance

表 3　横向偏差对比

Table 3　Comparison of lateral deviation

算 法	横向偏差平均值/m	横向偏差标准差/m
SAC_COE	0.07	0.05
DDPG_COE	0.19	0.10
PID	0.12	0.11

快的反应速度，在弯道处的横向偏差最小，而且有最小的标准差。综合结果表明：SAC_COE 有最强的车道保持能力和最好的稳定性。

3.2　决策层训练

为了验证整体算法的性能，将训练好的 SAC_COE 参数固定，作为换道策略的执行器，然后将换道策略更改为 DQN 算法在 npc 车辆数目为 25 的 town 05 地图中训练 1 000 个回合。并将其与基于简单换道规则的 SAC_COE 和不进行换道的 SAC_COE 进行比较，训练结果如图 6 所示。

实验结果表明：虽然决策层在训练前期会作出不正确的换道决策从而导致碰撞，导致累计奖励较低，但是在训练后期随着经验的累积，决策层逐渐学会了选择合适的变道时机，累计奖励值也超过了两种对比策略。

随后使用训练好的整体算法在 npc 车辆数目为 25 的 town 04 地图中进行 100 回合的测试。表 4 为测试结果，不选择换道的策略可能会导致跟随低速车辆，使平均累计奖励较低。简单的换道规则对平均累计奖励有所提升，但是效果仍然一

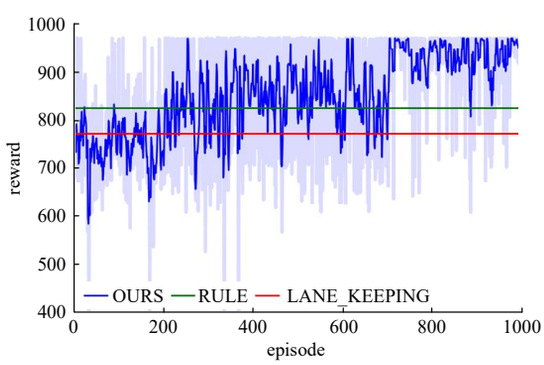


图 6　决策层训练过程

Fig. 6　Training process of decision layer

表 4　决策层测试结果

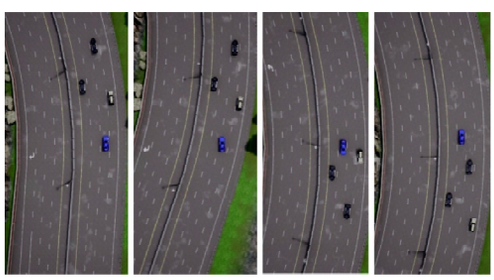
Table 4　Test results of decision makers

算 法	成功率/%	平均累计奖励	累计奖励标准差
OURS	100	925.1	56.9
RULE	97	825.6	148.4
LANE_KEEPING	100	773.2	183.6

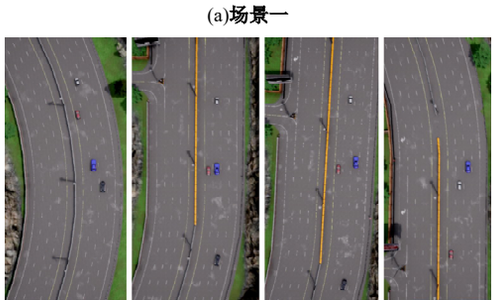
般,基于DQN的换道策略有最高的平均累计奖励和最好的稳定性,说明本文算法有良好的稳定性和泛化性。

为了充分展示本文算法在高速公路环境中的性能,设计了4个典型工况,以测试整体算法在不同交通场景下的表现,结果如图7所示。

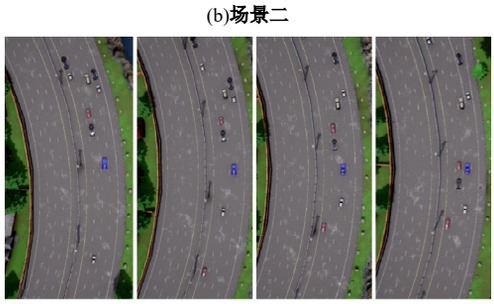
图中蓝色车辆使用本文算法控制,其他车辆为npc车辆。场景一中,除了自车左侧车道,前方的车道都被低速车辆占据。在该状态下,算法生成了向左变道的指令,安全高效地完成了变道,并成功超越了3台低速车辆。场景二模拟了自车车道和左侧车道前方都有低速车辆的情况,并且自车与右后方的车辆距离较近,不具备变道条件。等右后方车辆距离变远后,算法立即选择向右变道,成功完成了变道操作并超越了3台低速车辆。



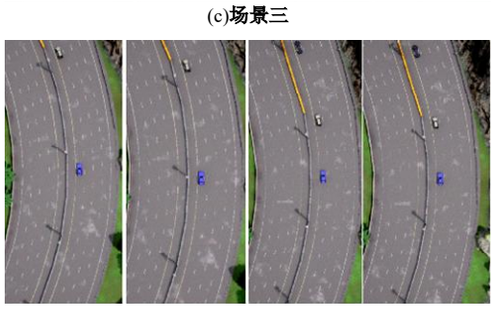
(a)场景一



(b)场景二



(c)场景三



(d)场景四

图7 典型工况测试结果

Fig. 7 Typical operating condition test results

场景三模拟了交通拥挤状态,在该状态下,算法一直选择车道保持,而且与前车保持一定的安全距离。场景四模拟了自车以80 km/h接近前方静止车辆,前方车辆进入检测范围后,自车立刻进行紧急制动并成功刹停。实验结果表明:本文算法在不同工况下都可以安全通过,并且有良好的高效性和安全性。

4 结束语

本文将分层强化学习思想应用于自动驾驶领域,使用两层的强化学习框架代替自动驾驶的决策层和控制层,降低了强化学习求解的复杂度,并为两层模型分别设计了状态空间、动作空间和奖励函数。为解决强化学习未能充分利用先验经验的问题,提出了一种在线专家和强化学习相结合的训练方法,应用在了控制层的训练中。最后使用Carla搭建了高速公路仿真环境验证了算法的性能。

结果表明:控制层可以出色完成轨迹跟踪和跟车任务,其收敛速度和跟踪误差也远远超过基线算法。结合决策层的整体算法可以安全高效地通过典型场景。未来将基于车载实际要求,完善轨迹点生成模块,并进行实车验证。

参考文献:

[1] 张羽翔. 基于知识与机理增强强化学习的智能车辆决策与控制[D]. 长春:吉林大学汽车工程学院, 2022.
Zhang Yu-xiang. Domain knowledge and physical-enhanced reinforcement learning for intelligent vehicles decision-making and control[D]. Changchun: School of Transportation, Jilin University, 2022.

[2] Schwarting W, Alonso M J, Rus D. Planning and decision-making for autonomous vehicles[J]. Annual Review of Control, Robotics, and Autonomous Systems, 2018, 1: 187-210.

[3] Paden B, Čáp M, Yong S Z, et al. A survey of motion planning and control techniques for self-driving urban vehicles[J]. IEEE Transactions on Intelligent Vehicles, 2016, 1(1): 33-55.

[4] 王景珂. 基于学习的自动驾驶行为决策研究[D]. 杭

州: 浙江大学控制科学与工程学院, 2021.
Wang Jing-ke. Research on autonomous driving behavior decison-making based on learning[D]. Hangzhou: College of Control Science and Engineering, Zhejiang University, 2021.

[5] Sutton R S, Barto A G. Reinforcement Learning: An Introduction[M]. Cambridge: MIT Press, 2018.

[6] Kendall A, Hawke J, Janz D, et al. Learning to drive in a day[C]//International Conference on Robotics and Automation, Montreal, Canada, 2019: 8248-8254.

[7] Wurman P R, Barrett S, Kawamoto K, et al. Outracing champion gran turismo drivers with deep reinforcement learning[J]. Nature, 2022, 602: 223-228.

[8] 李文礼, 邱凡珂, 廖达明, 等. 基于深度强化学习的高速公路换道跟踪控制模型[J]. 汽车安全与节能学报, 2022, 13(4): 750-759.
Li Wen-li, Qiu Fan-ke, Liao Da-ming, et al. Highway lane change decision control model based on deep reinforcement learning[J]. Journal of Automotive Safety and Energy, 2022, 13(4): 750-759.

[9] Kiran B R, Sobh I, Talpaert V, et al. Deep reinforcement learning for autonomous driving: a survey [J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(6): 4909-4926.

[10] Zhu M, Wang Y, Pu Z, et al. Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving[J]. Transportation Research Part C: Emerging Technologies, 2020, 117: 102662.

[11] Huang Z, Wu J, Lyu C. Efficient deep reinforcement learning with imitative expert priors for autonomous driving[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(10): 7391-7403.

[12] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518: 529-533.

[13] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//International Conference on Machine Learning, Stockholm, Sweden, 2018: 1861-1870.

[14] Liang X, Wang T, Yang L, et al. Cirl: controllable imitative reinforcement learning for vision-based self-driving[C]//Proceedings of the European Conference on Computer Vision, Munichi, Germany, 2018: 584-599.

[15] Hester T, Vecerik M, Pietquin O, et al. Deep q-learning from demonstrations[C]//Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, USA, 2018: 3223-3230.

[16] Elallid B B, Benamar N, Hafid A S, et al. A comprehensive survey on the application of deep and reinforcement learning approaches in autonomous driving [J]. Journal of King Saud University-Computer and Information Sciences, 2022, 34(9): 7366-7390.

[17] Kulkarni T D, Narasimhan K, Saeedi A, et al. Hierarchical deep reinforcement learning: integrating temporal abstraction and intrinsic motivation[J]. Advances in Neural Information Processing Systems, 2016, 29: 990-998.

[18] Duan J, Eben L S, Guan Y, et al. Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data[J]. IET Intelligent Transport Systems, 2020, 14(5): 297-305.

[19] Dosovitskiy A, Ros G, Codevilla F, et al. CARLA: an open urban driving simulator[C]//Conference on Robot Learning, Mountain View, USA, 2017: 1-16.