计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.201000084

基于动作约束深度强化学习的安全自动驾驶方法

代珊珊[1]　刘　全[1,2,3,4]
1 苏州大学计算机科学与技术学院　江苏 苏州 215006
2 苏州大学江苏省计算机信息处理技术重点实验室　江苏 苏州 215006
3 吉林大学符号计算与知识工程教育部重点实验室　长春 130012
4 软件新技术与产业化协同创新中心　南京 210000
(20185427004@stu.suda.edu.cn)

摘　要　随着人工智能的发展,自动驾驶领域的研究也日益壮大。深度强化学习(Deep Reinforcement Learning,DRL)方法是该领域的主要研究方法之一。其中,安全探索问题是该领域的一个研究热点。然而,大部分DRL算法为了提高样本的覆盖率并没有对探索方法进行安全限制,使无人车探索时会陷入某些危险状态,从而导致学习失败。针对该问题,提出了一种基于动作约束的软行动者-评论家算法(Constrained Soft Actor-critic,CSAC),该方法首先对环境奖赏进行了合理限制。无人车动作转角过大时会产生抖动,因此在奖赏函数中加入惩罚项,使无人车尽量避免陷入危险状态。另外,CSAC方法又对智能体的动作进行了约束。当目前状态选择动作后使无人车偏离轨道或者发生碰撞时,标记该动作为约束动作,在之后的训练中通过合理约束来更好地指导无人车选择新动作。为了体现CSAC方法的优势,将CSAC方法应用在自动驾驶车道保持任务中,并与SAC算法进行对比。结果表明,引入安全机制的CSAC方法可以有效避开不安全动作,提高自动驾驶过程中的稳定性,同时还加快了模型的训练速度。最后,将训练好的模型移植到带有树莓派的无人车上,进一步验证了模型的泛用性。

关键词:安全自动驾驶;深度强化学习;软行动者-评论家;车道保持;无人车
中图法分类号　TP181

Action Constrained Deep Reinforcement Learning Based Safe Automatic Driving Method

DAI Shan-shan[1] and LIU Quan[1,2,3,4]
1 School of Computer Science and Technology,Soochow University,Suzhou,Jiangsu 215006,China
2 Provincial Key Laboratory for Computer Information Processing Technology,Soochow University,Suzhou,Jiangsu 215006,China
3 Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education,Jilin University,Changchun 130012,China
4 Collaborative Innovation Center of Novel Software Technology and Industrialization,Nanjing 210000,China

Abstract　With the development of artificial intelligence,the field of autonomous driving is also growing. The deep reinforcement learning (DRL) method is one of the main research methods in this field. DRL algorithms have been reported to achieve excellent performance in many control tasks. However,the unconstrained exploration in the learning process of DRL usually restricts its application to automatic driving. For example,in common reinforcement learning (RL) algorithms,an agent often has to select an action to execute in each state although this action may result in a crash,deteriorating the performance,or even failing the task. To solve the problem,this paper proposes a new method of action constrained with the soft actor-critic algorithm (CSAC) where the 'NO-OP'(NO-Option) identifies and replaces inappropriate actions,and we test the algorithm in the lane-keeping tasks. The method firstly limits the environmental reward reasonably. When the rotation angle of the driverless car is too large,it will shake,then a penalty term will be added to the reward function to avoid the driverless car falling into a dangerous state as far as possible. The contributions of this paper are as follows:first,we incorporates action constrained function with SAC algorithm,which achieves faster learning speed and higher stability;second,we propose a reward setting framework that overcomes the shaking and instability of driverless cars,achieving a better performance;finally,we trains the model in the unity virtual environment for

到稿日期:2020-10-16　返修日期:2021-03-04　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61772355,61702055,61502323,61502329);江苏省高等学校自然科学研究重大项目(18KJA520011,17KJA520004);吉林大学符号计算与知识工程教育部重点实验室资助项目(93K172014K04,93K172017K18);苏州市应用基础研究计划工业部分(SYG201422);江苏高校优势学科建设工程资助项目
This work was supported by the National Natural Science Foundation of China(61772355,61702055,61502323,61502329),Jiangsu Province Natural Science Research University Major Projects(18KJA520011,17KJA520004),Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education,Jilin University(93K172014K04,93K172017K18),Suzhou Industrial Application of Basic Research Program Part(SYG201422) and A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.
通信作者:刘全(liuquan@suda.edu.cn)

evaluating the performance and successfully transplant the model to a donkey driverless car.

Keywords Safe automatic driving, Deep reinforcement learning, Soft actor-critic, Lane-keeping, Driverless cars

1 引言

自动驾驶(automatic driving)将在未来的智能交通系统中发挥重要作用,安全自动驾驶则是自动驾驶步入现实的第一准则[1]。目前,自动驾驶技术应用最广的是基于规则的方法。基于规则的自动驾驶框架包括感知、规划和控制 3 个过程[2]。大多数基于规则的自动驾驶技术都是通过车载传感器来检测周围的环境,并根据捕捉到的相关信息构造外部环境的模型,而没有理解当前的环境[3]。由于无人车对传感器以及基于规则算法的依赖性,它只能在有限领域和特殊行业得到发展,无法在人类日常生活中得到广泛应用[4-5]。而且,基于规则算法的自动驾驶模块之间的融合问题至今是一个亟需解决的难题。

与基于规则的自动驾驶技术相比,端到端的自动驾驶避免了以上问题,它主要由监督学习方法和 DRL 方法实现。其中,监督学习的自动驾驶思想是收集人类驾驶数据作为学习资料,训练端到端神经网络模型,使得模型可以由当前环境直接映射到动作[6-7]。由于模型训练数据量过大,面对陌生环境时,模型的鲁棒性较差,需要重新采集数据来训练模型。因此,基于监督学习的自动驾驶方法并不适合复杂多变的现实世界[8-9]。但随着 DRL 技术的不断发展,基于 DRL 的自动驾驶方法(见图 1)摆脱了基于监督学习方法的局限性,更贴近于现实人类的驾驶决策方式,并成为目前的研究热点。该方法主要通过智能体(agent)直接与环境进行交互,以获得最大化累积回报为导向,从环境中寻求最优策略 π^*。它是一种基于理解环境并根据当前环境做出决策的方法,该方法更接近于人类的思维方式[10]。

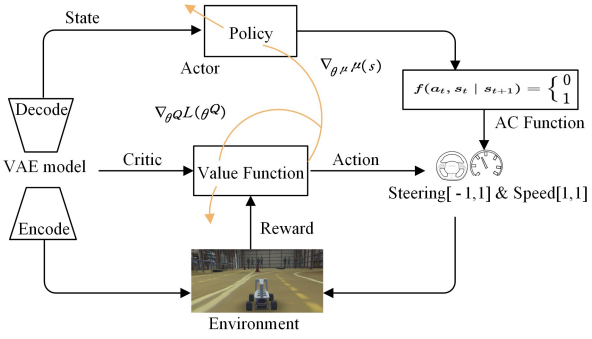


图 1 智能体与环境的交互框架

Fig. 1 Framework of agent and environments interaction

Maximilian 等[11]利用 A3C 算法学习如何在 TORCS 虚拟平台上驾驶赛车。该方法实现了无人车在不同曲线、坡道以及不同季节变换的道路上驾驶。Kendall 等[12]利用 DDPG 算法在驴车(donkey car)模拟器上进行学习,并在现实生活中使用电动无人车完成车道保持任务。Toromanoff 等[13]有效地利用了 RL 算法进行城市内的自动驾驶模拟训练,此方法实现了 RL 智能体在城市环境的驾驶,其框架如图 2 所示。Chen 等[14]提出了应用少量人工数据来提高策略模型训练的方法。以上基于 DRL 的自动驾驶方法实现了端到端的自动驾驶控制决策任务,但是存在学习速度慢、不稳定的缺点。

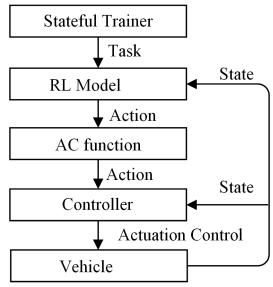


图 2 基于强化学习的自动驾驶框架

Fig. 2 Autonomous driving framework based on DRL

DRL 算法利用深度神经网络高容量函数的逼近能力和强化学习算法的决策能力,实现了自动化决策和控制任务[15]。SAC(Soft Actor-critic)[16-17]是一种基于最大熵强化学习框架的行动者-评论家算法。在这个框架中,其目标是期望得到最大化回报,同时最大化期望熵。该算法在 MuJoCo 模拟器的一系列连续控制任务中实现了最先进的性能,并且能更高效地利用样本,从而获得较高的参数鲁棒性,并超越了之前算法的成绩。近几年,DRL 算法已经应用于一系列具有挑战性的领域,研究者们开始关注 DRL 在自动驾驶领域的应用怎样从理论研究迈向现实生活。然而,实现理论研究到现实应用的同时,保证 Agent 决策的安全性是必要的。自动驾驶研究过程中发生的一系列事故,更凸显出了安全问题在控制决策中的重要性[18]。为了解决这一问题,有些专家提出了安全强化学习的思想。安全强化学习(Safe Reinforcement Learning,SRL)指在满足一定安全约束的条件下最大化累积回报。目前,针对强化学习 Agent 探索过程中存在的不安全问题,Garcia 等[19]认为安全强化学习方法可以分为两类。其中,一类是修改目标函数的方法;另一类是约束 Agent 探索空间的方法。对于安全性要求较高的系统,通常会选择第二类方法,通过改进探索过程中的外部因素进行安全约束。Garcia 等[20]针对 Agent 的"试错式"无约束探索问题,引入了基于策略改进的 PI-SRL(Policy Improvement Through Safe Reinforcement Learning)方法,改进了连续状态空间和动作空间的安全问题。Berkenkarmp 等[21]针对 Agent 无约束探索导致的不稳定性,将控制理论李亚普诺夫(Lyapunov)方法引入 RL 控制策略上,实现了高性能的控制策略,并有效提高了探索的稳定性。

本文针对基于 DRL 算法的无约束探索带来的安全性问题,引入了状态动作许可(state action permissibility)属性[22],并提出了一种通用、直接的动作约束 SRL 方法。状态动作许可表示选择当前状态 s_t 下的动作 a_t,到达新状态 s_{t+1},并判断该动作是否安全,如果安全则允许执行,否则将延迟该动作的执行。CSAC 算法是动作约束函数与 SAC 算法的结合,可将其用于自动驾驶车道保持任务。本文的主要贡献如下:

(1)针对 DRL 算法的无约束探索导致的不安全问题,提出了动作约束安全强化学习方法,此方法在 Agent 执行决策的不安全问题上进行了有效的约束。

(2)针对无人车驾驶过程中抖动和不稳定性问题，提出了全新的自动驾驶奖赏框架，当动作转角大于设定参数时，将给予惩罚。

(3)本文将 CSAC 算法与 SAC 算法在自动驾驶车道保持任务上进行对比实验，CSAC 算法的训练速度更快，而且实现了安全可靠性驾驶。最后，将 CSAC 模型移植到带有树莓派的无人车上测试，实验结果证明了模型的泛用性。

2 相关工作

2.1 强化学习

强化学习是机器学习的一个重要分支，最初是受到动物学习心理、优化算法、概率论以及控制理论等启发而得以发展，在人工智能领域也越来越受关注。在强化学习任务中，Agent 与环境交互，期望得到最大的累积回报。强化学习应用马尔可夫决策过程(Markov Decision Process，MDP)框架来定义 Agent 与环境之间的交互过程[10]。MDP 是强化学习在数学问题上的理论框架，也是经典的序贯决策学习框架。Agent 在随机环境中进行探索，其采取的动作同时影响着当前的立即奖赏以及后续的状态和动作。MDP 可以表示为标准的五元组($\mathcal{S},\mathcal{A},P,R,\gamma$)，五元组中的元素代表的意义如下：

(1)状态空间集合 $\mathcal{S}$：$s_t\in\mathcal{S}$；

(2)动作集合 $\mathcal{A}$：$a_t\in\mathcal{A}$；

(3)状态转移概率 $P:\mathcal{S}\times\mathcal{S}\times\mathcal{A}\to[0,1]$；

(4)奖赏函数 $R:\mathcal{S}\times\mathcal{S}\times\mathcal{A}\to\mathbb{R}$；

(5)折扣因子 γ：$0\leqslant\gamma\leqslant1$。

在强化学习任务中，时刻 t 开始到时刻 $\mathcal{T}$ 结束的累积奖赏函数定义为：

$$G_t=\sum_{l=t}^{T}\gamma^{l-t}r_l \tag{1}$$

其中，折扣因子 $0\leqslant\gamma\leqslant1$，目的是权衡未来获得的奖赏对目前的累积奖赏的影响力度。

$Q^\pi(s_t,a_t)$是状态动作值函数，指在当前的状态 s_t 下执行动作 a_t，Agent 所获得的累积奖赏表示为：

$$Q^\pi(s_t,a_t)=E_\pi[R_t\mid s_t,a_t,\pi] \tag{2}$$

$Q^\pi(s_t,a_t)$函数遵循贝尔曼方程，利用具有递归性质的贝尔曼方程进行迭代计算，直到 $Q^\pi(s_t,a_t)$最终收敛，从而求得最优策略。

$$Q^*(s_t,a_t)=\max_\pi\mathbb{E}[R_t\mid s_t,a_t,\pi] \tag{3}$$

对于离散动作空间内的强化学习任务，求解状态动作值 $Q^\pi(s_t,a_t)$函数是很好的方法[10,14-15]，然而，对于连续动作空间强化学习任务，策略梯度方法则是最佳选择[23-25]。

2.2 最大熵强化学习

最大熵强化学习优化策略的准则是最大化期望回报，同时最大化期望熵。标准强化学习的目标是获得最大的期望回报 $R=\sum\mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[r(s,a)]$，最终得到最优策略 $\pi^*(a_t\mid s_t)$。而熵是策略中随机性的一种度量，设 x 为随机变量，概率密度函数为 P，熵 H 的计算式如下[26]：

$$H(P)=\underset{x\sim P}{E}[-\log P(x)] \tag{4}$$

在最大熵强化学习中，带有最大熵的目标函数通过增加一个熵项，使得最优策略的熵在每次迭代下达到最大。

$$\pi^*=\arg\max_\pi\sum_t\mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[R(s_t,a_t)+\alpha H(\pi(a_t\mid s_t))] \tag{5}$$

其中，α 是温度参数，它决定了熵的重要性，并控制了最优策略的随机性[16]。当 $\alpha=0$ 时，最大熵强化学习与传统的 RL 算法相同，如果 $\alpha>0$，则最大熵 RL 鼓励探索。在实践应用中，最大熵的优点包括：1)它鼓励更广泛的探索，同时放弃多余的样本；2)该策略可以得到多个近似最优动作；3)该方法可以有效地提高训练速度，是优化传统 RL 目标函数的一种新方法。

由最大熵目标函数可以定义一个新的值函数 $V^\pi(s_t)$，$V^\pi(s_t)$中添加了一个对熵的奖赏：

$$V^\pi(s_t)=\underset{\tau\sim\pi}{\mathrm{E}}[\sum_{t=0}^{\infty}\gamma^t(R(s_t,a_t,s_{t+1})+\alpha H(\pi(a_t\mid s_t)))] \tag{6}$$

类似地，带有熵的 Q 函数 $Q^\pi(s_t,a_t)$添加一个对熵的奖赏：

$$Q^\pi(s_t,a_t)=\underset{\tau\sim\pi}{\mathrm{E}}[\sum_{t=0}^{\infty}\gamma^tR(s_t,a_t,s_{t+1})+\alpha\sum_{t=1}^{\infty}\gamma^tH(\pi(a_t\mid s_t))] \tag{7}$$

因此，$V^\pi(s_t)$和 $Q^\pi(s_t,a_t)$的关系可以写为：

$$V^\pi(s_t)=\underset{a\sim\pi}{E}[Q^\pi(s_t,a_t)]+\alpha H(\pi(\cdot\mid s_t)) \tag{8}$$

对于 Bellman 方程，Q^π可以定义为：

$$Q^\pi(s_t,a_t)=\underset{s\sim P}{\mathrm{E}}[R(s_t,a_t,s_{t+1})+\gamma V^\pi(s_{t+1})+\alpha H(\pi(\cdot\mid s_{t+1}))] \tag{9}$$

最大熵强化学习已经被成功地应用在了许多算法中，包括 Levine 等[27]利用最大熵强化学习引导策略探索，实现了感知和控制系统任务。O'Donoghue 等[28-29]应用最大熵强化学习正则化策略梯度方法，在离散空间任务中取得了优异的成绩。Haarnoja 等[30]利用最大熵强化学习方法，将行动者网络(Actor)作为近似取样器，其框架如图 3 所示。本文中的 SAC 方法利用最大熵强化学习算法改进探索，易于参数的微调，是目前最优的离策略深度强化学习方法。

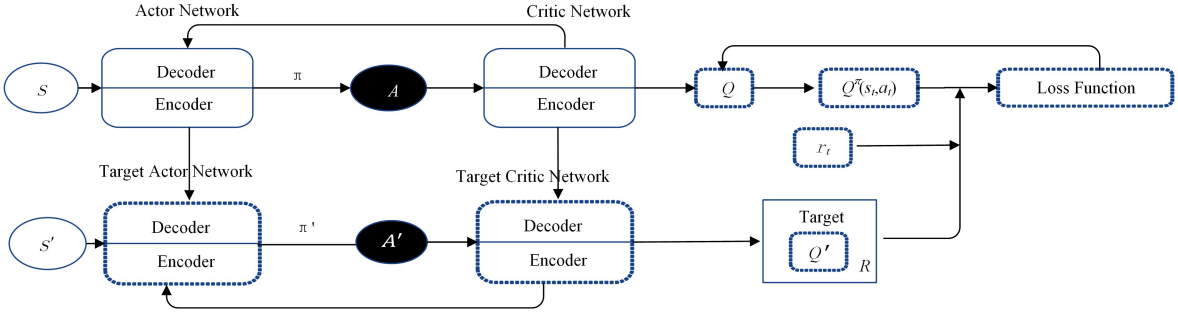


图 3 Actor-critic 框架

Fig. 3 Architecture of Actor-critic

3 基于动作约束的软行动者评论家算法

对于安全性要求较严格的自动驾驶领域，如何保证驾驶策略的安全性是关键。本节介绍安全性约束的软行动者评论家算法，首先介绍动作约束方法[25]，然后介绍 CSAC 算法框架。

3.1 动作约束方法

为保证自动驾驶决策的安全性，提出了动作约束的方法，该方法启发于人脑在面临决策时的网络模型，其在每一步的动作执行之前都要进行判断，如果当前动作会使无人车陷入灾难，则这个动作将被禁止执行。

3.1.1 大脑决策

人类大脑决策执行原理中，基底神经元丘脑(Cortico-Basal-Ganglia-Thalamic，CBGT)网络是根据人类大脑皮层神经元之间的反射原理构造的，它定义了大脑的决策模型[31-32]。在生物学神经网络中，决策的形式可以分为直接决策和间接决策。

对于 RL 问题，Agent 通常需要长时间的训练和优化，在得到最优策略后可以直接进行决策。但对于不确定问题，通常采取随机选择动作执行的方法。而对于哺乳动物大脑 CBGT 通路，当决策不确定时，通常选择延迟决策，选择等待并观察的方式[33]。在现实中，自动驾驶任务利用这一原理来限制不正确的动作，并延迟决策的执行。

3.1.2 动作约束函数

自动驾驶的动作空间是一个二维连续空间，包括转向 $S\in[-1,1]$和加速度 $A\in[-1,1]$。动作约束方法仅考虑横向的转向空间，即选择性约束转角动作 $S\in[-1,1]$[22]。标准的强化学习 Agent 包括探索和利用两个过程，探索是帮助 Agent 更了解状态空间，而利用则是更好地保证 Agent 获得更高的奖赏。该方法对无人车的探索和利用均进行了限制。

定义 1(动作约束性) 如果 Agent 从状态 s_t 到状态 s_{t+1} 的动作 a_t 导致车辆偏离道路或发生碰撞，则该动作在状态 s_t 中将被约束，将约束动作存入集合 X_t。

定义 2(动作允许性) 如果 Agent 从状态 s_t 到状态 s_{t+1} 的动作 a_t 不在状态为 s_t 的约束集合 X_t 中，则认为它是允许的。

定义 3(动作约束函数) 如果一个动作被约束，则 $f(a_t,s_t|s_{t+1})=0$，此时执行“NO-OP”命令，即当前状态的动作不执行任何操作，并等待下一步的操作。

当无人车发生碰撞时，情节将被终止，并设置 $\delta_{\text{track}}=1$。其中，δ_{track}表示是否偏离路线或者发生碰撞，$\delta_{\text{track}}=1$ 表示碰撞为真，反之 $\delta_{\text{track}}=0$。当动作 a_t 在状态 s_t 下向左偏离道路时，则 s_t 状态的动作约束区间 $X_t=[-1,a_t]$，随着训练的进行，如果动作 a_{t1} 在状态 s_t 下向左偏离，且 $a_{t1}>a_t$，则更新为 $X_t=[-1,a_{t1}]$。如向右偏离，则约束动作区间设置为 $X_t=[a_t,1]$。例如，图 4 所示的车道偏离，在状态 s_1 下执行动作 a_1，导致下一状态 s_2 偏离车道，则状态 s_1 下的动作选择 a_1 将被约束。当 $\delta_{\text{track}}=1$ 时，在状态 s_1 下执行动作 a_1 将会发生碰撞。

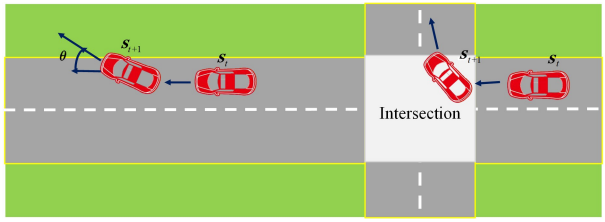


图 4 车道偏离

Fig. 4 Lane departure

动作约束函数如下：

$$f(a_t,s_t|s_{t+1})=\begin{cases}0, & \text{if } a_t\in X_t \text{ or } \delta_{\text{track}}=1\\ 1, & \text{otherwise}\end{cases} \tag{10}$$

当 $f(a_t,s_t|s_{t+1})=0$ 时，说明当前的动作 a_t 属于被约束的动作，执行动作为 False。当 $f(a_t,s_t|s_{t+1})=1$ 时，执行动作为 True。因此，对于每个给定的状态 s_t，约束动作的属性是局部的。这种方法可以有效地避开自动驾驶过程中的危险动作，起到有效的安全保护作用。约束函数 f 是一个条件函数，其允许执行符合状态 s_t 的 a_t，而限制不符合状态 s_t 的 a_t。

3.2 奖赏设置

RL 中奖赏的设置是获得最优策略的关键因素之一。本文将奖赏定义为行驶的距离，即 $v\times t$。由于无人车在驾驶过程中选择动作的转角过大会导致车辆抖动问题，因此存在很大的安全隐患。为了解决上述问题，定义了表示舒适度的参数 ∂。如果 $a_{t+1}-a_t>\partial$，则进行奖赏的惩罚。

奖赏函数如下：

$$R=\begin{cases}v\times t-\dfrac{1}{2}\left(\dfrac{1}{1+e^{\cos\theta}}\right), & a_{t+1}-a_i\geqslant\partial,\delta_{\text{track}}\neq1\\ v\times t, & a_{t+1}-a_t<\partial,\delta_{\text{track}}\neq1,\\ -10, & \delta_{\text{track}}=1\end{cases} \tag{11}$$

其中，v 是无人车的速度，t 是时间步长。当 $a_{t+1}-a_t>\partial$时，将利用 sigmoid 函数对抖动进行惩罚。实验对比了 sigmoid[34] 函数和 probit[35] 函数的效果，最终我们选择 sigmoid 函数，它可以有效地将惩罚的值映射在区间(0,1)，在特征值差别较小时，应用效果较好，因此奖赏函数巧妙地对转角过大时带来的抖动现象进行惩罚。图 4 中，θ 是汽车中心线 D 与道路中心线 d 形成的角度，$\cos\theta$ 参数表示方向盘角度的余弦值。

SAC 使用函数逼近的方法同时学习策略 $\pi_\theta(a_t|s_t)$和两个 Q 函数 $Q_{\phi_1}(s_t,a_t)$，$Q_{\phi_2}(s_t,a_t)$，这些网络的参数分别是 θ，ϕ_1 和 ϕ_2。接下来介绍更新规则。

(1)学习 Q 函数。SAC 中 Q 函数的学习类似于双延迟 DDPG(Twin Delayed Deep Deterministic Policy Gradient Algorithm，TD3)[25]，同时学习两个 Q 函数 $Q_{\phi_1}(s_t,a_t)$，$Q_{\phi_2}(s_t,a_t)$，两个 Q 函数使用同一个目标，计算两个 Q 函数得出较小的一个目标值：

$$y(r,s_{t+1},d)=r+\gamma(1-d)\min_{i=1,2}Q_{\phi_{i,\text{targ}}}(s_{t+1},a_{t+1}(s_{t+1})) \tag{12}$$

通过迭代目标函数学习两个 Q 函数，然后逐步计算，最终使用较小的 Q 值作为目标值。此方法有助于避免 Q 函数的过高估计问题。

$$L(\phi_1,\mathscr{D})=\mathop{\mathrm{E}}_{(s_t,a_t,r,s_{t+1},d)\sim\mathscr{D}}[(Q_{\phi_1}(s_t,a)-y(r,s_{t+1},d))^2]$$

$$L(\phi_2,\mathscr{D})=\mathop{\mathrm{E}}_{(s,a,r,s',d)\sim\mathscr{D}}[(Q_{\phi_2}(s_t,a)-y(r,s_{t+1},d))^2] \tag{13}$$

从递归的 Bellman 方程开始，对 $Q^\pi(s_t,a_t)$ 进行熵正则化校正，然后通过使用熵的定义可以写为[36]：

$$\begin{aligned}Q^\pi(s_t,a_t)&=\mathop{\mathrm{E}}_{\substack{s_{t+1}\sim P\\a_{t+1}\sim\pi}}[R(s_t,a_t,s_{t+1})+\gamma(Q^\pi(s_{t+1},a_{t+1})+\alpha H(\pi(\cdot\,|s_{t+1})))]\\&=\mathop{\mathrm{E}}_{\substack{s_{t+1}\sim P\\a_{t+1}\sim\pi}}[R(s,a,s_{t+1})+\gamma(Q^\pi(s_{t+1},a_{t+1})-\alpha\log\pi(a_{t+1}|s_{t+1}))]\end{aligned} \tag{14}$$

以上可以得到在 SAC 中 Q 网络的损失函数如下：

$$L(\phi_i,\mathscr{D})=\mathbb{E}_{(s_t,a_t)}[\frac{1}{2}(Q_{\phi_i}(s,a)-(r+\gamma\,\mathbb{E}_{s_{t+1}\sim p}[V_{\psi_{\mathrm{targ}}(s_{t+1})}]))^2] \tag{15}$$

(2)学习策略 π。策略 $\pi(a|s)$ 表示状态 s 映射到每个可能的动作 a 的概率。策略在每个状态下发挥作用，在最大化期望回报的同时最大化期望熵[16]，即它应该最大化 $V^\pi(s_t)$，将其扩展为：

$$\begin{aligned}V^\pi(s_t)&=\mathop{\mathrm{E}}_{a_t\sim\pi}[Q^\pi(s_t,a_t)]+\alpha H(\pi(\cdot\,|s_t))\\&=\mathop{\mathrm{E}}_{a_t\sim\pi}[Q^\pi(s_t,a_t)-\alpha\log\pi(a_t|s_t)]\end{aligned} \tag{16}$$

策略的优化如下：

$$\max_\theta\mathop{\mathrm{E}}_{\xi\sim\mathscr{D},\xi\sim\mathscr{N}}[Q_{\phi_1}(s,\tilde{a}_q(s,x))-a\log p_q(\tilde{a}_q(s,x)|s)]$$

$$\tilde{a}_\theta(s,\xi)=\tanh(\mu_\theta(s)+\sigma_\theta(s)\odot\xi),\xi\sim\mathscr{N}(0,I) \tag{17}$$

其中，$\mathscr{D}$ 是经验回放缓冲池，ξ 是基于高斯分布 $\mathscr{N}$ 的噪声。SAC 克服了两个主要困难，一是高样本复杂度，二是对超参数的适用性。该算法更适用于自动驾驶任务的学习。

基于动作约束的 SAC 算法的流程如算法 1 所示。

算法 1 CSAC

输入：状态

输出：最优策略

1. 初始化：策略网络参数 θ；Q 值函数和目标网络参数 ϕ_1，ϕ_2；Q 目标网络参数 $\phi_{\mathrm{targ},1}\leftarrow\phi_1$，$\phi_{\mathrm{targ},2}\leftarrow\phi_2$；V 函数和目标网络参数 ψ；V 目标网络参数 $\psi_{\mathrm{targ}}\leftarrow\psi$；初始化经验回放缓冲池 $\mathscr{D}$；设置超参数 δ_{limit}，$f(a_t,s_t,|s_{t+1})=1$
2. for t=1 to N do：
3. if $f(a_t,s_t,|s_{t+1})=1$：
4. 执行动作
5. else：continue
6. 观察 s′，r 和 done
7. if done：reset，$\delta_{\mathrm{track}}=1$
8. if $\delta_{\mathrm{track}}=1$ 或者 $a_t-a_{t+1}>\delta_{\mathrm{limit}}$：
9. $f(a_t,s_t|s_{t+1})=0$，store
10. $(s_t,a_t,r,s_{t+1},\mathrm{done},f(a_t,s_t|s_{t+1}))$ in $\mathscr{D}$
11. if it is time to update then：
12. for every step do：
13. 随机从 C(a batch sample)采样
14. 计算目标 Q 值函数
15. $K=\min_{i=1,2}Q_{\phi_{\mathrm{targ},i}}(s_t,\tilde{a}')-\alpha\log\pi_\theta(\tilde{a}'|s_t)\,y(r,s_t,d)=r+\gamma(1-d)K$
16. 每步梯度下降更新 Q 值函数

$$M=Q_{\phi_i}(s_t,a_t)-y(r,s_{t+1},d)$$

$$\nabla_{\phi_i}\frac{1}{|C|}\sum_{(s,a,r,s',d)\in C}(M)^2\ \text{for i=1,2}$$

17. 每步梯度下降策略

$$N=\min_{i=1,2}Q_{\phi_i}(s,\tilde{a}_\theta(s))-\alpha\log\pi_\theta(\tilde{a}_\theta(s)|s)\nabla_\theta\frac{1}{|B|}\sum_{s\in B}(N)$$

18. 更新目标值网络

$$\psi_{\mathrm{targ}}\leftarrow\rho\psi_{\mathrm{targ}}+(1-\rho)\psi$$

19. End for
20. End

CSAC 算法利用两个 Q 函数来减少策略改进步骤的正偏差，正偏差可以降低值函数的性能。两个 Q 函数独立训练，取其中较小的以优化目标函数。动作约束的集合将被写入 done 信号中，并存入经验回放池。CSAC 方法不仅可以学习具有挑战性的任务，而且可以保证机器学习过程中的安全探索。由于 SAC 是无模型(model-free)DRL 算法，因此约束函数 f 的条件建立需要从零开始训练。通过不断的试错，寻找安全驾驶的边界条件，并根据边界条件指导动作的选择。在经验回放池的样本中，边界条件随着训练的不断完善，同时对违规动作的执行进行了约束，加快了训练的速度。

4 实验

实验的主要目的是使无人车保持在车道内驾驶，以及在各种曲折、交错的道路上更加平稳地驾驶。实验所使用的无人车安装有一个前置高清 1080P 摄像头，未安装导航设备，无人车从起点到终点不断尝试寻找最优策略。实验中，CSAC 算法与 SAC 算法均存在 Actor-critic 框架，为了对比实验的公平，采用相同的参数设置。虚拟环境应用驴车模拟器的两个主要环境为："Generated Road"和"Generated Track"。

4.1 车道保持

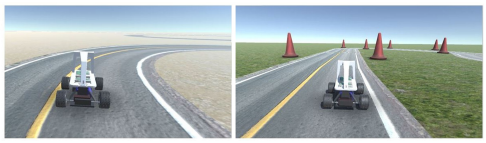
车道保持(lane keeping)指无人车必须严格按照规定保持在车道内行驶。基于规则的自动驾驶中，车道保持辅助系统的主要作用是帮助驾驶员保持在某个车道内行驶，提升了行驶安全性和便利性。本文中的车道保持是 DRL 算法通过与环境交互，来不断自主学习如何在车道内保持规范驾驶[37-38]。

Agent 从零开始学习，并最终学会在难易不同的路线上保持车道内驾驶。图 4 给出了在双车道和交叉口处，车辆从状态 s_t 到下一个状态 s_{t+1} 的动作选择错误，导致偏离道路的现象。如果 Agent 偏离道路，此情节将被终止，并给予负奖赏。

4.2 实验参数设置

驴车模拟器是基于游戏平台的自动驾驶仿真器，它提供了物理引擎、图形化驱动环境以及难度不同的仿真环境。如图 5 所示，在模拟器中无人车仅使用高清的前置摄像头来理解当前的驾驶环境，相机的原始帧的大小设置为 160×210 的

RGB图像。实验前，首先采集驴车模拟器的两个环境的道路图片，然后将采集的数据用于VAE模型训练。VAE模型[39-40]由4层卷积神经网络构成，它可以对图像进行特征提取和去除噪声，同时还提高了图像数据的训练效率。



(a)"Generated Road" (b)"Generated Track"

图5 模拟实验环境

Fig. 5 Simulation experiments environments

CSAC方法的参数设置：首先将动作约束集设置为空，即$X_t=[\ ]$，以及$\delta_{\text{track}}=0$。随着训练数据在经验缓冲池的更新，数据集被进一步优化。Actor的学习率设为0.000 01，Critic网络的学习率为0.000 1。实验中，SAC算法涉及的其他重要参数设置如表1所列。

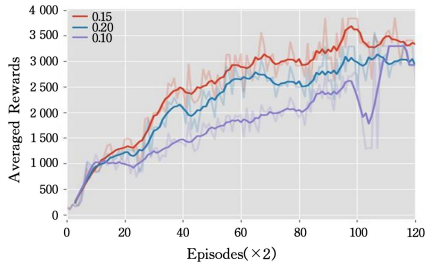
表1 实验环境参数

Table 1 Experimental environment hyperparameters

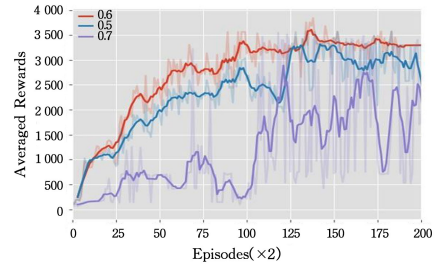
Parameter	SAC-Value
optimizer	Adam
discount γ	0.9
learning rate	0.000 3
replay buffer size	300 000
train frequency	3 000
gradient steps	100
learning starts	1 000
number of hidden units per layer	256
number of samples per minibatch	64
nonlinearity	ReLU
target update interval	10

4.3 模拟实验

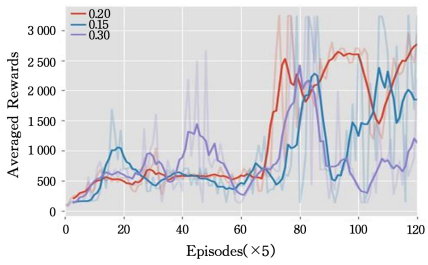
为了验证动作约束函数的有效性，在驴车模拟器中进行训练和测试。实验中，由于方向盘角度变化过大，对无人车的稳定性影响很大。设∂为连续两个动作转角的差值，控制$|a_{t+1}-a_t|\leqslant\partial$。如图6所示，实验中比较了参数$\partial$的效果，0.15是"Generated Road"环境的较好的选择，0.2是"Generated Track"较好的选择。



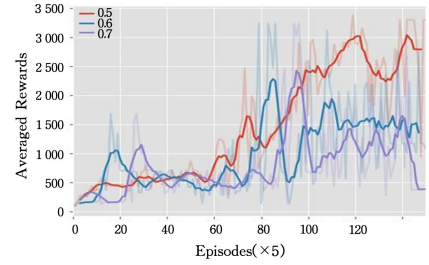
(a)Generated Road环境∂值实验对比图



(b)Generated Road环境 MAX_THROTTLE对比图



(c)Generated Track环境∂值实验对比图



(d)Generated Track环境 MAX_THROTTLE对比图

图6 自动驾驶虚拟环境下动作转角大小控制参数和最大油门参数的对比实验

Fig. 6 Comparison experiments of angle control and maximum throttle parameters in automatic drive simulator environment

实验中，动作约束考虑了横向的转角变化，针对纵向的加速度，设置"Generated Road"环境$MAX_THROTTLE=0.6$，"Generated Track"环境$MIN_THROTTLE=-0.2$，$MAX\_THROTTLE=0.5$。Generated Track环境设置的动作转角参数为$\partial=0.2$时可以得到更高的回报。

4.3.1 Generated Road

"Generated Road"虚拟环境可以随机生成不同难易程度的道路，如图7所示，其中包括急转弯和交叉路口。无人车的任务不仅需要保持在车道内平稳驾驶，而且需要在没有导航的情况下穿过交叉路口。实验过程中，无人车行驶的距离越远，速度越快，抖动越少，得到的奖赏就越高。如果无人车在交叉路口选择了错误的车道或驶离车道，则情节将被终止。

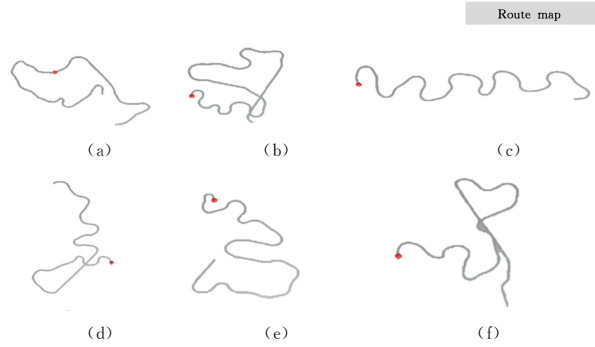


图7 "Generated Road"环境下的训练路线

Fig. 7 Training roads in Generated Road

为了保证模型的鲁棒性，实验采用了6种不同的训练路线。无人车在一条路径上训练成功后，会切换至另一条道路

继续训练。实验中，所有的模型进行 100 万个时间步训练，最终以记录的情节(Episode)数和每个情节的平均奖赏作为评估的指标。CSAC 和 SAC 算法实验中的每个情节得到的平均奖赏越高，效果就越好。图 8 给出了“Generated Road”环境下对图 7 中的道路分别进行训练的结果，其中横坐标是情节数，纵坐标是每个情节所得到的平均奖赏。由图 8 可以看出，CSAC 算法更快到达终点，训练速度也明显比 SAC 算法快，所获得的奖赏较 SAC 算法有明显提高。动作约束函数的应用能够快速地确认安全驾驶动作的边界，这样节省了 DRL 算法在不断“试错式”学习的时间。例如，当动作执行转角值 $a_t=0.6$，发生无人车向右偏离车道，则边界区间 $X_t=[0.6,1]$之内的动作值将不会再执行，从而节省了训练的时间，也使得 Agent 更快找到最优策略。当遇到急转弯或交叉路口时，动作约束的方法能够帮助 Agent 更快地通过，SAC 算法则需要更长的时间试错。实验中，测试道路不再局限于图 6 所示的路径，无人车从起点开始到情节结束，同时记录驾驶过程中的平均奖赏。表 2 中记录任务中的总路线数 N，无人车可以由起点成功驾驶到终点的路线数记为 Sc，Sc/N 的百分比作为任务衡量标准。

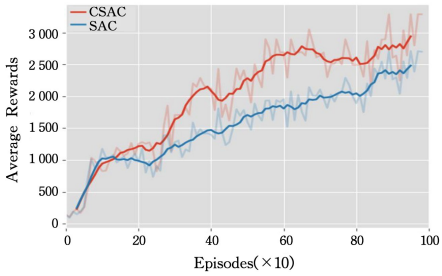


图 8 “Generated Road”环境的训练表现

Fig. 8 Generated Road training performance

表 2 “Generated Road”模拟环境中的测试

Table 2 Comparison of testing in Generated Road

Algorithm	Success rate/%	Average Reward	Max Reward
SAC	76.9	2908	3169
CSAC	81.4	3061	3180

表 2 列出了 CSAC 算法和传统的 SAC 算法在“Generated Road”模拟环境中的性能。测试 100 万步，实验结果表明 CSAC 算法的性能优于传统的 SAC 算法。由于自动驾驶任务较为复杂，使得应用了安全机制的动作约束方法比原始算法的优势明显，其中 CSAC 模型完成任务的成功率可以达到 81.4%。

4.3.2 Generated Track

“Generated Track”虚拟环境图 5(b)是路边设有路障，有两个急转弯的环路。对于此环境，连续的急转弯是无人车完成任务的难点。每当无人车碰到路障或者偏离道路，将从新开始下一个情节，避免无人车陷入更加困难的场景。

实验中，模型进行 100 万步的训练，并记录实验过程的每个情节和平均奖赏。图 9 给出了“Generated Track”环境下的训练结果，其中横坐标是情节数，纵坐标是每个情节所得到的平均奖赏。图 8 给出了 CSAC 算法和传统 SAC 算法的训练效果，可以看出基于动作约束的 CSAC 算法的效果更突出，主要体现在面对急转弯的学习能力方面，加入安全约束的方法有更好的适用性。

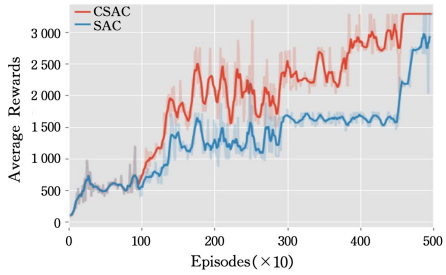


图 9 “Generated Track”环境的训练表现

Fig. 9 Generated Track training performance

表 3 列出了 CSAC 算法和 SAC 算法在“Generated Track”模拟环境中的性能。测试 100 万步，实验结果表明，CSAC 算法的测试成功率为 79.1%，比 SAC 算法高 2.8%，且 CSAC 算法的平均值与最大奖赏都高于 SAC 算法。由于“Generated Track”模拟环境存在路障的干扰，以及较大角度的急转弯，使得 SAC 算法的训练时间较长，而动作约束函数有效地限制了急转弯的转角，使得 CSAC 算法的效果较好，驾驶稳定性更好。

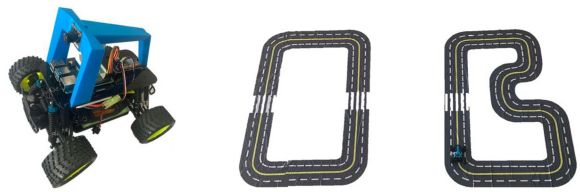
表 3 “Generated Track”模拟环境的测试

Table 3 Comparison of testing in Generated Track

Algorithm	Success rate/%	Average Reward	Max Reward
SAC	76.3	2012	3340
CSAC	79.1	2407	3362

4.4 现实中的无人车实验

为了进一步验证 CSAC 自动驾驶模型的泛用性，我们自制了一辆无人车，车上装有树莓派，配置包括 Pi-4B、4G 主板、前置高清摄像头。实验中，将训练好的“Generated Track”环境模型移植到无人车的树莓派上，并使用图 10 所示的两条 3 m×2 m 的道路进行测试。首先应用无人车实时采集这两条路径的图像数据并进行 VAE 模型训练。然后，将训练后的模型直接导入无人车。无人车的参数设置为：左右转角范围设置为[−0.7，0.7]，最大油门设置为 0.4，其他模型的参数与模拟器环境设置一致。



(a)Donkey-car (b)Testing road1 (c)Testing road2

图 10 无人车测试环境示意图

Fig. 10 Autonomous driving testing environment diagrams

在这个实验中，当无人车偏离车道进入无法恢复的位置时，将人为终止这个情节。然后，无人车被送到起始车道的中心，开始下一情节。由实验结果可知，表 4 中无人车在图 10 路线 1 驾驶，总长度为 10 m，平均需要 12 s 来完成驾驶任务。路线 2 长度为 10 m，平均需要 28 s 来完成驾驶任务，结果表明，转弯角度较大时，无人车需要更长的时间尝试通过。无人

车模型的成功移植凸显了 CSAC 算法的泛化性和通用性。

表 4 无人车测试

Table 4 Test in Donkey car

Road	Success time/s	Speed/(m/s)
1	12	1.2
2	28	2.8

由于无人车和模拟器的驾驶场景、地面摩擦力、车辆性能、视觉复杂度等因素存在差异，因此现实中无人车的测试效果比模拟器的测试效果更差。然而，预先训练 VAE 模型对成功移植有很大的帮助。

结束语 在模拟器环境之外，基于 DRL 的自动驾驶方法应用于现实世界的局限性很大，主要原因是学习过程中缺乏安全保障。因此，安全性问题是自动驾驶领域值得关注的问题。本文针对 DRL 算法的无约束探索问题，提出了带有动作约束的 CSAC 安全自动驾驶方法，并针对车辆的抖动问题提出了新的奖赏设置框架。在驴车模拟器仿真环境的车道保持任务中，CSAC 算法在训练速度和所得奖赏方面都高于传统的 SAC 算法。

测试实验中，CSAC 算法的成功率远高于 SAC 算法，充分体现了带有动作约束的 CSAC 算法的稳定性和学习能力都超越了传统的 SAC 算法。在未来的工作中，我们将不断地完善自动驾驶的方法与模型，将本文方法扩展到现实应用场景中。此外，我们还将继续在自动驾驶方向上进行其他的相关研究与探索。

参 考 文 献

[1] ORT T, PAULL L, RUS D. Autonomous vehicle navigation in rural environments without detailed prior maps[C]// 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 2040-2047.

[2] PENDLETON S D, ANDERSEN H, DU X X, et al. Perception, Planning, Control, and Coordination for Autonomous Vehicles [J]. Machines, 2017, 5(1): 6.

[3] CAPORALE D, SETTIMI A, MASSA F, et al. Towards the Design of Robotic Drivers for Full-Scale Self-Driving Racing Cars [C]// 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019: 5643-5649.

[4] ZHUANG L, ZHANG Z, WANG L. The automatic segmentation of residential solar panels based on satellite images: A cross learning driven U-Net method [J]. Applied Soft Computing, 2020, 92: 106283.

[5] VEDDER B, SVENSSON B J, VINTER J, et al. Automated Testing of Ultrawideband Positioning for Autonomous Driving [J]. Journal of Robotics, 2020, 2020: 1-15.

[6] BOJARSKI M, DEL TESTA D, DWORAKOWSKI D, et al. End to End Learning for Self-Driving Cars[J]. arXiv: 1604.07316, 2016.

[7] XU H, GAO Y, YU F, et al. End-to-End Learning of Driving Models from LargeScale Video Datasets [C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017: 2174-2182.

[8] CHEN L, WANG Q, LU X, et al. Learning Driving Models From Parallel End-to-End Driving Data Set[J]. Proceedings of the IEEE, 2020, 108(2): 262-273.

[9] CODEVILLA F, MULLER M. End-to-end driving via conditional imitation learning[C]// 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 4693-4700.

[10] SUTTOM R S, BARTO A G. Reinforcement learning: An introduction [M]. MIT Press, 1998.

[11] MAXIMILIAN J, RAOUL D, MARIN T, et al. End-to-End Race Driving with Deep Reinforcement Learning[C]// International Conference on Robotics and Automation (ICRA). IEEE, 2018: 2070-2075.

[12] KENDALL A, HAWKE J, JANZ D, et al. Learning to Drive in a Day [C]// 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2019: 8248-8254.

[13] TOROMANOFF M, WIRBEL E, MOUTAR-DE F. End-to-End Model-Free Reinforcement Learning for Urban Driving Using Implicit Affordances[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 7153-7162.

[14] CHEN S, WANG M, SONG W, et al. Stabilization Approaches for Reinforcement Learning-Based End-to-End Autonomous Driving[J]. IEEE Transactions on Vehicular Technology, 2020, 69(5): 4740-4750.

[15] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[16] 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 ICML. 2018.

[17] SHI W, SONG S, WU C. Soft Policy Gradient Method for Maximum Entropy Deep Reinforcement Learning [C]// International Joint Conference on Artificial Intelligence (IJCAI). 2019.

[18] ZHU F, WU W, FU Y C, et al. Security depth reinforcement learning method based on double depth network[J]. Acta Computerica, 2019, 42(8).

[19] GARCI A J, FERNÁNDEZ F. A comprehensive survey on safe reinforcement learning [J]. Journal of Machine Learning Research, 2015, 16(1): 1437-1480.

[20] GARCIA J, FERNANDEZ F. Safe Exploration of State and Action Spaces in Reinforcement Learning[J]. Journal of Artificial Intelligence Research, 2014, 45(1).

[21] BERKENKARMP F, TURCHETTA M, SCHOELLIG A P, et al. Safe model-based reinforcement learning with stability guarantees[J]. arXiv: 1705.08551, 2017.

[22] MAZUMDER S, LIU B, WANG S, et al. Action permissibility in deep reinforcement learning and application to autonomous driving[C]// KDD'18 Deep Learning Day. 2018.

[23] LIU Q, ZHAI J W, ZHANG Z, et al. A review of deep reinforcement learning[J]. Acta Computerica Sinica, 2018, 41(1): 1-27.

[24] LEE K, SAIGOL K, THEODOROU E A. Early Failure Detection of Deep End-to-End Control Policy by Reinforcement Learning[C]// 2019 International Conference on Robotics and

Automation (ICRA). IEEE,2019.

[25] FUJIMOTO S, HOOF H, MEGER D. Addressing function approximation error in actor-critic methods[C]// International Conference on Machine Learning. PMLR,2018:1587-1596.

[26] ZIEBART B D, MAAS A L, BAGNELL J A, et al. Maximum entropy inverse reinforcement learning[C]// AAAI Conference on Artificial Intelligence (AAAI). 2008:1433-1438.

[27] LEVINE S, FINN C, DARRELL T, et al. End-to-End Training of Deep Visuomotor Policies[J]. Journal of Machine Learning Research,2015,17(1):1334-1373.

[28] O'DONOGHUE B, MUNOS R, KAVUKCUOGLU K, et al. PGQ: Combining policy gradient and Q-learning[J]. arXiv: 1611.01626,2016.

[29] NACHUM O, NOROUZI M, XU K, et al. Bridging the gap between value and policy based reinforcement learning[C]// Advances in Neural Information Processing Systems (NIPS). 2017:2772-2782.

[30] HAARNOJA T, TANG H, ABBEEL P, et al. Reinforcement learning with deep energy-based policies[C]// International Conference on Machine Learning (ICML). 2017:1352-1361.

[31] MINK J W. The basal ganglia: focused selection and inhibition of competing motor programs[J]. Progress in Neurobiology, 1996,50(4):381-425.

[32] LIPTON Z C, AZIZZADENESHELI K, KUMAR A, et al. Combating reinforcement learning's sisyphean curse with intrinsic fear[J]. arXiv:1611.01211,2016.

[33] AGARWAL A, ABHINAU K V, DUNOVAN K, et al. Better Safe than Sorry: Evidence Accumulation Allows for Safe Reinforcement Learning[J]. arXiv:1809.09147,2018.

[34] REN J, MCLSAAC K A, PATEL R V, et al. A potential field model using generalized sigmoid functions[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2007,37(2):477-484.

[35] GOMES G S, LUDERMIR T B. Complementary log-log and probit: activation functions implemented in artificial neural networks[C]// 2008 Eighth International Conference on Hybrid Intelligent Systems. IEEE,2008:939-942.

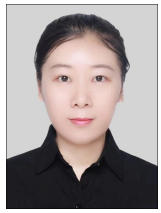
[36] SCHULMAN J, ABBEEL P, CHEN X. Equivalence between policy gradients and soft Q-learning[J]. arXiv: 1704.06440, 2017a.

[37] CHEN Z, HUANG X. End-to-end learning for lane keeping of self-driving cars[C]// 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE,2017.

[38] BADRINARAYANAN V, KENDALL A, CIPOLLA R. Segnet: A deep convolutional encoder-decoder architecture for scene segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2017,39(12):2481-2495.

[39] SILVER D, HUANG A, MADDISON C J A, et al. Mastering the game of go with deep neural networks and tree search[J]. Nature,2016,529(7587):484-489.

[40] SILVER D, HUBERT T, SCHRITTWIESER I J, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm[C]// CoRR. 2017.



DAI Shan-shan, born in 1990, postgraduate candidate. Her main research interests include reinforcement learning, deep reinforcement learning and automatic drive.

LIU Quan, born in 1969, Ph.D, professor, supervisor. His main research interests include reinforcement learning, deep reinforcement learning and automated reasoning.