面向自动驾驶的混合架构哈密顿-雅可比-贝尔曼近端策略优化方法研究

王金强,宋利蓉,蒋远博,雍宾宾,李 妍,周庆国*
(兰州大学信息科学与工程学院,甘肃兰州 730000)

摘 要: 深度强化学习(Deep Reinforcement Learning,DRL)为解决自动驾驶中复杂的序列决策问题提供了强大的端到端学习框架,但车辆控制策略的安全性仍是一个核心难题,基于哈密顿-雅可比-贝尔曼(Hamilton-Jacobi-Bellman,HJB)方程的物理信息强化学习(Physics-Informed Reinforcement Learning,PIRL)方法展现了巨大潜力。然而,这类方法在实践中受限于选用神经网络的性能。采用传统的多层感知机(MultiLayer Perceptron,MLP)时,难以为HJB物理约束提供高保真的梯度信号,从而引发训练不稳定和模型效率低下问题。为解决这一难题,本文提出了一种面向自动驾驶任务的混合架构哈密顿-雅可比-贝尔曼近端策略优化(Mixed Architecture Hamilton-Jacobi-Bellman Proximal policy Optimization,MAHPO)算法,该方法创新性地构建了一个异构的Actor-Critic框架,其策略网络(Actor)采用MLP以保证决策效率,而值函数网络(Critic)采用柯尔莫哥洛夫-阿诺德网络(Kolmogorov-Arnold Network,KAN)网络进行近似。进一步地,通过训练值函数表征网络KAN的内部可学习光滑B样条函数,可利用轨迹数据自适应地学习非线性变换,从而高效地建模复杂的价值函数及其平滑的梯度场,确保策略网络稳定更新。在自动驾驶模拟环境MetaDrive中的实验结果表明:相较于基线算法,MAHPO算法在任务成功率、碰撞率和离路率等关键性能指标上均取得明显提升,相较于最优基准的软演员-评论家算法(Soft Actor-Critic,SAC)在平均成功率上提升了5.88%,离路率相较于原始HJBPPO算法下降了约78.22%。

关键词: 深度强化学习(DRL);自动驾驶;哈密顿-雅可比-贝尔曼(HJB)方程;混合架构;近端策略优化(PPO)
基金项目: 兰州大学中央高校基本科研业务费专项资金(No.lzujbky-2024-eyt01);国家自然科学基金(No.61402210);甘肃省拔尖领军人才项目
中图分类号: TP273.5 **文献标识码:** A **文章编号:** 0372-2112(2026)03-1024-12
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20250977

Research on Mixed Architecture Hamilton-Jacobi-Bellman Proximal Policy Optimization Method for Autonomous Driving

WANG Jinqiang, SONG Lirong, JIANG Yuanbo, YONG Binbin, LI Yan, ZHOU Qingguo*
(*School of Information Science and Engineering*, *Lanzhou University*, *Lanzhou*, *Gansu* 730000, *China*)

Abstract: Deep reinforcement learning (DRL) provides a powerful end-to-end learning framework for addressing complex sequential decision-making problems in autonomous driving, but the safety of vehicle control policies remains a core challenge. physics-informed reinforcement learning (PIRL) methods based on the hamilton-jacobi-bellman (HJB) equation have demonstrated significant potential. However, such methods are severely limited in practice by the performance of the selected neural networks. Conventional multilayer perceptrons (MLPs) struggle to provide high-fidelity gradient signals for HJB physical constraints, thereby leading to training instability and model inefficiency issues. To address this challenge, we proposes a mixed architecture Hamilton-Jacobi-Bellman proximal policy optimization (MAHPO) algorithm tailored for autonomous driving tasks. This method innovatively constructs a heterogeneous Actor-Critic framework. Its policy network (Actor) uses an MLP to ensure efficient decision-making, while the value function network (Critic) is approximated by a kolmogorov-arnold network (KAN). Furthermore, the KAN-based value function representation network employs internal learnable smooth B-spline functions that can adaptively learn nonlinear transformations from trajectory data. This capability enables efficient modeling of complex value functions and their smooth gradient fields, thereby ensuring stable policy network updates. Experimental results in the MetaDrive simulation environment validate the efficacy of the MAHPO algorithm, which yields significant improvements over baselines across key performance metrics such as success rate, collision rate, and off-road rate. It has an average success rate improvement of 5.88% compared with the optimal benchmark soft ac-

收稿日期:2025-12-11;录用日期:2026-01-06;责任编委:吕金虎
*通信作者:周庆国

tor-critic (SAC), and the off-road rate has decreased by about 78.22% compared with the original HJBPPO algorithm.

Keywords: deep reinforcement learning (DRL); autonomous driving; Hamilton-Jacobi-Bellman (HJB) equation; mixed architecture; proximal policy optimization(PPO)

Foundation Item(s): Fundamental Research Funds for the Central Universities (No.lzujbky-2024-eyt01); National Natural Science Foundation of China (No.61402210); Gansu Province's Top Leading Talents

0 引言

深度强化学习(Deep Reinforcement Learning,DRL)[1-2]作为人工智能领域目前最具颠覆性的技术之一,不仅在棋类[3-4]、游戏[5]、科学[6]、机器人[7]、大语言模型[8]和电子信息[9-10]领域取得了突破性的进展,还逐步成为解决复杂决策问题的核心工具。这项技术的成功主要源自DRL成功地将深度神经网络与强化学习结合,通过深度神经网络强大的表征能力来学习高维状态空间,并利用强化学习的试错机制来优化长期决策策略。

在众多研究领域中,自动驾驶是DRL面临的最难、最具挑战性的应用场景之一。通常情况下,该任务本质上是一个以安全为基础的连续序列决策问题,行驶的车辆需在复杂多变的交通环境中实时感知周围状况,同时通过车辆的自身状态来预测其他交通参与者的行为,并作出安全、高效的驾驶决策。然而,在实际的驾驶策略模型训练过程中,传统强化学习算法通常依赖最大化任务奖励,难以提供严格的安全性保证,且面临收敛性不稳定问题,容易陷入局部最优解困境。为应对这一挑战,以哈密顿-雅可比-贝尔曼(Hamilton-Jacobi-Bellman,HJB)方程理论为基础的最优控制理论方法被集成到近端策略优化(Proximal Policy Optimization,PPO)算法[11]上,形成了HJBPPO算法。该方法创新性地将最优控制理论中的HJB方程作为一种物理约束,旨在通过优化值函数来弥补动态系统中连续时间和离散时间计算回报间的信息差距,并利用微分消除梯度误差,为策略学习提供更加精准的反馈。

然而,在使用传统深度神经网络近似策略网络和值函数网络时,神经元往往依赖固定的、非自适应的激活函数,其存在内在梯度表征缺陷,例如修正线性单元(Rectified Linear Unit,ReLU)和双曲正切函数(Hyperbolic Tangent,Tanh)等激活函数产生的分段常数梯度,这限制了神经网络高保真地拟合目标函数的导数,即平滑且复杂的值函数梯度场。归根结底,传统多层感知机(MultiLayer Perceptron,MLP)通常通过组合预设的、刚性的非线性单元来逼近目标函数,从而对于非平滑的值函数梯度场产生了一种次优的权衡,即智能体为满足该约束而采取的规避行为缺乏效率和平滑性,在自动驾驶任务中反而产生了更高的碰撞率和不稳定。

为了解决强化学习模型中采用HJB方程产生的不平滑梯度场问题,以及使用有限差分计算动态系统产生的误差和平滑性问题,本文提出了一种面向自动驾驶任务的混合架构哈密顿-雅可比-贝尔曼近端策略优化(Mixed Architecture Hamilton-Jacobi-Bellman Proximal policy Optimization,MAHPO)算法。具体的,MAHPO由策略网络和值函数网络两部分组成,其中,策略网络使用传统的MLP实现,其利用当前车辆状态求取最优控制动作。值函数网络通过基于Kolmogorov-Arnold表示定理(Kolmogorov-Arnold Represe-ntation Theorem,KART)的柯尔莫哥洛夫-阿诺德网络(Kolmogorov-Arnold Network,KAN)[12]实现。KAN的核心组件是训练可学习的光滑B样条函数,它能根据数据自适应地学习最适合的非线性变换函数,捕捉价值函数的局部细节和剧烈变化。更进一步地,通过值函数网络可近似拟合准确的优势函数,更好地学习策略网络,从而提升样本学习效率和模型最终的性能。

综上所述,本文的主要贡献如下:

(1)提出了一种新的基于Actor-Critic方法的混合架构MAHPO。在该架构中,策略网络基于依赖通用近似定理的MLP网络,值函数网络通过基于KART定理和可学习B样条的KAN网络。该结构能够有效解决值函数的梯度场问题,为策略网络的更新提供精确的指导。

(2)提出了使用可学习B样条函数作为强化学习值函数的近似方法,为自动驾驶等复杂问题的建模和可解释性提供了新方案。

1 相关工作

DRL为自动驾驶的序列决策任务提供了端到端的学习框架[13-14],这类方法能够从高维的感知信息学习最优车辆控制策略,实现平稳驾驶。例如,Kendall等人[15]使用单个单目图像作为输入,在少数数据中学习车道跟随策略,通过使用无模型的DRL算法摆脱了对驾驶逻辑规则的依赖。这也是现有大多数基于连续动作控制算法[16-18]实现端到端自动驾驶的关键。一般的,强化学习算法在最大化任务奖励的同时通常难以提供严格的安全性保证,这在安全攸关的自动驾驶场景中是根本性缺陷。为解决该问题,一个热门的研究方向是将安全约束显式地融入学习框架中,例如通过将控制屏障函数(Control Barrier Function,

CBF)[19]集成到安全状态集中,并通过修正策略模型的输出来确保决策动作始终在动作空间内。此外,Feng等人[20]提出了一种基于密度DRL的自动驾驶方法,该方案可通过移除非安全关键状态并重新连接关键状态来重构马尔可夫决策过程,从而使训练数据中的信息得到密集化。此外,另一种经典方法是将传统控制理论中的安全性(如HJB方程)与强化学习框架结合,引入基于物理约束的正则项,以在理论层面提供更强的安全保障。

综上所述,上述方法在解决端到端自动驾驶任务中,主要依赖于传统的MLP近似策略值函数和价值函数,针对异构网络架构组成方面的研究相对较少,这也是本文聚焦于混合架构设计的主要原因。

2 问题描述与方法建模

2.1 问题描述

考虑一个自动驾驶环境,其车辆动力学的组成方程为

$$\dot{x}=f(x,u) \tag{1}$$

其中,$f(\cdot)$表示自动驾驶车辆的动态系统。t时刻车辆的动力学信息为$\boldsymbol{x}_t^{\text{ego}}\in\mathbb{R}^6$,表示一个6维向量,包含了车辆自身的关键动力学与姿态信息,具体包括当前速度、航向角、相对于车道中心的横向偏离、当前转向角、上一时间步的油门/刹车指令以及横摆角;t时刻智能体的状态信息变量为$\boldsymbol{x}_t=(\boldsymbol{x}_t^{\text{lidar}},\boldsymbol{x}_t^{\text{ego}},\boldsymbol{x}_t^{\text{lane}},\boldsymbol{x}_t^{\text{nav}})$,其中$\boldsymbol{x}_t^{\text{lidar}}\in\mathbb{R}^{240}$是一个240维向量,表示从激光雷达传感器获取的原始数据,用于检测车辆周围50 m范围内的其他车辆、障碍物及其相对位置;$\boldsymbol{x}_t^{\text{lane}}\in\mathbb{R}^3$为一个3维的向量,用于描述当前车道的结构,包含了关于左侧黄线、右侧路沿以及车辆在车道内横向位置的信息;$\boldsymbol{x}_t^{\text{nav}}\in\mathbb{R}^{10}$为一个10维的向量,由全局路径规划器提供,包含了引导车辆到达最终目标点的导航指令和路径点信息;$\boldsymbol{u}_t\in\mathbb{R}^3$表示智能体控制车辆的动作,由方向转角$u_s$、加速度$u_a$和刹车$u_b$三部分组成,因此最优值函数可以表示为

$$V^*(x)=\sup_u\int_{t_0}^{\infty}\gamma^t R\big(x(\tau;\,t_0,x_0,u(\cdot)),u(\tau)\big)\mathrm{d}\tau \tag{2}$$

从式(2)可得,最优的动作函数为

$$u(x)=\arg\max_a\pi_\theta(a|x) \tag{3}$$

本文的研究目标是如何通过交互学习训练得到一个最优车辆控制模型,使得车辆在道路行驶过程中安全可靠,对应到强化学习模型中,需通过大量的轨迹数据来拟合得到最大化累计期望奖励来指导模型的训练。

2.2 方法建模

本文所提MAHPO的架构如图1所示,其中,①表示MetaDrive环境中自动驾驶的地图组成;②表示策略网络Actor的内部组成原理,其依赖于一个MLP神经网络;③表示智能体,其内部包含交互数据的轨迹;④表示整个智能体的组成部分,包含策略网络和值函数网络,以及模型的训练流程;⑤表示自动驾驶车辆的任务轨迹,即需要智能体控制车辆在未发生碰撞和离路的情况下安全到达目的地;⑥表示基于KAN网络近似的值函数网络结构内部示意图。

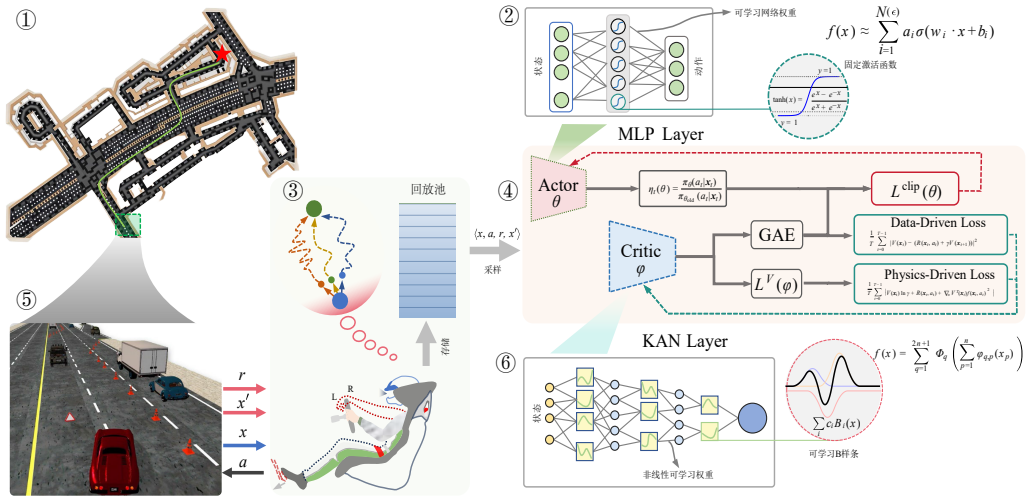


图1 MAHPO架构

Figure 1 MAHPO method

2.2.1 策略网络架构

策略网络π_θ的结构如图1中②所示,由一个三层MLP神经网络组成,输入层为智能体的状态信息,输出层为车辆控制动作。根据t时输入的状态$\boldsymbol{x}_t=$

$(\boldsymbol{x}_t^{\text{lidar}},\boldsymbol{x}_t^{\text{ego}},\boldsymbol{x}_t^{\text{lane}},\boldsymbol{x}_t^{\text{nav}})$近似得到当前动作为$\pi_\theta(a_t|\boldsymbol{x}_t)$,通过前后两步之间的更新来引导策略学习,PPO中定义了概率比例$r_t(\theta)$,表示为

$$r_t(\theta)=\frac{\pi_\theta(a_t|\boldsymbol{x}_t)}{\pi_{\theta_{\text{old}}}(a_t|\boldsymbol{x}_t)} \tag{4}$$

进一步可以得到策略网络的损失函数为

$$L^{\text{CLIP}}(\theta)=\hat{E}_t\left[\min\left(r_t(\theta)\hat{A}_t,\text{clip}(r_t(\theta),1-\varepsilon,1+\varepsilon)\hat{A}_t\right)\right] \tag{5}$$

其中,$\hat{A}_t$采用通用优势估计(Generalized Advantage Estimation,GAE)[21]方法计算,表示为

$$\hat{A}_t=\delta_t+(\gamma\lambda)\delta_{t+1}+\cdots+(\gamma\lambda)^{T-t+1}\delta_{T-1} \tag{6}$$

此处时序差分误差(Temporal Difference,TD)表示为

$$\delta_t=r_t+\gamma V(\boldsymbol{x}_{t+1})-V(\boldsymbol{x}_t) \tag{7}$$

为计算$V(\boldsymbol{x}_t)$,需要通过MLP神经网络近似状态和决策动作之间的关系。

2.2.2　值函数架构

值函数网络$V_{\boldsymbol{\Phi}}(x)$的结构如图1中⑥所示,其由一个三层的KAN神经网络组成,输入层维度为状态信息和车辆控制动作信息,输出层为值函数,根据t时刻输入的状态$\boldsymbol{x}_t=(\boldsymbol{x}_t^{\text{lidar}},\boldsymbol{x}_t^{\text{ego}},\boldsymbol{x}_t^{\text{lane}},\boldsymbol{x}_t^{\text{nav}})$,可以通过KAN近似得到当前的值函数为$V_{\boldsymbol{\Phi}}(x)$。当值函数$V_{\boldsymbol{\Phi}}(x)$满足HJB方程时,最优值函数的成立依赖于式(8):

$$\begin{aligned}&V(x)\ln\gamma+R\left(x,u(x)\right)+\nabla_x V^T(x)f\left(x,u(x)\right)\\&\approx V(x)\ln\gamma+\sup_{u\in U}R(x,u)+\nabla_x V^T(x)f(x,u)\end{aligned} \tag{8}$$

其中,T表示回合中的时间步数;$\nabla_x V^T$表示对系统通过自动微分得到的值函数梯度,那么基于HJB方程的损失函数变为

$$\begin{aligned}\text{MSE}_f=\frac{1}{T}\sum_{t=0}^{T-1}|&V\left(\boldsymbol{x}_t\right)\ln\gamma+R\left(\boldsymbol{x}_t,a_t\right)\\&+\nabla_x V^T\left(\boldsymbol{x}_t\right)f\left(\boldsymbol{x}_t,a_t\right)|^2\end{aligned} \tag{9}$$

其中,使用有限差分近似$f\left(\boldsymbol{x}_t,a_t\right)$,则式(9)可以重写为

$$\begin{aligned}\text{MSE}_f=\frac{1}{T}\sum_{t=0}^{T-1}|&V\left(\boldsymbol{x}_t\right)\ln\gamma+R\left(\boldsymbol{x}_t,a_t\right)\\&+\nabla_x V^T\left(\boldsymbol{x}_t\right)\left(\frac{\boldsymbol{x}_{t+1}-\boldsymbol{x}_t}{\Delta t}\right)|^2\end{aligned} \tag{10}$$

则基于数据驱动的值函数近似误差表示为

$$\text{MSE}_u=\frac{1}{T}\sum_{t=0}^{T-1}\left|V(\boldsymbol{x}_t)-\left(R(\boldsymbol{x}_t,a_t)+\gamma V(\boldsymbol{x}_{t+1})\right)\right|^2 \tag{11}$$

考虑使用KAN近似值函数过程,由于有界域上的一个多元连续函数可写为由单变量连续函数和二元加法运算组成的有限复合函数,因此对于一个光滑函数V:$[0,1]^n\to R$来说,存在

$$V(x)=\sum_{q=1}^{2n+1}\boldsymbol{\Phi}_q\left(\sum_{p=1}^{n}\phi_{q,p}(x_p)\right) \tag{12}$$

其中,$\phi_{q,p}$:$[0,1]\to R$和$\boldsymbol{\Phi}_q$:$R\to R$,值函数$\boldsymbol{V}(x)$网络结构表述为

$$V(x)=\boldsymbol{\Phi}_{\text{out}}\circ\boldsymbol{\Phi}_{\text{in}}\circ x \tag{13}$$

其中,$\boldsymbol{\Phi}_{\text{in}}$表示网络的输入为$n$、输出为$2n+1$,对应的$\boldsymbol{\Phi}_{\text{out}}$的输入则为$2n+1$,输出为1,具体表述为

$$\boldsymbol{\Phi}_{\text{in}}=\begin{pmatrix}\phi_{1,1}(\cdot)&\cdots&\phi_{1,n}(\cdot)\\\vdots&\vdots&\vdots\\\phi_{2n+1,1}(\cdot)&\cdots&\phi_{2n+1,n}(\cdot)\end{pmatrix} \tag{14}$$

$$\boldsymbol{\Phi}_{\text{out}}=\left(\boldsymbol{\Phi}_1(\cdot),\boldsymbol{\Phi}_2(\cdot),\cdots,\boldsymbol{\Phi}_{2n+1}(\cdot)\right) \tag{15}$$

组合起来得到

$$\boldsymbol{\Phi}=\begin{pmatrix}\phi_{1,1}(\cdot)&\cdots&\phi_{1,n_{\text{in}}}(\cdot)\\\vdots&\vdots&\vdots\\\phi_{n_{\text{out}},1}(\cdot)&\cdots&\phi_{n_{\text{out}},n_{\text{in}}}(\cdot)\end{pmatrix} \tag{16}$$

最后值函数网络可以表示为$V_{\boldsymbol{\Phi}}(x)$:

$$V_{\boldsymbol{\Phi}}(x)=\boldsymbol{\Phi}_{L-1}\circ\boldsymbol{\Phi}_1\circ\boldsymbol{\Phi}_0\circ x \tag{17}$$

其中,图1中⑥的非线性表示模块里的激活函数采用B样条函数,其为一个分段多项式函数,表示为基函数$B_i(x)$及其对应系数c_i的线性组合:

$$\text{spline}(x)=\sum_i c_i B_i(x) \tag{18}$$

具体的,B样条基函数通过Cox-de Boor递归公式[22]构造,对于递归从阶数为$p=0$的分段常数B样条曲线表示为

$$B_{i,0}(x)=\begin{cases}1, & \text{if } t_i\leqslant x<t_{i+1}\\0, & \text{otherwise}\end{cases} \tag{19}$$

则高阶B样条曲线(当$p>0$时)由低阶样条曲线递归生成,计算过程为

$$B_{i,p}(x)=\frac{x-t_i}{t_{i+p}-t_i}B_{i,p-1}(x)+\frac{t_{i+p+1}-x}{t_{i+p+1}-t_{i+1}}B_{i+1,p-1}(x) \tag{20}$$

最后,依据式(9)和式(11)得到值函数的总的损失函数表示为

$$J(\varphi)=0.5\text{MSE}_u+\lambda_{\text{HJB}}\text{MSE}_f \tag{21}$$

2.3　算法执行细节

MAHPO算法由MLP实现的策略网络Actor与环境交互收集一批经验轨迹。接着利用Critic计算出的优势函数来构建并优化PPO的裁剪目标损失。其中,Critic更新的总损失由标准贝尔曼误差损失和HJB物理约束损失两部分构成。为了计算后者,本文利用KAN架构的内在可微性,解析地计算出值函数关于状态的高质量梯度$\nabla_x V_\varphi\left(\boldsymbol{x}_t\right)$,并据此构建HJB残差。最终,通过对这个混合损失函数进行梯度下降更新Critic的参数,具体执行过程如算法1所示。

算法1 MAHPO算法训练流程

输入：环境$\mathcal{E}$，动力学$f(x,a)$，奖励函数$r(x,a)$，折扣因子γ，KAN网络的层数L，B-spline阶数k，网格数G，超参数ε，λ_{HJB}，以及学习率α_1,α_2

输出：优化的策略网络π_θ^*，值网络V_φ^*

初始化策略网络π_θ

初始化层L，B-spline阶数k，网格数G，节点向量$\mathcal{T}$值网络V_φ

FOR iteration = 1, 2, ⋯ DO

运行策略π_θ收集轨迹$\mathcal{D}=\left\{\left(x_t,a_t,r_t,x_{t+1}\right)\right\}(t=1,2,\cdots,T)$

// KAN前向传播B-spline基函数展开

FOR 每个状态$x_t\in\mathcal{D}$ DO

$x^{(0)}\leftarrow \boldsymbol{x}_t$

FOR $l=1,2,\cdots,L$ DO

$u\leftarrow \text{normalize}\left(x^{(l-1)}\right)$

$x^{(l)}\leftarrow \sum_{i,j}\phi_{i,j}^{(l)}(u_i)$，其中$\phi_{i,j}^{(l)}(u)=\sum_{p=0}^{G+k-1}c_{i,j,p}^{(l)}N_p^k(u;\mathcal{T})$

END FOR

$V_\varphi(x_t)\leftarrow x_1^{(L)}$

计算优势函数A，重要性采样比$r_t(\theta)$

// 更新策略网络

$L(\theta)\leftarrow \frac{1}{T}\sum_t\min\left[r_t(\theta)A_t,\text{clip}\left(r_t(\theta),1-\varepsilon,1+\varepsilon\right)A_t\right]$

$\theta\leftarrow\theta+\alpha_1\nabla_\theta L(\theta)$

// 计算基于KAN的值网络损失

$\text{MSE}_u\leftarrow\frac{1}{T}\sum_t\left(V_\varphi(\boldsymbol{x}_t)-R_t\right)^2$

// HJB约束损失

FOR $\left(\boldsymbol{x}_t,a_t\right)\in\mathcal{D}$ DO

$\nabla_s V_\varphi(\boldsymbol{x}_t)\leftarrow\frac{\partial V_\varphi}{\partial x_t}$

$\frac{\partial\phi_{i,j}^{(l)}}{\partial a}=\sum_p c_{i,j,p}^{(l)}\cdot N_p^{k-1}(a)$

$\mathcal{R}_{\text{HJB}}(t)\leftarrow\nabla_s V_\varphi(x_t)^T f\left(x_t,a_t\right)+r\left(x_t,a_t\right)$

END FOR

$\text{MSE}_f\leftarrow\frac{1}{T}\sum\mathcal{R}_{\text{HJB}}(t)^2$

$J(\varphi)\leftarrow\text{MSE}_u+\lambda_{\text{HJB}}\cdot\text{MSE}_f$

$\varphi\leftarrow\varphi-\alpha_2\nabla_\varphi J(\varphi)$

END FOR

3 实验结果

3.1 实验环境

为验证所提MAHPO算法的性能表现，本文在自动驾驶MetaDrive仿真环境[23]中进行了训练和评估测试。具体而言，训练环境由100张不同地图组成，每个地图包含4个地图块，这些块包含了C型（环形路）、X型（十字交叉路）、O型（环岛）、R型（外坡道）和T型（岔路）等路状，如图2所示。在训练过程中按照环境的默认设置进行了配置，测试过程使用随机的由相同路状的20张地图作为评估环境。

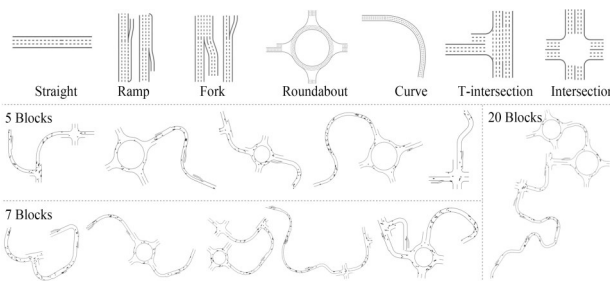


图2 实验环境的路状组成原理块[23]

Figure 2 Principle block of road condition composition in the experimental environment[23]

此外，针对奖励函数设计，本文采用了MetaDrive环境默认的奖励函数设置[23]，奖励函数组成为

$$R=c_1R_{\text{driving}}+c_2R_{\text{lateral}}+c_3R_{\text{heading}}+c_4R_{\text{steering}}-c_5P_{\text{collision}}+R_{\text{termination}} \tag{22}$$

其中：驾驶奖励$R_{\text{driving}}=d_t-d_{t-1}$，$d_t$和$d_{t-1}$分别表示目标车辆的纵向移动，通过记录连续两个时间步的轨迹，提供密集奖励以鼓励智能体朝着目的地移动；横向奖励R_{lateral}鼓励智能体尽量靠近参考轨迹驾驶，当车辆与参考轨迹之间的距离大于阈值时，回合将因车辆驶离道路而终止；朝向奖励R_{heading}要求智能体的朝向与车道上某一点的方向保持一致，该点的位置通过每一步计算车辆与参考线之间的相对位置来获取；转向奖励R_{steering}是为了在速度较高时奖励大转向角，速度越高，转向角度大的惩罚就越大；碰撞惩罚$P_{\text{collision}}$是对与特定交通参与者发生碰撞的惩罚，它是一个固定的常量；终止奖励$R_{\text{termination}}$包含一组稀疏奖励。在回合结束时，其他稠密奖励将被禁用，仅一项稀疏奖励将给予智能体，依据其终止状态，在情节结束时，使用成功奖励和离路惩罚作为环境奖励的设置，这项惩罚将以负奖励的形式给出。

3.2 评价指标

为了对所提方法进行公平测试，本文采用式（23）中所示的四类指标评估算法性能：

$$\begin{cases}\text{AR}=\dfrac{1}{N}\sum_{i=1}^{N}R_i\\ \text{SR}=\dfrac{1}{N}\sum_{i=1}^{N}S_i\times100\%\\ \text{CR}=\dfrac{1}{N}\sum_{i=1}^{N}C_i\times100\%\\ \text{OR}=\dfrac{1}{\text{N}}\sum_{i=1}^{N}O_i\times100\%\end{cases} \tag{23}$$

其中：AR表示平均回报；SR表示成功率；CR表示碰撞率；OR表示离路率；$N\in\mathbb{N}^+$表示总测试次数；$R_i\in\mathbb{R}$表示第i次测试的累积回报；$S_i,C_i,O_i\in0,1$分别表示

第i次测试的成功、碰撞、离路标志。

3.3 对比实验

为了对MAHPO方法进行公平对比测试,在硬件方面,本文采用了双路128核心和两块Nvidia 4090显卡计算平台进行实验,并选取了当前应用在MetaDrive环境中的主流强化学习方法PPO[10]、软演员-评论家算法(Soft Actor-Critic,SAC)[24]和HJBPPO[25]算法进行测试,相应的实现均采用原始论文的官方代码执行测试,其中策略网络和值函数网络的网络结构参数如表1所示。

表1　MAHPO模型网络结构参数表

Table 1　Parameters of the network architecture of the MAHPO

网络	层名称	输入维度	输出维度	激活函数
策略网络(MLP)	输入层	259	—	—
	隐含层	128	128	Relu
	输出层	128	3	Tanh
值网络(KAN)	输入层	262	—	—
	隐含层	262	64	B-spline (G=8,k=7)
	输出层	64	1	线性

此外,为了能够有效地评估算法训练的效果,本文在训练过程中每隔30个回合进行一次评估,每次评估均采用20个种子,求取平均回报值、成功率、碰撞率和离路率。如图3所示,实验结果表明:SAC算法在学习速度和最终性能上均优于其他方法,在PPO家族内部,MAHPO算法在训练后期超越了标准PPO,展现了更优的性能。而HJBPPO算法表现最差,其学习曲线早早陷入低水平平台期。表明HJBPPO的实现工具(如MLP)无法有效利用HJB约束,这为本文引入KAN架构以解决此问题提供了关键的实验支持。

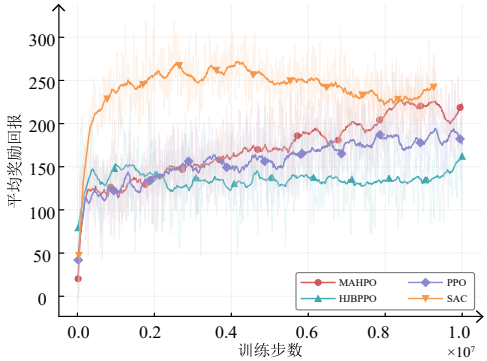


图3　各类方法的平均回报性能曲线图

Figure 3　Average return performance curves of various methods

图4揭示了各类算法在成功率上的性能表现,其中MAHPO凭借最稳健的学习曲线,持续提升并最终在训练结束时达到了约0.65的最高任务成功率。相比之下,初期领先的SAC虽然在训练中期(约0.5×10^7步)迅速达到约0.62的峰值,但随后却出现了明显性能衰退,最终成功率降至0.5以下,这种训练后期性能下降可能源于SAC的熵正则化机制,即SAC通过熵正则化鼓励策略探索,但固定的熵系数α可能导致训练后期探索过度,使得策略无法充分利用已学到的知识进行探索[24]。在复杂的决策环境中,过高的策略熵可能导致智能体做出次优决策,从而表现为性能的持续衰退。而HJBPPO在这一关键指标上始终表现最差,最终仅稳定在约0.32的低水平,表明基于MLP的实现无法有效指导策略学习。这为本文引入KAN架构以修复其梯度表征瓶颈提供了决定性的实验依据。

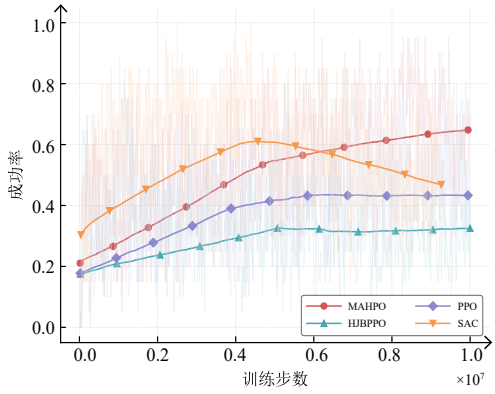


图4　各类算法的成功率性能曲线图

Figure 4　Success rate performance curves of various algorithms

再者,碰撞率与离路率也是衡量算法质量的重要指标,为分析各算法所学策略在安全维度上存在的显著差异,本文分别从碰撞率和离路率两个维度对所提方法进行了分析,如图5和图6所示。具体地,HJBPPO展现出一种矛盾的性能,在整个训练过程中维持了近乎为零的碰撞率(0.02),展现了最优的碰撞规避能力。然而,这种表现是以完全牺牲任务执行能力为代价的,其离路率稳定在不可接受的高水平(0.65)。使用MLP实现HJB约束虽然能够有效生成对危险状态的强排斥力,但由于梯度表征的粗糙性,该信号缺乏必要的精细度以指导智能体在遵守安全约束的同时执行有效的驾驶行为,从而导致策略收敛至一种无效的、纯粹的避险模式。与此相反,SAC学习到一种高风险策略,它能迅速将离路率降至接近零,表现出卓越的路径跟踪能力。然而,随着碰撞率在训练后期急剧攀升至所有算法中的最高点(>0.5),反映出其策略为最大化短期奖励(如循迹精度)而牺牲了关键的长期安全性。相比之下,MAHPO和PPO实现了对两个安全指标更为均衡的优化,且MAHPO在训练后期成功地将离路率降低到接近零的同时,将碰撞率维持在相对可控的水平(0.3),最终获得了最高的任务成功率。

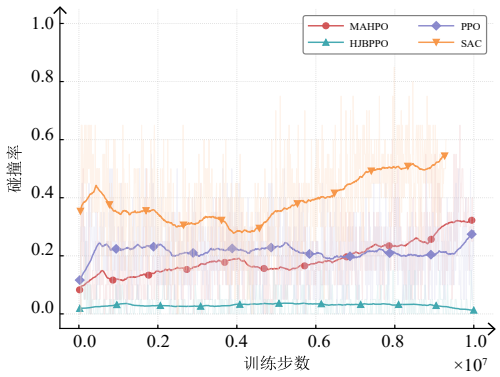


图5 各类算法碰撞率性能曲线图

Figure 5 Collision rate performance curves of various algorithms

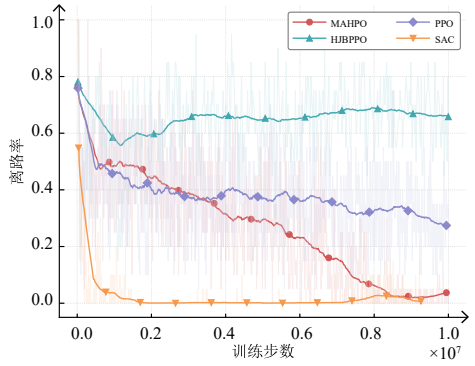


图6 各类算法的离路率性能变化图

Figure 6 Performance variation of off-road rate for various algorithms

此外，为了验证最终训练模型的性能，本文利用最优模型在评估环境中进行10次重复测试，每次测试采用50个回合求平均值，最后求取10次测试的平均指标。实验结果如表2所示，MAHPO展现了最优的整体性能，其成功率达到了0.792，在所有算法中性能最优，这一成功源于其在各项关键指标上实现了最佳的平衡：其在保持极低的离路率（0.044）的同时，将碰撞率维持在相对较低的水平（0.148），并且其平均回报奖励（236.372）不仅高，而且标准差（±47.669）远小于SAC，表明其策略具有高度的鲁棒性和一致性。与此形成鲜明对比的是，SAC展现出一种高风险、高回报的“激进”策略。尽管SAC的平均回报奖励最高（259.014）伴随偏离道路的行为（离路率=0.0），但这是以最高的碰撞率（0.252）为代价的。此外，值得注意的是HJBPPO的表现，与训练曲线中性能垫底的情况不同，在最终的泛化测试中，HJBPPO的性能（成功率0.648）显著超越了标准PPO基线（成功率0.478）。HJBPPO成功地将PPO极高的离路率（0.438）大幅降低至0.202，尽管其碰撞率（0.15）略高于PPO（0.084）。这一现象表明：HJB约束虽然可能在训练初期对探索造成干扰，但最终赋予了策略一种更强的泛化能力，使其在未见过的随机环境中能更好地维持在道路上。PPO虽然碰撞率最低，但由于无法有效完成任务，其整体性能最差。

表2 在评估环境中的四类指标测试结果

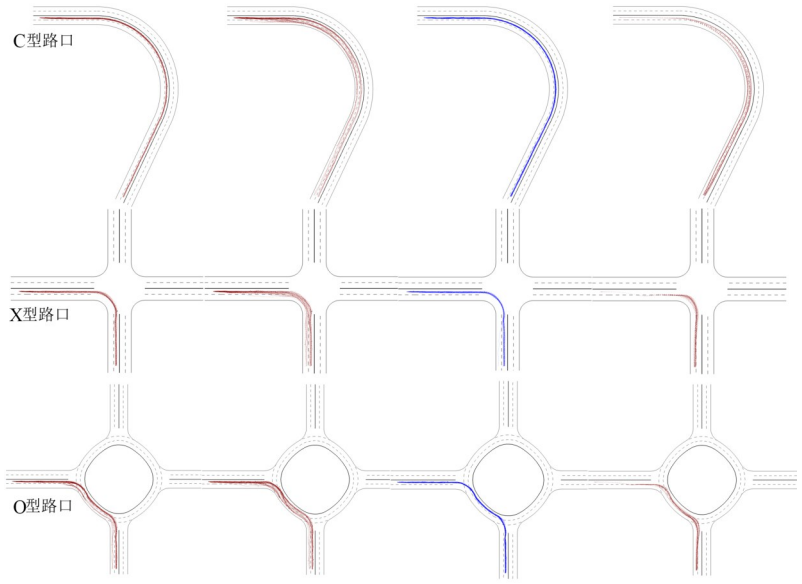
Table 2 Test results of the four types of indicators in the environment

方法名称	成功率(↑)	平均回报奖励(↑)	碰撞率(↓)	离路率(↑)
SAC	0.748 ± 0.003	**259.014 ± 133.07**	0.252 ± 0.003	0.0 ± 0.0
PPO	0.478 ± 0.006	148.307 ± 186.978	**0.084 ± 0.002**	0.438 ± 0.007
HJBPPO	0.648 ± 0.004	190.601 ± 198.837	0.15 ± 0.005	0.202 ± 0.004
MAHPO	**0.792 ± 0.002**	236.372 ± 47.669	0.148 ± 0.001	**0.044 ± 0.001**

注：±后的数值表示10次运行的标准差；加粗字体表示最优结果。

为验证所提方法在决策动作方面的性能，本文分别从简单和复杂组合的两种路状对四类算法进行了评估测试。图7展示了在单一路状中，PPO、HJBPPO、MAHPO和SAC四种算法在C型弯道、X型路口及O型环岛三种路况下的轨迹可视化结果图。其中，MAHPO的表现最佳，其轨迹在三种路况下，包括最具挑战性的O型环岛，均展现出平滑度和一致性。这直观地证明了MAHPO学习到的策略具备强大的泛化能力，能够稳定地适应不同的道路形态。同时，SAC也展现了高质量的控制策略，其轨迹平滑且循迹能力强。然而，其激进的驾驶风格在所有场景中均清晰可见，尤其是在环岛的入口和出口处。这种行为模式虽然能够最大化通行效率（对应其高奖励和低离路率），但也使其在面对动态障碍时缺乏足够的安全冗余，这正是其高碰撞率的根源。相比之下，PPO作为基准算法，其控制能力的不足在复杂路况下被进一步放大，其轨迹不仅在C型弯道中显得松散和抖动，在X型和O型路口中也表现出明显的不稳定和缺乏一致性，进一步表明PPO学习到的策略泛化能力有限，难以在多变的道路环境中维持精确控制。

最后，就HJBPPO算法而言，其轨迹可视化结果表明：在所有场景中轨迹均呈现出极度的发散和不连贯，尤其是在需要决策的X型路口和O型环岛，其策略网络输出的动作近乎丧失了控制，间接表明基于MLP的HJBPPO难以将HJB方程蕴含的信息进行有效的近似，致使学习的策略丧失了执行任务的能力。在组合路状评估中，本文分别在“ROT型路口”（环岛与多路口复合）和“CRTOX型路口”（弯道、环岛与多路



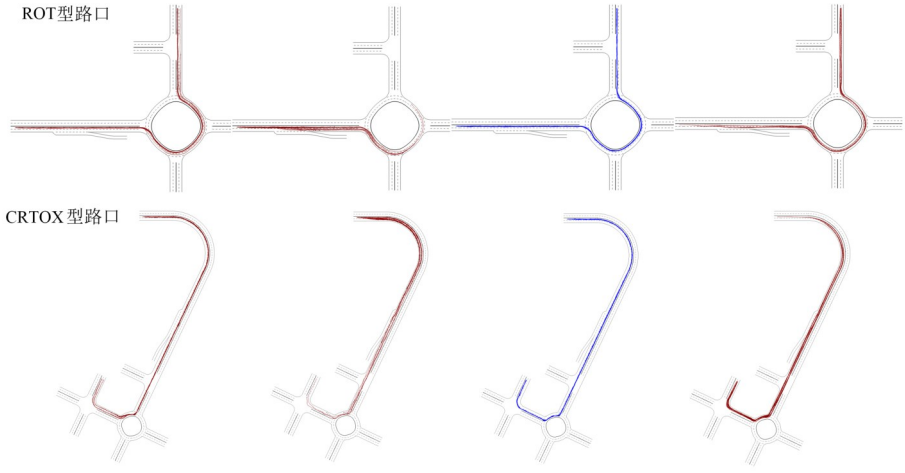
(a) PPO　(b) HJBPPO　(c) MAHPO　(d) SAC

图7　单一路状各类算法测试性能图

Figure 7　Performance Evaluation of Various Algorithms under a Single Road Condition

口序列)。两种复合场景中对PPO、HJBPPO、MAHPO和SAC四种算法进行了评估。图8实验结果表明:MAHPO的轨迹在两个最为复杂的场景中都展现了最优的控制水平,其轨迹如同一条精确绘制的示范路线,平滑、稳定且高度一致。尤其是在需要连续决策和精细操作的ROT和CRTOX序列中,MAHPO的轨迹表现出优秀的鲁棒性和泛化能力,证明其学习到了一个高质量、可信赖的驾驶策略,这与其在量化评估中的结论一致。就SAC算法而言,虽然SAC也成功地通过了这些复杂路况,但其策略的内在缺陷被清晰地暴露出来。其轨迹虽然大体平滑,但一致性明显劣于MAHPO,且其激进的动作依然存在。更重要的是,在CRTOX场景的末端,可以观察到SAC的轨迹出现了明显的决策失误(一条轨迹错误地急转弯),表明了策略的不稳定性。再者,PPO作为基准,在复杂路况下控制能力具有较大局限性。其轨迹不仅松散、抖动,在需要决策的环岛和路口处还表现出明显的不一致,轨迹束发散严重。进一步表明PPO学习到的策略难以应对复杂序列任务决策。最后HJBPPO的表现最差,在ROT和CRTOX两个场景中,其策略模型输出的动作相对发散混乱,这表明基于MLP的HJBPPO难以有效地对细致的梯度场中的信息学习。

就训练时间而言,四种算法的训练时长如图9所示,实验结果表明:四种算法在训练时间消耗方面存在显著



(a) PPO　(b) HJBPPO　(c) MAHPO　(d) SAC

图8　组合路状各类算法测试性能图

Figure 8　Performance evaluation of various algorithms under combined road conditions

差异。MAHPO算法的训练时长最高，达到5 429 min，相较于原始的HJBPPO算法（1 488 min）增加了约265%的时间消耗。这一结果说明：虽然将HJBPPO中的神经网络替换为KAN结构后实现了性能上的提升，但代价是显著增加了训练时长。KAN网络由于其特殊的网络架构和可学习的激活函数，需更多的计算资源和迭代时间来优化参数。SAC算法耗时4 429 min，PPO算法用时3 774 min，两者的训练时间处于中等水平。相比之下，HJBPPO算法以1 488 min的训练时长展现出最优的时间效率。本文提出的基于KAN网络的强化学习方法在Metadrive环境中具有优势，但该方法仍存在一些潜在的局限性值得讨论。首先，KAN网络相比传统的MLP引入了额外的计算开销。由于KAN在每条边上使用可学习的样条函数[12]，其参数量和前向传播时间复杂度通常高于同等规模的MLP网络。本文在状态空间（259维状态输入）下验证了方法的有效性，当状态维度显著增加时（如高维机器人控制任务），计算开销问题可能会更加突出。此外，KAN网络在高维状态空间下的可扩展性仍需进一步验证，因为样条函数在高维空间中可能面临“维度灾难”问题[26]，这可能导致训练收敛速度降低和参数量快速增长。

3.4 消融实验

3.4.1 不同网格大小对平均回报的影响分析

为验证MAHPO方法的关键参数对控制策略的影响，本文对KAN网络中的B样条函数的网格点数（Grids）进行了消融实验，由于其直接决定了值函数表征能力（即模型容量）以及学习的复杂度。图10中的实验结果表明：Grids=10是算法在环境中的最优参数，其平均回报和成功率均为最佳水平，相应的其成功率最终稳定在0.8左右，这主要源于其在离路率指标上实现了最快的优化，致使将该比率降至接近于零，进一步说明了其模型容量足以精确学习复杂的循迹策略。同时，其碰撞率维持在0.1~0.2的可控范围内，实现了安全与效率的有效平衡。

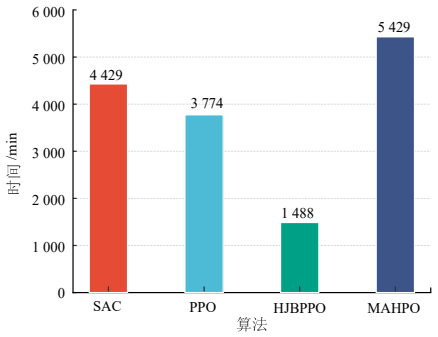
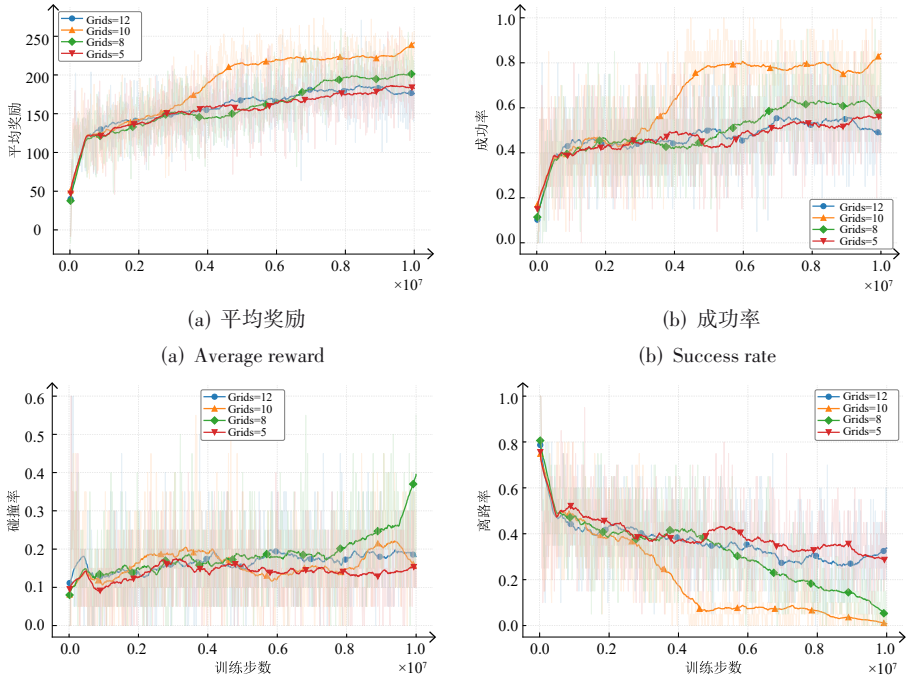


图9 组合路状各类算法测试性能图

Figure 9 Performance evaluation of various algorithm tests for combined road conditions

相比之下，Grids=5和8表现出欠拟合的特征，其较高的离路率限制了其学习有效控制策略，导致整体



(a) 平均奖励 (a) Average reward

(b) 成功率 (b) Success rate

(c) 碰撞率 (c) Collision rate

(d) 离路率 (d) Off-road rate

图10 不同B样条网格大小对性能的影响分析图

Figure 10 Analysis of the impact of different B-spline grids on performance

的成功率偏低。此外,虽然 Grids=12 的表达能力最强,但性能表现不佳,表明过高的模型复杂度反而损害了策略的泛化能力。综上所述,Grids=10 被证明是在模型复杂度与任务需求间的最佳平衡点,为客观描述消融参数对实验性能的影响,本文依据四个性能指标,对实验序列的最后 20 个数据求取平均值来反映性能,结果如表 3 所示。

表 3　网格数量对实验性能的影响结果

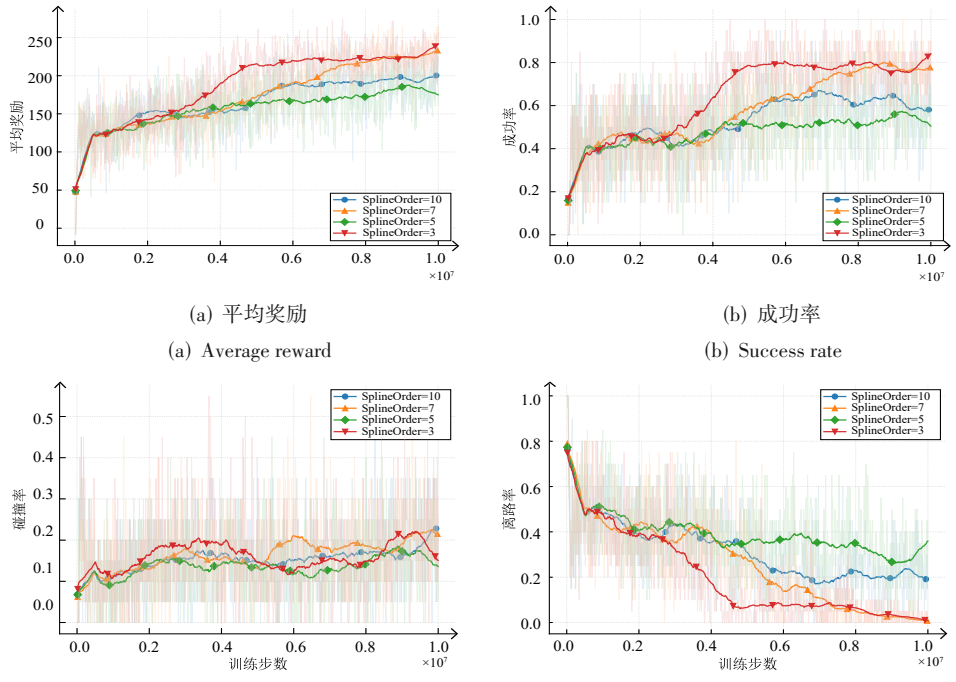
Table 3　Effects of grid number on experimental performance

参数(Grids)	平均回报	成功率	碰撞率	离路率
5	177.246	0.520	0.157	0.322
8	208.468	0.662	0.260	0.080
10	226.691	0.772	0.207	0.020
12	168.129	0.477	0.212	0.312

实验结果表明:当 Grids=10 时,算法达到最优性能,平均回报高达 226.691,成功率达到 0.772,同时碰撞率和离路率分别控制在 0.207 和 0.020 的较低水平,展现出最佳的任务完成能力和安全性。相比之下,当 Grids=5 时,虽然离路率较低(0.322),但成功率仅为 0.520,表明网格粒度不足导致策略表达能力受限;而当 Grids=12 时,尽管参数量更大,但成功率反而下降至 0.477,离路率上升至 0.312,可能是由于过细的网格划分引入了过拟合或训练不稳定性。Grids=8 作为次优配置,各项指标较为均衡,成功率达 0.662。综合来看,适中的网格参数(Grids=10)能够在模型复杂度与泛化能力间取得最佳平衡,为 KAN 网络在强化学习中的应用提供了重要的超参数选择依据。

3.4.2　不同的样条阶对平均回报的影响分析

SplineOrder(样条阶数)也是影响 MAHPO 所学习函数局部平滑度的关键因素。为探索该参数对算法的性能影响,本文在 MetaDrive 环境中进行了消融实验研究,图 11 中的实验结果表明:一个较低的样条阶数(SplineOrder=3)展现了最优的综合性能,该配置在平均回报和成功率上均收敛至最高水平,且成功率在训练结束时达到了 0.9 左右。这一优异表现主要归功于其在离路率指标上的优势,如图 11(d)所示,其离路率曲线下降得最快,并最终收敛至接近于零的最低水平。相比之下,更高的样条阶数(如 5 和 10),反而表现不佳。综上所述,对于连续控制任务的 MetaDrive 环境而言,一个更平滑的值函数表示(由低阶样条强制施加)构成了一种有益的归纳偏置,尤其是 SplineOrder=3 带来的平滑性正则化效应,有效防止了值函数学习到不必要的局部振荡,从而提供了更稳定、更可靠的梯度信号,也揭示了强制施加平滑性约束对于学习鲁棒控制策略可能至关重要。



(a) 平均奖励　(a) Average reward

(b) 成功率　(b) Success rate

(c) 碰撞率　(c) Collision rate

(d) 离路率　(d) Off-road rate

图 11　不同B样条阶数对性能的影响分析图

Figure 11　Analysis of the impact of different B-spline orders on performance

为进一步客观陈述消融参数SplineOrder对实验性能的影响，本文依据四个性能指标，对实验序列的最后20个数据求取平均值来反映性能，结果如表4所示。实验结果表明：当SplineOrder=7时算法达到最优性能，平均回报高达230.42，成功率达到0.777，离路率仅为0.005，展现出最佳的导航精度和路径保持能力。相比之下，当SplineOrder=3时虽然成功率也较高（0.772），但离路率达到0.020，平均回报略低（226.691），说明较低阶样条函数的拟合能力相对不足。SplineOrder=5和10的表现均劣于3阶和7阶，其中5阶的成功率为0.535、离路率高达0.282，而10阶的成功率仅为0.595、离路率为0.202，这可能是因为过低阶数限制了函数表达能力，而过高阶数则引入了不必要的复杂度，导致训练难度增加和泛化性能下降。综合来看，适中偏高的样条阶数（SplineOrder=7）能够在函数逼近精度与模型稳定性之间取得最佳平衡。

表4 不同样条阶对实验性能的影响结果

Table 4 The impact of different spline orders on experimental performance

参数(SplineOrder)	平均回报	成功率	碰撞率	离路率
3	226.691	0.772	0.207	0.020
5	188.896	0.535	0.185	0.282
7	230.420	0.777	0.217	0.005
10	198.653	0.595	0.205	0.202

4 结论

本文研究了DRL不同模型架构组成对自动驾驶任务中的最优控制策略影响问题。首先，明确了面向连续状态和动作空间的PPO方法面临的挑战，以及通过向损失函数中添加HJB方程来提高模型安全性；其次，提出了应用于策略和值函数的混合架构方法MAHPO，有效地对策略网络和值函数网络通过不同的网络表征方法进行表示。实验结果表明：本文所提出的MAHPO方法在现有的基准算法SAC上的平均成功率提升了5.88%，离路率相比于原始算法HJBPPO下降了约78.22%，证明了本文所提方法的有效性。未来将进一步考虑混合架构算法的训练效率问题。

参考文献

[1] 刘全, 翟建伟, 章宗长, 等. 深度强化学习综述[J]. 计算机学报, 2018, 41(1): 1-27.
Liu Quan, Zhai Jianwei, Zhang Zongzhang, et al. A survey on deep reinforcement learning[J]. Chinese Journal of Computers, 2018, 41(1): 1-27. (in Chinese)

[2] François-Lavet V, Henderson P, Islam R, et al. An introduction to deep reinforcement learning[J]. Foundations and Trends in Machine Learning, 2018, 11(3/4): 219-354.

[3] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature, 2016, 529(7587): 484-489.

[4] Silver D, Hubert T, Schrittwieser J, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play[J]. Science, 2018, 362(6419): 1140-1144.

[5] Vinyals O, Babuschkin I, Czarnecki W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning[J]. Nature, 2019, 575(7782): 350-354.

[6] Fawzi A, Balog M, Huang A, et al. Discovering faster matrix multiplication algorithms with reinforcement learning[J]. Nature, 2022, 610(7930): 47-53.

[7] Singh B, Kumar R, Singh V P. Reinforcement learning in robotic applications: A comprehensive survey[J]. Artificial Intelligence Review, 2022, 55(2): 945-990.

[8] Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback[C]//Proceedings of the 36th International Conference on Neural Information Processing Systems. New York: Curran Associates Inc., 2022: 2011.

[9] 胡瑜洪, 王德光, 杨明, 等. 基于强化学习的离散事件系统最优定向监控[J]. 电子学报, 2024, 52(9): 3172-3184.
Hu Yuhong, Wang Deguang, Yang Ming, et al. Optimal directed control of discrete event systems based on reinforcement learning[J]. Acta Electronica Sinica, 2024, 52(9): 3172-3184. (in Chinese)

[10] 陈爽, 田烨, 付莹. 基于强化学习的免调参即插即用单光子图像重建方法[J]. 电子学报, 2024, 52(10): 3600-3612.
Chen Shuang, Tian Ye, Fu Ying. Reinforcement learning based tuning-free plug-and-play image reconstruction method for single photon imaging[J]. Acta Electronica Sinica, 2024, 52(10): 3600-3612. (in Chinese)

[11] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[PP/OL]. V2. arXiv (2017-08-28) [2025-10-21]. https://arxiv.org/abs/1707.06347.

[12] Liu Z M, Wang Y X, Vaidya S, et al. KAN: Kolmogorov-Arnold networks[C]//Proceedings of the Thirteenth International Conference on Learning Representations. Singapore: OpenReview.net, 2025: 70367-70413.

[13] Kiran B R, Sobh I, Talpaert V, et al. Deep reinforcement learning for autonomous driving: A survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(6): 4909-4926.

[14] 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.

[15] Kendall A, Hawke J, Janz D, et al., Learning to drive in a day[C]//Proceedings of the International Conference on Robotics and Automation (ICRA). Piscataway: IEEE, 2019: 8248-8254.
[16] You C X, Lu J B, Filev D, et al. Highway traffic modeling and decision making for autonomous vehicle using reinforcement learning[C]//Proceedings of the IEEE Intelligent Vehicles Symposium (IV). Piscataway: IEEE, 2018: 1227-1232.
[17] Mirchevska B, Pek C, Werling M, et al. High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning[C]//Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC). Piscataway: IEEE, 2018: 2156-2162.
[18] Da C, Qian Y S, Zeng J W, et al. ST-PPO: A spatio-temporal attention enhanced proximal policy optimization algorithm for autonomous driving in complex traffic scenarios[J]. Machine Learning, 2025, 114(11): 245.
[19] Zhang C Z, Dai L F, Zhang H, et al. Control barrier function-guided deep reinforcement learning for decision-making of autonomous vehicle at on-ramp merging[J]. IEEE Transactions on Intelligent Transportation Systems, 2025, 26(6): 8919-8932.
[20] Feng S, Sun H W, Yan X T, et al. Dense reinforcement learning for safety validation of autonomous vehicles[J]. Nature, 2023, 615(7953): 620-627.
[21] Schulman J, Moritz P, Levine S, et al. High-dimensional continuous control using generalized advantage estimation[PP/OL]. V6. arXiv (2018-10-20) [2025-10-21]. https://arxiv.org/abs/1506.02438.
[22] De Boor C. Package for calculating with B-splines[J]. SIAM Journal on Numerical Analysis, 1977, 14(3): 441-472.
[23] Li Q Y, Peng Z H, Feng L, et al. MetaDrive: Composing diverse driving scenarios for generalizable reinforcement learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(3): 3461-3475.
[24] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//Proceedings of the 35th International Conference on Machine Learning. Stockholm: PMLR, 2018: 1861-1870.
[25] Mukherjee A, Liu Jun. Bridging physics-informed neural networks with reinforcement learning: Hamilton-Jacobi-bellman proximal policy optimization (HJBPPO)[C]//Proceedings of the Workshop on New Frontiers in Learning, Control, and Dynamical Systems at the International Conference on Machine Learning. Honolulu: PMLR, 2023.
[26] Tsitsiklis J N, Van Roy B. Feature-based methods for large scale dynamic programming[C]//Proceedings of 1995 34th IEEE Conference on Decision and Control. Piscataway: IEEE, 1995: 565-567.

作者简介

王金强　男,1993年4月出生于甘肃省定西市。现为兰州大学核科学与技术学院萃英博士后。主要研究方向为深度强化学习、AI4Science和机器人。

E-mail: jqwang16@lzu.edu.cn

雍宾宾　男,1988年11月出生于河南省商丘市。现为兰州大学信息科学与工程学院副教授,硕士生导师。主要研究方向为深度学习、并行计算和自动驾驶。

E-mail: yongbb@lzu.edu.cn

宋利蓉　女,2000年10月出生于青海省海东市。现为兰州大学信息科学与工程学院硕士研究生。主要研究方向为深度强化学习、自动驾驶。

E-mail: songlr2023@lzu.edu.cn

李　妍　女,1976年10月出生于甘肃省武威市。现为兰州大学信息科学与工程学院副教授,硕士生导师。主要研究方向为自然语言处理、深度强化学习。

E-mail: liyan_2007@lzu.edu.cn

蒋远博　男,1999年7月出生于河南省平顶山市。现为兰州大学信息科学与工程学院博士研究生。主要研究方向为深度强化学习、自动驾驶。

E-mail: jyuanbo2025@lzu.edu.cn

周庆国　男,1973年10月出生于福建省三明市。现为兰州大学信息科学与工程学院教授,博士生导师。主要研究方向为嵌入式系统、网络安全、具身智能。

E-mail: zhouqg@lzu.edu.cn