基于改进 TD3 算法的移动机器人路径规划

李明明，潘子豪

（西安科技大学通信与信息工程学院，陕西 西安 710600）

摘　要： 传统的移动机器人路径规划算法通常需要在有地图的条件下才能有效规划路径。相比之下基于深度强化学习（DRL）的路径规划因其在无地图条件下的导航能力而备受关注。然而，传统的 DRL 路径规划算法往往存在样本利用率低、训练速度慢、泛化能力不足等问题。针对上述问题，对双延迟深度确定性（TD3）策略梯度算法进行改进以提高其在移动机器人路径规划中的性能。首先，针对 TD3 算法持续探索空间能力有限的问题，对其探索策略进行改进，通过使用具有时间相关性的粉红噪声来增强算法的持续探索空间能力。其次，结合 n 步方法和损失调整近似 Actor 优先（LA3P）经验回放方法，n 步方法将经验回放池中的即时奖励扩展为 n 步的累计折扣奖励，能够更准确地捕捉长期奖励信号，而 LA3P 方法通过对 n 步经验的高效利用，提高样本利用率和算法的性能。最后，通过在 Gazebo 中搭建了 3 个不同的环境进行实验，并和多种算法进行比较。实验结果表明，改进算法在训练时间、平均成功率、平均距离等方面更具优势，证明了改进算法的有效性。

关键词： 路径规划；深度强化学习；双延迟深度确定性策略梯度；粉红噪声；损失调整近似 Actor 优先经验回放

中图分类号： TP242.6　　**文献标志码：** A　　**DOI：** 10.19678/j.issn.1000-3428.0070256

Mobile Robot Path Planning Based on Improved TD3 Algorithm

LI Mingming, PAN Zihao

(College of Communication and Information Engineering, Xi′an University of Science and Technology, Xi′an 710600, Shaanxi, China)

【Abstract】 The traditional path-planning algorithm of a mobile robot usually requires a map to effectively plan the path. By contrast, path planning based on Deep Reinforcement Learning (DRL) does not require maps for navigation, owing to which it has received considerable attention. However, traditional DRL path-planning algorithms often face challenges such as low sample utilization, slow training speed, and insufficient generalization ability. To solve these issues, the Twin Delayed Deep Deterministic (TD3) policy gradient algorithm is improved to enhance its path-planning performance for mobile robots. First, to solve the problem of the TD3 algorithm having limited ability for continuous space exploration, the exploration strategy is improved and the continuous space exploration ability of the algorithm is enhanced using pink noise with time correlation. Second, the n-step method is combined with the Loss-Adjusted Approximate Actor Prioritized (LA3P) experience replay method. The n-step method can capture the long-term reward signal more accurately by expanding the immediate reward in the experience replay buffer to the n-step cumulative discount reward, whereas the LA3P method can improve the sample utilization and performance of the algorithm by efficiently using the n-step experience. Finally, three different environments are built in Gazebo for the experiments and compared with various algorithms. The experimental results show that the improved algorithm has more advantages in terms of training time, average success rate, and average distance, which proves the effectiveness of the improved algorithm.

【Key words】 path planning; Deep Reinforcement Learning (DRL); Twin Delayed Deep Deterministic (TD3) policy gradient; pink noise; Loss-Adjusted Approximate Actor Prioritized (LA3P) experience replay

0　引言

路径规划是移动机器人的一个重要研究领域，它是移动机器人完成其他任务的基础前提。常见的路径规划算法有 A*算法[1]、人工势场法[2]、遗传算法[3]和蚁群算法[4]等。在这些方法中，环境地图的构建和维护是昂贵的，对激光传感器的精度也有依赖，且容易受传感器噪声的影响[5]。近年来，基于深度强化学习（DRL）的路径规划算法的研究受到研究人员的关注。DRL 是一种直接和环境进行交互，不断试错，根据环境给出的反馈来不断优化自身策略的机器学习方法。这种基于学习的方法，在一定程度上可以直接从原始数据中学习，不需要构建先验的环境地图[6]，同时降低了对高精度传感器的要

基金项目： 国家自然科学基金（62401459）。

作者简介： 李明明，女，副教授、硕士，主研方向为机器人技术；潘子豪，硕士研究生。

收稿日期： 2024-08-14　　**修回日期：** 2024-10-19　**E-mail：** pzh1162024@163.com

求，因此基于 DRL 的路径规划算法得到了广泛的关注。传统的强化学习(RL)方法无法有效处理高维问题。深度学习(DL)为 RL 解决维数灾难问题提供了有效方案，在 DL 中的神经网络拥有强大的特征提取能力，RL 拥有强大的决策能力[7]，两者的结合可以使移动机器人直接端到端地输出动作，从而实现路径规划。

LILLICRAP 等[8]提出了深度确定性策略梯度(DDPG)算法，它是在确定性策略梯度(DPG)[9]算法的基础上提出来的。为了缓解 DDPG 算法存在的高估偏差，FUJIMOTO 等[10]提出了双延迟深度确定性(TD3)策略梯度算法。然而，将这些传统的深度强化学习方法直接应用到移动机器人路径规划中，往往存在局部最优性、样本利用率低、训练速度慢和泛化能力不足等问题。

CIMURS 等[11]以 TD3 算法为基础，通过从环境中获取可能的导航方向的兴趣点并选择最佳航路点，缓解了局部最优问题。HU 等[12]在 D3QN 中加入了优先经验回放(PER)[13]，并将其用于移动机器人自主导航，以提高样本利用率。MARCHESINI 等[14]提出了一种改进的 PER，并将其命名为多批次 PER，他们引入了 3 个不同的批次，取得了比 PER 更好的性能。SAGLAM 等[15]详细分析了 PER 用于连续动作空间的离线策略演员-评论家(Actor-Critic)方法中导致算法性能下降的原因，提出了一种损失调整近似 Actor 优先(LA3P)经验回放方法，提高了基础算法的性能。ZHANG 等[16]在 Double DQN(DDQN)[17]算法中引入 n 步方法，提高了训练过程中对目标 Q 值估计的准确性。YANG 等[18]在 Soft Actor-Critic(SAC)[19]中考虑了 n 步方法，使机器人能够进行更长远的规划并具有灵活调整策略的能力。CHOWDHURY 等[20]改进了 TD3 算法的探索策略，提出一种熵最大化 TD3 算法，通过基于熵的随机 Actor 保证初始阶段的充分探索。YIN 等[21]提出了一种基于奖励的 ε-greedy 探索策略，并提出了一种 ε 计算公式，该策略提高了算法的性能，使得机器人在选择动作时更加智能。LIU 等[22]在 SAC 算法中加入了具有时间相关的 Ornstein-Uhlenbeck (OU) 噪声，以提高智能体持续探索空间的能力。EBERHARD 等[23]对有色噪声作为深度强化学习中连续控制任务的动作噪声进行了全面的实验评估，并推荐粉红噪声作为连续控制任务中的动作噪声。

本文对 TD3 算法进行改进，引入时间相关性的粉红噪声来提高移动机器人的探索能力，将 n 步方法和 LA3P 方法相结合来准确捕捉长期奖励信号，提高样本利用率，增强移动机器人在实际任务中的表现能力。

1 改进 TD3 算法

本节首先描述 TD3 算法，然后对改进 TD3 算法的各模块进行介绍，包括粉红噪声、n 步方法和 LA3P 方法，最终得到本文改进的 TD3 算法。

1.1 TD3 算法

TD3 算法对 DDPG 算法进行了改进。它通过选择 2 个目标价值网络较小的一方来计算目标值，从而减小高估偏差[10]，如式(1)所示：

$$y_{t:t+1}=r_{t+1}+\gamma \min_{i=1,2} Q_{\theta_i'}(s_{t+1},\tilde{a}_{t+1}) \tag{1}$$

式中：r_{t+1} 是在状态 s_t 下执行动作 a_t 所获得的奖励；γ 为折扣因子；θ_i' 为目标价值网络的参数；s_{t+1}、$\tilde{a}_{t+1}$ 分别为下一时刻的状态和动作。

1.2 改进 TD3 探索策略

在 DRL 的连续控制任务中最简单且常用的探索方式就是加入动作噪声，通过这种方式促进智能体的探索，从而发现有用的信息，如 TD3 算法中独立采样的高斯噪声。

如果从某个随机过程提取的信号 $\varepsilon(t)$ 满足下面的性质：

$$|\hat{\varepsilon}(f)|^2 \propto f^{-\beta} \tag{2}$$

则该随机过程称为具有颜色参数 β 的有色噪声。其中，f 为频率，$\hat{\varepsilon}(f)$ 为信号 $\varepsilon(t)$ 的傅里叶变换，$|\hat{\varepsilon}(f)|^2$ 称为功率谱密度(PSD)[23]。当 $\beta=1$ 时，有色噪声就变为了粉红噪声。

图 1 所示为生成独立高斯噪声和粉红噪声的采样图(彩色效果见《计算机工程》官网 HTML 版，下同)。从图中可以看出，独立高斯噪声没有时间相关性，每个时间点上的噪声值都是相互独立的，而粉红噪声则是一种时间相关的噪声，时间序列中的噪声值在不同时间点上具有一定的相关性。此外独立高斯噪声的噪声值大多集中在均值(通常为 0)附近，而在远离均值的地方出现概率较低，因此在 TD3 算法输出的确定性动作中加入独立高斯噪声，会导致探索主要集中在输出动作附近，对于远离输出动作的区域探索不足[24]。粉红噪声不会大量集中在均值附近，而是在远离均值的地方也有较多的分布，从而对于偏远的动作也能较好地探索到。

独立高斯噪声的 PSD 是一个常数，这意味着其在频域上没有明显的集中，表现为随机变化。根据式(2)，粉红噪声的 PSD 在低频上较大，在高频上较小，这意味着粉红噪声的变化趋势较为缓慢，波动相

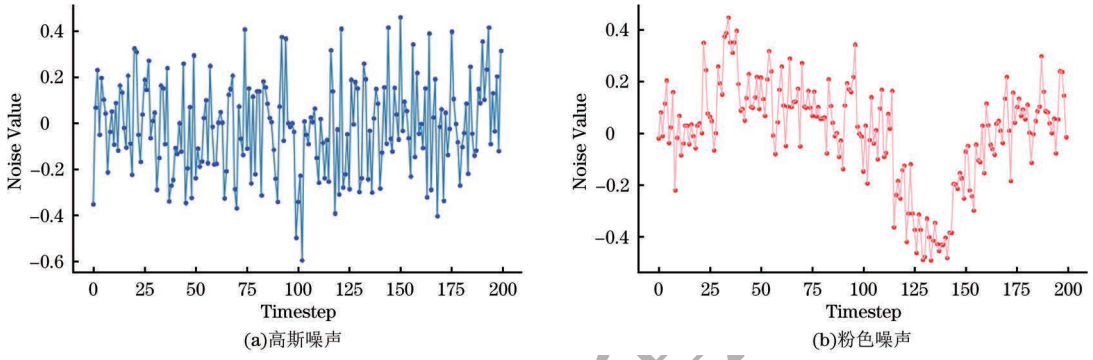


图 1 不同噪声的采样图

Fig. 1 Sampling graphs of different noises

对连续和平滑。在连续控制问题中，系统的动作通常是连续且平滑的，因此粉红噪声相比独立高斯噪声能够减少频繁的剧烈波动，较少出现前后噪声值相差过大的情况，从而提供更平滑的探索。

1.3 *n* 步 TD3 算法

原始的 TD3 算法只考虑了一步即时奖励 r_{t+1} 来计算目标，如式(1)所示。本节将考虑 TD3 算法的 n 步形式。1 步方法和蒙特卡洛(MC)是 n 步方法的 2 个极端[25]。

1 步目标可以用式(3)表示，它只包含一个时间步的奖励。

$$y_{t:t+1}=r_{t+1}+\gamma Q(s_{t+1},a_{t+1}) \tag{3}$$

MC 是基于完整回合的奖励来计算折扣回报，如式(4)所示，它依赖于完整回合中的所有时间步的奖励。

$$G_t=r_{t+1}+\gamma r_{t+2}+\gamma^2 r_{t+3}+\cdots+\gamma^{T-t-1}r_T \tag{4}$$

通常适当调整 n 值的大小可以加快学习速度[25]，n 步目标可以用式(5)表示，它包含 n 个时间步的折扣奖励。

$$y_{t:t+n}=r_{t+1}+\gamma r_{t+2}+\cdots+\gamma^{n-1}r_{t+n}+\gamma^n Q(s_{t+n},a_{t+n}) \tag{5}$$

于是 n 步 TD3 算法的式(1)变为式(6)：

$$y_{t:t+n}=r_{t+1}+\cdots+\gamma^{n-1}r_{t+n}+\gamma^n \min_{i=1,2} Q_{\theta_i'}(s_{t+n},\bar{a}_{t+n}) \tag{6}$$

1.4 LA3P 算法

经验回放的应用打破了训练样本之间的相关性，通过重复利用过去的经验从而提高了样本利用率，这些样本被以相同的概率采样，但是实际上像 TD 误差更大的样本可能对于算法的训练是很重要的，它们应该以更大的概率采样，于是针对这样的问题，PER 被提出。PER 在 DQN 等算法中取得了成功[13,26]，它以更高的概率采样 TD 误差较大的经验，加快了算法训练速度，提高了算法性能。然而在连续动作空间离线策略的 Actor-Critic 方法中，PER 却降低了算法性能[15,27]。FUJIMOTO 等[27]认为当 PER 和 MSE 损失函数结合使用时，PER 在估计 TD 目标的期望值时可能会偏向异常值，而不是学习平均值，从而对算法性能造成影响，因此提出了 LAP(Loss-Adjusted Prioritized)函数，如式(7)所示：

$$p(i)=\frac{\max(|\delta(i)|^{\alpha},1)}{\sum_j \max(|\delta(j)|^{\alpha},1)},\quad L_{\text{Huber}}(\delta(i))=\begin{cases}0.5\delta(i)^2, & |\delta(i)|\leqslant 1\\ |\delta(i)|, & \text{others}\end{cases} \tag{7}$$

式中：$p(i)$是第 i 个经验样本的优先级；$|\delta(i)|$是 TD 误差的绝对值；α 决定优先级的使用程度。当 $|\delta(i)|\leqslant 1$ 时，损失函数 Huber 变成了 MSE，此时应该对经验样本均匀采样，以防止 PER 和 MSE 结合导致偏差；当$|\delta(i)|>1$ 时，根据优先级非均匀采样，进一步消除 PER 算法中偏向于异常样本的偏差[15,27]。

此外，非均匀采样数据上计算的损失函数都可以转换为另一个具有相同期望梯度的均匀采样损失函数，并推导出了 LAP 的镜像损失函数 PAL(Prioritized Approximation Loss)，该函数应该与均匀采样相结合，以消除均匀采样中异常值的偏差[15,27]，如式(8)所示：

$$\lambda=\frac{\sum_j \max(|\delta(j)|^{\alpha},1)}{N},\quad L_{\text{PAL}}(\delta(i))=\frac{1}{\lambda}\begin{cases}0.5\delta(i)^2, & |\delta(i)|\leqslant 1\\ \dfrac{|\delta(i)|^{1+\alpha}}{1+\alpha}, & \text{others}\end{cases} \tag{8}$$

式中：N 为批次大小。

SAGLAM 等[15]从理论上分析了 PER 导致连续动作空间离线策略 Actor-Critic 方法性能变差的原因，提出了一种改进的 LA3P 方法，主要内容如下：

1)如果使用具有较大 TD 误差的经验样本对 Actor 网络进行优化,至少对于当前或后续样本的近似策略梯度可能会偏离实际梯度。建议使用低 TD 误差的经验样本训练 Actor 来克服这个问题,对于低 TD 误差的经验样本,通过从优先级经验回放池中使用逆采样来实现,如式(9)所示:

$$I\sim\bar{p}(i)=\frac{p_{\max}}{p(i)}=\max_i\left(\frac{\max(|\delta(i)|^{\alpha},1)}{\sum_j\max(|\delta(j)|^{\alpha},1)}\right)\cdot\frac{\sum_j\max(|\delta(j)|^{\alpha},1)}{\max(|\delta(i)|^{\alpha},1)} \tag{9}$$

式中:$p_{\max}$ 是先前存储的优先级的最大值。高 TD 误差的经验样本来训练 Critic,并结合 LAP 函数。采样的经验样本大小为$(1-\omega)\cdot N$,其中,$\omega\in[0,1]$表示均匀采样样本的比例。

2)Actor 和 Critic 的经验样本分别使用逆优先级和优先级进行采样,它们可能不会看到相同的样本,从而违反了 Actor-Critic 理论[28],于是结合 PAL 函数,使用 $\omega\cdot N$ 大小的批次进行均匀采样。

2 关键元素设计

状态空间、动作空间和奖励函数是 DRL 的关键元素。这一节对本文的这 3 个要素进行介绍,同时对网络结构进行设计,用于对机器人收集到的环境信息进行处理。

2.1 状态空间

状态空间是机器人感知环境的基础,决定了机器人能够获取到的环境信息,因此状态空间应该包含足够的关键信息,本文的状态空间由以下 3 个部分组成:

1)激光雷达数据。本文没有考虑机器人的后退,因此将激光雷达扫描范围限制在$-90°\sim90°$,同时考虑到原始数据量较大等因素,总共选取了 20 个激光雷达状态,分别记为 $L_1,L_2,\cdots,L_{20}$,激光雷达的测量距离限制在 0.12~3.5 m。

2)目标点信息。目标点信息是状态空间的重要组成部分。目标点信息由机器人到目标点的欧氏距离 d 和机器人航向与目标点的夹角 β 组成。

3)机器人速度信息。机器人速度信息由上一时刻的线速度 ν_{t-1} 和角速度 ω_{t-1} 组成。

综上,最终机器人的状态空间是由 24 个特征的 $S=[L_1,L_2,\cdots,L_{20},d,\beta,\nu_{t-1},\omega_{t-1}]$组成。

2.2 动作空间

动作空间决定了机器人可采取行动的集合。本文的动作空间 $a=[\nu_t,\omega_t]$,它由机器人线速度 ν_t 和角速度 ω_t 组成,由于只考虑了机器人前方的激光雷达数据,因此将线速度 ν_t 限制在 0~0.5 m/s,角速度 ω_t 限制在$-1\sim1$ rad/s。

2.3 奖励函数

强化学习的目标是最大化所获得的奖励[25]。本文的奖励函数由 4 个部分组成:第 1 个部分是目标奖励,即机器人到达目标所获得的即时奖励,它告诉机器人全局的主要任务;第 2 个部分是碰撞惩罚,即机器人碰撞到障碍物所得到的惩罚,它告诉机器人应该避免碰到障碍物这种行为;第 3 个部分是最大步数惩罚,即机器人在一回合中达到最大步数时的惩罚,它告诉机器人在一回合训练中应该尽量避免达到最大步数这种行为;第 4 个部分是其他奖励,这部分是缓解奖励稀疏问题的关键。综合上面的考虑,本文的奖励函数 r_t 如下:

$$r_t=\begin{cases}100, & d_t<0.15\ \text{m}\\ -200, & d_{\text{laser}}<0.15\ \text{m}\\ -200, & s_{\text{steps}}==400\\ 20\cdot(d_{t-1}-d_t)-20\cdot r_d, & \text{others}\end{cases}$$

$$r_d=\begin{cases}0.3-d_{\text{laser}}, & d_{\text{laser}}<0.3\ \text{m}\\ 0, & \text{others}\end{cases} \tag{10}$$

式中:d_{t-1} 表示上一时刻机器人到目标点的欧氏距离;d_t 表示当前时刻机器人到目标点的欧氏距离;d_{laser} 表示激光雷达测得的离障碍物的最小距离;s_{steps} 表示机器人当前已走的步数;r_d 的作用是让机器人和障碍物保持一定的安全距离。

2.4 网络结构

TD3 算法是一种基于 Actor-Critic 框架的 DRL 算法,它由 Actor 和 Critic 组成,同时结合了价值学习和策略学习的优点。演员根据状态采取行动,评论家则对演员所采取动作的质量进行评估,演员根据评论家的评价来更新自己的策略,以便能获得评论家更高的评价。本文算法网络结构如图 2 所示。

对于 Actor 网络来说,输入的是第 2.1 节中的状态空间,它经过 2 个包含 256 个神经元的全连接层,每层后面使用 ReLU 激活函数,最后经过一个包含 2 个神经元的全连接层,并使用 Tanh 激活函数输出动作 $a=[\nu,\omega]$,它包含线速度和角速度。

对于 Critic 网络来说,输入的是第 2.1 节状态空间和第 2.2 节动作空间的组合,它经过两个包含 256 个神经元的全连接层,每层后面使用 ReLU 激活函数,最后经过线性激活函数输出 Q 值。

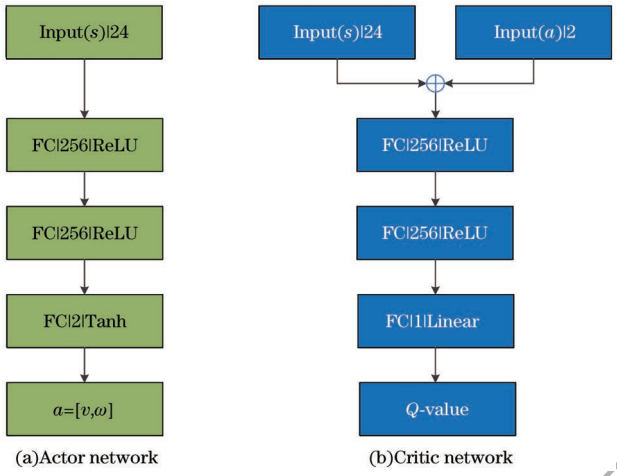


图 2 本文算法网络结构

Fig. 2 Network architecture of proposed algorithm

3 实验流程

本文算法的实验流程如图 3 所示，具体步骤为：

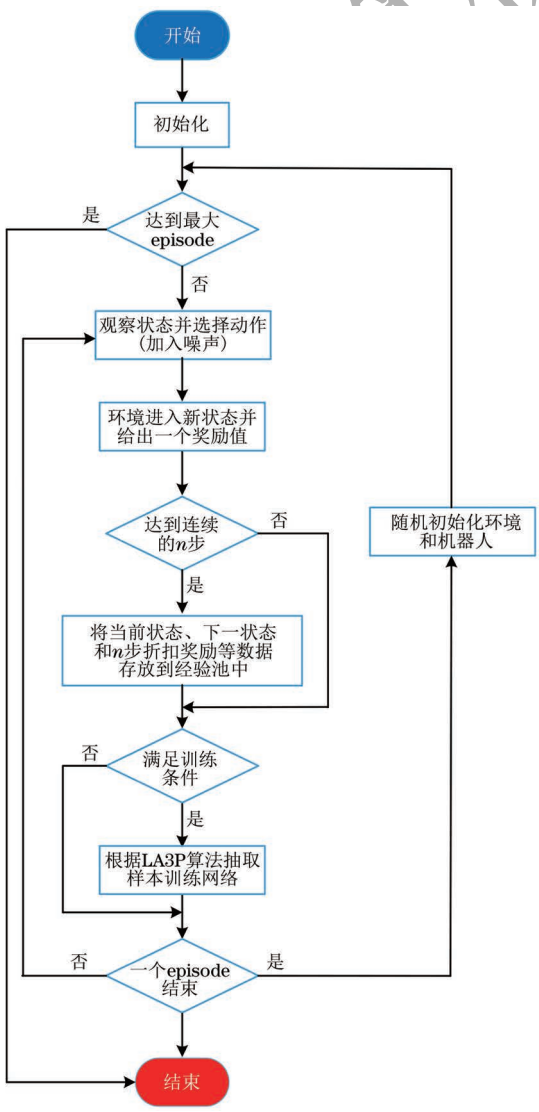


图 3 本文算法实验流程

Fig. 3 Experimental procedure of proposed algorithm

1）首先初始化 Actor 网络、Critic 网络和目标网络等。

2）进入算法的主循环，判断是否达到了最大 episode。若是，则进入第 7）步；否则，进入下一步。

3）机器人观察当前状态并选择动作，环境进入新的状态并给予机器人一个奖励值。

4）判断是否已经执行了连续的 n 步。若是，则将当前状态、动作、n 步折扣奖励和下一状态等存储到经验池中，然后进入下一步；否则，直接进入下一步。

5）判断是否需要开始训练，若满足训练条件，则按照第 1.4 节中的 LA3P 算法抽取样本并对网络进行训练，然后进入下一步；若不满足，则直接进入下一步。

6）判断当前 episode 是否已经结束，结束标志包括机器人到达目标点、碰撞障碍物和达到最大步数。若是，则随机初始化环境和机器人，并进入第 2）步；否则进入第 3）步。

7）训练结束。

4 训练和结果分析

本文实验是在 ROS 中进行，使用 Python 3.8.10 和 PyTorch 1.10 实现算法，使用 Gazebo 搭建训练和测试场景。实验电脑配置为：CPU i7-12700H，GPU RTX 3060 6 GB，RAM 16 GB。训练的参数如表 1 所示。

表 1 训练参数

Table 1 Training parameters

参数	参数值
经验池大小	5×10^5
批次大小	64
最大 episode	2 000
优化器	Adam
学习率	0.001
α	0.4
ω	0.5
γ	0.99
软更新率 τ	0.005
策略噪声 σ	0.2
策略噪声截断值 c	0.5

4.1 训练环境搭建

本文搭建的训练环境如图 4 所示，环境大小为 10 m×10 m。其中障碍物类型有多种，包括矩形墙壁、三角形墙壁、十字形墙壁和一个圆柱。为了增加仿真环境的多样性，另外添加了 4 个随机分布的小

箱子，这些小箱子不会分布在任何障碍物上面。此外机器人和目标点的位置随机分布但不会分布在任何障碍物上面。

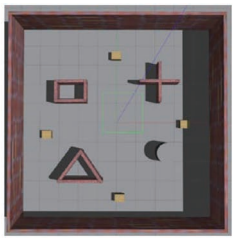


图 4 训练环境

Fig. 4 Training environment

4.2 粉红噪声实验

本节对第 1.2 节的内容进行了实验，并把加入粉红噪声的 TD3 算法记为 TD3PN，结果如图 5 所示。从图中可以看出，在训练过程中，绝大多数情况下 TD3PN 算法的奖励都高于 TD3 算法。并且，TD3PN 算法在大约在第 1 400 episode 之后逐渐趋于稳定，而 TD3 算法在大约在第 1 720 episode 之后才逐渐趋于稳定，这在一定程度上说明粉红噪声有利于机器人的探索。

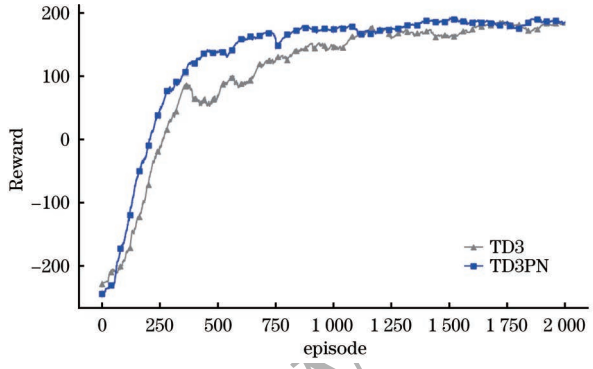


图 5 TD3 算法和 TD3PN 算法的奖励曲线

Fig. 5 Reward curves for TD3 algorithm and TD3PN algorithm

图 6 绘制了两种算法在训练过程中的步数变化曲线。从图中可以看出，在训练过程中两种算法的步数都是先上升后逐渐下降，说明算法逐渐学会了导航到目标点的策略。最后，两种算法都稳定在 125 episode 左右，但 TD3PN 算法的稳定速度更快。

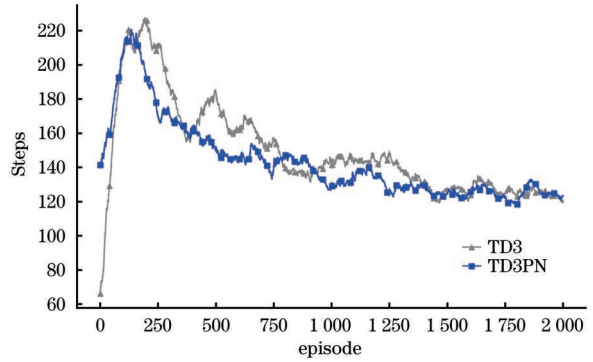


图 6 TD3 算法和 TD3PN 算法的步数曲线

Fig. 6 Steps curves for TD3 algorithm and TD3PN algorithm

两种算法训练的总步数和总时间如表 2 所示。从表中可以看出，TD3 算法训练的总步数为 303 553 步，训练总时间为 16.45 h，而 TD3PN 算法训练的总步数为 284 583 步，训练总时间为 14.18 h，表现更优。

表 2 TD3 算法和 TD3PN 算法的总训练步数和总训练时间

Table 2 Total training steps and total training time for TD3 algorithm and TD3PN algorithm

算法	总训练步数/步	总训练时间/h
TD3	303 553	16.45
TD3PN	**284 583**	**14.18**

4.3 n 值的选取

在这部分，将对 n-step 方法中的 n 值进行选取，考虑的 n 值包括 1、3、5、7 和 9。绘制了不同 n 值的奖励曲线，如图 7 所示。

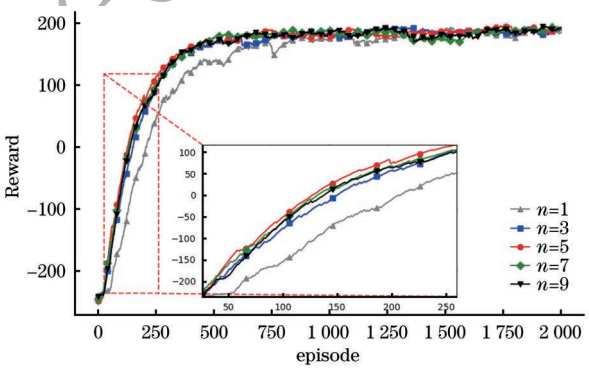


图 7 不同 n 值的奖励曲线

Fig. 7 Reward curves for different n values

从图中可以看出，除了 $n=1$ 的曲线外，其他几个 n 值的曲线之间差别不大，但 $n=5$ 的曲线在第 50～250 episode 之间的上升速度更快。最后，$n=3$ 的曲线大约在第 780 episode 后逐渐趋于稳定，$n=5$ 的曲线大约在第 720 episode 后逐渐趋于稳定，$n=7$ 的曲线大约在第 820 episode 后逐渐趋于稳定，$n=9$ 的曲线大约在第 710 episode 后逐渐趋于稳定。

图 8 所示为记录了训练过程中不同 n 值的步数变化。所有算法的步数都在逐渐减少，这表明算法逐渐学会了导航到目标点的能力。与其他 n 值相比，$n=5$ 的曲线在大多数 episode 下步数较少。除 $n=1$ 曲线外，所有曲线的步数最终都稳定在 120 左右。

在表 3 中记录了不同 n 值的训练总步数和总时间，从中可以看到，当 $n=5$ 时，算法的总训练步数和总时间都是最少的。当 $n=9$ 时，算法的总训练

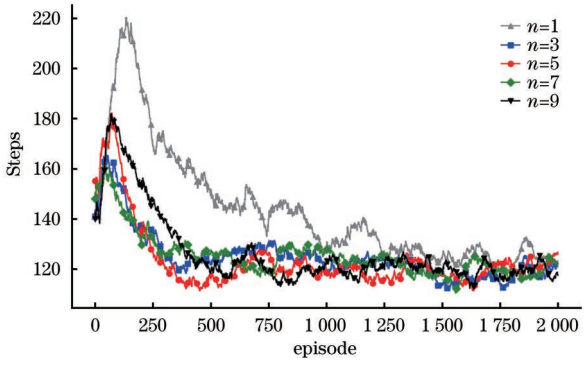


图 8 不同 n 值的步数曲线

Fig. 8 Steps curves for different n values

步数和总时间略高于 $n=3$ 和 $n=7$。当 $n=1$ 时，算法表现最差。

表 3 不同 n 值的总训练步数和总训练时间

Table 3 Total training steps and total training time for different n values

n	总训练步数/步	总训练时间/h
1	284 583	14.18
3	248 062	11.87
5	**244 294**	**11.63**
7	248 074	11.89
9	250 561	12.07

综上，经过上述的实验分析，当 $n=5$ 时，算法的整体性能更好，因此选择 $n=5$，并将本节的算法命名为 5-TD3PN。

4.4 提出的方法

本节引入第 1.4 节中的 LA3P 算法，形成本文最终的改进算法，命名为 5-TD3PN-LA3P。此外，还与 TD3-PER[10,13]、SAC[19]、DDPG[8] 等算法进行了对比实验。图 9 所示为不同算法的奖励曲线。

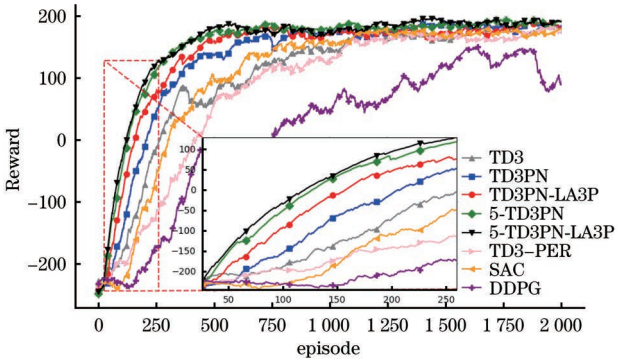


图 9 不同算法的奖励曲线

Fig. 9 Reward curves for different algorithms

从图中可以看出，本文算法在第 50～250 episode 之间比其他算法增长更快，大约在第 570 episode 后逐渐稳定下来。5-TD3PN 算法大约在第 720 episode 后逐渐稳定下来。TD3PN 算法大约在第 1 400 episode 后逐渐趋于稳定。TD3PN-LA3P 算法大约在第 750 episode 后逐渐趋于稳定。TD3-PER 算法大约在第 1 930 episode 后逐渐趋于稳定，比 TD3 算法的 1 720 episode 要慢，并且在整个过程中总体表现不如 TD3 算法。SAC 算法大约在第 1 220 episode 后逐渐趋于稳定。而 DDPG 算法在整个训练过程中无法收敛到一个稳定的值。

图 10 所示为不同算法在训练过程中步数的变化情况。从图中可以看到，本文算法在一开始就保持了较低的步数，这与其他算法相比更具优势。TD3-PER 算法在整个训练过程中几乎始终保持比 TD3 算法更高的步数，最终稳定在 130 步左右。SAC 算法在一开始就保持着比 TD3 算法更高的步数，但随后下降较快，稳定的步数也低于 TD3-PER 算法。而 DDPG 算法则一直保持着较高的步数，最终稳定在 170 步左右，表现差于其他算法。

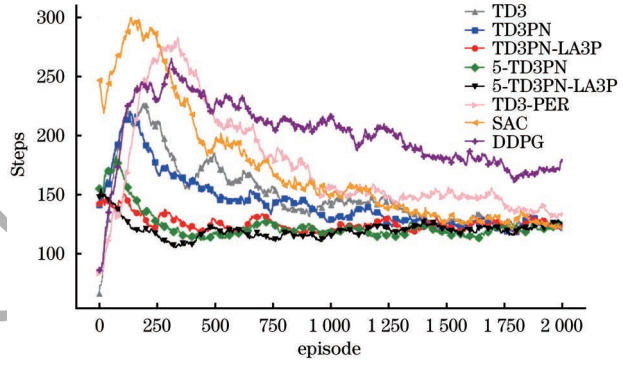


图 10 不同算法的步数曲线

Fig. 10 Steps curves for different algorithms

表 4 所示为不同算法的总训练步数和总训练时间，从表中可以看出，本文算法的总训练步数和总训练时间最少。TD3-PER 算法由于增加了 PER，训练步数和训练时间有所增加。TD3PN-LA3P 算法的训练步数和训练时间低于 TD3PN，但高于 5-TD3PN。SAC 算法的训练步数和训练时间高于 TD3 算法，但低于 TD3-PER 算法。另一方面，DDPG 算法的性能最差。

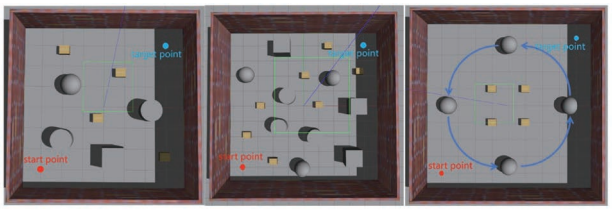
表 4 不同算法的总训练步数和总训练时间

Table 4 Total training steps and total training time for different algorithms

算法	总训练步数/步	总训练时间/h
TD3	303 553	16.45
TD3PN	284 583	14.18
TD3PN-LA3P	249 021	12.22
5-TD3PN	244 294	11.63
5-TD3PN-LA3P	**237 472**	**11.35**
TD3-PER	352 745	18.18
SAC	332 394	17.42
DDPG	411 371	21.56

5 测试和结果分析

本节将测试本文算法在未知环境中的泛化能力，同时将其与第 4.4 节中的其他算法进行比较。首先，在 Gazebo 中创建了 3 个测试环境，如图 11 所示。环境 a 的大小为 8 m×8 m，机器人起始点在(−3，−3)，如图中红点，目标点在(3，3)，如图中蓝点。环境 b 的大小为 10 m×10 m，机器人起始点在(−4，−4)，如图中红点，目标点在(4，4)，如图中蓝点。环境 c 是一个动态环境，其大小也是 10 m×10 m，机器人起始位置在(−4，−4)，如图中红点，目标点在(4，4)，如图中蓝点。对于图中的 4 个动态小球通过编写 gazebo 插件来控制它们的运动，4 个小球初始的 x、y 坐标分别为(3.5，0)、(0，3.5)、(−3.5，0)和(0，−3.5)，它们以 0.2 rad/s的角速度逆时针旋转。



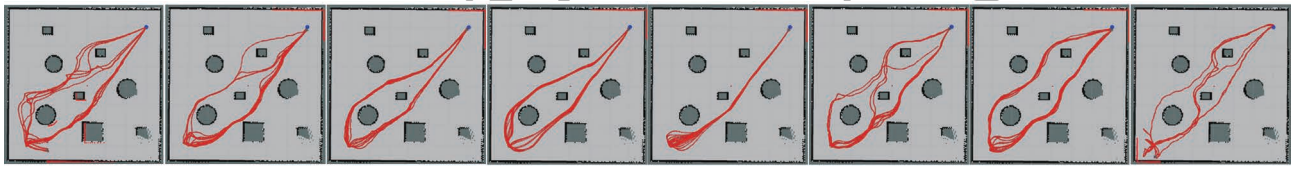
(a)静态环境1 (b)静态环境2 (c)动态环境

图 11 测试环境

Fig. 11 Test environments

在这 3 个场景中对每个算法测试 20 轮。

各算法在环境 a 中的测试结果如图 12 所示。为了能够更好地展示机器人的轨迹，本文采用了 Gmapping 算法进行建图，地图仅用于显示效果，没有其他任何作用。从图中可以看出，本文算法选取的路径相对较好。TD3 算法、TD3-PER 算法和 DDPG 算法的路径相对曲折。



(a)TD3 (b)TD3PN (c)TD3PN-LA3P (d)5-TD3PN (e)5-TD3PN-LA3P (f)TD3-PER (g)SAC (h)DDPG

图 12 各算法在环境 a 中的轨迹

Fig. 12 The trajectory of each algorithm in environment a

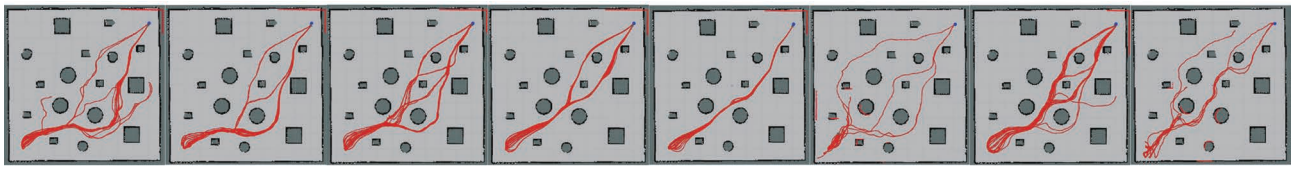
表 5 所示为每种算法的测试结果。从表中可以看出，本文算法在平均距离、平均时间方面都是最优的。TD3PN 算法的测试结果也优于 TD3 算法，这说明了添加粉红噪声的有效性。TD3PN-LA3P 算法的平均距离和平均时间优于 TD3PN，但平均成功率略低于 TD3PN。TD3-PER 算法的平均成功率低于 TD3 算法，但平均距离和平均时间都优于 TD3 算法。SAC 算法的整体表现优于 TD3 算法和 TD3-PER 算法，但平均时间高于 TD3-PER 算法。而 DDPG 算法表现最差。

各算法在环境 b 中的轨迹如图 13 所示。从图中可以看出，本文算法选择的路径相对较好，比较平滑。而 TD3 算法、TD3-PER 算法和 DDPG 算法选择的路径则较为曲折。

表 5 各算法在环境 a 中的测试结果

Table 5 Test results of each algorithms in environment a

算法	平均成功率/%	平均距离/m	平均时间/s
TD3	70	9.29	25.91
TD3PN	100	9.05	21.51
TD3PN-LA3P	95	9.03	20.60
5-TD3PN	100	9.00	20.22
5-TD3PN-LA3P	100	**8.77**	**20.13**
TD3-PER	60	9.25	23.13
SAC	100	9.21	23.51
DDPG	30	9.87	29.42



(a)TD3 (b)TD3PN (c)TD3PN-LA3P (d)5-TD3PN (e)5-TD3PN-LA3P (f)TD3-PER (g)SAC (h)DDPG

图 13 各算法在环境 b 中的轨迹

Fig. 13 The trajectory of each algorithms in environment b

每种算法的测试结果记录在表 6 中，可以看出，本文算法总体性能最佳。TD3PN-LA3P 算法的平均距离和平均时间优于 TD3PN 算法，但成功率略低于 TD3PN 算法。TD3PN 和 TD3 算法的对比结果表明，粉红噪声对 TD3 算法的性能有一定的改善。5-TD3PN 和 TD3PN 算法的比较结果表明，多

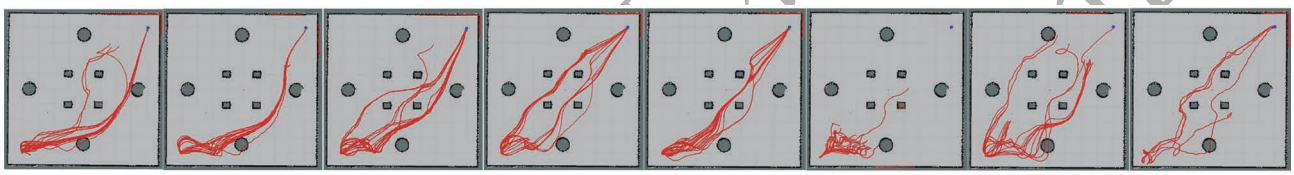
步骤方法也有利于提高算法性能。与 TD3 算法相比，TD3-PER 算法的性能更差，平均成功率只有 15%。DDPG 算法虽然平均距离和平均时间都优于 TD3 算法，但平均成功率却远低于 TD3 算法的平均成功率。

由于环境 c 是一个动态环境，因此无法在文中实时显示障碍物的情况，于是对环境 c 的初始状态进行建图，如图 14 所示，图中显示了各算法的测试结果。由于该环境是一个相对复杂的动态环境，各算法选择的路径也较为曲折，主要原因是为了避开动态障碍物。

表 6 各算法在环境 b 中的测试结果

Table 6 Test results of each algorithms in environment b

算法	平均成功率/%	平均距离/m	平均时间/s
TD3	70	13.33	34.29
TD3PN	100	13.15	31.04
TD3PN-LA3P	95	12.63	28.66
5-TD3PN	100	11.82	26.64
5-TD3PN-LA3P	100	**11.73**	**26.54**
TD3-PER	15	13.56	40.44
SAC	95	12.38	32.79
DDPG	15	13.08	34.13



(a)TD3 (b)TD3PN (c)TD3PN-LA3P (d)5-TD3PN (e)5-TD3PN-LA3P (f)TD3-PER (g)SAC (h)DDPG

图 14 各算法在环境 c 中的轨迹

Fig. 14 The trajectory of each algorithms in environment c

表 7 记录了每种算法在环境 c 中的测试结果。从表中可以看出，在环境 c 中，几乎每种算法的性能都有所下降，但本文算法性能最好。TD3PN-LA3P 算法的成功率高于 5-TD3PN 算法，但在平均距离和平均时间上表现不如 5-TD3PN 算法。在 TD3PN 中加入粉红噪声对 TD3 算法的性能有一定提高，但平均距离有所增加。而 5-TD3PN 算法的性能则优于 TD3PN 算法，这说明了多步方法的有效性。在这种情况下，TD3-PER 算法任何一次任务都无法完成，表现最差。SAC 算法的性能不如 DDPG 算法，但 DDPG 算法的性能不如 TD3 算法。

表 7 各算法在环境 c 中的测试结果

Table 7 Test results of each algorithms in environment c

算法	平均成功率/%	平均距离/m	平均时间/s
TD3	25	12.89	32.39
TD3PN	40	13.31	31.11
TD3PN-LA3P	75	12.70	29.60
5-TD3PN	65	12.38	29.19
5-TD3PN-LA3P	**85**	**12.06**	**28.64**
TD3-PER	0	—	—
SAC	5	13.93	37.58
DDPG	20	13.36	34.36

各算法训练完成后，将其和环境交互所得到的样本数量记录在了表 8 中。除 5-TD3PN 和 5-TD3PN-LA3P 算法外，在其他算法下机器人每采取一次行动都会被记录在经验池中，所以训练完成后经验池中的样本数量等于总训练步数。从表中可以看出，5-TD3PN-LA3P 算法训练完成后，经验池中的样本数量是最少的，这表明该算法使用更少的样本就可以完成最终的训练，同时结合前面的实验，5-TD3PN-LA3P 算法的整体表现更优，从而这在一定程度上说明本文算法拥有更高的样本利用率。

表 8 各算法训练完成后经验池中的样本数量

Table 8 Number of samples in experience replay buffer after training for each algorithms 单位：个

算法	样本数量
TD3	303 553
TD3PN	284 583
TD3PN-LA3P	249 021
5-TD3PN	236 297
5-TD3PN-LA3P	**229 478**
TD3-PER	352 745
SAC	332 394
DDPG	411 371

6 结束语

基于深度强化学习的无地图机器人路径规划方法已经得到了大量的研究，本文关注提高 TD3 算法在这一领域的实用性能，提出了一种改进的 TD3 算法，它包括对 TD3 算法探索策略的改进，以及有效结合 n 步方法和 LA3P 方法。首先对本文改进算法和其他一些现有的算法进行训练，本文算法在训练时间和训练步数上更具优势，超过了 DDPG、SAC 等算法。为了测试本文算法实际的泛化能力，构建了 3 种不同的未知环境，本文算法在平均成功率、平均距离和平均时间三方面更具优势，整体性能超过了 DDPG、SAC 等算法，表明本文算法泛化能力更强，并拥有更

高的样本利用率。由于本文算法对动态环境的自适应能力还需要进一步提高,本文只是在仿真环境下进行了实验,但仿真与现实往往存在差异,所以下一步研究应考虑本文算法在实际环境中的表现能力。

参考文献

[1] 王洪斌, 尹鹏衡, 郑维, 等. 基于改进的 A*算法与动态窗口法的移动机器人路径规划[J]. 机器人, 2020, 42(3): 346-353.
WANG H B, YIN P H, ZHENG W, et al. Mobile robot path planning based on improved A* algorithm and dynamic window method[J]. Robot, 2020, 42(3): 346-353. (in Chinese)

[2] 薛光辉, 王梓杰, 王一凡, 等. 基于改进人工势场算法的煤矿井下机器人路径规划[J]. 工矿自动化, 2024, 50(5): 6-13.
XUE G H, WANG Z J, WANG Y F, et al. Path planning of coal mine underground robot based on improved artificial potential field algorithm[J]. Journal of Mine Automation, 2024, 50(5): 6-13. (in Chinese)

[3] 王潇洒, 刘丽星, 杨欣, 等. 改进遗传算法的果园割草机作业路径规划[J]. 重庆理工大学学报(自然科学), 2024, 38(6): 227-233.
WANG X S, LIU L X, YANG X, et al. Improved genetic algorithm for orchard lawn mower operation path planning [J]. Journal of Chongqing University of Technology (Natural Science), 2024, 38(6): 227-233. (in Chinese)

[4] 朱敏, 胡若海, 卞京. 基于改进蚁群算法的移动机器人路径规划[J]. 现代制造工程, 2024, 52(3): 38-44.
ZHU M, HU R H, BIAN J. Path planning for mobile robots based on improved ant colony algorithm [J]. Modern Manufacturing Engineering, 2024, 52(3): 38-44. (in Chinese)

[5] HAMZA A. Deep reinforcement learning for mapless mobile robot navigation[D]. Luleå, Sweden: Luleå University of Technology, 2022.

[6] JIANG H, WANG H, YUAN W Y, et al. A brief survey: deep reinforcement learning in mobile robot navigation[C]// Proceedings of the 15th IEEE Conference on Industrial Electronics and Applications. Washington D. C., USA: IEEE Press, 2020: 592-597.

[7] 刘红伟, 潘灵, 吴明钦, 等. 一种 FPGA 集群轻量级深度学习计算架构设计及实现[J]. 电讯技术, 2024, 64(1): 14-21.
LIU H W, PAN L, WU M Q, et al. Design and implementation of lightweight deep learning computing architecture for FPGA cluster [J]. Telecommunication Engineering, 2024, 64(1): 14-21. (in Chinese)

[8] LILLICRAP T P. Continuous control with deep reinforcement learning[EB/OL]. [2024-07-10]. https://arxiv.org/abs/1509.02971v2.

[9] SILVER D, LEVER G, HEESS N, et al. Deterministic policy gradient algorithms[C]//Proceedings of the 31st International Conference on Machine Learning. Washington D. C., USA: IEEE Press, 2014: 387-395.

[10] FUJIMOTO S, HOOF H, MEGER D. Addressing function approximation error in actor-critic methods[C]//Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden: [s. n.], 2018: 1587-1596.

[11] CIMURS R, SUH I H, LEE J H. Goal-driven autonomous exploration through deep reinforcement learning[J]. IEEE Robotics and Automation Letters, 2021, 7(2): 730-737.

[12] HU W, ZHOU Y, HO H W. Mobile robot navigation based on noisy n-step dueling double deep Q-network and prioritized experience replay[J]. Electronics, 2024, 13(12): 2423.

[13] SCHAUL T, QUAN J, ANTONOGLOU I, et al. Prioritized experience replay[C]//Proceedings of the 2016 International Conference on Learning Representations. Washington D. C., USA: IEEE Press, 2016: 322-355.

[14] MARCHESINI E, FARINELLI A. Discrete deep reinforcement learning for mapless navigation [C]// Proceedings of the 2020 IEEE International Conference on Robotics and Automation. Washington D. C., USA: IEEE Press, 2020: 10688-10694.

[15] SAGLAM B, MUTLU F B, CICEK D C, et al. Actor prioritized experience replay [J]. Journal of Artificial Intelligence Research, 2023, 78: 639-672.

[16] ZHANG X, SHI X, ZHANG Z, et al. A DDQN path planning algorithm based on experience classification and multi steps for mobile robots[J]. Electronics, 2022, 11(14): 2120.

[17] VAN HASSELT H, GUEZ A, SILVER D. Deep reinforcement learning with double Q-learning[C]//Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix, USA: AAAI Press, 2016: 2094-2100.

[18] YANG J, NI J, LI Y, et al. The intelligent path planning system of agricultural robot via reinforcement learning[J]. Sensors, 2022, 22(12): 4316.

[19] 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. New, York, USA: ACM Press, 2018: 1861-1870.

[20] CHOWDHURY M A, LU Q. A novel entropy-maximizing TD3-based reinforcement learning for automatic PID tuning[C]// Proceedings of the 2023 American Control Conference. Washington D. C., USA: IEEE Press, 2023: 2763-2768.

[21] YIN Y, CHEN Z, LIU G, et al. A mapless local path planning approach using deep reinforcement learning framework[J]. Sensors, 2023, 23(4): 2036.

[22] LIU L, CHEN J, ZHANG Y, et al. Unmanned ground vehicle path planning based on improved DRL algorithm[J]. Electronics, 2024, 13(13): 2479.

[23] EBERHARD O, HOLLENSTEIN J, PINNERI C, et al. Pink noise is all you need: colored noise exploration in deep reinforcement learning[C]//Proceedings of the 12th International Conference on Learning Representations. Washington D. C., USA: IEEE Press, 2023: 457-466.

[24] 王涛, 张卫华, 蒲亦非. 适用于强化学习惯性环境的分数阶改进 OU 噪声[J]. 四川大学学报(自然科学版), 2023, 60(2): 57-63.
WANG T, ZHANG W H, PU Y F. An improved Ornstein-Uhlenbeck exploration noise based on fractional order calculus for reinforcement learning environments with momentum[J]. Journal of Sichuan University (Natural Science Edition), 2023, 60(2): 57-63. (in Chinese)

[25] SUTTON R S, BARTO A G. Reinforcement learning: an introduction[M]. Cambridge, USA: MIT Press, 2018.

[26] HESSEL M, MODAYIL J, VAN HASSELT H, et al. Rainbow: combining improvements in deep reinforcement learning[C]//Proceedings of the 32nd AAAI Conference on Artificial Intelligence. Palo Alto, USA: AAAI Press, 2018: 332-343.

[27] FUJIMOTO S, MEGER D, PRECUP D. An equivalence between loss functions and non-uniform sampling in experience replay[C]//Proceedings of the Advances in Neural Information Processing Systems. Cambridge, USA: MIT Press, 2020: 14219-14230.

[28] KONDA V, TSITSIKLIS J. Actor-Critic algorithms [C]// Proceedings of the Advances in Neural Information Processing Systems. Cambridge, USA: MIT Press, 1999: 212-222.

文字编辑 索书志
栏目编辑 赖玉玲