计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
[doi: 10.15888/j.cnki.csa.010041] [CSTR: 32024.14.csa.010041]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于分位数回归的多智能体强化学习①

张志文, 周长东, 马裕博, 张　博

(大连海事大学 人工智能学院, 大连 116026)
通信作者: 张　博, E-mail: bzhang@dlmu.edu.cn

摘　要: 多智能体强化学习是多智能体系统研究的重要组成部分, 在复杂协同任务中成效显著. 然而, 在需要长期决策的场景下, 由于长期回报的估计难度更大, 且难以对环境中的不确定性进行精准建模, 多智能体的表现往往不佳. 为解决上述问题, 本文提出了一种基于分位数回归的多智能体记忆强化学习模型. 该模型不仅选择性地利用了历史决策经验用于辅助长期决策, 还通过分位数函数对回报分布进行建模, 从而有效地捕捉了回报的不确定性. 该模型由记忆索引模块、隐式分位数决策网络和值分布分解模块这 3 部分组成, 其中记忆索引模块利用历史决策经验生成内在奖励, 促进智能体充分利用已有经验. 隐式分位数决策网络通过分位数回归, 对奖励分布进行建模, 为长期决策提供有力支持. 值分布分解模块将整体的回报分布分解为单个智能体的回报分布, 用于辅助单个智能体策略的学习. 本文的算法在星际争霸环境中进行了广泛的实验, 实验结果表明, 本文提出的方法提升了智能体在长期决策任务中的表现, 并具有较快的收敛速度.

关键词: 多智能体; 值分布强化学习; 分位数函数; 历史决策经验; 奖励分布

引用格式: 张志文,周长东,马裕博,张博.基于分位数回归的多智能体强化学习.计算机系统应用. http://www.c-s-a.org.cn/1003-3254/10041.html

Multi-agent Reinforcement Learning Based on Quantile Regression

ZHANG Zhi-Wen, ZHOU Chang-Dong, MA Yu-Bo, ZHANG Bo

(College of Artificial Intelligence, Dalian Maritime University, Dalian 116026, China)

Abstract: Multi-agent reinforcement learning (MARL) is a crucial part of multi-agent system research, demonstrating remarkable effectiveness in complex collaborative tasks. However, in scenarios requiring long-term decision-making, multi-agent systems often underperform due to the difficulty in estimating long-term returns and accurately modeling environmental uncertainties. To this end, this study proposes a multi-agent memory-reinforcement learning model based on quantile regression. The model not only selectively utilizes historical decision-making experience to assist long-term decision-making but also employs quantile functions to model the return distribution, thereby effectively capturing return uncertainties. The model comprises three components, including a memory indexing module, an implicit quantile decision network, and a value distribution decomposition module. Specifically, the memory indexing module generates intrinsic reward by adopting historical decision-making experience to enhance the agents' full utilization of existing experience. The implicit quantile decision network models reward distribution via quantile regression, providing powerful support for long-term decision-making. The value distribution decomposition module decomposes overall return distributions into the distribution of an individual agent to support single-agent strategy learning. Extensive experiments conducted in StarCraft II environments demonstrate that the proposed method enhances the performance of agents in long-term decision-making tasks, with fast convergence rates.

Key words: multi-agent; distributional reinforcement learning; quantile function; historical decision-making experience; reward distribution

① 收稿时间: 2025-05-31; 修改时间: 2025-07-07; 采用时间: 2025-07-29; csa 在线出版时间: 2025-11-17

协作多智能体强化学习[1−3]的目标是探索并设计一种高效策略, 来促使智能体之间协同工作, 以高效完成各类任务. 这一方法近年来取得了显著进展, 在多个关键领域展现了巨大的应用潜力, 在智能交通系统中, 该方法能够优化交通流量, 提高整体运输效率[4], 在无人船协同作业中, 该方法能够实现多艘无人船之间的精准配合, 完成复杂的水上任务[5], 而在机器人协作领域, 该技术则能增强机器人团队的整体协作能力, 提升工作效率和安全性[6]. 当前, 集中式训练与分布式执行 (centralized training with decentralized execution, CTDE) 范式[7−9]已成为多智能体强化学习领域的主流框架之一. 在此范式下, 智能体在训练阶段利用全局信息辅助学习, 而在实际执行时仅依据各自的局部观测做出决策. 在 CTDE 范式下, 众多算法通过值分解, 经验回放等方法, 进一步提升了智能体的决策能力, 其中 QPLEX[10]使用双工决斗网络扩展了值分解的表达范围, 使得智能体能够更准确地评估联合动作的价值. RODE[11]通过为智能体赋予不同角色提高了模型的泛化能力, 使其能够适应更加复杂多变的环境. EMC[12]通过经验回放提高对历史决策经验的利用效率, 从而加速了智能体的学习过程. EMU[13]通过记忆产生的内在奖励激励智能体向最优轨迹靠近, 进一步提升了学习效果. LAGMA[14]在分类全局状态后通过匹配历史最优轨迹生成额外奖励, 为智能体提供了更加丰富的反馈信息. 最近几年, 值分布强化学习方法在单智能体强化学习领域取得了引人注目的成果[15−17], 受此启发, DFAC 等方法[18−20]将值分布强化学习的思想引入多智能体值分解领域, 通过考虑智能体联合动作的回报分布情况, 而非仅关注其期望值, 从而提升了多智能体在复杂环境中的表现.

多智能体强化学习方法在追求高效策略的过程中, 通常面临着长期决策表现不佳的问题. 这是因为在学习过程中, 多个智能体在同一环境中并行进行策略学习, 导致训练过程具有更强的随机性[21], 在探索过程中, 每个智能体都需要不断尝试不同的行为, 并基于这些行为所带来的回报来调整其策略. 同时, 智能体还需学习如何与其他智能体进行有效的协调与合作, 以实现共同的任务目标. 然而, 由于马尔可夫过程状态转移的随机性以及智能体联合决策所带来的不确定性, 准确估计长期回报变得尤为困难[22].

为解决上述问题, 本文提出了一种包含记忆索引的多智能体值分布强化学习方法 (distributional value functions with episodic memory construction, DEC). 该模型由记忆索引模块、隐式分位数决策网络和值分布分解模块这 3 部分构成. 在学习过程中, 决策网络以局部观测为输入, 通过分位数回归建模奖励, 输出每个动作对应的回报分布, 这是本文通过建模回报的不确定性提升长期决策能力的关键步骤. 值分布分解模块通过将整体回报分布分解为回报的均值部分和偏离回报分布期望的不确定性部分, 分别进行处理, 在回报均值部分, 采用值分解方法实现智能体间的效益分配; 在偏离回报均值部分, 则通过累积单个智能体对回报不确定性的估计, 将整体回报分布进一步分解为单个智能体的回报分布, 以辅助单个智能体的策略学习. 此外, 本文通过记忆索引模块生成内在奖励, 该模块模拟索引与查阅信息的过程, 专注于历史决策经验的存储与利用. 该模块首先将原始场景编码为记忆索引, 以便于对记忆进行快速检索和匹配, 然后, 利用历史决策经验数据库为内在奖励的生成提供依据. 通过选择性地给予额外的奖励, 激励智能体有效利用过往经验, 从而辅助智能体进行更加明智的长期决策, 并提升算法收敛速度.

1 背景

1.1 多智能体马尔可夫决策过程

多智能体强化学习任务通常被建模为部分可观测的马尔可夫决策过程$\langle N,S,O,A,P,R,\gamma\rangle$, 其中$N$代表$n$个智能体, $s\in S$代表一个全局状态, $o_i\in O$代表第i个智能体的局部观测, $a_i\in A$代表第i个智能体的动作, $P(s'|s,a)$为状态转移函数, $R(s,a)$为奖励函数, γ为衰减因子. 在每一个时间步, 全局状态为s_t, 智能体i接受局部观测o_i, 依据决策模型采取动作a_i, 所有智能体的联合动作被执行后, 跳转到下一个状态s_{t+1}, 并获得衰减后的奖励r, 任务的目标是最大化累积奖励$\sum_{t=0}^{t_{\max}}\gamma^t r_t$.

1.2 值分布强化学习

值分布强化学习直接建模回报的分布$Z(s,a)$, 而不仅是传统的期望值$Q(s,a)$, 即动作价值函数. 这种方法通过捕捉回报的完整分布, 能够提供更丰富的信息, 从而提升算法的性能和鲁棒性. 值分布强化学习中对分布贝尔曼最优算子T^*[23]的定义如下:

$$T^*Z(s,a)=R(s,a)+\gamma Z(s',a'^*) \tag{1}$$

其中, s'为下一状态, a'^*为当前策略下, 下一状态的最

优联合动作. 初始化分布$Z(s,a)$之后, 不断应用贝尔曼最优算子, $Z(s,a)$会依瓦森斯坦度量收敛到最优策略下的回报分布[23], 如引理 1 所示.

引理 1. 设 $Z(s,a)$ 表示回报分布, 其期望为:

$$Q(s,a)=E[Z(s,a)] \tag{2}$$

定义分布贝尔曼最优算子T^*并取瓦森斯坦距离W_p作为度量, 则Q在T^*作用下收敛到最优值函数Q^*. 其中瓦森斯坦距离为:

$$W_p(Y,Z)=\left(\int_0^1 |Y(w)-Z(w)|^p \mathrm{d}w\right)^{\frac{1}{p}} \tag{3}$$

在本文中, 通过分位数函数来建模回报分布, 其中, 分位数函数是累积分布函数 (CDF) 的广义逆函数, 其具体定义如下:

$$F_X^{-1}(\omega)=\inf\{x\in R: \omega \leqslant F_X(x)\},\ \forall \omega\in[0,1] \tag{4}$$

当累积分布函数可逆时, 分位数函数即是累计分位函数的逆函数. 在本文中使用$Q(s,a,w)$指代回报分布$Z(s,a)$, 其含义为, 此后的总回报小于等于$Q(s,a,w)$的概率为w.

2 方法

本文提出的 DEC 方法旨在通过建模回报的不确定性并有效利用历史决策经验, 提高多智能体系统在长期决策任务中的表现. 其整体架构如图 1 所示, 由以下 3 个模块构成: 记忆索引模块, 隐式分位数决策网络和值分布分解模块.

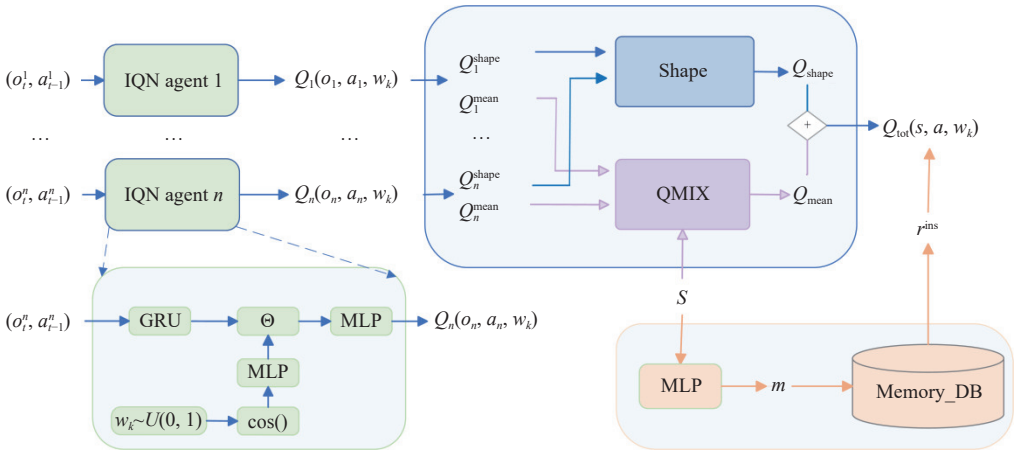

图 1 DEC 结构示意图

记忆索引模块通过检索记忆索引查找相似状态, 并根据这些相似状态的最高奖励及统计胜率赋予内在奖励. 隐式分位数决策网络以局部观测为输入, 通过采样分位数, 输出每个动作对应的回报分布. 值分布分解模块则负责将整体回报分布拆解为单个智能体的回报分布, 以此指导单智能体策略的学习过程. DEC 方法通过记忆索引模块建立并有效利用历史经验, 通过使用分位数函数建模奖励回报的方式捕捉了回报的不确定性, 从而能提供更准确的回报估计. 具体如算法 1 所示.

算法 1. DEC 算法

初始化: 设定智能体数量n, 训练总轮数L, 每轮时间步数$t_{\max}$, 批次大小bs, 初始化经验回放池B, 记忆索引库M, 隐式分位数决策网络g及网络参数θ, 值分布分解网络$mixer$及网络参数ω.

1) For e=0 to L−1 do
2) For t=0 to $t_{\max}$−1 do
3) 从环境中获得每个智能体局部观测$\{o_i\}_{i=1:n}$
4) 得到动作价值估计$Q_i(o_i,a_i,w_k)\leftarrow g_\theta(o_i,h_t)$
5) 依据动作价值估计选取动作a_i
6) 执行联合动作$\{a_i\}_{i=1:n}$获得奖励r^t
7) End for
8) 将轨迹$(s^t,o_i^t,a_i^t,r^t)_{t=1:t_{\max}}$存入$B$并更新$M$
9) 从B采样, 得到$(s^t,o_i^t,a_i^t,r^t)_{t=1:t_{\max}}^{b=1:bs}$用于训练
10) For t=0 to $t_{\max}$−1 do
11) 获得每个智能体局部观测$\{o_i\}_{i=1:n}$
12) 动作价值估计$Q_i(o_i,a_i,w_k)\leftarrow g_\theta(o_i,h_{t-1})$
13) 依据式 (10) 估计整体动作价值$Q_{\text{tot}}(s,a,w_k)$
14) 目标网络得到$\hat{Q}_{\text{tot}}(s,a,w_k)$
15) 依据式 (5) 查询每个状态s的索引m
16) 查询最相似的记忆记录$(\hat{s},\hat{m},\hat{v},\hat{\eta})$
17) 若m与$\hat{m}$足够近, 按式 (6) 给予内在奖励r^{ins}
18) 计算目标网络总Q值为$\hat{Q}_{\text{tot}}+r^{\text{ins}}$
19) 依据式 (11) 计算 TD-error
20) 依据式 (12) 计算$loss(\theta,\omega)$
21) 反向传播更新参数$\theta,\omega\leftarrow loss(\theta,\omega)$
22) End for
23) End for

2.1 记忆索引模块

记忆索引模块包括记忆存储与记忆利用两个子模块, 其中记忆存储模块负责存储历史决策经验, 而记忆利用模块依据记忆索引查找相似状态的历史记录并给予该状态额外奖励. 具体来说, 当多智能体系统处于状态s时, 记忆索引网络f_φ会根据该状态生成一个记忆索引m, 记忆利用子模块会利用该索引在记忆存储中搜索与当前状态最为相似的历史记录, 依据其中的历史最高回报与相似状态下智能体的历史胜率生成内在激励, 促进智能体高效利用历史决策经验.

● 记忆存储模块: 在记忆存储模块中, 为衡量两个状态的相似性, 并高效地实现对相似记忆的检索, 引入了记忆索引网络f_φ将全局状态映射为相应的索引, 进而利用两个索引之间的欧氏距离来评估不同状态之间的相似程度. 在记忆索引网络中, 本文设计了全局状态自编码器, 其潜在变量m被用作全局状态s的记忆索引, 其中,

$$m = f_\varphi(s) \tag{5}$$

这一设计能够有效地将高维的全局状态转换为低维的索引表示, 便于后续的相似度计算和记忆检索. 为了充分利用历史决策信息, 需要在智能体与环境交互的过程中存储部分关键数据. 具体而言, 每一状态及其对应的最高回报(s,v)都被记录下来. 此外, 为了防止智能体过于依赖记忆奖励而陷入局部最优解, 本文还额外记录了每个状态的统计胜率η. 这一指标用于控制内在激励的大小, 确保智能体在利用记忆信息时能够保持谨慎. 在智能体与环境完成一轮交互过程后更新记忆存储模块的内容, 这一过程包括计算每个状态的回报, 从而生成一系列(s,v)对, 接着将全局状态s映射为记忆索引m, 并搜索与记忆索引m最近的已有索引$\hat{m}$, 得到记忆元组$(\hat{s},\hat{m},\hat{v},\hat{\eta})$, 若$\hat{m}$与$m$的欧氏距离大于阈值, 则将$m$视为一个新的记忆索引, 并存储$(s,m,v,\eta)$. 若$\hat{m}$与$m$的距离小于一定阈值则将两者认定为同一索引, 并对相应的记忆元组进行更新, 更新操作具体分为两步: 首先对统计胜率η进行更新, 其次, 若v大于$\hat{v}$则对v值进行更新.

可学习的记忆索引映射的问题是如何在记忆映射函数f_φ更新为f'_φ后仍然保持记忆的一致性, 本文的方法为每经过一定轮次后, 更新(s,m,v,η)组中的m, 在记忆网络更新后, 使用更新后的记忆嵌入网络作用于原全局状态, 来将整个记忆更新为$(s,f'_\varphi(m),v,\eta)$, 由于记忆索引只用于查找相似状态, 因此这种更新方式能够保持记忆的一致性.

● 记忆利用模块: 在记忆利用模块中, 依据已有记忆生成内在奖励. 通过一个外部记忆模块记录并索引探索过程中遇到的高价值经验, 这些关键轨迹可被反复调用用于指导当前策略更新, 实现经验的充分复用. 为价值估计提供了额外的内在奖励, 避免智能体因遗忘过去的成功经验而偏离探索方向. 对于状态s, 经由f_φ获取记忆索引m, 随后根据最近的已有记忆索引$\hat{m}$检索出对应的$(\hat{s},\hat{m},\hat{v},\hat{\eta})$, 仅当前状态与已有记忆足够相似, 即$m$与$\hat{m}$之间的欧氏距离小于设定的阈值时, 才会产生内在激励, 内在奖励的计算公式为:

$$r^{\text{ins}} = ReLU(\hat{\eta}(\hat{v} - v)) \tag{6}$$

其中, v为对当前状态s状态价值的估计, 由公式可知, 内在奖励始终为非负值, 当前状态在引入内在奖励后的总价值不超过该状态的历史最优回报. 该内在奖励不仅使用该状态历史最高价值与当前价值的差距为该状态价值提供补充, 还额外考虑了该状态的统计胜率以防止草率地利用记忆而陷入局部最优解.

2.2 隐式分位数决策网络

为了高效利用奖励, 本文设计了隐式分位数决策网络, 采用分位数函数拟合奖励分布, 该智能体接受局部观测作为输入, 经过随机采样分位数之后, 输出每一可取动作的预期回报分布, 随后智能体通过比较不同分布的期望值采取动作. 在多智能体任务中, 由于部分可观察和其他智能体策略变化导致环境具有非平稳性, 单一的期望回报估计难以全面反映回报的随机性. 基于分位数的分布式值函数能够刻画回报的不确定性和整个分布形态, 为决策提供更丰富的信息.

隐式分位数决策网络接受局部观测作为输入, 经过 GRU 后得到隐藏状态h. 为了深入捕捉回报分布的特性, 本文采取了n_q次分位数采样, 针对每次采样得到的分位数w_k, 利用余弦编码技术将其转换并扩展为与 GRU 隐藏态相同的m维向量q, 其每一维度具体数据为$q_j = \cos(jw_k\pi)$.

随后将经过编码后的分位数经过 MLP 后与 GRU 隐藏状态h作哈达玛积, 经过 MLP 后得到结果$Q_i(o_i, a_i, w_k)$, 其含义为智能体i接受观察o_i后采取动作a_i得到的回报低于$Q_i(o_i,a_i,w_k)$的概率为w_k.

2.3 值分布分解模块

值分布分解模块负责将全局奖励分布分解为各个单一智能体的奖励分布, 以此为单一智能体策略的学习提供辅助, 为了实现全局奖励回报值分布向单一智能体回报分布的分解, 本模块设置了两个子模块, 分别对回报均值和偏离回报期望的部分进行处理, 进而完成回报分布的分解. 一个分布可以如式 (7) 所示进行分解[18].

$$Z = E[Z] + (Z - E[Z]) \tag{7}$$

本文通过值分解的方法获取对回报分布均值的估计, 即式 (7) 中的期望部分, 同时, 借助扰动模块建模整体回报的不确定性, 即式 (7) 中偏离期望的部分. 值分布分解模块接受每个智能体对回报的估计$Q_i(o_i,a_i,w_k)$为输入, 输出整体的奖励回报分布$Q_{\text{tot}}(o,a,w_k)$. 首先, 在分位数的维度上求均值得到每个智能体对回报预测的期望$Q_i(o_i,a_i)$.

$$Q_i^{\text{mean}}(o_i,a_i) = \sum_{k=1}^{n_q} Q_i(o_i,a_i,w_k) \tag{8}$$

接着得到对应的偏离回报均值的部分:

$$Q_i^{\text{shape}}(o_i,a_i,w_k) = Q_i(o_i,a_i,w_k) - Q_i^{\text{mean}}(o_i,a_i) \tag{9}$$

随后使用 QMIX[24]处理所有智能体的Q值, 得到整体分布的主体部分, 即式 (7) 的期望部分. 单个智能体的回报偏移不能直接通过 QMIX 网络而得到整体回报偏移, 是因为通过 QMIX 后不能保证其期望为 0, 为了保持这一性质, 这里直接将它们累加得到整体分布的偏移部分, 即式 (7) 偏离期望的部分. 随后得到整体的回报分布$Q_{\text{tot}}(s,a,w_k)$为:

$$Q_{\text{tot}} = QMIX(Q_1^{\text{mean}},\cdots,Q_n^{\text{mean}},s) + \sum_{i=1}^{n} Q_i^{\text{shape}} \tag{10}$$

2.4 损失函数

通过隐式分位数函数建模回报分布后, 考虑了内在奖励的 TD-error[13,25]定义为:

$$\delta^{w,w'} = r + r^{\text{ins}} + \gamma Q_{\text{tot}}(s',a'^{*},w') - Q_{\text{tot}}(s,a,w) \tag{11}$$

其中, s'为下一状态, a'^{*}为当前策略下, 下一状态下的最优动作, r^{ins}为式 (6) 中的内在奖励. 为了拟合分位数函数, 采取阈值为 1 的 Huber Loss 函数[26]计算分位数拟合损失, $L_{\text{huber}}(\delta^{w,w'})$具体定义为:

$$\begin{cases} \dfrac{1}{2}|w - U(\delta^{w,w'})|(\delta^{w,w'})^2, & \text{if } |\delta^{w,w'}| \leqslant 1 \\ |w - U(\delta^{w,w'})|\left(|\delta^{w,w'}| - \dfrac{1}{2}\right), & \text{if } |\delta^{w,w'}| > 1 \end{cases} \tag{12}$$

其中,

$$U(\delta^{w,w'}) = \begin{cases} 1, & \text{if } \delta^{w,w'} \leqslant 0 \\ 0, & \text{if } \delta^{w,w'} > 0 \end{cases} \tag{13}$$

2.5 收敛性分析

在本节考察分布强化学习在多智能体情形下的最优性与收敛性, 首先通过对分布求期望来将分布强化学习的收敛转化为通常强化学习的收敛, 由引理 1, 在单智能体的情形下, 分布强化学习的最优性与收敛性已被证明.

对于多智能体的情形, 一般通过假设训练过程中每个智能体都能够利用全局状态信息, 并满足个体-全局最优一致性 (individual-global max, IGM) 的条件将强化学习的收敛性扩展到多智能体的情形. 此时协作多智能体决策问题在形式上可以等价为一个具有联合动作的大规模单智能体决策过程. 其中 IGM 条件为:

$$\arg\max_a Q_{\text{tot}}(s,a) = \{\arg\max_{a_1} Q_1,\cdots,\arg\max_{a_n} Q_n\} \tag{14}$$

定理 1. 假设动作价值函数的分解满足 IGM 条件, 则在式 (10) 的分布情形下, IGM 条件依然满足.

$$\begin{aligned} &\arg\max_a\{E(Q_{\text{tot}}(s,a,w))\} \\ &= \arg\max_a\{E(Q^{\text{mean}}(s,a)) + E(Q^{\text{shape}}(s,a,w))\} \\ &= \arg\max_a\{E(QMIX(Q_1^{\text{mean}},\cdots,Q_n^{\text{mean}},s) + 0\} \\ &= \{\arg\max_{a_1} Q_1^{\text{mean}},\cdots,\arg\max_{a_n} Q_n^{\text{mean}}\} \end{aligned}$$

由于在式 (10) 中每个状态-动作的回报分布已被显式地分解为确定性的分布的期望部分和随机的均值为 0 的部分, 能够保证期望值的估计不受分布形状近似误差的影响. 本节从理论上证明了在合理假设下多智能体分位数分布强化学习的收敛性与最优性, 为分布式方法的合理性提供了直接证据.

3 实验

在本节中, 评估了 DEC 在星际争霸实验环境中的表现. 首先将 DEC 与典型的 MARL 模型进行了对比实验. 之后, 对 DEC 的组件进行了消融实验, 以验证模型的有效性.

3.1 实验设置

为了验证所提出的算法的有效性, 本文在星际争霸环境 (the StarCraft multi-agent challenge, SMAC)[27]中进行了广泛的实验, 场景示例如图 2 所示. SMAC 具有庞大的状态与观测空间, 对智能体之间的协同配合

提出了挑战. 该环境提供了多样化的遭遇战地图, 要求智能体采取策略, 与敌方单位展开对抗并取得胜利, 测试地图涵盖了从短期决策场景 (例如 5m_vs_6m 地图) 到中长期决策场景 (如 3s5z_vs_3s6z 地图) 的多种类型. 在对比方法中, 本文选取了多种前沿算法进行评估. 这些算法包括: 融合了好奇心与记忆机制的多智能体协同算法 EMC, 该算法通过智能体的探索能力和优化记忆能力来提升协同效果; 优化记忆机制的 EMU, 它通过选择性地激励, 鼓励智能体向最优行为轨迹靠近; 值分布分解多智能体强化学习算法 DMIX, 它利用值分布的分解技术来优化多智能体的决策过程; 完全去中心化隐式分位数强化学习算法 DIQL, 该算法采用隐式分位数智能体, 实现了高效的去中心化学习; 以及值分解领域的经典算法 QMIX, 作为基准算法, 用于对比和评估其他算法的性能.

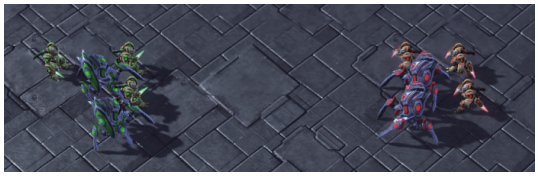

图 2 SMAC 场景图

3.2 实验结果

图 3 直观展示了 DEC 与基线方法在各类 SMAC 场景下的测试胜率对比情况, 实验数据是基于在不同随机种子下重复进行的 10 次实验的中位数统计结果. 表 1 中是详细测试胜率结果.

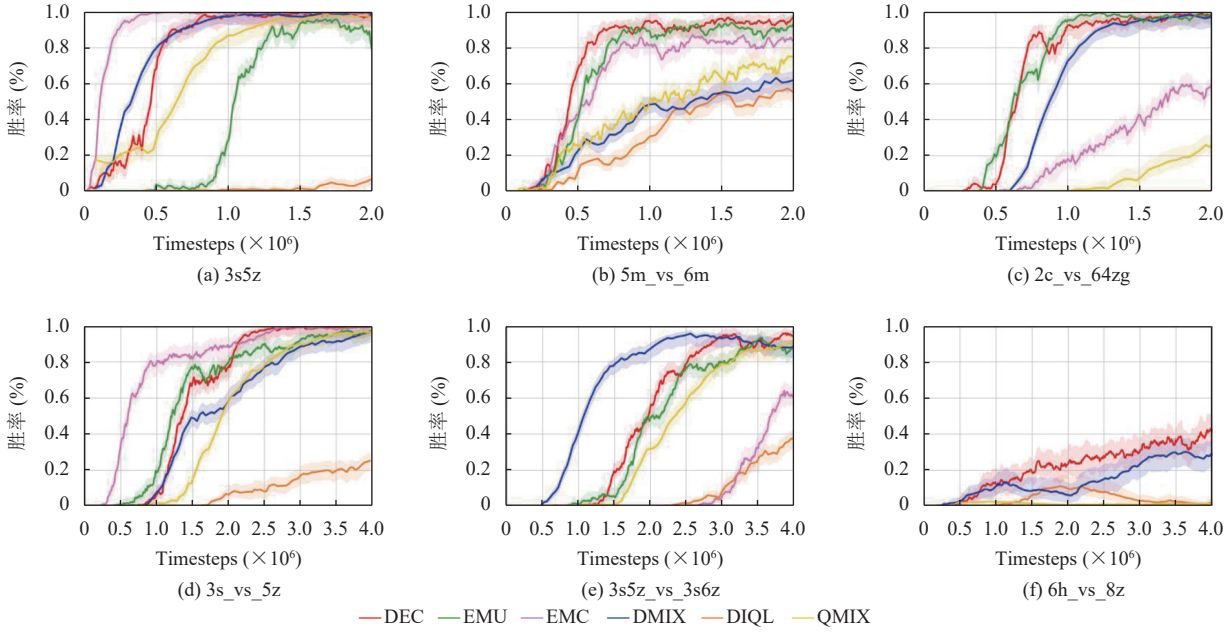

图 3 SMAC 场景的测试胜率

表 1 SMAC 场景测试胜率 (%)

模型	场景类别						平均胜率
	3s5z	5m_vs_6m	3s_vs_5z	2c_vs_64zg	3s5z_vs_3s6z	6h_vs_8z	
DEC	96	96	99	96	91	47	88
EMU[13]	81	89	96	95	88	0	75
EMC[12]	97	83	97	56	55	0	65
DMIX[18]	97	61	97	93	84	26	76
DIQN[18]	3	53	31	0	41	0	21
QMIX[24]	93	72	99	28	92	3	65

值得注意的是, 采用了记忆机制的方法, 例如 EMC 和 EMU, 通常展现出更快的收敛速度. 然而, 这些方法在超难地图如 6h_vs_8z 上表现不佳. 原因在于, 记忆机制虽然增强了智能体对历史经验的利用效率, 却未能同步提升其探索能力. 这导致智能体虽能在相对简单的环境中迅速找到解决方案, 并通过充分利用历史经验来提升胜率, 但在面对复杂环境时, 却往往缺乏探索更优解决方案的能力. 并且由于对过往经验的过度依赖, 智能体容易陷入局部最优解, 从而限制了其在高难度任务中的表现. 另一方面, 基于值分布分解的方法 DMIX 相比于其基线方法 QMIX, 在大部分场景中都有更优秀的表现, 这是因为 DMIX 直接建模了回报分布, 从而能够更精确地预估中长期回报并进行有效探

索, 在包括 6h_vs_8z 在内的超难环境中, 这种提升尤为显著. 但是这种方法在部分场景, 如 5m_vs_6m 中表现不佳, 这可能是因为在这类环境中更强调利用以往经验进行更精细的操作, 因此基于记忆的方法如 EMC、EMU 和 DEC 在这类地图中表现更好. 在算法收敛速度方面, DEC 在 5m_vs_6m 和 6h_vs_8z 地图上收敛速度最快, 在 2c_vs_64zg 和 3s5z_vs_3s6z 地图上也取得了次优的收敛速度. 为进一步评估模型的鲁棒性, 本文比较了各方法在多种场景中完成任务所需的平均决策步数, 如图 4 所示. 实验结果表明, DEC 能以更少的步数完成任务并实现与其他方法相当甚至更优的胜率, 体现出其策略的高效性与执行阶段的稳健性. 本文提出的 DEC 不仅通过直接建模回报分布提升了智能体在复杂环境下预测长期回报的能力, 还通过引入记忆索引模块有选择性地强化历史最优状态, 从而高效地利用历史决策经验. 这些综合策略的实施, 提高模型在处理中长期决策任务时的整体性能, 并且提升了算法的收敛速度.

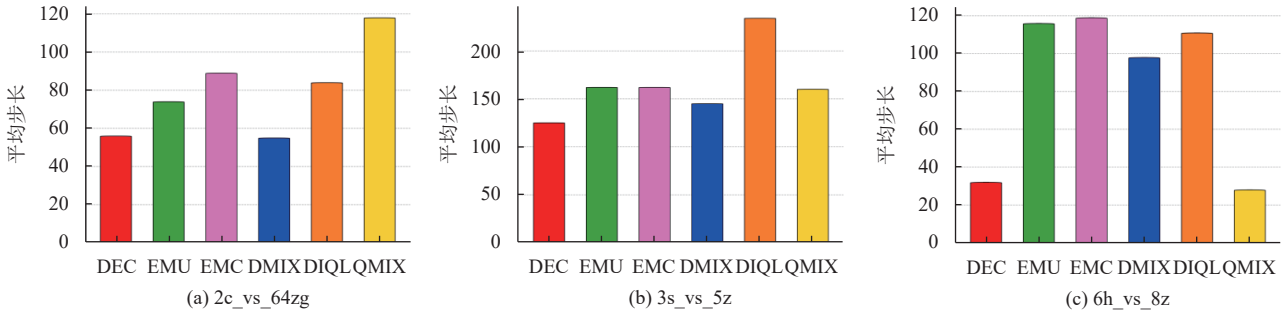

图 4 SMAC 场景决策步长实验示意图

3.3 计算开销分析

本文在多种实验场景下分别统计了 Agent 模块, Mixer 模块以及记忆索引模块的参数规模与整队智能体的每步浮点运算量. 各子模块的平均开销列于表 2.

表 2 DEC 的参数量与 FLOPs

网络结构	参数量 ($\times 10^4$)	浮点数运算 (FLOPs)
Agent 模块	3.60	4.01×10^5
Mixer 模块	3.57	9.54×10^5
记忆索引模块	1.22	1.21×10^4
DEC 模型	8.39	1.37×10^6

表 2 中, Mixer 模块与 Memory 模块只在训练阶段调用, 执行部署阶段只需要 Agent 模块进行前向传播. 综合来看, DEC 的总体参数量与所需浮点运算量显著低于当前主流机器人计算模块 (如 NVIDIA Jetson Xavier NX, NVIDIA Jetson Orin NX 16 GB) 的算力上限, 具备良好的实际部署潜力.

3.4 消融实验

为了验证算法的有效性, 本文在 3s_vs_5z、5m_vs_6m 和 6h_vs_8z 地图上进行了消融实验, 分别探究了移除记忆模块 (DEC without memory module, 记为 DEC-wo-MM) 和移除值分布模块 (DEC without distributional module, 记为 DEC-wo-DM) 对算法性能的影响. 为确保实验结果的稳健性, 每种配置均在不同随机种子的条件下独立运行 10 次, 并取其中位结果作为最终评估依据. 消融实验的结果如图 5 所示, 它们揭示了不同模块对算法性能的关键作用.

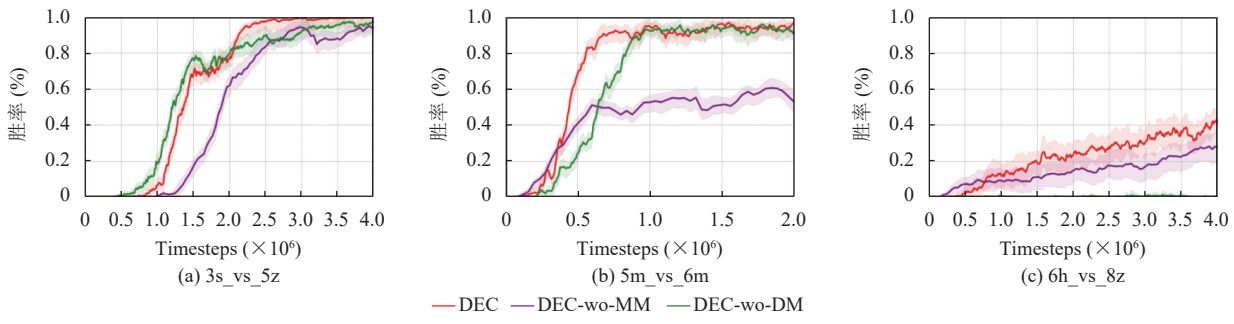

图 5 BTM 组件消融实验示意图

具体而言, 移除记忆模块后, 算法在需要精细操控的场景 (如 5m_vs_6m 和 6h_vs_8z 地图) 中表现出显著下滑, 胜率分别下降了 41% 和 18%. 这一结果可能归因于记忆模块的缺失导致模型无法有效利用历史决策经验, 进而难以通过实施精细的策略优化来进一步提升智能体在这些场景中的表现. 另一方面, 移除值分布模块对算法在超难地图 (如 6h_vs_8z) 上的性能产生了显著负面影响. 这可能是因为值分布模块的移除削弱了模型对回报分布进行建模的能力, 使得智能体在高度随机和动态变化的环境中难以对中长期回报进行准确评估, 导致了模型性能的下降. 具体结果如表 3.

表 3 消融实验 (%)

实验配置	场景类别		
	3s_vs_5z	5m_vs_6m	6h_vs_8z
DEC	99	96	47
DEC-wo-DM	97	90	3
DEC-wo-MM	94	55	29

4 结束语

本文提出了 DEC 模型来解决多智能体强化学习在长期决策中表现不佳的问题, DEC 由记忆索引模块, 隐式分位数决策网络和值分布分解模块 3 个模块组成. 其中, 记忆索引模块通过回顾并利用历史决策经验, 为长期决策提供有力支持; 隐式分位数决策网络则通过直接建模回报分布, 有效地捕捉了回报的不确定性; 而值分布分解模块则负责将回报分布进行细致分解, 从而辅助单个智能体进行更为精准的策略学习.本文在 SMAC 的多种场景下进行了实验, 实验结果表明, DEC 模型在算法收敛速度和最终效果上均显著优于基线方法. 该模型通过高效利用历史决策经验并直接建模回报分布, 为多智能体强化学习在长期预测场景的应用提供了新的方法. 接下来的研究包括分析不同的记忆索引方式和记忆利用方式对实验效果的影响, 以及通过引入新的值分解方法提高值分布分解的可解释性, 从而进一步提升模型的性能和可解释性.

参考文献

1 丁世飞, 杜威, 张健, 等. 多智能体深度强化学习研究进展. 计算机学报, 2024, 47(7): 1547–1567.

2 Oroojlooy A, Hajinezhad D. A review of cooperative multi-agent deep reinforcement learning. Applied Intelligence, 2023, 53(11): 13677–13722. [doi: 10.1007/s10489-022-04105-y]

3 Ning ZP, Xie LH. A survey on multi-agent reinforcement learning and its application. Journal of Automation and Intelligence, 2024, 3(2): 73–91. [doi: 10.1016/j.jai.2024.02.003]

4 Shou ZY, Chen X, Fu YJ, *et al*. Multi-agent reinforcement learning for Markov routing games: A new modeling paradigm for dynamic traffic assignment. Transportation Research Part C: Emerging Technologies, 2022, 137: 103560. [doi: 10.1016/j.trc.2022.103560]

5 任璐, 柯亚男, 柳文章, 等. 基于优势函数输入扰动的多无人艇协同策略优化方法. 自动化学报, 2025, 51(4): 824–834.

6 Orr J, Dutta A. Multi-agent deep reinforcement learning for multi-robot applications: A survey. Sensors, 2023, 23(7): 3625. [doi: 10.3390/s23073625]

7 Lowe R, Wu Y, Tamar A, *et al*. Multi-agent actor-critic for mixed cooperative-competitive environments. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6283–6393.

8 Rashid T, Farquhar G, Peng B, *et al*. Weighted QMIX: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 855.

9 Wang JH, Zhang Y, Gu YJ, *et al*. SHAQ: Incorporating shapley value theory into multi-agent Q-learning. Proceedings of the 36th International Conference on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2022. 430.

10 Wang JH, Ren ZZ, Liu T, *et al*. QPLEX: Duplex dueling multi-agent Q-learning. Proceedings of the 9th International Conference on Learning Representations. Vienna: OpenReview.net, 2021.

11 Wang TH, Gupta T, Mahajan A, *et al*. RODE: Learning roles to decompose multi-agent tasks. Proceedings of the 9th International Conference on Learning Representations. Vienna: OpenReview.net, 2021.

12 Zheng LL, Chen JR, Wang JH, *et al*. Episodic multi-agent reinforcement learning with curiosity-driven exploration. Proceedings of the 35th International Conference on Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2021. 287.

13 Na H, Seo Y, Moon IC, *et al*. Efficient episodic memory utilization of cooperative multi-agent reinforcement learning. Proceedings of the 12th International Conference on

Learning Representations. Vienna: OpenReview.net, 2024.

14 Na H, Moon IC. LAGMA: Latent goal-guided multi-agent reinforcement learning. Proceedings of the 41st International Conference on Machine Learning. Vienna: OpenReview.net, 2024.

15 Schneider L, Frey J, Miki T, *et al*. Learning risk-aware quadrupedal locomotion using distributional reinforcement learning. Proceedings of the 2024 IEEE International Conference on Robotics and Automation. Yokohama: IEEE, 2024. 11451–11458.

16 Kim D, Lee K, Oh S, *et al*. Trust region-based safe distributional reinforcement learning for multiple constraints. Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2024. 874.

17 Wang KW, Zhou K, Wu RZ, *et al*. The benefits of being distributional: Small-loss bounds for reinforcement learning. Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2023. 107.

18 Sun WF, Lee CK, Lee CY. DFAC framework: Factorizing the value function via quantile mixture for multi-agent distributional Q-learning. Proceedings of the 38th International Conference on Machine Learning. JMLR, 2021. 9945–9954.

19 陈妙云, 王雷, 盛捷. 基于值分布的多智能体分布式深度强化学习算法. 计算机系统应用, 2022, 31(1): 145–151. [doi: 10.15888/j.cnki.csa.008237]

20 Du XQ, Chen HC, Wang C, *et al*. Robust multi-agent reinforcement learning via Bayesian distributional value estimation. Pattern Recognition, 2024, 145: 109917. [doi: 10.1016/j.patcog.2023.109917]

21 Zhu CX, Dastani M, Wang SH. A survey of multi-agent deep reinforcement learning with communication. Autonomous Agents and Multi-agent Systems, 2024, 38(1): 4. [doi: 10.1007/s10458-023-09633-6]

22 Luis CE, Bottero AG, Vinogradska J, *et al*. Value-distributional model-based reinforcement learning. The Journal of Machine Learning Research, 2024, 25(1): 298.

23 Bellemare MG, Dabney W, Munos R. A distributional perspective on reinforcement learning. Proceedings of the 34th International Conference on Machine Learning. Sydney: JMLR, 2017. 449–458.

24 Rashid T, Samvelyan M, De Witt CS, *et al*. Monotonic value function factorisation for deep multi-agent reinforcement learning. The Journal of Machine Learning Research, 2020, 21(1): 178.

25 Dabney W, Ostrovski G, Silver D, *et al*. Implicit quantile networks for distributional reinforcement learning. Proceedings of the 35th International Conference on Machine Learning. Stockholmsmässan: PMLR, 2018. 1104–1113.

26 Dabney W, Rowland M, Bellemare MG, *et al*. Distributional reinforcement learning with quantile regression. Proceedings of the 32nd AAAI Conference on Artificial Intelligence. New Orleans: AAAI Press, 2018. 2892–2901.

27 Samvelyan M, Rashid T, de Witt CS, *et al*. The StarCraft multi-agent challenge. Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. Montreal: International Foundation for Autonomous Agents and Multiagent Systems, 2019. 2186–2188.

(校对责编: 张重毅)