小 型 微 型 计 算 机 系 统
Journal of Chinese Computer Systems
DOI:10.20009/j.cnki.21-1106/TP.2024-0487
2025 年 12 月 第 12 期
Vol.46 No.12 2025

时空注意力驱动的分组异步多智能体强化学习框架

陈 涛[1],唐静峰[1],成科扬[1],彭长生[2]

[1](江苏大学 计算机科学与通信工程学院,江苏 镇江 212013)
[2](江苏科海智能系统有限公司,江苏 镇江 212009)
E-mail:kycheng@ujs.edu.cn

摘 要: 在无人驾驶、智能制造和自动化物流等实际应用中,智能体需要高效协同以应对复杂多变的场景.然而,现有的合作模型对合作动态变化的刻画仍显不足.为解决这一问题,提出一种结合动态时空注意力机制和分组异步学习策略的多智能体强化学习框架.该框架能够更好地捕捉智能体之间的时空协作特性,并提高系统的训练效率与稳定性.框架中的动态时空注意力网络通过时域卷积网络分析智能体的轨迹,扩展卷积范围以捕捉更大范围的依赖关系,即使没有显式位置编码,也能通过多层卷积逐步聚合上下文信息,提升时空特征的表达能力.通过计算智能体间的动态影响权重,模型能够优化关键注意力的分配,从而提升多智能体的协作效率,特别是在复杂动态合作任务中.此外,分组异步更新模块通过将智能体分组并异步更新,显著提高训练效率和稳定性.组内智能体采用同步更新策略,组间则采用异步更新,从而减少梯度波动,增强系统的鲁棒性.实验结果表明,该方法在保持高效性和鲁棒性的同时,能够更全面地建模智能体间复杂的协作动态关系.

关 键 词: 多智能体系统;多智能体强化学习;动态时空注意力;分组异步更新;协作智能体
中图分类号: TP18 **文献标识码:** A **文 章 编 号:** 1000-1220(2025)12-2876-08

Spatio-temporal Attention-driven Grouped Asynchronous Multi-intelligent Body Reinforcement Learning Framework

CHEN Tao[1],TANG Jingfeng[1],CHENG Keyang[1],PENG Changsheng[2]

[1](School of Computer Science and Communication Engineering,Jiangsu University,Zhenjiang 212013,China)
[2](Jiangsu Kehai Intelligent System Co.,Ltd,Zhenjiang 212009,China)

Abstract: In practical applications such as driverless driving,intelligent manufacturing,and automated logistics,intelligent agents need to collaborate efficiently to cope with complex and changing scenarios. However,existing cooperation models still fall short in describing dynamic changes of cooperation. To address this issue,a multi-agent reinforcement learning framework combining dynamic spatiotemporal attention mechanism and grouped asynchronous learning strategy is proposed. This framework can better capture spatiotemporal collaboration characteristics between agents and improve training efficiency and stability of system. The dynamic spatiotemporal attention network in framework analyzes trajectories of agents through Temporal Convolutional Network(TCN),extending convolution range to capture wider range of dependencies. Even without explicit position encoding,multi-layer convolution gradually aggregates contextual information,enhancing expression ability of spatiotemporal features. By calculating dynamic influence weights between agents,model can optimize allocation of key attention,thereby improving collaboration efficiency of multi-agents,especially in complex dynamic cooperation tasks. Additionally,grouped asynchronous update module significantly enhances training efficiency and stability by grouping agents and updating them asynchronously. Agents within group adopt synchronous update strategy,while agents between groups use asynchronous updates,reducing gradient fluctuations and enhancing robustness of system. Experimental results demonstrate that this method can more comprehensively model complex collaborative dynamic relationships between agents while maintaining efficiency and robustness.

Keywords: multi-intelligent systems;multi-intelligent reinforcement learning;dynamic spatio-temporal attention;grouped asynchronous updating;collaborative intelligences

0 引 言

在多智能体强化学习(MARL)的研究领域中,合作与协调机制的设计长期以来都占据着至关重要的位置.然而,随着多智能体系统在无人驾驶、智能制造和虚拟环境等实际应用中的需求日益增长,现有方法在面对复杂动态环境时暴露出了一些明显的局限性.这些局限性不仅影响了系统的学习效率,还对智能体间协作的有效性提出了挑战.

收稿日期:2024-10-12 收修改稿日期:2024-12-02 基金项目:国家自然科学基金项目(62372215,61972183)资助;江苏省科技计划专项资金项目(BE2022781)资助. 作者简介:**陈 涛**,男,1999 年生,硕士研究生,CCF 学生会员,研究方向为机器学习;**唐静峰**,男,1998 年生,硕士研究生,CCF 会员,研究方向为计算机视觉;**成科扬**(通信作者),男,1982 年生,博士研究生,教授,CCF 会员,研究方向为计算机视觉;**彭长生**,男,1967 年生,硕士研究生,讲师,研究方向为模式识别.

早期研究主要聚焦于集中式学习策略,例如 Lowe 等人提出的 MADDPG,通过集中训练、分散执行的框架,在训练阶段智能体可以访问全局信息,而在执行阶段则只依赖局部观测[1-4]. 这种方法显著减少了学习负担,为该领域的进展铺设了基石. 然而,随着应用场景的复杂性增加,单纯依赖集中训练的方法难以适应高度动态和不确定的环境,特别是在智能体需要实时作出决策的情况下,这种方法的局限性愈加明显.

随后,注意力机制的引入标志着 MARL 领域的一个转折点[5-10]. 通过增强对智能体之间交互的细腻程度,研究者如 Sunehag 等人的 VDN 和 Rashid 等人的 QMIX 利用价值分解方法应对多智能体环境的复杂性[11,12]. 然而,这些方法仍然存在协作不足的问题,智能体往往倾向于独立行动而非协同作业,无法充分发挥多智能体系统的整体优势[11-14].

为了进一步增强智能体之间的协作能力,研究者们引入了基于通信的多智能体强化学习策略. 早期的工作,如 Foerster 等人的研究,通过引入差分通信机制,试图在保证信息有效传递的同时减少不必要的数据传输,以提升系统的效率[15]. 然而,随着智能体数量的增加,通信的复杂性也急剧上升,导致信息过载成为系统扩展性的主要障碍. 为了缓解这一问题,Zambaldi 等人提出了通过限制通信范围的方式,以邻近智能体为主,来缓解信息过载的问题[16]. 然而,在复杂应用场景中,如何选择合适的代理邻域仍然是一大挑战,且通信范围的限制可能导致信息共享不足,从而影响整体协作效果.

近年来,图神经网络(GNNs)的兴起为智能体之间的合作建模提供了新的视角[17-23]. Jiang 等人提出的图卷积强化学习模型通过将智能体间的交互关系映射为图结构,捕捉了更丰富的合作特性[24]. 此后,Wang 等人进一步拓展了这一思路,提出了一种结合时间和空间维度的图卷积网络,以动态调整智能体之间的合作关系[25,26]. 这些方法通过引入图结构来捕捉智能体之间的复杂互动关系,然而,它们主要聚焦于当前时间步骤的特征,未能充分考虑时间维度对于合作权重学习的重要性. 而许多合作行为往往是跨越多个时间步骤的,这意味着忽视时间维度的合作建模在动态环境中会显著降低协作的效率和效果.

基于通信的 MARL 策略与基于 GNNs 的学习方法各有侧重,适用于不同的应用场景. 基于通信的 MARL 策略在智能体数量较少时能有效促进信息传递和协作,但在大规模系统中面临信息过载和通信成本高的问题. 而 GNNs 方法虽然在处理大规模多智能体系统时展现了更高的效率和鲁棒性,但其模型复杂度高,训练难度大,且在时间维度上的建模尚需进一步完善.

在策略更新方法方面,Actor-Critic(AC)方法作为一种结合值函数逼近于策略梯度的技术,在多智能体强化学习中寻求平衡直接策略探索与价值估计的优点[27]. AC 框架通过反馈循环迭代更新策略生成的 actor 部分和评估策略价值的 critic 部分,尽管取得了一定的成就,但仍然面临计算效率低下和策略多样性不足的问题. 特别是在多智能体环境中,每一轮迭代中计算优势函数的需求可能导致学习速率的缓慢. A2C 通过多线程同步策略优化,依据所有执行者的反馈进行调整,而 A3C 通过异步更新策略进一步加速学习,增强策略多样性,但它们仍然受到异步更新可能导致的策略不一致性与梯度偏差积累的影响,并且在大规模部署中可能加剧资源竞争和通信成本[28]. 这些方法在探索与利用之间寻找平衡点时常遇到困难,有时会陷入局部最优解,延长收敛时间,限制了系统的整体表现.

针对这些现有方法在处理动态多智能体系统中的局限性,本文提出了一种新的多智能体强化学习方法—时空分组异步演员-评论家(Space-Time Grouping Asynchronous Actor-Critic,STGA-AC). 该方法采用动态时空注意力网络来提取智能体间的时空合作关系,并通过分组异步更新方法对模型参数进行高效更新. 这种创新方式不仅适应环境的快速变化,还能够通过优化智能体间的合作策略,实现更高的学习效率和决策质量,解决了传统方法在可扩展性和实时更新方面的问题. 本文的主要贡献如下:

1)将动态注意力机制与时间卷积网络(TCN)[29-33]相结合,不仅捕捉智能体行为的时间序列特征,还间接蕴含了智能体之间的空间关系,使模型能更精确地区分不同位置上智能体的行为模式. 通过这种结合,模型在低复杂度实验场景中的平均奖励值较基线方法提高了 15% 左右,有效提升了协作效率;

2)引入了分组内同步与组间异步更新策略,有效地平衡了训练效率与模型稳定性. 相比传统方法,该策略显著减少了计算资源竞争与通信瓶颈,在中等复杂度实验场景中使用了分组异步策略的 MADDPG-GA 对比原始的 MADDPG 算法,将平均奖励值提升约 45%;MADDPG-STGA 对比没有使用分组异步策略的 MADDPG-ST 在训练过程中将波动幅度减少了约 10%;

3)在多智能体粒子环境中对所提的 STGA-AC 方法进行了实验评估. 实验结果显示,STGA-AC 在各复杂度场景中的表现均优于其他基线方法,特别是在中等复杂度场景中,收敛成功率达到约 85%. 这些结果不仅验证了 STGA-AC 的高效性与优越性,还展示了其在复杂实际应用场景中的巨大潜力,为多智能体系统注入了新颖的交互理解与更新策略.

1　基础理论

1.1　部分可观测马尔科夫决策过程

多智能体强化学习(MARL)通过扩展经典的马尔可夫决策过程(Decentralized Partially Observable Markov Decision Process,Dec-POMDPs)[34]来对问题进行建模,以适应多智能体环境中的复杂交互. 这一框架可形式化为一个 N 元组的分量表示,具体形式为 $<N,S,\{A_i\}_{i=1}^{N},T>$,其中 N 代表参与的智能体总数,S 定义了所有可能的状态空间,而每个智能体 i 拥有自己的动作集合 $A_i(i=1,\cdots,N)$. 奖励函数 $R_i:S\times A_1\times\cdots\times A_N\rightarrow[0,1]$ 映射出智能体 i 的回报,而状态转移函数 $T=S\times A_1\times\cdots\times A_N\rightarrow[0,1]$ 描述了状态变化的概率,决定下一步的可能性.

具体地,聚焦于部分可观测的马尔可夫博弈场景,智能体各自接收到局部观测信息 o_i,并根据其局部观测值决定其动作 $a_i\in A$,联合动作表示为 $a=(a_1,\cdots,a_N)\in A$,智能体会根据此构建策略 $\pi_i:o_i\rightarrow P(A_i)$,其中 $P(A_i)$ 表示行动的概率分布,P 是一个状态转移函数. 这意味着每个智能体基于其有限的局部视角去选择行动. 智能体追求的目标是最大化其累积奖

励,定义为 $R_i=\sum_{t=0}^{T}\gamma^T r_i^T$,这里是折扣因子,位于[0,1]区间内,以平衡即时与长远利益.

在本研究中,设定一个完全合作的环境,每个智能体 i 仅能接收到局部观测 o_i. 对于任一时序 t,每个智能体基于其策略确定动作 π_i,集体行动后,环境回馈一个全局奖励 r. 该奖励信号被用来驱动网络的更新过程,优化策略,以期达成更高效的合作与奖励收获.

1.2 演员-评论家(actor-critic)算法

actor-critic 算法是一种结合价值方法和策略梯度方法优势的强化学习技术,旨在高效地学习多智能体在复杂环境中的最优策略. 它通过分离学习过程为两个关键角色来实现这一目标:一个是负责探索和决策的"演员"(actor),另一个是评估这些决策质量的"评论家"(critic).

演员部分对应于策略函数 $\pi_\theta(a|s)$,其任务是基于当前状态 s 学习选择最佳动作 a 的策略. 简而言之,演员就像是舞台上的演员,依据剧本(策略)表演,其目标是最大化长期奖励. 在 actor-critic 框架中,演员根据评论家提供的反馈来调整其策略参数,以便逐渐提升其在环境中的表现. actor 的更新通常基于梯度原则,通过策略梯度来优化策略 θ,使得期望回报最大. 在策略梯度更新中,优势函数 $A(s,a)$ 起着关键作用,衡量相对于当前策略的额外收益,优势函数可以用动作价值函数 $Q(s,a)$ 和状态价值函数 $V(s)$ 来计算:

$$A(s,a)=Q(s,a)-V(s) \tag{1}$$

策略梯度的更新公式可以表示为:

$$\nabla J(\theta)=\mathrm{E}_{s\sim d^\pi,a\sim\pi_\theta}[\nabla_\theta\log\pi_\theta(a|s)A(s,a)] \tag{2}$$

这里 d^π 是遵循策略 π 下的状态分布,$s\sim d^\pi$ 表示从遵循策略 π 的状态分布中抽取的一个状态 s. 而 $a\sim\pi_\theta$ 表示在状态 s 下从一个遵循策略 π_θ 的动作分布中抽取一个动作 a.

评论家则扮演评估者的角色,它估计当前策略下状态的价值函数 $V(s)$ 或动作价值函数 $Q(s,a)$. 这为演员提供了关于其行动质量的即时反馈,类似于观众或影评人对演员表现的评判. 评论家利用时序差分(Temporal Difference,TD)学习来更有效地估计值函数,从而减少策略梯度方法中常见的高方差问题,加快学习进程,TD 误差的定义如下:

$$\delta=r+\gamma V(s_{t+1})-V(s_t) \tag{3}$$

评论家更新状态价值函数 $V(s)$ 或动作价值函数 $Q(s,a)$ 的规则为:

$$V(s)\leftarrow V(s)+\beta\delta \tag{4}$$

$$Q(s,a)\leftarrow Q(s,a)+\beta\delta \tag{5}$$

其中,β 控制了当前值函数 $V(s)$ 更新的幅度.

结合二者,Actor-Critic 算法能够实现在线更新,即在每个时间步根据最新的状态和奖励信息调整策略,无需等待整个序列结束. 这显著提高了学习效率,尤其是在处理具有大量状态或连续动作空间的任务时.

2 时空分组异步演员-评论家(STGA-AC)方法

2.1 STGA-AC 框架总览

STGA-AC 的框架结构如图 1 所示,这是一个结合动态时空注意力机制和分组异步更新方法的多智能体强化学习框架. 首先,智能体的观测轨迹通过时间卷积网络(TCN)提取出时间特征. 接着,利用权重网络计算智能体间的动态影响权重,形成注意力矩阵,从而建模智能体之间的合作关系. 通过软注意力机制和特征更新步骤,每个智能体根据其他智能体的重要性自我更新. 最后,使用分组内同步与组间异步更新对智能体的学习策略进行更新.

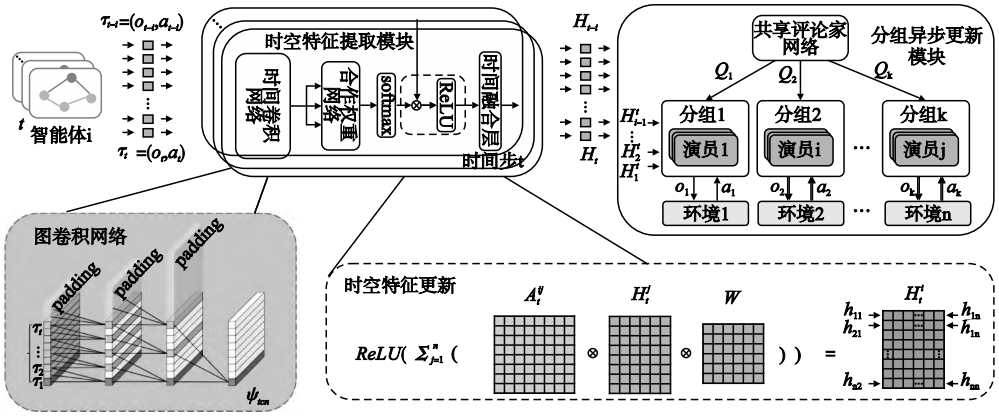

图 1 STGA-AC 示意图

Fig. 1 Illustration of STGA-AC

2.2 动态时空注意力

多智能体强化学习中,智能体的长期行为对于多智能体之间的合作关系建模至关重要. 在多智能体系统中,每个智能体会从其他智能体那里收集关于其观察到的环境和采取的行动的信息,本文将利用这些数据对智能体之间的时空合作关系进行建模. 这一过程中,计算每个代理 i 的时空间依赖特征 H_t^i,需要综合所有智能体的局部观测 $o=(o_1,\cdots,o_N)$ 和行动 $a=(a_1,\cdots,a_N)$,其中 $i\in\{1,\cdots,N\}$.

下面将详细描述动态时空注意力网络的构建过程. 首先,将每个智能体的观测轨迹 τ_i 输入到一个时间卷积网络中 ψ_{tcn},该模型负责从时间序列中提取合作特征 f_{tc_i}. 这些特征在不同的时间步 t 被送入一个权重网络 φ_w,计算智能体 i 在该时间步相对于其他智能体 j 的影响权重 w_t^{ij}. 这样,对于每个时间步 t,输出的权值向量 w_t^i 包括了与所有其他智能体的权

重 $w_t^{i1}, w_t^{i2}, \cdots, w_t^{iN}$,反映了智能体 i 在多智能体环境中与其他智能体的动态合作关系.

通过卷积运算,TCN 模型可以灵活地处理时间序列数据,适合捕捉局部时间特征中的合作信息. TCN 模型由 N 层时间卷积层构成,每一层都包括了一个扩展的一维卷积滤波器 W_{Ft}和残差连接 W_{Fr},同时使用 ReLU(·)激活函数. 具体来说,对于输入的序列特征 $x \in \boldsymbol{R}^{\mathrm{n}}$的第 i 个元素 x_i,扩展卷积的计算公式 F 可以表达为:

$$F(i) = (x \times W_{Ft})(i) = \sum_{j=0}^{k-1} W_{Ft} \cdot x_{i-d \cdot j} \tag{6}$$

其中 $x_{i-d \cdot j}$表示在时间步向前看 $d \cdot j$ 个时间步的输入特征,d 是膨胀因子,k 是滤波器的大小.

每个时间卷积(TC)层接收前一层的输出作为其输入. 特别地,第 1 层以智能体的历史轨迹作为输入特征. 在每层中,使用 ReLU 激活扩展卷积滤波器的输出 H_{TCN_k},计算如下:

$$\hat{H}_{TCN_k} = ReLU(W_{Ft} * H_{TCN_{k-1}} + b_t) \tag{7}$$

$$H_{TCN_k} = H_{TCN_{k-1}} + W_{Fr} * \hat{H}_{TCN_k} + b_r \tag{8}$$

其中 $*$ 表示卷积算子,$W_{Ft} \in \mathrm{R}^{3 \times n_f \times n_f}$为核数为 3 的扩展卷积滤波器的权重,$n_f$ 为该滤波器的个数. $W_{Fr} \in R^{1 \times n_f \times n_f}$为残差连接中 1 维卷积的权值,$b_t$ 和 b_r 是偏置项. 通过扩大卷积和增加 TC 层数可以扩大模型的感受野,公式表示为:

$$rf(k) = 3^{(k+1)} - 1 \tag{9}$$

在此基础上,本文提出一种新的方法,以动态注意力机制构建智能体间的合作关系. 此方法利用 TCN 模型输出的权重 w_i 来形成一个动态注意力权重矩阵 A_t^i. 应用 softmax 函数,为每个智能体分配对其他智能体的关注程度. 计算公式如下:

$$A_t^{ij} = \frac{\exp(w_t^{ij})}{\sum_{k=1}^{n} \exp(w_t^{ik})} \tag{10}$$

其中 A_t^{ij} 表示智能体 i 在时间 t 对智能体 j 的注意力权重. 使用 softmax 确保每个智能体的注意力权重在所有智能体上的总和为 1,从而提供一种归一化的方式来解释这些权重.

通过使用动态注意力矩阵和智能体的特征,可以对智能体的特征进行更新. 这一步骤模拟智能体根据其他智能体的重要性进行自我更新的过程. 特征更新如下:

$$H_t^i = RELU(\sum_{j=1}^{n} A_t^{ij} H_t^j W) \tag{11}$$

其中,智能体 i 的特征向量矩阵 H_t^i 表示基于当前时间步所有其他智能体的特征向量矩阵 H_t^j 加权求和,并结合可学习参数矩阵 W.

为了进一步利用时间信息,增加一个时间融合层来整合各时间步的特征,增强模型对时间动态的理解. 这通过以下方式实现:

$$H_t^i = RELU(TemporalConv([H_t^i, H_{t-1}^i])) \tag{12}$$

其中 TemporalConv(·)是一个一维卷积操作,目的是整合智能体过去的特征信息,以加强其时间序列的连贯性.

时空动态注意力模块输出的特征 H_t^i 通过整合每个智能体的时间和空间信息,将捕捉智能体之间复杂的交互关系. 这些特征反映了智能体在环境中的动态合作情况,为后续多智能体强化学习模型提供智能体之间的时空特征支持.

为了实现上述机制,时间卷积网络(TCN)设定了以下参数:卷积核大小为 3,使用 3 层时间卷积层,滤波器数量分别为 64、128、256. 每层卷积后采用 ReLU 激活函数,并使用 0.2 的 dropout 率以防止过拟合. 权重网络采用两层全连接结构,包含 128 和 64 个神经元,每层后均使用 ReLU 激活. 最终生成的注意力矩阵维度为 $n \times n$,其中 n 为智能体数量,动态调整智能体间的合作关系.

2.3　分组异步更新

本节将详细介绍如何对 actor 进行分组,并异步更新全局模型的过程.

为了充分利用计算资源并减少同步更新时的等待延迟,本文将工作线程按组分配,每个组包含若干个线程. 分组的目的是减少更新的等待时间,并充分利用计算资源. 设 N 为总线程数,K 为组数(可以根据任务的并发性和每组的计算能力灵活调整组的大小和数量),则每组大约包含 $N \div K$ 个线程. 接着,每个线程独立执行并与环境交互,收集一系列的轨迹:

$$\tau_k^g = (a_0, o_0, \cdots, a_T, o_T) \tag{13}$$

其中,τ_k^g 表示第 k 个线程在组 g 中采集的轨迹数据. 每个线程计算其轨迹的累积奖励和梯度.

在分组异步更新算法中,优势函数 $A(s_t, a_t)$是核心组成部分,用于指导策略更新. 优势函数定义为动作的期望回报与当前状态的值函数的差值,即:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t) \tag{14}$$

然而,由于直接计算 $Q(s_t, a_t)$需要未来所有回报的信息,此处采用 TD error(Temporal Difference error)作为优势函数的近似:

$$r_t + \gamma V(s_{t+1}) - V(s_t) \tag{15}$$

其中 γ 是折扣因子.

每个线程根据自己从环境中采样的轨迹 τ_i 计算策略和价值函数的梯度. 这里的优势函数用于指导组内策略(actor)的梯度计算:

$$\nabla_\theta L_{actor,k}^g(\theta) = \sum_t \nabla_\theta \log \pi_\theta(a_t | s_t)(r_t + \gamma V_\theta(s_{t+1}) - V_\theta(s_t)) \tag{16}$$

对于价值函数(critic)的更新,则利用均方误差损失:

$$\nabla_{\theta_v} L_{critic,k}^g(\theta_v) = \sum_t 2(V_{\theta_v}(s_t) - (r_t + \gamma V_{\theta_v}(s_{t+1})))\nabla_\theta V_{\theta_v}(s_t) \tag{17}$$

其中,$L_{actor,k}^g$和 $L_{critic,k}^g$分别表示组 g 中第 k 个线程计算的策略网络和价值网络的损失.

组内所有线程的策略和价值函数梯度被汇总,并平均化以减少单个样本的噪声影响:

$$\Delta\theta_g = \frac{1}{M} \sum_{k=1}^{M} \nabla_\theta L_{actor,k}^g(\theta) \tag{18}$$

$$\Delta\theta_{v_y} = \frac{1}{M} \sum_{k=1}^{M} \nabla\theta_v L_{critic,k}^g(\theta_v) \tag{19}$$

该方法的主要动机是利用组内的数据多样性来减少梯度估计的方差,并避免频繁的全局同步,从而加速学习过程而不牺牲性能.

组内所有线程完成任务并汇总梯度后,使用这些累积的梯度异步更新全局模型:

$$\theta \Leftarrow \theta + \alpha \Delta\theta_g \tag{20}$$

$$\theta_v \Leftarrow \theta_v + \alpha_v \Delta\theta_{v_y} \tag{21}$$

其中 α 和 α_v 分别是策略和价值函数的学习率,这一更新过程依然保持异步的特性,即不同组之间的更新是独立的.

更新全局模型后,组内的所有线程从全局模型中同步最新的网络参数 θ 和 θ_v. 这确保了所有线程都能基于最新学到的策略继续进行探索和学习,同时保证了全局知识的及时传播.

在具体实现中,STGA-AC 使用了基于组内同步和组间异步更新的策略. 通过动态调整每组智能体的大小和更新频率,框架在提升训练速度的同时,保持了模型的稳定性. 这种策略有效地减少了智能体之间的策略冲突,并显著加快了在复杂环境中的学习过程.

3 实验与分析

本节在多智能体粒子仿真环境(MPE)[35,36]中测试了 STGA-AC 处理多种复杂任务的表现,这是 MARL 领域中被广泛使用以评估算法性能的仿真环境. 首先在 3 种不同环境复杂度的捕食者-猎物任务中进行对比实验,以证明 STGA-AC 的整体表现;然后在其中两种可以相对明显表现学习表现的场景中进行消融实验,来验证动态时空注意力和分组异步更新方法对性能的影响.

3.1 捕食者-猎物任务

实验环境为二维连续的捕食者-猎物环境,如图 2 所示. 场景中包含 4 种类型的实体:F 片森林、S 个食物、Z 个障碍物和智能体(K 个捕食者和 L 个猎物). 捕食者需要合作追赶猎物,成功捕获猎物会获得奖励,反之则会受到惩罚. 环境中的捕食者可以利用环境中的固定实体为己方获取优势,包括利用森林进行埋伏,以及将猎物驱赶至障碍物处以限制其活动范围,但同时猎物也可以利用这些实体进行躲藏. 环境中的捕食者可以利用环境中的固定实体为己方获取优势,包括利用森林进行埋伏,以及将猎物驱赶至障碍物处以限制其活动范围,但同时猎物也可以利用这些实体进行躲藏.

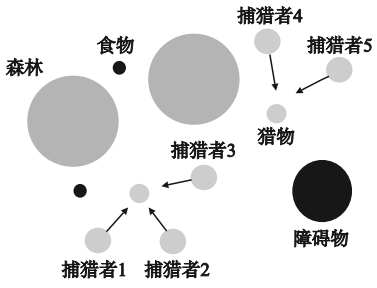

图 2 捕食者-猎物场景示例

Fig. 2 Example of the predator-prey scenario

通过调整场景中实体的数量,设计了 3 种不同复杂度的仿真场景. 所有实验均在 CPU Intel Xeon Gold 5218R 和 GPU Nvidia RTX 3090 上使用 5 个随机种子构建,具体实验配置如表 1 所示.

3.2 对比实验

本实验场中猎物的策略使用了 DQN[37],而本文提出的 STGA-AC 将与多种基线方法作为捕食者的策略进行对比实验. 实验选择了 MADDPG[1]、反事实多智能体策略梯度方法(counter-factual multi-agent policy gradient, COMA)[4]、使用软注意力的多智能体强化学习方法(multi-agent actor-critic, MAAC)[38]这 3 种基于 CTDE 框架的 MARL 算法作为基线方法.

表 1 捕食者-猎物环境的实验设置

Table 1 Experimental configuration for predator-prey environments

K	L	F	S	Z	环境复杂度
4	3	2	–	–	低
4	3	2	1	–	中
6	2	2	2	1	高

在 3 个不同复杂度的捕食者-猎物场景中进行 1500 轮训练,同时记录每种方法的平均奖励值. 在实验中,为确保 STGA-AC 与基线方法的公平比较,所有模型均使用相同的超参数设置. 具体包括:评论家网络和行动者网络的学习率均设为 0.001,优化器使用 Adam. Adam 优化器结合了动量和 RMSprop 的优点,能够自适应地调整每个参数的学习率,从而加速收敛并提高模型性能. 在多智能体系统中,尤其是在复杂的捕食者-猎物实验场景中,状态和动作空间都非常大,模型需要在大量数据和复杂交互中进行有效学习. Adam 优化器的自适应学习率特性使得模型能够在训练初期快速收敛,并在后期保持稳定的性能,这对于长时间的训练过程尤为重要. 此外,Adam 优化器计算效率高,对超参数不敏感,具有良好的泛化能力和稳定的收敛性,特别适合处理大规模数据集和复杂模型. 折扣因子(γ)设为 0.99,探索率(ε)初始值为 1.0,并且每 100 轮训练后减小 0.05,直到达到最小值 0.1. 在具体实现中,STGA-AC 框架采用了动态分组策略,每组初始包含 2 个智能体,组内同步更新每 5 个时间步执行一次,组间异步更新每 15 个时间步执行一次. 组内的成员可以根据训练中的表现和任务需求进行调整,以提高不同智能体之间的协作效率. 本实验中使用随机打乱的规则进行重分组,这种策略的主要依据是提高智能体之间的合作多样性和适应性,防止因固定分组而产生的局部最优问题,同时通过探索更多的合作方式,进一步提升多智能体系统的整体协作效率和鲁棒性.

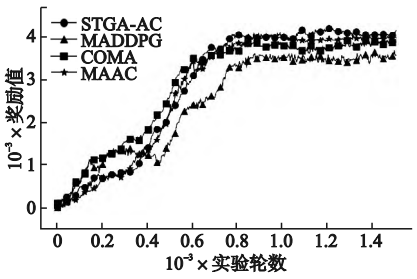

图 3 低复杂度环境下的性能对比

Fig. 3 Performance comparison on low complexity environments

图 3 ~ 图 5 展示了保持训练参数相对一致时,各个方法在捕食者-猎物场景中奖励得分的平均值. 在简单复杂度的场景中,本文提出的 STGA-AC 刚开始的表现并不突出,同样使用注意力机制的 MAAC 也有相同的问题,但最终这两种方法均能收敛到更高的平均奖励值. STGA-AC 对比 MAAC,更复杂的网络设计带来前期的学习速度相对落后,后期则占据了

得分上的优势;在中等复杂度的场景中,各个方法之间的差距体现得更加明显,没有使用注意力机制的 MADDPG 和 COMA 相对表现不佳,COMA 使用了反事实的思想来区分单个智能体对系统奖励的贡献,一定程度上带来了性能上的提升,这也体现在了奖励值曲线上. 而使用了注意力机制的 STGA-AC 和 MAAC 均取得了不错的得分. STGA-AC 由于特征提取网络更加复杂,动态时空注意力模块能够有效捕捉智能体之间的时空依赖关系,使得智能体在决策时充分考虑到其他智能体的状态和行动,从而在复杂环境中获得更优的策略. 然而,这种复杂的网络设计在训练初期导致了学习速度较慢,表现并不佳,并且中期出现了较大的波动. 尽管如此,随着训练的深入,STGA-AC 逐渐提取出更有效的特征,并利用这些特征在中等复杂度的环境中找到了独立性与合作性之间的平衡,最终后期依然取得了最高的平均奖励值且收敛到了相对稳定的策略;在复杂度最高的场景中,4 种方法的表现都不佳,MADDPG 甚至难以学习到有效的策略,相比之下其他 3 种方法虽然最终获得了一定的得分,但是依旧无法收敛到平稳的策略,STGA-AC 相对于其他 3 种方法虽然最终获得的平均奖励值略高,但也没有取得明显优势.

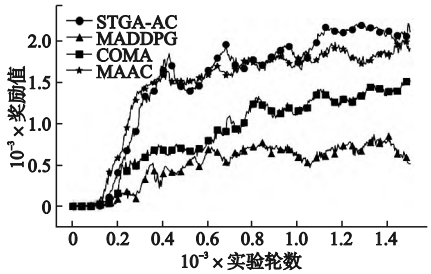

图 4　中等复杂度环境下的性能对比

Fig. 4　Performance comparison on medium complexity environments

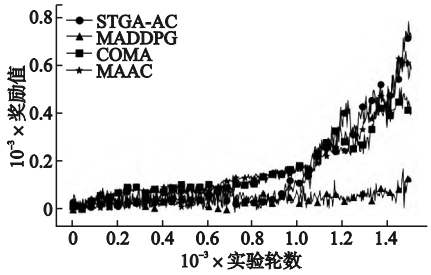

图 5　高复杂度环境下的性能对比

Fig. 5　Performance comparison on high complexity environments

由此可见,STGA-AC 在处理场景相对复杂的任务时,有着很明显的性能优势. 这主要得益于其动态时空注意力机制和出色的特征提取能力,使得模型能够更好地捕捉智能体间复杂的时空合作关系,从而在多变的环境中更有效地学习合作策略. 然而,在低复杂度的场景中,由于 STGA-AC 的网络结构较为复杂,模型需要更多的时间来调整和适应其复杂的特征提取与注意力机制,因此需要更长的训练时间才能达到策略收敛. 而在处理更高复杂度的任务时,尽管 STGA-AC 具备优势,但由于环境的不确定性和智能体间高度动态的交互,模型在学习过程中可能会出现波动和不稳定的情况,类似的问题也在其他基线算法中出现,表明在极其复杂的环境中,寻找有效策略的学习依然充满挑战.

3.3　消融实验

为了测试动态时空注意力模块和分组异步更新方法对性能提升的影响,本文设计了消融实验. 实验对比了原始的 MADDPG,使用动态时空注意力模块而不使用分组异步更新方法的 MADDPG-ST,使用分组异步更新方法而不使用动态时空注意力模块的 MADDPG-GA,以及使用了动态时空注意力模块和分组异步更新方法的 MADDPG-STGA. 实验环境选择了可以相对明显表现学习曲线变化的低复杂度及中等复杂度的捕食者-猎物环境.

图 6、图 7 展示了不同多智能体粒子环境下的消融实验. 首先,比较 MADDPG 和 MADDPG-ST 的性能表现. 实验结果表明,在低复杂度的捕食者-猎物环境中,MADDPG-ST 对比原始的 MADDPG 有着一定的性能提升,但是 MADDPG-ST 的策略收敛速度却并不占优势,这可能是复杂的网络结构为训练带来了负担. 在中等复杂度的场景中,MADDPG 无法学习到比较好的策略,问题根源可能在于在复杂环境里,MADDPG 中的价值函数估计存在偏差, 导致智能体行为偏离最优路径,甚至学会非最优策略. 此外,MADDPG 在执行阶段缺乏有效的通信学习机制,每个智能体的决策主要依赖于自己的局部观测,没有充分利用其他智能体的信息,而智能体的最佳行动通常依赖于其他智能体的状态和行动. 相比之下,MADDPG-ST 充分利用了其他智能体的观测,带来显著的性能提升,最终也学习到了相对稳定的策略.

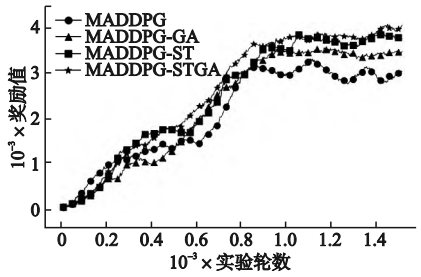

图 6　低复杂度环境下的消融实验

Fig. 6　Ablation experiments on low complexity environment

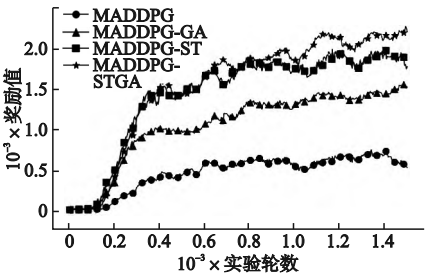

图 7　中等复杂度环境下的消融实验

Fig. 7　Ablation experiments on medium complexity environment

进一步,验证分组异步更新方法的有效性. 对比 MADDPG 和 MADDPG-GA,在低复杂度的捕食者-猎物环境中,

MADDPG-GA 对比原始的 MADDPG 性能提升的同时也实现了更快且更稳定的策略收敛. 在中等复杂度的环境中,分组异步更新带来的性能提升更加明显.

最后,在两个场景中,同时使用动态时空注意力和分组异步更新方法的 MADDPG-STGA 均获得了最佳的平均得分. 在低复杂度的场景中,MADDPG-STGA 能够快速收敛到最佳且平稳的策略,这主要得益于动态时空注意力机制的作用,该机制使智能体能够有效捕捉和利用其他智能体的行为信息,优化协作关系. 在中等复杂度的环境中,尽管 MADDPG-STGA 的学习过程中出现了一定的波动,但相比仅使用动态时空注意力的方法,分组异步更新策略的引入有效减少了学习过程中的波动幅度,最终实现了最高的平均得分,并收敛到相对稳定的策略. 可见,动态时空注意力和分组异步更新方法的结合对性能提升有显著作用,且二者共同作用的改进效果远胜于单独使用其中任一方法.

4 结 论

本文提出一种结合动态时空注意力机制与分组异步更新策略的多智能体强化学习框架,该框架通过时间卷积网络(TCN)有效提取智能体行为的序列特征,最终生成的特征不仅反映时间上的动态变化,还间接蕴含空间关系. 动态注意力机制不仅促进了智能体间的高效合作学习,还考虑了智能体在特定时间步长的相对重要性. 分组异步更新策略通过智能体的分组内同步与组间异步更新,可显著提高学习效率与模型的稳定性. 实验结果证实,所提方法在复杂多变环境下展现出优越的协作性能与学习效率,为智能体的动态合作关系建模提供了新的视角.

未来的工作将从 3 个方面进一步拓展与深化:首先,探索更先进的图神经网络结构以更好地捕捉智能体间的复杂依赖关系,特别是在大规模多智能体系统中;其次,探索并优化更合理的分组和重分组策略,以提高多智能体系统在动态环境中的协作效率,并增强模型的适应能力;最后,研究更为细致的奖励机制设计,以促进智能体间的公平合作与长期策略一致性,确保系统整体效益最大化.

References:

[1] Lowe R, Wu Yi, Tamar A, et al. Multi-agent actor-critic for mixed cooperative competitive environments [C]//Proceedings of the 31st Int Conf on Neural Information Processing Systems, 2017: 6382-6393.

[2] Wan K, Wu D, Li B, et al. ME-MADDPG: an efficient learning based motion planning method for multiple agents in complex environments[J]. International Journal of Intelligent Systems, 2022, 37(3): 2393-2427.

[3] Al Saffar M, Gül M. Data-efficient MADDPG based on self-attention for IoT energy management systems[J]. IEEE Access, 2023, 11: 109379-109389, doi: 10.1109/access.2023.3322193.

[4] Chen L, Zhang H, Xiao J, et al. Counterfactual critic multi-agent training for scene graph generation[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 4613-4623.

[5] Cheng J, Li N, Wang B, et al. High sample efficient multiagent reinforcement learning for navigation and collision avoidance of uav swarms in multitask environments[J]. IEEE Internet of Things Journal, 2024, 11(22): 36420-36437.

[6] Pu Z, Wang H, Liu Z, et al. Attention enhanced reinforcement learning for multi agent cooperation[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 34(11): 8235-8249.

[7] Wen M, Kuba J, Lin R, et al. Multi-agent reinforcement learning is a sequence modeling problem[J]. Advances in Neural Information Processing Systems, 2022, 35: 16509-16521.

[8] Dabre R, Fujita A. Recurrent stacking of layers for compact neural machine translation models[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2019: 6292-6299.

[9] Xie J, Ajagekar A, You F. Multi-agent attention-based deep reinforcement learning for demand response in grid-responsive buildings[J]. Applied Energy, 2023, 342: 121162, doi: 10.1016/j.apenergy.2023.121162.

[10] Wang Y, Shang F, Lei J, et al. Dual-attention assisted deep reinforcement learning algorithm for energy-efficient resource allocation in industrial internet of things[J]. Future Generation Computer Systems, 2023, 142: 150-164, doi: 10.1016/j.future.2022.12.009.

[11] Sunehag P, Lever G, Gruslys A, et al. Value-decomposition networks for cooperative multi-agent learning[C]//Proceedings of the 17th Int Conf on Autonomous Agents and Multi Agent Systems, 2018: 2085-2087.

[12] Rashid T, Samvelyan M, Schroeder C, et al. Qmix: monotonic value function factorisation for deep multi-agent reinforcement learning [C]//Proc of the 35th Int Conf on Machine Learning, 2018: 4295-4304.

[13] Rashid T, Farquhar G, Peng B, et al. Weighted qmix: expanding monotonic value function factorisation for deep multi-agent reinforcement learning[J]. Advances in Neural Information Processing Systems, 2020, 33: 10199-10210, doi: 10.48550/arxiv.2006.10800.

[14] Hostallero W J K D E, Son K, Kim D, et al. Learning to factorize with transformation for cooperative multi-agent reinforcement learning[C]//Proc of the 36th Int Conf on Machine Learning, 2019: 5887-5896.

[15] Foerster J, Assael I A, De Freitas N, et al. Learning to communicate with deep multi-agent reinforcement learning[J]. Advances in Neural Information Processing Systems, 2016, 29, doi: 10.48550/arxiv.1605.06676.

[16] Zambaldi V, Raposo D, Santoro A, et al. Relational deep reinforcement learning[J]. arXiv e-prints, arXiv: 1806.01830.2018.

[17] Xiao J, Wang Z, He J, et al. A graph neural network based deep reinforcement learning algorithm for multi-agent leader-follower flocking[J]. Information Sciences, 2023, 641: 119074, doi: 10.1016/j.ins.2023.119074.

[18] Du W, Ding S, Zhang C, et al. Multiagent reinforcement learning with heterogeneous graph attention network[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 34(10): 6851-6860.

[19] Liu L, Gurney N, McCullough K, et al. Graph neural network based behavior prediction to support multi-agent reinforcement learning in military training simulations[C]//Winter Simulation Conference (WSC), 2021: 1-12.

[20] Bernárdez G, Suárez Varela J, López A, et al. Magnneto: a graph neural network-based multi-agent system for traffic engineering

[J]. IEEE Transactions on Cognitive Communications and Networking,2023,9(2):494-506.

[21] Wu Z,Pan S,Chen F,et al. A comprehensive survey on graph neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems,2020,32(1):4-24.

[22] Zhou J,Cui G,Hu S,et al. Graph neural networks:a review of methods and applications[J]. AI Open,2020,1:57-81,doi:10.48550/arxiv.1812.08434.

[23] Ding S,Du W,Ding L,et al. Learning efficient and robust multi-agent communication via graph information bottleneck[C]//Proceedings of the AAAI Conference on Artificial Intelligence,2024:17346-17353.

[24] Jiang J,Dun C,Huang T,et al. Graph convolutional reinforcement learning[J]. arXiv preprint arXiv:1810.09202,2018.

[25] Wang Y,Xu T,Niu X,et al. STMARL:a spatio-temporal multi-agent reinforcement learning approach for cooperative traffic light control[J]. IEEE Transactions on Mobile Computing,2020,21(6):2228-2242.

[26] Ivanovic B,Pavone M. The trajectron:probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision,2019:2375-2384.

[27] Konda V R,Tsitsiklis J N. Actor-citic agorithms[C]//Proceedings of the 12th International Conference on Neural Information Processing Systems,1999:1008-1014.

[28] Mnih V,Badia A P,Mirza M,et al. Asynchronous methods for deep reinforcement learning[C]//International Conference on Machine Learning,2016:1928-1937.

[29] Hewage P,Behera A,Trovati M,et al. Temporal convolutional neural(TCN)network for an effective weather forecasting using time-series data from the local weather station[J]. Soft Computing,2020,24(21):16453-16482.

[30] Ma Y,Shen M,Zhang N,et al. OM-TCN:a dynamic and agile opponent modeling approach for competitive games[J]. Information Sciences,2022,615:405-414,doi:10.1016/j.ins.2022.08.101.

[31] Zhang Y,Gu T,Zhang X. MDLdroid:a ChainSGD-reduce approach to mobile deep learning for personal mobile sensing[J]. IEEE/ACM Transactions on Networking,2021,30(1):134-147.

[32] Fan J,Zhang K,Huang Y,et al. Parallel spatio-temporal attention-based TCN for multivariate time series prediction[J]. Neural Computing and Applications,2023,35(18):13109-13118.

[33] Meydani A,Shahinzadeh H,Ramezani A,et al. Comprehensive review of artificial intelligence applications in smart grid operations[C]//9th International Conference on Technology and Energy Management,2024:1-13.

[34] Dibangoye J S,Amato C,Buffet O,et al. Optimally solving Dec-POMDPs as continuous-state MDPs[J]. Journal of Artificial Intelligence Research,2016,55:443-497,doi:10.1613/jair.4623.

[35] Luke S,Cioffi Revilla C,Panait L,et al. Mason:a multiagent simulation environment[J]. Simulation,2005,81(7):517-527.

[36] Ma Y,Shen M,Zhao Y,et al. Opponent portrait for multiagent reinforcement learning in competitive environment[J]. International Journal of Intelligent Systems,2021,36(12):7461-7474.

[37] Mordatch I,Abbeel P. Emergence of grounded compositional language in multi-agent populations[C]//Proceedings of the AAAI Conference on Artificial Intelligence,2018,doi:10.1609/aaai.v32i1.11492.

[38] Iqbal S,Sha F. Actor-attention-critic for multi-agent reinforcement learning[C]//International Conference on Machine Learning,2019:2961-2970.