DOI:10.19652/j.cnki.femt.2305370

基于注意力机制的信息预处理多智能体强化学习算法

杜泳韬 赵岭忠 翟仲毅
(桂林电子科技大学计算机与信息安全学院 桂林 541004)

摘 要:多智能体强化学习在群体控制领域具有广泛应用,然而传统的强化学习方法(如Q-Learning或策略梯度)在多智能体环境中表现不佳。在训练过程中,每个智能体的策略不断变化。当一个智能体基于环境信息做出决策时,其他智能体的决策可能已经影响了环境信息,导致智能体感知的转移概率分布和奖赏函数发生变化,使得环境变得非平稳,训练无法有效进行。为了缓解这一问题,研究了一种基于多头自注意力的多智能体强化学习算法。该方法考虑了其他智能体的行动策略,利用多头自注意力算法使智能体能够学习对决策影响最大的因素,成功地学习了复杂的多智能体协调策略。在实验结果中平均回报达值到了0.82,远高于传统算法的表现。实验结果表明,所提出的基于多头自注意力的多智能体强化学习算法能够有效解决环境不平稳导致的多智能体学习困难问题,提高了多智能体强化学习算法的收敛速度和平稳性。

关键词:多智能体强化学习;多头自注意力;信息预处理;策略梯度;非平稳
中图分类号:TP301 **文献标识码**:A **国家标准学科分类代码**:520.10

Attention-based information preprocessing multi-agent reinforcement learning algorithm

Du Yongtao Zhao Lingzhong Zhai Zhongyi
(School of Computer and Information Security, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: Multi-agent reinforcement learning has a broad range of applications in group control. However, traditional reinforcement learning methods, such as Q-learning or policy gradient, prove unsuitable for multi-agent environments. As training progresses, the strategy of each agent undergoes changes. When one agent makes decisions based on environmental information, the decisions of other agents may have already influenced the environment's information, leading to changes in the transition probability distribution and the reward function perceived by the agent. This renders the environment non-stationary, hindering the training process. To address these issues, this paper explores a multi-agent reinforcement learning algorithm based on multi-head self-attention. The approach considers the action strategies of other agents and utilizes a multi-head self-attention algorithm to enable agents to learn the most influential factors in the environment, successfully acquiring complex multi-agent coordination policies. In the experimental results, the average return value reaches 0.82, which is much higher than the performance of traditional algorithm. Experimental results demonstrate the effectiveness of the proposed multi-agent reinforcement learning algorithm based on multi-head self-attention in overcoming challenges related to the non-stationary environment, thereby enhancing the convergence speed and stability of the multi-agent reinforcement learning algorithm.

Keywords: multi-agent reinforcement learning; multi-head self-attention; information preprocessing; policy gradient; non-stationary

0 引 言

多智能体强化学习在近年来受到广泛关注和研究[1]。在多智能体环境[2]中,智能体互相协作[3]、相互合作、竞争,以实现特定任务的目标。在这一过程中,不同的智能体需要考虑其他智能体的行动策略[4],以及全局奖励的变化,使得多智能体强化学习问题变得更加复杂。为了解决这些问题[5],研究人员提出了各种方法[6-7]。其中包括基于智能体通信学习的方法,如每个智能体独立训练从门机制中学习如何通信IC3Net(individualized controlled con-

收稿日期:2023-07-31

tinuous communication model)[8]，使用连续通信网络(communication neural net，CommNet)[9]等。这些方法允许智能体之间相互通信，提高协同能力。然而，这些方法通常会占用大量带宽，容易导致通信错误和消息冗余等问题，从而限制了它们的应用范围。另一方面，基于协作学习的方法包括价值分解网络(value-decomposition networks，VDN)[10]，混合网络将局部价值函数组合的QMIX[11]，以及改进后通过约束条件来调整局部价值的QTRAN[12]，使用演员评论家方法的方法，如添加反事实基线的COMA[13]，混合场景下的多智能体深度确定性策略MADDPG[14]。这些方法允许智能体之间共享信息，提高协同能力。然而，这些方法仍然存在一些不足，例如难以处理全局奖励取平均的信息、价值函数分解问题、局部最优问题，以及难以应用于大规模智能体环境等。为了解决复杂环境中信息冗余问题[15]，本文提出了一种基于多头自注意力[16]的多智能体强化学习算法。该方法考虑了其他智能体的行动策略，并对存放在经验池中的信息进行信息预处理，以使每个智能体在训练期间能够收集到更有效的信息。此外，采用多头自注意力算法，使智能体能够学习环境中对决策影响最大的因素，从而成功地学习复杂的多智能体协调策略。

1 相关知识

1.1 预训练的多头自注意力模型

多头自注意力模型是一种广泛应用于处理多信息输入的注意力机制变体。它通过计算不同信息之间的相似性来获取关键信息，减少其他因素的干扰。本文基于多头注意力机制模拟智能体之间的信息交互，有效降低模型训练和智能体决策的复杂度。其注意力机制原理如下：

$$attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{1}$$

设计中，矩阵$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$由多个智能体状态共同输入产生。通过计算点乘$\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}$来衡量相似性，为避免数值过大，进行了$\sqrt{d_k}$的缩放，并使用softmax函数进行归一化，最后与矩阵V相乘，得到各智能体状态的权重。

多头自注意力机制是注意力机制的变体，在计算机视觉领域取得了显著成功。和普通注意力机制相似，输入为$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$矩阵，不同之处在于对这3个矩阵进行多次计算，融合不同的注意力信息，从而提取更详细的特征。融合信息公式如下：

$$\boldsymbol{Y}=\underset{i\in[h]}{concat}[attation^{i}(\boldsymbol{X})] \tag{2}$$

式中：$\boldsymbol{Y}$表示融合的信息；$\boldsymbol{X}$表示输入的各智能体的状态；h表示注意力头的数量；$\boldsymbol{W}_0$表线性变化；$attation^{i}(\boldsymbol{X})$表示第$i$个注意力头的计算结果。

在信息输入中，输入向量是由每个智能体的策略信息向量组成的矩阵，经过线性变换得到查询向量、键向量和值向量。查询向量与键向量的点积被用作相似性得分，用于计算值向量的加权和，得到加权值向量。这样，每个智能体都能从其他智能体的策略信息中提取相关特征，更好地理解其他智能体的策略，并在多智能体环境中做出更好的决策。

1.2 多智能体AC(actor-critic)算法

AC算法[17]是一种结合了值和策略的强化学习算法，用于解决连续动作控制问题。包含两个网络，Actor网络和Critic网络。Actor网络用于输出决策策略，而Critic网络则用于估计当前状态下的值函数，对Actor网络的决策进行评估和指导。

AC算法使用策略$\pi(a\mid s)$表示输出动作的概率，$Q^{\pi(a|s)}$表示采取策略获得的奖励值，策略梯度更新如下：

$$g=E\left[\sum_{t=D}^{\infty}\varphi_t\nabla_{\theta}\log\pi_{\theta}(\boldsymbol{a}_t\mid\boldsymbol{s}_t)\right] \tag{3}$$

本文提出了多智能体AC算法，其本质为了扩展Critic对于其他智能体策略的信息学习功能[18]，每个智能体能够对其他智能体的策略进行函数逼近。假设有n个智能体，其策略化参数为$\theta=\{\theta_1,\theta_2,\cdots,\theta_n\}$，策略$\pi=\{\pi_1,\pi_2,\cdots,\pi_n\}$，对于每个智能体，其策略更新梯度为：

$$\nabla_{\theta_i}J(\theta_i)=$$
$$E_{s\sim p^{\pi},a_i\sim\pi_i}[\nabla_{\theta_i}\log\pi_i(a_i\mid\boldsymbol{o}_i)Q_i^{\pi}(\boldsymbol{x},a_1,\cdots,a_n)] \tag{4}$$

式中：$Q_i^{\pi}(\boldsymbol{x},a_1,\cdots,a_n)$是集中化的动作值函数，所有智能体动作$\boldsymbol{a}_1,\cdots,\boldsymbol{a}_n$的输入，$x$是状态信息，输出为所有智能体的$Q$值[19]。每个智能体都有自己的$Q_i^{\pi}$函数，因此可以有不同的奖励函数，从而完成合作与竞争的任务。将式(4)的策略梯度算法扩展到确定性的策略μ_{θ_i}，得到新的策略梯度为：

$$\nabla_{\theta_i}J(\mu_i)=$$
$$E_{x,a\sim D}[\nabla_{\theta_i}\mu_i(a_i\mid\boldsymbol{o}_i)\nabla_{a_i}Q_i^{\mu}(\boldsymbol{x},a_1,\cdots,a_n)] \tag{5}$$

式中：$D=(\boldsymbol{x},\boldsymbol{x}',a_1,\cdots,a_n,r_1,\cdots,r_n)$为经验存储；$Q_i^{\mu}$为每个智能体的值函数，解决了从单智能体到多智能体的功能转换问题。

用$\bar{Q}_i^{\mu'}$表示目标网络，$\mu'=[\mu'_1,\cdots,\mu'_n]$表示目标策略的滞后，集中式Critic的更新公式如下：

$$\begin{cases}L(\theta_i)=E_{x,a,r,x'}[(Q_i^{\mu}(\boldsymbol{x},a_1,\cdots,a_n)-y^2]\\ y=r_i+\bar{Q}_i^{\mu'}(\boldsymbol{x}',a'_1,\cdots,a'_n)\mid_{a'_j=\mu'_j(o_j)}\end{cases} \tag{6}$$

从上述的多智能体AC算法可以看出，Actor仅使用了局部观察信息，Critic使用了全部的信息来进行学习。即使每个智能体的策略信息不断更新，整个环境也能保持稳定。

2 基于信息预处理的多智能体强化学习算法征提取

2.1 环境模型

如图1所示，多智能体AC算法融合了多头自注意力模型，以实现更高效的多智能体协同学习。

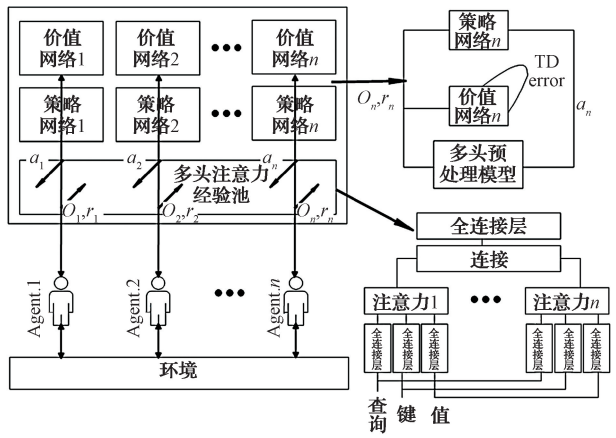

图1 多智能体交互模型

首先，每个智能体观测到周围环境信息，随后引入多头自注意力模型对信息进行处理。该模型计算不同信息的相似性，提取更细致的特征，促进智能体之间的信息交互。多头自注意力模型使用矩阵 $\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$ 的点积来衡量不同信息之间的相似性。这种方法允许模型在多个头之间共享信息，通过在不同的子空间学习不同的特征，提高特征的表征能力。处理好的向量信息随后存放于经验池，供已训练的模型使用。这样的方式使得智能体能够更好地理解其他智能体的策略，从而在多智能体环境中做出更好的决策。

其次，采用 AC 算法，结合值和策略的方法，实现单步更新深度强化学习。Actor 网络根据当前状态选择动作，Critic 网络评估动作值函数。这使得智能体能够在每一步中更新策略，实现更快的学习。Critic 网络被扩展以学习其他智能体的策略信息，通过引入其他智能体的状态和动作作为额外输入。这样，Critic 网络中引入额外的输入来实现的，这些输入包括其他智能体的状态和动作。这样，Critic 网络能够学习其他智能体的策略信息[20]，并用于评估动作值函数。采用集中化的动作值函数和确定性策略梯度，促使多智能体之间的协同学习。集中化的动作值函数使得智能体在训练过程中能够共享信息，更好地协同学习。确定性策略梯度通过计算梯度的期望值来更新策略，降低训练过程中的方差。

结合多头注意力机制的 AC 模型，能够降低模型训练和智能体决策的复杂度。该方法提高了算法的效率。实验结果表明，与传统的 AC 算法相比，多智能体 AC 算法能够稳定地收敛，并在多种多智能体环境中取得了优异的性能。引入多头自注意力机制后，算法的收敛速度更快，在更短的时间内获得更好的性能表现。

2.2 信息预处理方法

首先，将多智能体的状态输入到多头自注意力层中，使用多头自注意力机制来提取有用的信息，然后将其输出到输出层。输出层通常用于将多头自注意力层的输出转换为所需的输出格式，例如动作。最终，将输出的动作返回给环境，让智能体执行这些动作，从而在环境中获得奖励并更新模型参数。该算法通过引入多头自注意力模型计算不同信息的相似性，提取更细节的特征，实现智能体之间的信息交互。这种方法带来了多重优势，不需要智能体之间的通信，减轻了通信负担，解决了全局奖励问题，并且适用于大规模智能体的环境。本文提出的基于多头自注意力的多智能体强化学习算法的算法框图如图 2 所示。该算法框架包括 3 个主要部分，多头自注意力模型、多智能体 AC 算法和重放缓冲区。在每个时间步骤中，智能体观察到环境状态，并将其作为输入提供给多头自注意力模型。该模型将多智能体状态作为输入，计算多头自注意力的输出，然后将其作为多智能体 AC 算法的输入，以计算动作和奖励。智能体采取动作并接收奖励，并将其存储在重放缓冲区中，供后续使用。

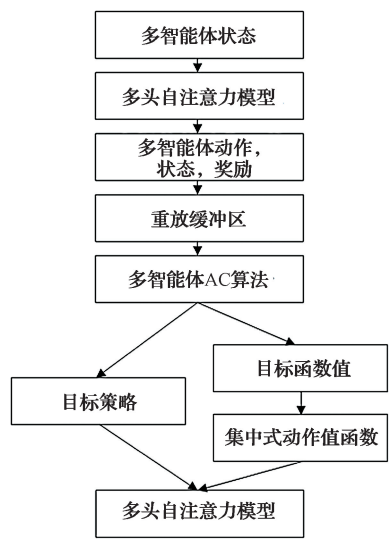

图2 多头自注意力的多智能体强化学习算法的算法框图

MA-HAN 算法是一种基于多智能体强化学习的算法，利用多头自注意力模型提取多智能体之间的信息交互，从而实现更好的学习效果。该算法由如下步骤组成。

1）初始化，在开始训练前，对算法进行初始化设置，包括网络架构和超参数等方面的设置。获取状态和观测，在每一步训练中，每个智能体需要获取当前状态 $\boldsymbol{X}$ 和其他智能体的观测 $\boldsymbol{O}_i$，这些信息将用于选择动作和更新策略。

2）选择动作和探索，每个智能体使用其策略函数 $\mu\theta_i$ 来选择动作 a_i，并使用随机过程 N 来进行探索，以便发现新的状态和奖励。

3）执行动作序列和观察新状态，在执行动作序列后，每个智能体会收到一个奖励 r，并观察到新的状态 $\boldsymbol{X}'$。

4）存储经验，智能体将经验存储到回放缓冲区中，以便在以后的训练中进行学习。

5）随机采样经验样本，从回放缓冲区中随机采样一批经验样本，用于更新每个智能体的 Q 值函数。每个经验

样本包含一个状态 $\boldsymbol{X}$、一个动作 A、一个奖励 R、下一个状态 $\boldsymbol{X}'$和一个标记 S，表示该经验样本是否是最后一个样本。

6)是否继续采样，如果已经采样了足够的经验样本，则进入下一步骤；否则返回步骤2)进行下一次动作选择。

7)更新智能体的 Q 值函数，使用采样得到的经验样本，以更好地预测每个动作的价值。

8)更新策略参数，使用更新后的 Q 值函数，更新每个智能体的策略参数，以提高策略的质量。

9)更新目标网络参数，为了减少更新引起的高度相关性和不稳定性，使用目标网络参数来更新智能体的 Q 值函数和策略参数。

随机采样的大小可以根据需要进行设置，通常选择一定数量的经验样本进行训练，以确保智能体在训练过程中具有足够的样本来学习和调整策略参数。

综上所述，MA-HAN算法是一种基于多智能体强化学习的算法，使用多头自注意力模型对多智能体信息进行加权，使用回放缓冲区存储每个智能体的经验，并通过 Q 值函数和策略函数来更新每个智能体的策略参数。通过使用目标网络参数来减少更新引起的高度相关性和不稳定性，该算法在复杂的多智能体环境中实现了协作和学习，以达成共同目标。

2.3 MA-HAN算法

首先，该算法采用了一个信息预处理阶段，将输入向量转换为查询向量、键向量和值向量。这一步骤可以帮助更好地处理输入向量，并减少了输入向量的维度。通过这种方式，可以更好地理解其他智能体的策略，并在多智能体环境中做出更好的决策。例如，在多智能体环境中，不同智能体之间的策略和行为可能存在相互影响的情况。通过使用查询向量、键向量和值向量，可以更好地理解这些影响，并在决策时考虑它们。

MA-HAN算法

1 对于 episode=1 到 M 进行每回合探索
2 初始化一个随机过程 N 进行动作探索
3 接受初始状态 $\boldsymbol{X}$
4 从 t 时开始到最大时限
5 每个智能体 i 选择动作 $a_i=\mu_\theta(o_i)+N_i w.r.t$ 当前策略和探索
6 执行动作序列 $a=(a_1,\cdots,a_N)$ 并观察奖励 r 和新状态 $\boldsymbol{X}'$
7 将(状态 $\boldsymbol{X}$，动作 a_0，回报 r，新状态)放入多头注意力模型中
8 从多头注意力模型分析得到数据存储到回放缓冲区中
9 状态 $\mathbf{X}$ 变为 $\mathbf{X}'$
10 对于每个智能体来说
11 从 D 中随机采样 S 个样本 (x^j,a^j,r^j,X'^j)
12 设置 $y^j=r_i^j+rQ_i^{\mu'}(X'^j,a'_1,\cdots,a'_n)|a'_k=\mu'_k(o_k^j)$
13 通过最小化损失来更新 $critic\ L(\theta)=1/S\ \Sigma_j(y^j-Q_i^\mu(X^j,a_1^j,\cdots,a_k^j))^2$
14 通过采样策略梯度更新 $\nabla_{\theta i}J\approx 1/S\ \Sigma_j\nabla\theta_i\mu_i(o_i^j)\nabla aiQ_i^\mu(x^j,a_1^j,\cdots a_n^j)|\ a_i=\mu_i(o_i^j)$
15 更新每个代理 i 的目标网络参数
16 $\theta'_i\leftarrow J\theta_i+(1-\tau)\theta'_i$

其次，算法采用了多头注意力机制，将输入向量分成多个头，以获得更好的特征表示。每个头都有自己的查询向量、键向量和值向量，然后使用点积计算相似性得分并进行加权计算。这样，可以从多个角度更好地理解输入向量，并利用这些信息做出更好的决策。例如，在多智能体环境中，不同智能体可能存在不同的策略和行为。通过使用多头注意力机制，可以更好地理解这些策略和行为，并在决策时采取更合理的行动。

综上所述，算法的着重点在于采用了信息预处理阶段和多头注意力机制，以更好地处理输入向量并获得更好的特征表示。有助更好地理解多智能体环境中的情况，并在决策时做出更好的决策。为多智能体强化学习算法的研究和应用提供了新的思路和方向。

3 实 验

3.1 实验环境介绍

为了验证本文提出的基于多头自注意力的多智能体强化学习算法，采用了文献[21]提出的实验环境。该环境包含了 N 个智能体和 L 个地标，这些智能体和地标都在连续的空间和离散时间的二维世界中。在环境中，智能体具有一定概率执行物理动作，或者与其他智能体进行信息交互。同时，本文考虑了智能体之间的合作与竞争关系，即在某些情境下，智能体之间需要进行一定的通信才能获得最大奖励，而在其他情境下，智能体之间可能存在相互冲突的目标。

1) 协作沟通

(1)环境由说话者和聆听者两个智能体组成，放置在一个拥有3种颜色地标的环境中。

(2)在每个回合中，聆听者需要移动到特定颜色的地标，并根据到指定地标的距离分配奖励。

(3)说话者需要根据聆听者的动作来指定地标的颜色，以实现协作沟通的目标。

2) 协作导航

(1)在这个环境中，智能体需要通过物理协作到达一组地标。

(2)目标智能体需要观察其他智能体和地标的位置，根据智能体与各地标的距离获得相同的奖励。

3) 远离

(1)场景包括 L 个地标和 $N+M$ 个智能体。其中，有一个是目标地标。

(2)N 智能体需要尽可能靠近目标地标以获得最大奖励。

(3)M 个智能体必须要阻止 N 个智能体靠近目标地标,但是每个智能体都不知道其他智能体是合作还是竞争关系,需要从智能体的动作中进行推断。

4) 物理欺骗

(1)N 个智能体合作以到达目标地标,从而获得奖励。

(2)对手智能体也想到达目标地标,但不知道其位置。

(3)为了欺骗对手,合作的智能体需要分散覆盖所有地标,否则会受到惩罚。

5) 捕食者与被捕食者

N 个相对较慢的合作智能体需要在环境中追逐速度较快的对手,但会有 L 个大地标阻碍。当合作的智能体与对手相遇时,智能体会得到奖励,对手则受到惩罚。

捕食者与被捕食者场景下的实验效果如图 3 所示。

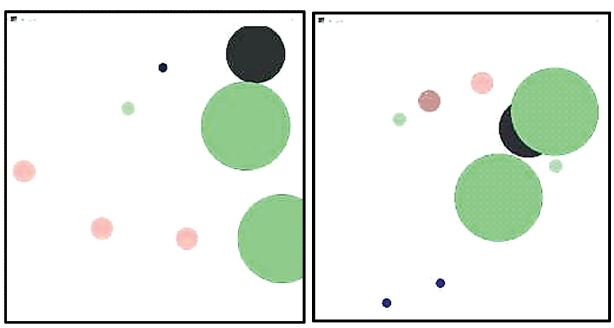

图 3 捕食者与被捕食者实验

3.2 实验模型训练

在模型训练中,使用了一个多智能体环境(MultiAgentEnv)来模拟不同智能体之间的交互。在每个 episode 开始时,采用 Ornstein-Uhlenbeck 随机过程作为探索策略,以探索智能体的动作空间。每个智能体选择一个动作,执行这些动作并观察奖励和新的状态。

本文使用 PyTorch 实现了基于多头自注意力的多智能体强化学习算法。多头自注意力层接收每个智能体的观察作为输入,并输出每个智能体的编码。通过引入多头自注意力模型,能够加权不同智能体的信息。接着,使用 Q 值函数和策略函数来更新每个智能体的策略参数。采用 Adam 优化器和 0.001 的学习率来更新目标网络。采用 0.1 的 τ 值来更新目标网络参数。设置 $\gamma=0.92$ 来平衡奖励的未来价值。

在训练中,随机选择了 10 个不同的种子进行训练,这些环境(合作沟通,物理欺骗)有明显的成功/失败条件。当重放缓冲区中的 episode 数量达到 512 个后,更新网络参数。通过大量实验验证了算法在不同场景下的性能,并使用不同的种子值以确保结果的可靠性。

1)协作沟通

在一个 3×3 的网格世界中,每个智能体位于不同的网格中,通过通信寻找彼此的位置。在 100 个 episode 的训练后,两位智能体能够准确地定位并会合在一起。

2)物理欺骗

两个智能体被放置在一个带有障碍物的房间中,其中一个智能体知道出口的位置,另一个不知道。在 200 个 episode 的训练后,导航智能体能够通过多头自注意力有效地向被困智能体提供出口信息,从而两个智能体成功离开房间。

3.3 实验结果分析

本文实验使用了 3 种不同的算法,传统 AC 算法、多智能体 AC 算法和多智能体 AC +多头自注意力算法。在实验中,随机选择了 10 个不同的种子进行训练,这些环境(合作沟通,物理欺骗)有明显的成功/失败条件。每个算法在每个环境中都进行了 10 000 个 episode 的训练,以评估其性能。

本文实验结果如图 4 所示。

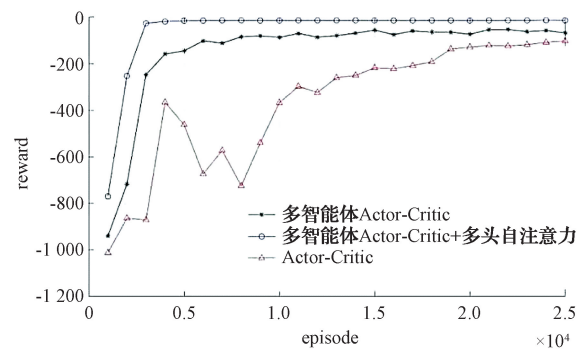

图 4 对比实验结果

从图 4 可以看出,传统的 AC 算法收敛速度较慢,且不够稳定。多智能体 AC 算法相比传统的 AC 算法,能够稳定的收敛,且收敛速度更快。

加入多头自注意力算法后,算法在 10 000 个 episode 前就达到了收敛状态,且非常稳定。然后通过回报含函数作为评价指标,计算平均回报和标准差:

$$y = r_i + \cdot\bar{Q}_i^{\mu'}(\boldsymbol{x}', a'_1, \cdots, a'_n)\big|_{a'_j=\mu'_j(o_j)} \tag{7}$$

每个场景的表现如表 1 所示。

表 1 不同场景下实验比较

实验场景	算法	平均回报	标准差
协作沟通	AC	0.54	0.11
	AC +多头自注意力	0.82	0.05
物理欺骗	AC	0.49	0.14
	AC +多头自注意力	0.82	0.04
远离	AC	0.49	0.07
	AC +多头自注意力	0.81	0.06
协作导航	AC	0.41	0.09
	AC +多头自注意力	0.82	0.06
捕食者	AC	0.39	0.06
	AC +多头自注意力	0.80	0.07

从表1可以看出，在所有场景中，使用多头自注意力的多智能体AC算法的平均回报都明显高于传统的AC算法。同时，标准差也更小，表明该算法更加稳定。特别是在物理欺骗和网格世界这两个场景中，多头自注意力算法的表现尤为突出，平均回报达到了0.82，远高于传统算法的表现。这些实验结果说明，多头自注意力算法可以有效地提高多智能体强化学习算法的性能，并使其更加稳定。

在协作沟通场景中，3种算法的性能都比较接近，但是多智能体AC＋多头自注意力算法仍然表现得更好。接下来将本文方法在MPE的多智能体环境中进行对比试验，设置智能体数量为5。

本文算法在spread中环境中与MADDPG算法的对比如图5所示，其收敛效果也能达到MADDPG算法相应的水准，比MADDPG算法更快收敛。

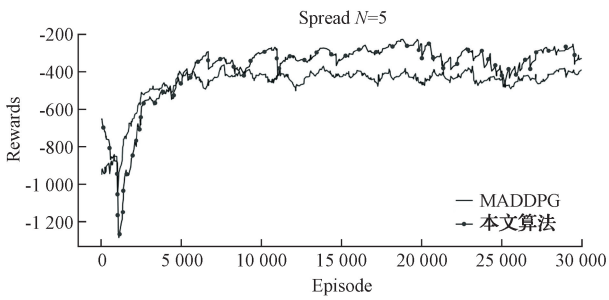

图5 Spread $N=5$环境中算法对比

3.4 实验总结

通过在多智能体环境下实现多头自注意力的多智能体强化学习算法，取得了显著的成果。该算法通过学习几何符号引导语言，表现出卓越的性能，具备成熟的沟通能力，并成功解决了复杂的导航和欺骗问题。实验结果表明，该算法在适用性和泛化能力上表现出色，展现了良好的应用前景。

本文实验进行了许多精心的设计，包括使用不同的种子值、调整批量大小、学习率和更新率等，以优化模型性能。同时，涵盖了多个场景和问题，以测试算法的通用性和鲁棒性。使用了PyTorch框架来实现本文提出的多头自注意力的多智能体强化学习算法。具体来说，使用PyTorch实现了多头自注意力层和输出层，并将其整合到多智能体AC算法中。然后，通过在不同环境中进行大量实验，验证了算法的性能以及在不同问题和场景下的可扩展性和适应性。

除了算法性能的分析，进行了大量的超参数调优和实验结果分析，以确保算法的可靠性和有效性。实验结果表明，本文提出的基于多头自注意力的多智能体强化学习算法能够有效地应对多智能体训练中的问题，并具有广泛的应用前景。

4 结 论

传统的强化学习方法在多智能体环境中面临着巨大的局限性。随着训练的进行，每个智能体的策略都在变化，使得环境变得非平稳。为了应对这一问题，本文提出了基于多头自注意力的多智能体强化学习算法，将多头自注意力算法与多智能体AC结合，使得智能体能够学习到对决策影响最大的特征，从而规避了复杂且繁多的环境影响因素，提高了多智能体强化学习的收敛速度和平稳性。

本文介绍了一种具备强大适用性的基于多头自注意力的多智能体强化学习算法，该算法可应用于合作沟通、物理欺骗、导航等多种场景。在训练过程中，使用多智能体环境模拟了多个智能体之间的交互，并采用了随机过程来探索策略。通过PyTorch框架实现了多头自注意力模型，并利用Q值函数和策略函数更新每个智能体的策略参数。进行了广泛的实验，验证了该算法在不同场景下的性能和可靠性。通过对实验结果的深入分析和比较，证明了该算法的有效性和优越性，为多智能体强化学习算法的研究和应用提供了新的思路和方向。结果表明，该算法能够有效地解决复杂问题，具备强大的通信能力和适应性。该算法在智能机器人、自主驾驶汽车等人工智能领域具有广泛应用前景。

尽管该算法取得了显著的效果，但仍存在一些潜在的限制和改进方向。在一些复杂场景中，算法的学习效率可能较低，需要更多的训练时间和数据。此外，在一些真实场景，算法需要考虑到实际环境的物理限制和安全性，需要进一步完善。

综上所述，本文提出的基于多头自注意力的多智能体强化学习算法不仅具有广泛的应用前景和较强的适应性，而且在智能机器人、自主驾驶汽车等领域有着广泛的应用潜力。本文研究提供了一种新的思路和方法，为多智能体强化学习算法的研究和应用提供了新的方向和参考。

参 考 文 献

[1] 邹启杰，蒋亚军，高兵，等. 协作多智能体深度强化学习研究综述[J]. 航空兵器，2022，29(6)：78-88.

[2] 段伟浩，赵瑾，梁家瑞，等. 基于深度强化学习的多智能体动态寻路算法[J]. 计算机仿真，2023，40(1)：441-446，473.

[3] 杨傲雷，陈燕玲，徐昱琳. 基于强化学习的机器人手臂仿人运动规划方法[J]. 仪器仪表学报，2021，42(12)：136-145.

[4] MNIH V，BADIA A P，MIRZA M，et al. Asynchronous methods for deep reinforcement learning[C]. Proceedings of the 33rd International Conference on International Conference on Machine Learning，2016，48：1928-1937.

[5] 金月，张旭东. 深度多智能体强化学习综述[C]. 第十

六届全国信号和智能信息处理与应用学术会议，2022：109-114.

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

[7] SUKHBAATAR S，KOSTRIKOV I，SZLAM A，et al. Intrinsic motivation and automatic curricula via asymmetric self-play [J]. 6th International Conference on Learning Representations，2018.

[8] SINGH A，JAIN T，SUKHBAATAR S. Learning when to communicate at scale in multiagent cooperative and competitive tasks [C]. 7th International Conference on Learning Representations，2019.

[9] SUKHBAATAR S，ROB F. Learning multiagent communication with backpropagation[J]. Advances in Neural Information Processing Systems，2016，29：2244-2252.

[10] SUNEHAG P，LEVER G，GRUSLYS A，et al. Value-decomposition networks for cooperaive multi-agent learning based on team reward[C]. Proceedings of the 17th Conference on Autonomous Agents and MultiAgent Systems，2018：2085-2087.

[11] RASHID T，SAMVELYAN M，SCHROEDER C，et al. QMIX：Monotonic value function factorisation for deep multi-agent reinforcement learning [J]. The Journal of Machine Learning Research，2020，21(1)：7234-7284.

[12] SON K，KIM D，KANG J W，et al. QTRAN：Learning to factorize with transformation for cooperative multi-agent reinforcement learning[C]. International Conference on Machine Learning，2019：5887-5896.

[13] FOERSTER J，FARQUHAR G，AFOURAS T，et al. Counterfactual multi-agent policy gradients [C]. Proceedings of the AAAI Conference on Artificial Intelligence，2018，32(1)：63-82.

[14] LOWE R，WU Y，TAMAR A，et al. Multi-agent actorcritic for mixed cooperative-competitive environments[J]. Advances in Neural Information Processing Systems，2017，30：6379-6390.

[15] VASWANI A，SHAZEER N，PARMAR N，et al. Attention is all you need [J]. Advances in Neural Information Processing Systems，2017，30：6000-6010.

[16] VOITA E，TALBOT D，MOISEEV F，et al. Analyzing multi-head self-attention：Specialized heads do the heavy lifting，the rest can be pruned[C]. ACL 2019-57th Annual Meeting of the Association for Computational Linguistics，Proceedings of the Conference，2020：5797-5808.

[17] 蔡军，苟文耀，刘颜. 基于 actor-critic 框架的在线积分强化学习算法研究[J]. 电子测量与仪器学报，2023，37(3)：194-201.

[18] 陈亮，梁宸，张景异，等. Actor-Critic 框架下一种基于改进 DDPG 的多智能体强化学习算法[J]. 控制与决策，2021，36(1)：75-82.

[19] MNIH V，KAVUKCUOGLU K，SILVER D，et al. Playing atari with deep reinforcement learning [J]. CoRR，2013，abs/1312.5602.

[20] 臧嵘，王莉，史腾飞. 基于注意力消息共享的多智能体强化学习[J]. 计算机应用，2022，42(11)：3346-3353.

[21] MORDATCH I，ABBEEL P. Emergence of grounded compositional language in multi-agent populations[C]. Proceedings of the AAAI Conference on Artificial Intelligence，2018.

作 者 简 介

杜泳韬，硕士研究生，主要研究方向为人工智能，形式化技术。

E-mail：1358796456@qq.com