面向多智能体协作的注意力意图与交流学习方法

俞文武[1,2] 杨晓亚[1,2] 李海昌[1] 王 瑞[1] 胡晓惠[1]

摘 要 对于部分可观测环境下的多智能体交流协作任务, 现有研究大多只利用了当前时刻的网络隐藏层信息, 限制了信息的来源. 研究如何使用团队奖励训练一组独立的策略以及如何提升独立策略的协同表现, 提出多智能体注意力意图交流算法 (Multi-agent attentional intention and communication, MAAIC), 增加了意图信息模块来扩大交流信息的来源, 并且改善了交流模式. 将智能体历史上表现最优的网络作为意图网络, 且从中提取策略意图信息, 按时间顺序保留成一个向量, 最后结合注意力机制推断出更为有效的交流信息. 在星际争霸环境中, 通过实验对比分析, 验证了该算法的有效性.

关键词 多智能体, 强化学习, 意图交流, 注意力机制

引用格式 俞文武, 杨晓亚, 李海昌, 王瑞, 胡晓惠. 面向多智能体协作的注意力意图与交流学习方法. 自动化学报, 2023, **49**(11): 2311−2325

DOI 10.16383/j.aas.c210430

Attentional Intention and Communication for Multi-agent Learning

YU Wen-Wu[1,2] YANG Xiao-Ya[1,2] LI Hai-Chang[1] WANG Rui[1] HU Xiao-Hui[1]

Abstract For multi-agent communication and cooperation tasks in partially observable environments, most of the existing studies only use the information of the hidden layer of the network at the current time, which limits the source of information. This paper studies how to use team rewards to train a set of independent policies and how to improve the collaborative performance of independent policies. A multi-agent attentional intention communication (MAAIC) algorithm is proposed to improve the communication mode, and an intention information module is added to expand the source of communication information. The network with the best performance in the history of an agent is taken as the intention network, from which the policy intention information is extracted. The historical intention information of the agent that performs best at all times is retained as a vector in chronological order, and combined with the attention mechanism and current observation history information to extract more effective information as input for decision-making. The effectiveness of the algorithm is verified by experimental comparison and analysis on StarCraft multi-agent challenge.

Key words Multi-agent, reinforcement learning, intention communication, attention mechanism

Citation Yu Wen-Wu, Yang Xiao-Ya, Li Hai-Chang, Wang Rui, Hu Xiao-Hui. Attentional intention and communication for multi-agent learning. *Acta Automatica Sinica*, 2023, **49**(11): 2311−2325

多智能体强化学习技术在现实世界中有着广泛的应用, 例如踢球机器人团队[1]、游戏智能[2]、自动驾驶等. 随着深度学习在语音识别[3]、文本翻译和目标检测[4] 等领域的发展, 多智能体强化学习同这些领域技术融合, 已取得了许多成果[5]. 然而, 多智能体强化学习领域仍存在许多开放问题, 比如训练过程非平稳性、维度爆炸和信用分配问题等[6].

本文针对部分可观测的多智能体合作任务, 采用基于值函数的强化学习方法进行研究. 独立 Q 学习 (Independent Q-learning, IQL)[7−8] 是将单智能体强化学习方法直接应用到多智能体问题上的典型代表, 虽然具有很好的扩展性, 但是由于其他智能体的策略在训练过程中不断发生变化, 对单个智能体而言, 所处环境是非平稳的, 在复杂任务上的表现往往不佳. 值函数分解方法在一定程度上能缓解智能体间信用分配和懒惰智能体的问题, 同时每个智能体学习到自身最优的局部值函数, 仅仅利用自身的局部值函数进行决策, 具有很好的扩展性. 因此, 本文基于值函数分解系列方法[9−11], 采用集中式训练和分布式执行训练模式[12]. 在这种模式下, 所有智能体在训练时被统一集中控制, 执行过程中各

收稿日期 2021-05-18 录用日期 2021-09-17
Manuscript received May 18, 2021; accepted September 17, 2021
国家重点研发计划 (2019YFB1405100), 国家自然科学基金 (61802380, 61802016) 资助
Supported by National Key Research and Development Program of China (2019YFB1405100) and National Natural Science Foundation of China (61802380, 61802016)
本文责任编委 金耀初
Recommended by Associate Editor JIN Yao-Chu
1. 中国科学院软件研究所天基综合信息系统重点实验室 北京 100190 2. 中国科学院大学 北京 100049
1. Science and Technology on Integrated Information System Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing 100190 2. University of Chinese Academy of Sciences, Beijing 100049

个智能体被单独控制.

本文在值分解框架的基础上进行改进, 对于维度爆炸问题, 本文训练的所有智能体网络参数共享. 而对于其他智能体的存在, 会导致训练不稳定现象. 本文的应对方案是对其他智能体的策略进行建模, 利用这部分信息可以让智能体处在相对稳定的环境下进行训练.

本文为每个智能体增加额外的公共网络, 将历史表现最优的策略网络保存下来. 在智能体网络参数共享的情况下, 历史性能最优的网络即代表最优的策略, 可以预示智能体将来所要达成的目标. 智能体网络训练最终目的为获取最优策略, 因此历史最优网络能够推测出其他智能体的未来信息. 本文将历史最优网络命名为意图网络, 从意图网络得到的信息称为意图信息. 因此整合这些意图信息作为交流信息, 可以让交流信息包含不同时间段最优网络对当前状态的指导信息, 智能体也可以在最初时, 有更多的基础信息筛选. 而历史最优网络是通过测试过程中智能体每一局平均得分的高低来筛选. 本文将这部分意图信息和观测信息整合起来, 并基于注意力机制进行信息提取, 继而利用整合提取后的信息进行决策, 加速智能体之间的合作学习. 对于历史最优网络是否可以代替意图网络, 本文在小规模的修改版捕食者游戏上, 从内在奖励方面来验证意图网络的可行性.

本文为每个智能体之间增加了交流通道, 通过沟通, 帮助智能体学会更好地合作. 传统的交流方法有直接固定交流信息、交流离散信息或对于连续交流信息做一个简单的加和, 这些方式过于简单, 不适用于复杂的合作任务. 智能体收到的交流消息会随着智能体数目的增加而增加, 过多的消息反而会引入一些干扰信息, 不利于智能体的决策. 因此本文提出为每个智能体增加一个交流通道, 智能体间的交流信息采用多头注意力机制进行提取. 这种基于注意力的交流模式可以更好地处理智能体数目的变化, 具有良好的可扩展性, 智能体的交流网络结构采用门控循环单元 (Gated recurrent unit, GRU)[13] 神经网络.

本文采用集中式训练、分布式执行的训练框架, 提出面向部分可观测合作问题的多智能体注意力意图交流学习算法 (Multi-agent attentional intention and communication, MAAIC). 在训练过程中, 利用额外的意图信息使训练更加高效, 同时改善了智能体间的交流信息结构. 本文算法的主要贡献有:

1) 增加额外的公共网络, 保存其他智能体的历史时刻最优策略, 为智能体决策提供了额外的信息来源;

2) 采用注意力机制, 对意图信息和交流信息进行处理, 从而能够提取到更为有效的信息;

3) 验证增加意图网络的可行性, 从内在意图奖励角度, 对智能体意图信息的可靠性提供理论依据.

本文结构安排如下: 第 1 节介绍相关工作; 第 2 节介绍背景知识; 第 3 节详细介绍本文算法的结构, 包括组成成分与训练过程; 第 4 节为具体实验分析; 第 5 节总结研究结果.

1 相关工作

继单智能体领域取得卓越的性能后[14], 研究人员转而朝向更有难度的多智能体环境上[15−16]. 最简单的多智能体学习方法是每个智能体独立训练学习, 早期的尝试是 IQL[17], 但一般在实际应用中表现不好, 由于其实现简单而且随着智能体数目的增加仍然具有很好的扩展性, 因此很多问题都是用独立智能体学习作为基线来进行实验对比.

对于多智能体合作任务[18−19] 的研究算法, 2017 年, Lowe 等[20] 提出多智能体深度确定性策略梯度, 将深度确定性策略梯度扩展和应用到多智能体领域, 每个智能体都有自己的动作和评论者网络. 在训练阶段, 评论者能够拿到所有智能体的动作, 因此能够用于混合环境下每个智能体都有自己局部奖励的问题. 在多智能体深度确定性策略梯度基础上, 基于存储的多智能体深度确定性策略梯度[21] 算法提出了共享内存作为一种交流模式, 比另一种受到信息丢弃启发的多智能体深度确定性策略梯度[22] 算法, 具有更好的鲁棒性.

另外一种处理全局奖励信用分配问题的方法是值函数分解, 2017 年, Sunehag 等[9] 提出值函数分解网络 (Value-decomposition networks, VDN) 算法, 将一个全局的 Q 函数分解成智能体的局部 Q 函数之和. Q 混合网络 (Q mixing network, QMIX)[10] 在 VDN 的基础上, 添加了一个混合网络, 为分解添加了非线性部分, 并且保证全局 Q 函数对于局部 Q 函数是单调的. 反事实多智能体[23] 算法利用反事实基线, 去衡量智能体在全局奖励中所做的贡献. 然后, 为适用于 QMIX 中满足了分解条件且不满足单调性的任务, Q 因式分解 (Q-transformation, QTRAN)[24] 算法弱化了 QMIX 的结构约束, 从而能够处理更加通用的任务. Q 注意力算法[25] 从理论上给出一种通用的值函数分解方式, 基于多头注意力显式建模智能体对整体的影响, 并将这种影响引入混合网络中, 实现了一种更精确的值函数表示. Q 路径分解算法[26] 提出一种利用积分梯度技巧, 沿轨迹路径分解全局 Q 值的新的值分解方法, 通过积分梯度衡量每个智能体对于全局 Q 值的贡献, 将这部分贡献作为局部Q 值进行监督学习.

在多智能体交流方面, 2016 年, Foerster 等[27] 提出离散和连续的智能体间交流两种方法, 是最早

在深度强化学习中引入交流信息, 旨在缓解离散交流通道的问题, 是深度 Q 网络 (Deep Q-network, DQN) 和 IQL 的结合应用到多智能体问题上. 同年, Sukhbaatar 等[28] 提出简洁的交流网络 (Communication neural net, CommNet), 使用了一个连续的交流通道, 智能体接受从其他智能体传来的信息之和, 算法允许多步交流, 梯度能够通过连续交流通道回传给各个智能体. 双向协作网络[29] 提出双向循环网络构建每个智能体, 智能体之间并不显示交流信息, 而是发生在隐藏空间. 2018 年, Singh 等[30] 提出的独立控制连续通信模型, 为每个智能体添加一个门控, 决定是否同其他智能体进行交流, 从而让智能体学会更好地交流.

在筛选有效信息方面, 注意力模型作为一种成功的方法, 被广泛应用于计算机视觉[31]、自然语言处理[32] 和强化学习. 2018 年, Jiang 等[33] 提出的注意力协作算法, 可以让智能体选择是否进行通信和同哪些智能体进行通信. 有目标的多智能体交流 (Targeted multi-agent communication, TarMAC)[34] 通过基于签名的软注意力机制来衡量消息的相关性, 并进行多轮的交流. 两步注意力图网络[35] 算法使用两阶段注意力网络模型, 分别利用硬注意力机制、确定交互的智能体和软注意力机制, 确定交互的权重, 自动学习大规模复杂游戏中不断变换的智能体间关系.

在多智能体建模领域, 以前的研究多是通过观测学习其他智能体的模型. 2018 年, Raileanu 等[36] 提出的自身模仿其他智能体算法, 利用自身策略, 去预测对手的动作, 从而推断其他智能体的目标信息, 再利用这部分目标信息做决策. 对于那些共享目标任务, 自身模仿其他智能体算法具有很好的表现. 社会性影响方法[37] 是通过一种统一的方法, 去实现多智能体的协调和沟通, 即给予智能体对其他智能体的行为产生因果影响的内在奖励. 但这两种方法都需要额外地使用监督学习方法, 去训练这个预测网络.

这些交流算法大多局限于当前时刻策略网络的隐藏层信息. 本文算法通过增加意图网络, 扩大了交流信息的来源, 且选择经典值分解系列的算法流程作为基础框架, 从意图信息模块和交流模块两个角度进行改善, 提取出有效的交流信息, 使智能体更好地协作. 相比于自身模仿其他智能体算法, 本文的意图网络是直接从历史上挑选出最优策略网络, 不需要额外训练网络.

2　背景知识

多智能体强化学习涉及多个智能体和多个状态, 是马尔科夫决策过程和矩阵博弈的结合, 其中马尔科夫决策过程包含一个智能体和多个状态, 矩阵博弈包含多个智能体和一个状态. 多智能体强化学习的发展与博弈论[38] 是分不开的, 部分可观测的多智能体合作问题可被定义如下: $< N, \mathcal{S}, \mathcal{A}, R, P, \mathcal{O}, \gamma >$, 其中 N 是智能体的数目, $\mathcal{S}$ 代表全局状态空间, $\mathcal{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_N\}$ 是所有智能体的动作空间, R 是奖励函数, $P : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \to \mathcal{S}$ 是环境状态转移函数, $\mathcal{O} = \{\mathcal{O}_1, \cdots, \mathcal{O}_N\}$ 代表所有智能体的观测空间, γ 是折扣因子. 智能体 i 的策略 $\pi_{\theta_i} : \mathcal{O}_i \times \mathcal{A}_i \to [0, 1]$, 智能体 π_i 执行在时刻 t 动作 a_i^t 后, 会从环境中获得奖励 $r_i^t : \mathcal{S} \times \mathcal{A}_i \to R$. 本文在纯合作任务上进行研究, 此时所有智能体获得的奖励相同即 $r_1^t = r_2^t = \cdots = r_N^t = r^t$, 整个任务目标为智能体学到最优合作策略, 从而最大化累计回报 $G = \sum_{t=0}^{T} \gamma^t r^t$.

VDN 算法是基于深度循环 Q 网络提出的值分解结构, 能够在仅有全局收益情况下, 去学习不同智能体的动作价值函数, 从而缓解由于部分可观测导致的伪收益和懒惰智能体问题. VDN 的值分解函数如下:

$$Q\left(\left(h^1, h^2, \cdots, h^d\right), \left(a^1, a^2, \cdots, a^d\right)\right) \approx \sum_{i=1}^{d} \widetilde{Q}_i\left(h^i, a^i\right) \tag{1}$$

QMIX 是在 VDN 基础上进行的改进版本, 采用一个混合网络, 对局部智能体函数进行合并, 并且在训练过程中, 利用全局状态信息为混合网络提供正向权重. 本文对于联合动作值取 arg max 等价于对于每个局部动作值函数求 arg max:

$$\arg\max_{u} Q_{tot}(\tau, u) = \begin{pmatrix} \arg\max_{u_1} Q_1(\tau_1, u_1) \\ \vdots \\ \arg\max_{u_n} Q_n(\tau_n, u_n) \end{pmatrix} \tag{2}$$

QMIX 将上式转换成为一种单调性的约束, 这种约束通过混合网络实现, 其中约束如下:

$$\frac{\partial Q_{tot}}{\partial Q_i} \geq 0, \quad \forall i \in \{1, 2, \cdots, n\} \tag{3}$$

3　多智能体注意力意图交流算法

为了更好地处理多智能体交流合作问题, 保证交流信息的充分性和有效性, 本文提出 MAAIC 算法, 为每个智能体增加能够表示其他智能体意图的网络信息输入, 同时使用注意力机制对意图信息和交流信息进行处理. 本节首先对算法思想和算法整体框架进行详细阐述; 然后, 介绍多个意图网络注意力单元、多头注意力的交流结构和算法的整体训练流程; 最后, 进一步探究意图网络的影响.

3.1 MAAIC 算法框架

在多智能体合作问题中, 在已知其他智能体状态与动作以及自身策略的信息基础上, 若能够知道其他智能体的目标, 就能更好地推测出其他智能体的意图. 基于此, 本文选取智能体历史上表现最好的网络结构, 这些网络中间层信息能够表示智能体的意图目标, 通过对这些意图进行注意力机制提取, 从而得到更加有效的信息. 将这部分信息加入到自身的决策中, 能够提升智能体的决策能力.

本文提出的多智能体注意力意图交流学习算法框架见图 1, 共分为 3 个阶段. 第 1 阶段是对信息特征的处理. 提出为智能体增加公共意图网络, 这些意图网络是由近阶段表现最好的网络复制而来. 首先将观测输入到全连接 (Fully connected, FC) 层得到初步的特征, 然后通过 GRU 循环神经网络, 对智能体的局部观测和动作进行处理, 得到包含历史的状态动作信息, 将这部分信息与通过意图网络得到其他智能体的意图信息进行注意力机制提取, 得到包含意图信息的状态信息. 第 2 阶段是交流模块. 智能体之间增加多头注意力机制的交流通道, 每个智能体将交流信息广播出去, 智能体根据自身信息和接收到其他智能体传来的交流信息进行多头注意力信息的提取, 从而得到对自身决策有效的信息输入给 GRU 单元. 为了让交流信息更加充分, 智能体作出决策前, 会同其他智能体进行多次迭代交流. 第 3 阶段将经过意图和交流提取到的高层信息, 经过一个全连接层, 得到智能体当前状态下的局部 Q 值函数, 进行一个简单的加和操作, 得到整体 Q^{sum} 值求和函数.

在上述算法框架中, 所有智能体都共享同一套网络参数, 根据不同时刻的观测输入和智能体编号的不同, 能够得到不同的信息. 本文每个智能体的网络参数共享, 主要有两个原因: 1) 大部分多智能体问题是一个开放的系统, 智能体进入或退出都有可能发生, 那么此时如果用独立的网络去训练智能体, 训练出来的网络容易对环境中智能体数目过拟合, 很难泛化; 2) 对多智能体环境, 如果采用独立的智能体网络, 需要很大精力去训练出一个好模型, 随着智能体数目的增加, 甚至有可能训练不出一个好模型. 为了缓和部分可观测带来的问题, 同时也会将智能体的动作和交流信息传入下一时刻的智能体信息中. 第 2 阶段交流模块所要经过的 2 个 GRU 循环神经网络也可以共享参数. 因此, 本文使用 1 个 GRU 循环神经网络来循环迭代交流.

整个网络的流程运行过程为: 智能体当前时刻得到观测后, 首先, 经过全连接层提取信息; 然后,

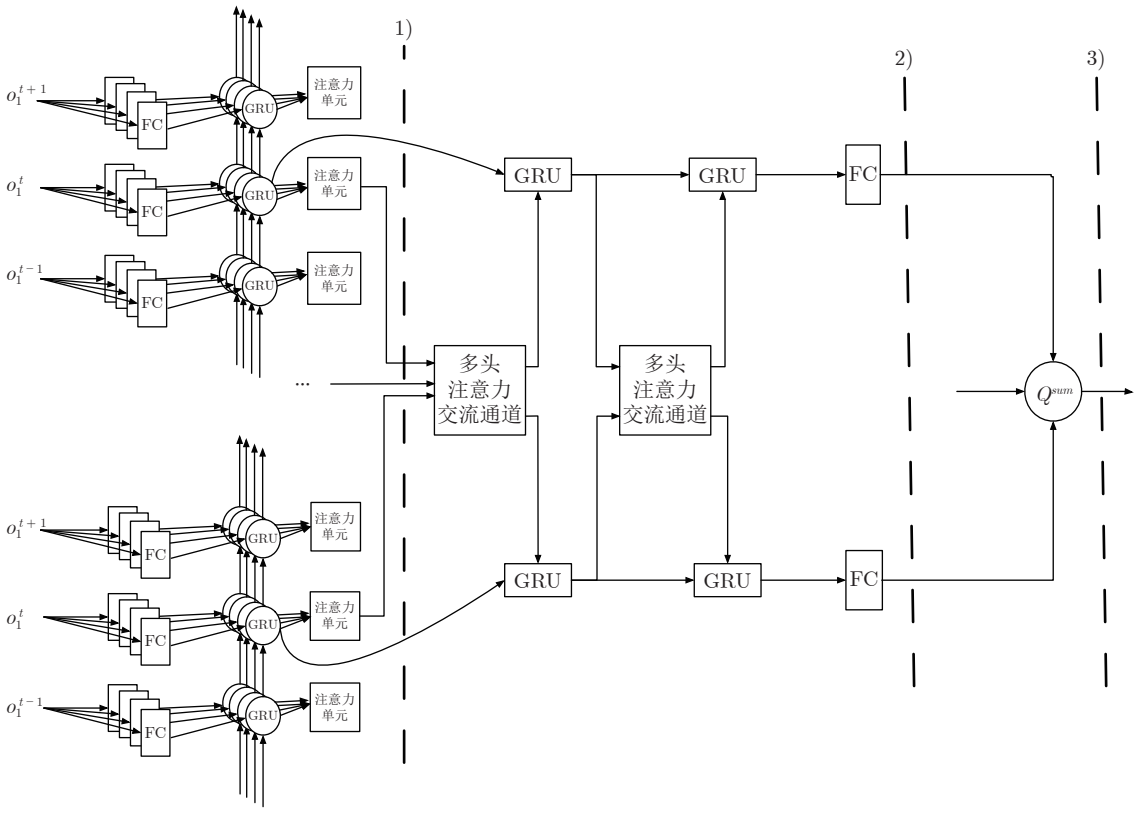

图 1 MAAIC 算法框架

Fig. 1 Overall framework of MAAIC algorithm

经过 GRU 循环神经网络, 得到包含过往历史的观测信息, 这部分历史观测信息在和过往的意图网络信息先做一个注意力机制信息提取后, 再同其他智能体进行一个多头注意力机制的信息交流; 在信息交流后, 经过一个全连接层, 从而得到智能体的状态动作值函数; 最后, 将这些状态动作值函数相加, 利用 DQN 的训练模式来学习策略.

3.2　意图网络的注意力机制

为了缓和多智能体环境不稳定的问题, 可以采取许多办法, 其中对其他智能体的行为进行建模, 推测出其他智能体的意图, 相比于简单地将其他智能体当作环境的一部分, 更有帮助. 认知科学研究发现, 人类使用与他们相交互的其他群体的目标、信念和喜好来帮助他们更好地做决策. 其中, 人类会从对他人的观测中模拟他人行为, 这种脑部过程能够帮助他们更好地知道他人的意图和行为, 从而在社会环境中作出正确行为.

因此, 为智能体增加多个意图网络. 本文的意图网络是历史近阶段最优网络的一个复制且依据每局的游戏等分来判断网络是否最优. 当有了最优网络, 智能体就能从中推测出其他智能体的意图信息. 这些网络接受智能体的观测 o_i^t, 首先, 经过全连接层和 ReLU 函数激活之后得到信息 $fc1_i^t$; 然后, 将这些信息输入到 GRU 循环神经网络, 得到输出 h_i^t. GRU 网络的隐藏状态来源于上一时刻对 o_i^t 信息处理后得到的 GRU 输出 h_i^{t-1}, 计算公式如下:

$$\begin{cases} r = \sigma\left(W_{ir} fc1_i^t + b_{ir} + W_{hr} h_i^{t-1} + b_{hr}\right) \\ z = \sigma\left(W_{iz} fc1_i^t + b_{iz} + W_{hz} h_i^{t-1} + b_{hz}\right) \\ h_i^{t-1'} = h_i^{t-1} \odot r \\ h_i' = \tanh\left(W_{ih} fc1_i^t + b_{ih} + W_{hh} h_i^{t-1'} + b_{hh}\right) \\ h_i^t = (1-z) \odot h_i^{t-1} + z \odot h_i' \end{cases} \tag{4}$$

式中, σ 和 tanh 代表要训练的权重矩阵和偏置. r 和 z 代表重置门和更新记忆门, 从而更好地利用历史信息, 缓和局部观测问题.

在训练过程当中, 有些历史阶段的网络能有好的表现, 也就是从一定程度上说明网络能够表示出智能体更好的意图, 因此将这些网络结构的参数保存下来, 再利用这些网络对当前时刻智能体的观测做进一步的信息补充, 从而让智能体能够更好地利用以往经验进行学习. 本文提出的多智能体注意力意图交流学习算法为每个智能体增加三个意图网络, 如图 2 第 1 阶段, 其中线性层 $Linear(Q)$ 表示 Q 为一个线性层, $Linear(K)$ 和 $Linear(V)$ 也是线性层. 这样对当前时刻智能体的观测就得到一个信息

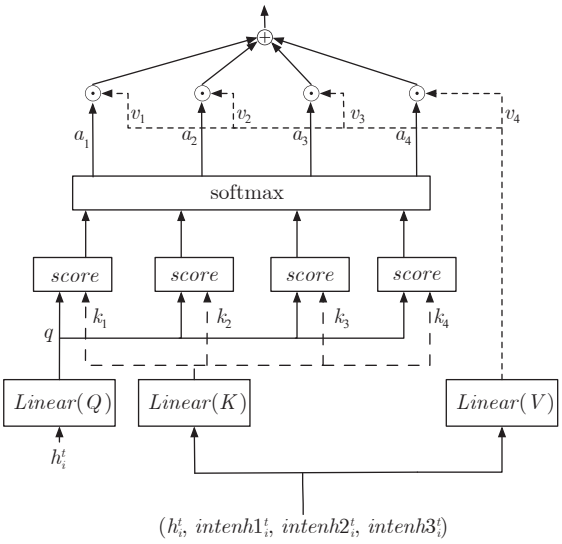

图 2　对意图网络进行自注意力信息的提取

Fig. 2　Extracting self attention information from intention network

向量 $(h_i^t, intenh1_i^t, intenh2_i^t, intenh3_i^t)$. 其中 h_i^t 代表当前网络的信息, $intenh_i^t$ 代表历史网络的信息, 该向量能够在一定程度上表示出对其他智能体的意图理解信息.

对经过加工的信息向量进行进一步加工处理的方式有 3 种: 第 1 种是简单地进行加和, $newh_i^t = h_i^t + intenh1_i^t + intenh2_i^t + intenh3_i^t$; 第 2 种是经过一个全连接层, 将信息向量拼接起来 $(h_i^{t\mathrm{T}}, intenh1_i^{t\mathrm{T}}, intenh2_i^{t\mathrm{T}}, intenh3_i^{t\mathrm{T}})$, 然后经过一个全连接网络, 得到信息输出 $newh_i^t = W_{hh'}(h_i^{t\mathrm{T}}, intenh1_i^{t\mathrm{T}}, intenh2_i^{t\mathrm{T}}, intenh3_i^{t\mathrm{T}}) + b_{hh'}$; 第 3 种是本文采用方式, 如图 2 所示, 用自注意力机制对这些意图信息进行处理, 能够从这些信息中, 提取出能够帮助智能体合作的更有价值的信息. 本文使用软性注意力机制, 给定信息输入 $X = (h_i^t, intenh1_i^t, intenh2_i^t, intenh3_i^t)$, 查询向量 $q = h_i^t$ 为当前时刻的智能体的信息, 打分机制选用缩放点积模型 $s(k, q) = k^{\mathrm{T}}q/\sqrt{d}$. 对于上述的信息采用注意力机制的处理公式如下:

$$\begin{cases} q = Qh_i^t \\ key = KX \\ value = VX \\ score = \mathrm{softmax}\left(\dfrac{q^{\mathrm{T}} key}{\sqrt{d}}\right) \\ a_i^t = value \times score \end{cases} \tag{5}$$

首先, 对查询向量即当前历史观测信息 h_i^t, 进行一个线性特征变换得到新的向量 q; 然后, 对于信息输入 X 经过矩阵变换后, 得到 key、$value$ 矩阵; 接着, 计算得分值, 计算 $value$ 上向量集不同的权重;

最后, 将这些向量分别和权重相乘并相加, 得到最终的注意力信息 a_i^t.

3.3 交流信息的注意力机制

交流是一种重要的获得信息方式, 作为智能体的一种必要手段, 能够帮助从其他智能体的经验中学习, 在部分可观测环境中, 交流能够帮助传递其他智能体的观测信息, 从而能够去缓和多智能体环境不稳定问题. 本文研究致力于为星际争霸这样复杂的环境, 寻找更好的处理方案, 让交流发生在隐藏空间, 进而让高层次信息能够传递到各个智能体中, 以及让每个智能体得到的策略梯度能够回传到整个网络中.

在多智能体合作环境下, 智能体 $agent_0$ 接受环境的观测 o_0^t, 通过自身的网络输出相应动作 a_0^t, 根据当前状态执行完动作后, 环境给出一个集体奖励. 集中式训练分布式执行的多智能体同环境交互见图 3, 智能体间可以选择是否利用有限通道进行交流信息. 本文从另一个角度出发, 为了更好地利用交流信息, 考虑到在不同时刻, 当前智能体策略对于其他智能体的关注点是不一样的, 因此在交流信息上选择使用注意力模型, 通过给其他智能体的交流信息分配不同权重, 提取出有效的交流信息.

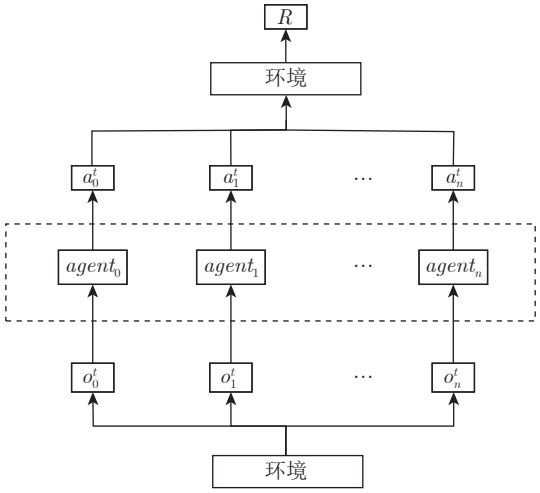

图 3 基于集中式训练分布式执行的多智能体同环境交互

Fig. 3 Multi-agent interaction with environment under centralized training and decentralized execution

本文智能体间交流信息计算采用多头注意力模型, 相比于单头注意力模型, 多头注意力模型本身为多个注意力的计算, 彼此之间相互独立, 起到一个集成的作用, 防止出现过拟合. 本文选择 GRU 神经网络处理智能体交流, 智能体间的交流可循环迭代 k 个时间步, GRU 网络隐藏层状态来自前一阶段 GRU 隐藏层状态输出. 考虑到智能体在实际问题中发送交流信息是需要一定时间的, 如果同步处理交流信息, 每次需要等到所有信息都发送过来, 决策效率必然不高. 因此实现交流信息时, 一般采用异步处理即当前时刻接受前一时刻的交流信息进行处理, 并且输出下一时刻的交流信息. 交流通道使用的多头注意力模型见图 4. 交流信息 c 的计算公式见式 (6), 其中 Q、K、V 根据交流时刻以及决策时刻存在不同的情况, 本文采用 8 头注意力模型, 映射权重矩阵为 $W_i^Q \in \mathbf{R}^{d_{model} \times d_k}$、$W_i^K \in \mathbf{R}^{d_{model} \times d_k}$、$W_i^V \in \mathbf{R}^{d_{model} \times d_k}$, 其中 d_{model} 为智能体意图融合向量 a_i^t 的维度, d_k 为每个注意力头部的维度.

$$\begin{cases} c = Multihead(Q,\, K,\, V) = \\ \quad \text{Concat}(head_1,\, head_2,\, \cdots,\, head_8) \\ head_i = Attention(A^{\mathrm{T}} W_i^Q,\, A^{\mathrm{T}} W_i^K,\, A^{\mathrm{T}} W_i^V) \end{cases} \tag{6}$$

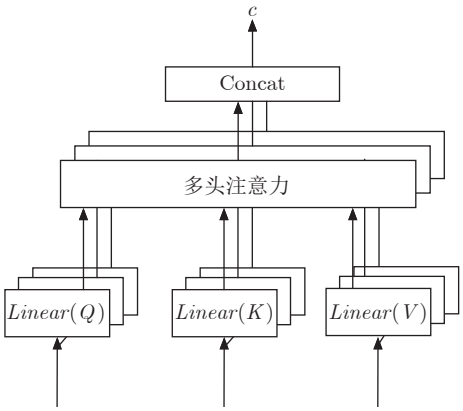

图 4 交流通道使用的多头注意力模型

Fig. 4 Multihead attention model used in communication channels

本文在每个决策时刻, 智能体之间可以交流 k 时间步, 每次交流的计算流程如下:

$$\begin{cases} h_i^{t^k} = GRU\,(c,\, h_{out})\,, & k = 0 \\ c = \begin{cases} Multihead\,(A^t,\, A^t,\, A^t)\,, & t = 0 \\ Multihead\left(A^t,\, H^{t-1^K},\, H^{t-1^K}\right), & t > 0 \end{cases} \end{cases} \tag{7}$$

$$\begin{cases} h_i^k = GRU\left(c,\, h_i^{t-1}\right), \quad k > 0 \\ c = Multihead\left(H^{t^{k-1}},\ H^{t^{k-1}},\ H^{t^{k-1}}\right) \end{cases} \tag{8}$$

式中, t 表示当前决策时刻, k 代表交流时间步, 智能体之间最多进行 k 步交流. 矩阵 A 由每个智能体经过意图网络注意力模型的输出 $(a_0^t, \cdots, a_n^t)$ 构成, 矩阵 H 由每个智能体经过 GRU 得到的隐藏层状态构成. 在本文提出的智能体交流结构中, 智能体会将上一时刻经过交流单元 GRU 后, 得到的隐藏层

信息加入当前时刻, 进行一个注意力的提取, 从而使整个智能体的决策与过去的信息更加紧密地结合起来. 交流信息 c 的计算共分为 3 种情况: 1) 智能体第 1 次同环境进行交互做决策, 在 $t=0$ 且 $k=0$ 时刻, 此时智能体没有前一时刻的交流信息, 输入给多头注意力交流通道的信息为 $(A^t,\ A^t,\ A^t)$. 2) 智能体在 $t>0$ 且 $k=0$ 时, 此时智能体的交流信息是由上一决策时刻的隐藏层向量给出. 在 $k=0$ 时刻, 智能体需要利用意图观测信息进行注意力的提取, 此时多头注意力单元的输入为 $(A^t,\ H^{t-1^K},\ H^{t-1^K})$. 3) 当交流时间步 $k>0$, 此时输入由当前时刻上一交流时间步的隐藏层信息 $(H^{t^{k-1}},\ H^{t^{k-1}},\ H^{t^{k-1}})$ 构成.

3.4　网络结构更新

本文算法采用离线更新方式, 将多个智能体同环境进行交互的状态、动作、奖励、终止状态等数据放到经验池中, 每次从经验池中选取批量完整的每局游戏过程进行学习. 算法同 DQN 一样, 构建一个目标网络, 每隔固定周期, 将当前网络参数复制, 用于计算下一时刻状态值, 用自举方式对状态值进行更新, 能够加速收敛并且有助于算法的稳定. 算法的损失函数 $loss$ 为:

$$\begin{cases} loss=\displaystyle\sum_{j=1}^{nbatch}\sum_{t=1}^{T}\left(\sum_{i=1}^{n}y_i^j-\sum_{i=1}^{n}Q_i^j\left(o_i^t,\ a_i^t;\theta\right)\right)^2 \\ y_i^j=\begin{cases} r^j, & \text{终止} \\ r^j+\gamma\max_{a'}Q^{target}\left(o_i^{t+1},\ a';\theta'\right) \end{cases} \end{cases} \tag{9}$$

多智能体注意力意图交流学习算法的流程见附录 A 中算法 1. 该算法是基于 Q 值算法, 在算法训练过程中, 通过对损失函数最小化, 找到最优价值函数估计, 最优策略来自对动作空间遍历后的最大 Q 值动作. 在智能体学习过程中, 会不断根据评估时刻智能体的表现, 将性能更好的网络替换过往的意图网络.

3.5　内在意图奖励

针对本文的一个最主要的假设, 意图网络是可以短暂地作为最优策略网络, 本文从内在奖励方面研究意图网络的影响, 为意图信息的有效性提供理论依据. 修改智能体 i 做出动作后得到的即时奖励:

$$r_i^t=\alpha e_i^t+\beta c_i^t \tag{10}$$

式中, e_i^t 是外在的环境奖励, c_i^t 是意图影响的内在奖励. 意图影响的本质是一个智能体局部最优策略对自身当前策略有指导作用, 就可以提供额外的奖励.

计算内在意图奖励需要知道智能体当前策略网络与意图网络在环境状态下作出的行为, 一种直观的奖励方式可以是当两者的行为 a 一致时, 给出正向奖励 $c_i^t=|e_i^t|$. 然后, 它可以奖励自己采取它认为最具影响力的行动. 这是一种自然而然的方式, 因为它类似于人类思考自己对他人的影响方式.

本文设置了多个意图网络且多个智能体是共享网络, 得到的反馈是团队奖励, 因此只有当所有智能体行为都与意图网络行为一致时 (多个意图网络只要有一个满足要求即可), 才会有内在意图奖励. 在 t 时刻得到的奖励为:

$$r^t=\begin{cases} \alpha e^t+\beta c^t, & Q(o^t,\ a^t;\theta)= \\ & Q^{Intention}\left(o^t,\ a^t;\theta\right) \\ \alpha e^t \end{cases} \tag{11}$$

式中, o 和 a 是所有智能体的观察和动作, $Q^{Intention}$ 代指多个意图网络中的一个.

4　实验

本节在星际争霸 SMAC[39] 和修改版的捕食者游戏[24] 上开展实验.

4.1　环境简介

在经过精心设计的 SMAC 游戏场景中, 智能体必须学会一种或多种微管理技术, 才能击败敌人. 每个场景都是两支部队之间的对抗, 每个部队的初始位置、数量、类型都随着场景的不同而发生变化. 其中第 1 支部队是由本文算法所控制的智能体构成, 第 2 支部队是内置游戏人工智能控制的敌方构成. 本文算法所使用的 SMAC 实验场景如表 1 所示. SMAC 环境下具体算法参数设置见附录 B.

在每个时间步长, 每个智能体都会获得其视野范围内的局部观测结果, 其中包含了每个单元圆形区域内的地图信息. 具体地, 每个智能体得到的特征向量包含视野范围内的友军和敌军的属性 (如距离、相对 x、相对 y、健康、盾牌、$unit\_type$), 其中盾牌是能够抵消伤害并可以在一段时间后重新生成的. 同时, 特征向量也包含周围地形特征以及可观测到友军的最后行动. 在本文的局部观测中, 智能体的观测无法区分其余智能体是不是在视野范围, 还是已经死亡.

4.2　实验结果分析

首先, 给出本文的多智能体注意力意图交流学习算法在 SMAC 五个实验场景下, 同基准算法的对比实验结果 (见图 5), 其中 MAAIC-VDN 是本文的多智能体注意力意图交流学习算法; 然后, 对这些实验结果进行分析.

表 1 SMAC 实验场景

Table 1 Experimental scenarios under SMAC

场景名称	我方单位	敌方单位	类型
5m_vs_6m	5 名海军陆战队	6 名海军陆战队	同构但不对称
3s_vs_5z	3 潜行者	5 狂热者	微型技巧: 风筝
2s_vs_1sc	2 缠绕者	1 脊柱爬行者	微技巧交火
3s5z	3 潜行者 & 5 狂热者	3 潜行者和 5 狂热者	异构且对称
6h_vs_8z	6 蛇蝎	8 狂热者	微招: 集中火力

实验场景任务目标是要学会让我方单位战胜敌方单位, 因此本文选取游戏胜率作为最终评估目标. 在算法训练过程中, 每间隔 *evaluate_cycle* 次, 算法会对智能体学习到的策略进行评估. 智能体会在相应场景游戏下进行 *evaluate_epoch* 轮的游戏测试, 通过统计胜利次数给出我方的游戏胜率. 为了不失一般性, 算法采用不同的随机种子, 进行 4 次重复实验, 且使用 95% 的置信区间. 每次完整训练进行 *n_epoch* 次数, 训练结束后会得到 *n_epoch*/*evaluate_cycle* 个游戏胜率数据用于绘制实验结果图.

图 5 给出了 5 个场景下, 本文多智能体注意力意图交流学习算法 MAAIC-VDN 的对比结果, 其中 2s_vs_1sc 和 3s5z 是简单场景, 算法能够获得比较高的胜率; 5m_vs_6m 和 3s_vs_5z 为困难场景, 算法需要学到更好的合作策略才能够取得胜利; 6h_vs_8z 是超级困难场景, 需要更长的训练时间以及需要学会更好的获胜技巧. 由图 5 可以看出, 本文算法 MAAIC-VDN 在所有场景下的收敛速度明显优于其他基线算法, 并且性能表现是最好的. 本文算法在场景 3s5z 和 3s_vs_5z 上, 相比于 VDN 获得了一定的性能提升, 特别是在超级困难场景 6h_vs_8z 中, 在其他算法都已经失效的情况下, 仍然获得了 25% 左右的胜率. IQL 除了在简单的场景 2s_vs_1sc 上, 能够最终获得和其他算法同样的性能, 在其他的场景下的结果, 都远远不如 VDN 和 MAAIC-VDN, 说明独立学习的智能体在复杂环境不能够表现出好的合作策略. VDN 算法在 2s_vs_1sc、3s5z、3s_vs_5z、5m_vs_6m 上, 都能获得不错的性能, 但是在复杂场景 6h_vs_8z 中, 基本没有学到任何策略.

同样在 QMIX 算法上修改, 图 6 给出了在 5 个场景下, 本文多智能体注意力意图交流学习算法 MAAIC-QMIX 算法与 QMIX 和 IQL 算法在 SMAC 上的实验结果, 其中在简单场景 3s5z 和困难场景 3s_vs_5z 下, MAAIC-QMIX 算法相较于其他算法收敛更快, 最终结果也更好. 在超级困难场景 6h_vs_8z 中, 其他算法结果都接近零, 而本文算法获得了 20% 左右的胜率. 图 7 给出了 5 个场景下, 基于 QTRAN 算法的多智能体注意力意图交流学习 MAAIC-QTRAN 算法在 SMAC 上的实验结果, 可以看出, 相较于其他基线算法, MAAIC-QTRAN 算法性能都有显著提升. 其中, 所有算法在简单场景 2s_vs_1sc 最终都能取得百分百胜率, 而本文 MAAIC-QTRAN 算法收敛更快. 值得注意的是, 在场景 3s_vs_5z 下, QTRAN 算法由于其对

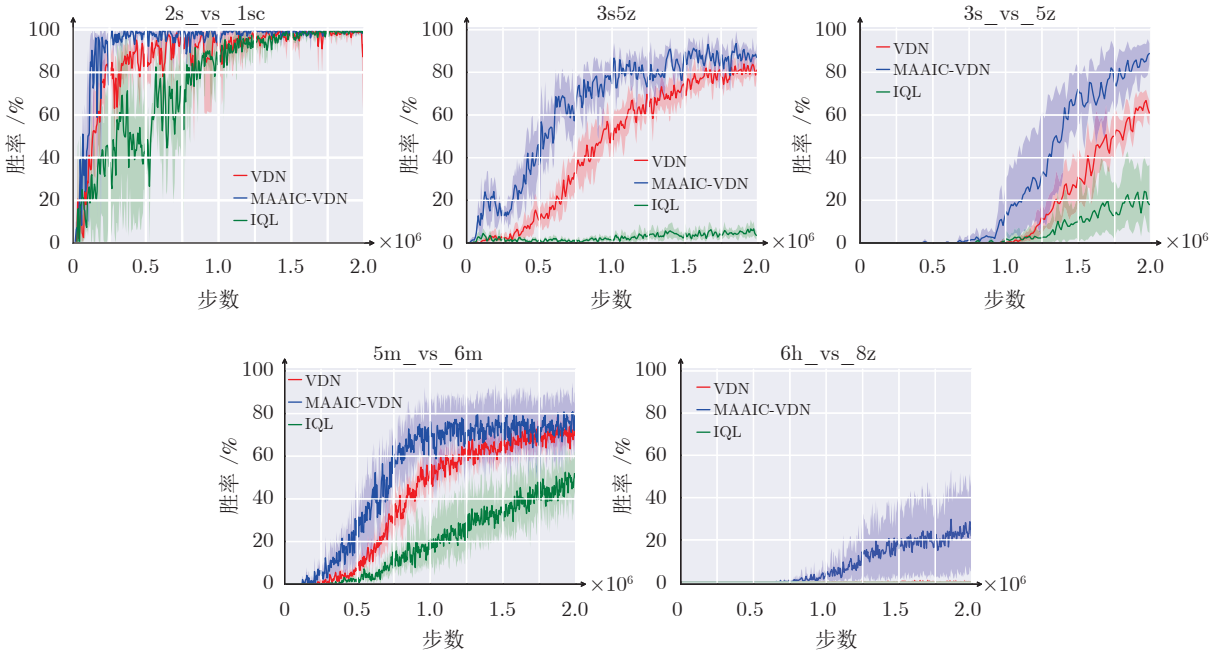

图 5 MAAIC-VDN 算法在 SMAC 上的实验结果

Fig. 5 Experimental results of MAAIC-VDN algorithm on SMAC

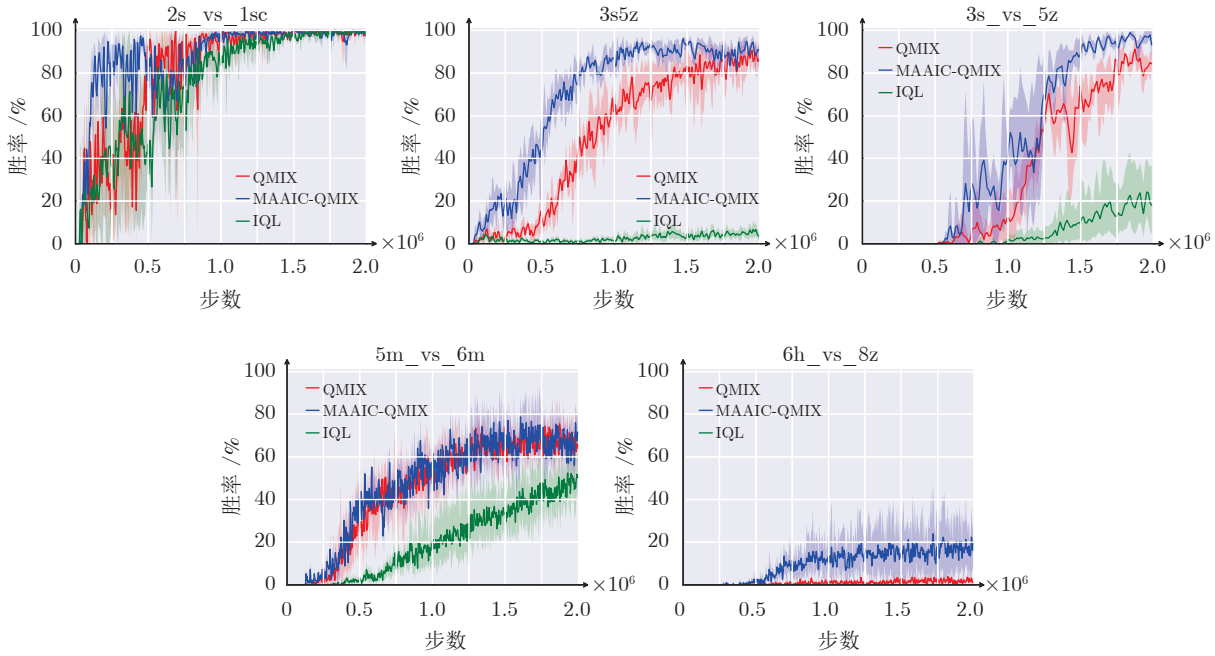

图 6　MAAIC-QMIX 算法在 SMAC 上的实验结果

Fig. 6　Experimental results of MAAIC-QMIX algorithm on SMAC

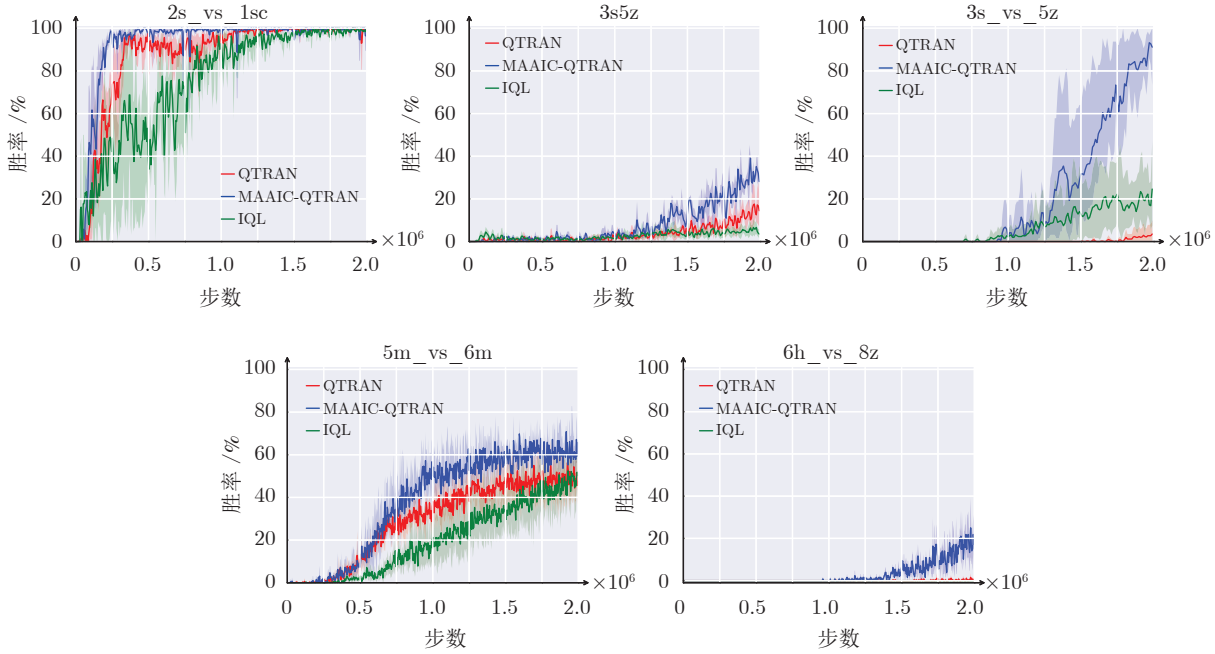

图 7　MAAIC-QTRAN 算法在 SMAC 上的实验结果

Fig. 7　Experimental results of MAAIC-QTRAN algorithm on SMAC

约束条件的放松, 性能不佳, 而本文 MAAIC 方法可以让智能体快速地找到最优策略.

表 2 给出了测试算法的最大中值实验结果, 最大中值为训练过程最后 250 k 步得到的所有测试结果. 实验结果表明, 启发式 (Heuristic) 算法即攻击最近敌方单位策略算法, 性能是最低的. 而在基础算法框架上应用本文 MAAIC 方法, 智能体之间能更好地协作, 获取更高性能. 除了 2s_vs_1sc 外, IQL 的胜率很低, 这是因为直接使用全局奖励更新策略会带来非平稳性, 当智能体数量增加时, 这种非平稳性会变得更严重. QTRAN 算法的表现也不是很好, 是因为实际场景下的宽松约束可能会阻碍

其更新的准确性[25].

在两个基线算法上加入本文的多智能体注意力意图交流模型, 其最终实验结果和收敛速度在大多数场景中都有明显的提升, 显示了本文算法能够从意图信息中学习到更多知识.

4.3 消融实验

本文进行三个方面消融实验分析: 1) 对交流结构的消融实验. 对比算法为去掉意图网络结构, 保留本文的多头注意力多步交流模块的 VDN with TarMAC 算法和去掉本文的交流信息的网络结构, 换成 CommNet 的交流结构的 VDN with CommNet 算法; 2) 对意图网络数目的消融性实验. 对比了 1、3、5 个意图网络的 MAAIC 算法; 3) 对比历史最优网络[40] 与历史最近邻网络[41], 哪个作为意图网络更好, 本文采用的是历史最优网络作为意图网络.

首先, 验证本文交流结构消融性的实验结果见图 8. 由图 8 可以看出, 在 6 个场景下, VDN with CommNet 性能是最差的, 说明本文多头注意力交流结构能够在复杂环境中提取到有效信息, 智能体如果只是简单地拿到所有智能体信息, 简单求和反而会让策略变差. 算法 VDN with TarMAC 除了在场景 3s5z 性能与本文算法相当, 在其他场景下, MAAIC-VDN 算法性能表现最好. 由场景 2s_vs_1sc 可以看出, 含注意力交流模块的 VDN with TarMAC 算法收敛速度和多智能体注意力意图算法是一样的, 相比 CommNet 交流结构, 收敛速度更快, 说明本文多头注意力交流模块对于提升收敛速度很有效. 由场景 5m_vs_6m 和场景 3m 可以看出, 多智能体注意力意图相比于没有意图的 VDN 算法, 性能要好一些, 从而说明本文提出的意图网络结构与注意力交流结构结合, 能够帮助改善算法性能. 在 3s_vs_5z 和 6h_vs_8z 场景下, 本文算法无论是收敛速度还是性能, 都比其他的交流结

表 2 测试算法的最大中值实验结果 (%)

Table 2 Maximum median performance of the algorithms tested (%)

场景	MAAIC-VDN	VDN	IQL	MAAIC-QMIX	QMIX	Heuristic	MAAIC-QTRAN	QTRAN
2s_vs_1sc	100	100	100	100	100	0	100	100
3s5z	90	87	9	**97**	91	42	31	20
5m_vs_6m	**87**	78	59	74	75	0	67	58
3s_vs_5z	**98**	73	46	**98**	97	0	97	15
6h_vs_8z	**55**	0	0	31	3	0	22	0

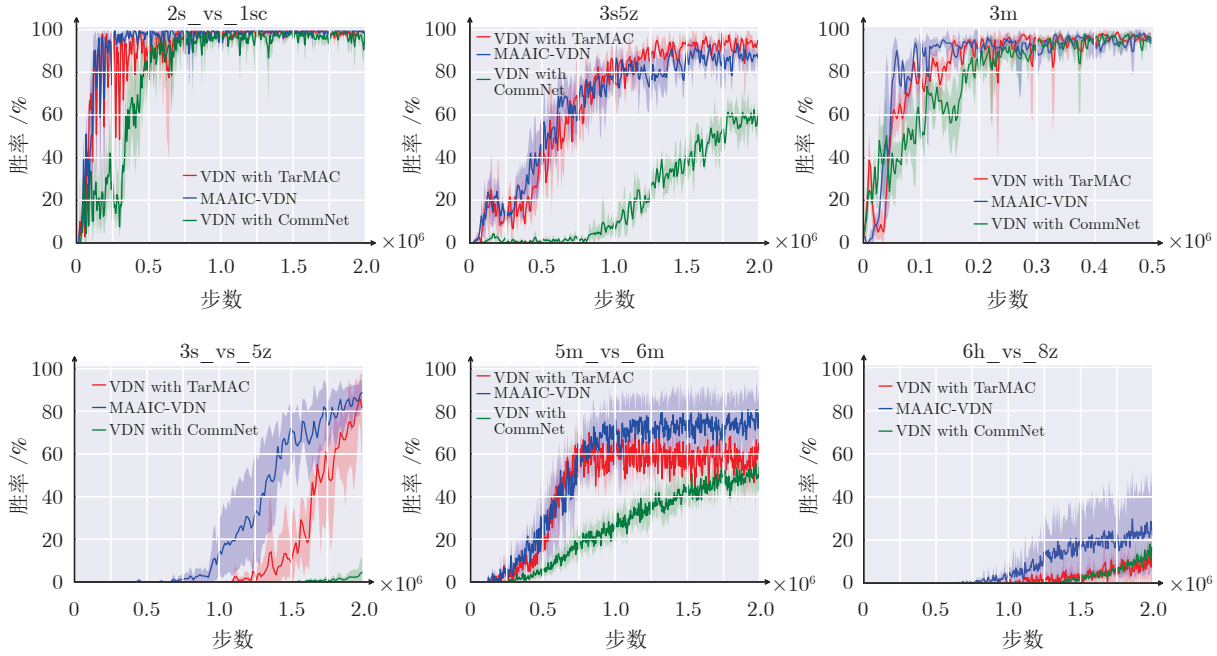

图 8 交流结构消融性实验结果

Fig. 8 Experimental ablation results of the communication structure

构要好. 这表明, 从额外的意图网络中提取意图信息作为交流信息, 比只在当前内部网络提取交流信息, 更具有价值, 验证了本文的多个意图网络和注意力交流模型的有效性.

接着, 验证意图网络数目对于性能的影响, 在 SMAC 的 3 个实验场景下, 图 9 给出了 1 个意图网络、3 个意图网络和 5 个意图网络的消融性实验结果. 由图 9 可以看出, 在简单场景 2s_vs_1sc中, 意图网络数并不影响最终性能, 算法都能很快地收敛到最好结果. 由场景 3s_vs_5z 可以看出, 5 个意图网络的性能大于 1 个意图网络和 3 个意图网络性能. 在场景 5m_vs_6m 中, 3 个意图网络性能最优. 在 3 个场景下, 有意图网络算法性能收敛都是比基线 VDN 算法要快且大多数最终结果也更好. 由实验结果可以看出, 意图网络对算法性能有明显提升, 能够处理应对复杂问题且多个意图网络性能优于 1 个意图网络.

表 3 为本文训练 1 000 轮后, MAAIC-VDN 算法在不同意图网络数的 GPU 上内存开销, 表 3 中数值为运行多次求平均值且去掉个位数得到的结果. 图 10 为 MAAIC-VDN 算法在不同意图网络数的时间开销, 纵坐标是平均每局所消耗的时间, 本文选择在场景 2s_vs_1sc 下测量, 这是由于所有算法都可以快速找到最优策略. 为了减少其他变量干扰, 本文只在 1 个 GPU 运行且只运行单个程序. VDN 算法内存开销和时间开销都是最小的, 而添加注意力机制的交流模块且无意图网络 VDN with TarMAC算法的内存和时间开销大幅度增加. 随着意图网络数量的增加, MAAIC-VDN 算法的内存和时间花费也越来越大, 总体上呈现线性增长. 其中注意力机制交流模块是必不可少的, 因此, 本文在衡量算法性能与消耗情况下, 选择使用 3 个意图网络.

图 11 给出了历史最优网络作为意图网络和历史最近邻网络作为意图网络 2 个算法的消融性对比实验. 可以看出, 在 3 个场景下, 2 个算法相比于算法 VDN, 其收敛速度和最终性能结果更好. 本文采用的历史最优网络作为意图网络算法, 在 3 个场景中性能表现最优, 这表明了使用历史最优网络作为意图网络, 得到的意图信息更为有效.

4.4　内在意图奖励实验

为了更好地探究意图网络的作用, 本文单独在小规模的球形环境 (即修改版捕食者 (Modified predator-prey, MPP) 游戏) 验证意图网络的性能, 实验参数见附录 C. 修改的捕食者游戏比经典的捕食者游戏更复杂. 两者状态空间和行动空间的构造是相同的, 捕获猎物就相当于将猎物置于代理人的观察视界内. 修改的捕食者游戏将经典的捕食者游戏扩展到只有当多个捕食者同时捕获猎物时, 才会给予积极奖励, 这需要更高程度的合作. 如果两个或两个以上捕食者同时捕获猎物, 捕食者将获得 +1 的团队奖励, 但如果只有一个捕食者捕获猎物, 捕食者将获得负奖励 P.

图 12 给出了 2 个智能体、不同惩罚值 P 下 VDN 与本文 VDN-Intention 的内在意图奖励实验结果, 其中意图网络数为 3. 图 12 中, 每个算法都绘制了

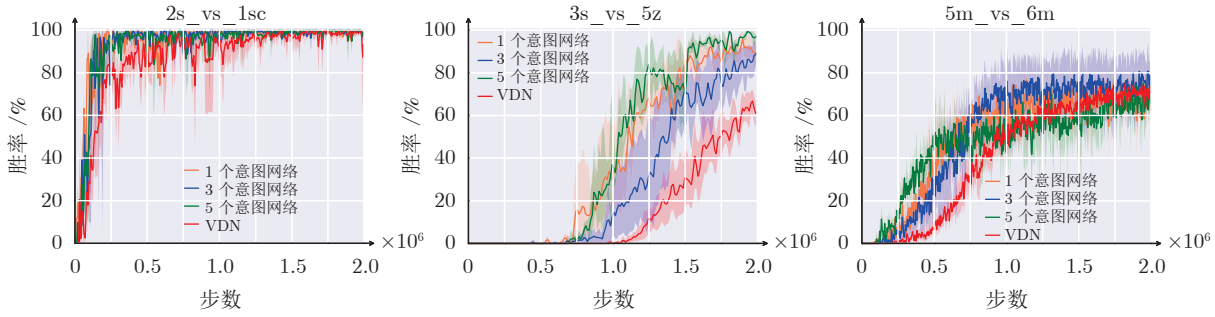

图 9　意图网络数的消融性实验结果

Fig. 9　Experimental ablation results of the number of intention networks

表 3　MAAIC-VDN 算法在不同意图网络数的 GPU 内存开销 (MB)

Table 3　GPU memory cost for different numbers of intention networks based on MAAIC-VDN algorithm (MB)

场景	5 个意图网络	3 个意图网络	1 个意图网络	VDN with TarMAC	VDN
2s_vs_1sc	1560	1510	1470	1120	680
5m_vs_6m	1510	1500	1500	1150	680
3s_vs_5z	2120	2090	2100	1480	730

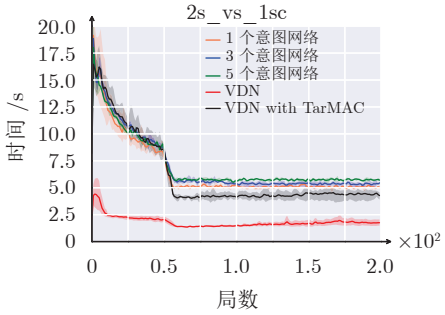

图 10 MAAIC-VDN 算法在不同意图网络数的时间开销

Fig. 10 Time cost for different numbers of intention units based on MAAIC-VDN algorithm

5 次测试的平均奖励和 95% 的置信区间. 可以看出, 基线算法 VDN 随着惩罚值 P 的增大, 训练难度增大, 得到的奖励减少. 而本文 VDN-Intention 算法是在 VDN 基础上, 只增加了内在意图奖励, 没有额外添加注意力机制和交流结构, 可以看出, 与基线算法 VDN 相比, 有较大的性能提升. 即使惩罚值 P 很大, VDN-Intention 算法中智能体还是能很好地合作捕捉猎物. 这充分表明本文最主要假设的可行性, 即意图网络是可以短暂地作为最优策略. 因此, 在大型复杂环境下, 可以从意图网络中抽取意图信息, 帮助智能体更好地决策.

5 结束语

本文提出一种意图存储机制, 引入额外的公共网络保存历史表现最好的策略网络, 以此建模其他智能体的意图信息. 同时引入交流模块, 采用多头注意力机制提高整个网络的表示能力. 本文的多智能体意图交流算法可以充分利用过往局部最优网络信息, 扩大了信息的来源渠道. 最后, 将本文的多智能体注意力意图交流学习算法分别在开源的星际争霸任务场景上和捕食者环境上进行验证和分析, 并同领域内的经典算法进行对比, 验证了本文算法的有效性. 通过在星际争霸环境上的消融性实验分析, 验证了本文提出的多智能体注意力意图结构和多头注意力交流结构的可行性, 并通过内在意图奖励方式, 验证了意图网络提供意图信息的可靠性.

附录A MAAIC 算法

算法 1. 多智能体注意力意图交流学习算法

1) 创建智能体的动作值函数 Q、目标动作值函数 Q^{target} 和 3 个意图网络 Q^I, 并且对网络 $Q(;\theta)$、$Q^{target}(;\theta')$、$Q^I(;\theta'_1)$、$Q^I(;\theta'_2)$、$Q^I(;\theta'_3)$ 进行随机初始化;

2) 初始化经验池 D, 容量设置为 N. 初始化数组 3 维的 $reward\_array$;

3) 初始化学习率 α, 训练时所需要的样本批数量 $batch\_$

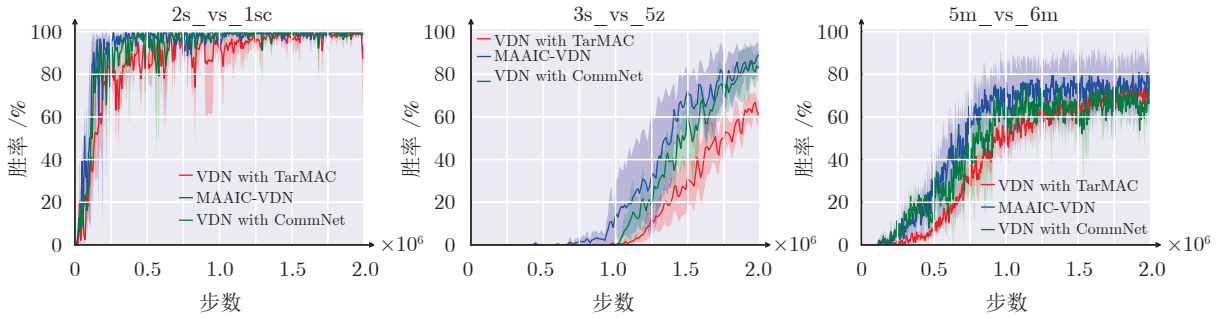

图 11 历史最优网络和最近邻网络作为 MAAIC 消融性实验结果

Fig. 11 Experimental ablation results of MAAIC with the best Q-network and the nearest Q-network

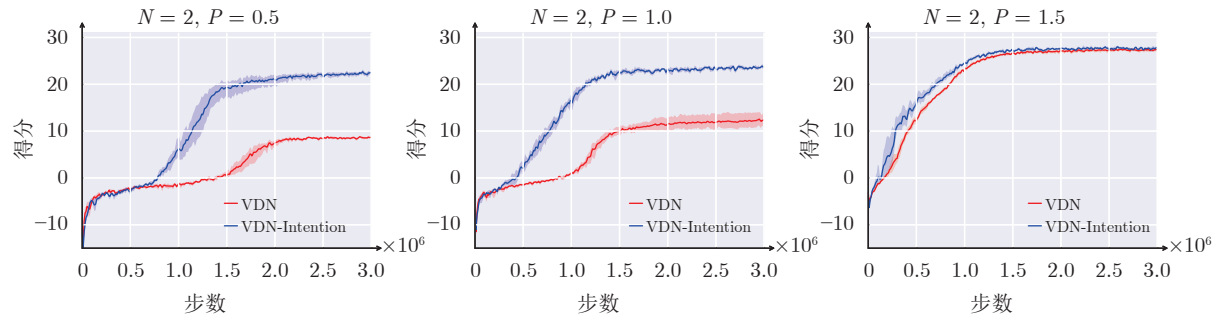

图 12 内在意图奖励实验结果

Fig. 12 Experimental results of intrinsic intention rewards

size, 迭代次数 *epoch* 等数据;

4) for $epoch = 1, M$ do;

5) 初始化环境, 每个智能体的初始观测为 $o^0 = (o_1^0, \cdots, o_n^0)$;

6) for $t = 1, T$ do;

7) 用 ϵ 概率随机选取动作 a_i^t, 否则选取 $a_i^t = \max_a Q^*(o_i^t, a; \theta)$;

8) 在仿真环境当中执行联合动作 $a^t = (a_1^t, \cdots, a_n^t)$, 得到集体奖励 r^t 以及下一时刻的观测 $o^{t+1} = (o_1^{t+1}, \cdots, o_n^{t+1})$;

9) 记录 $(o^t, a^t, r^t, o^{t+1}, t)$五元组信息到每集游戏变量中;

10) end for;

11) 计算测试过程每集游戏平均得分 r_sum, 与数组 $reward\_array$ 最小值比较, 当大于最小值时, 那么更新对应位置的意图网络, 即将当前策略网络的参数 $Q(;\theta)$ 复制到意图网络当中 $Q^I(;\theta_i')$;

12) 将整集游戏收集的经验数据放入经验池 D 中;

13) 从经验池 D 中, 批量 $mini\_batch$ 从每集游戏中采样;

14) for $t = 1, T$ do;

15) 对于每个 $(o^t, a^t, r^t, o^{t+1}, t)^j$, 用意图网络计算 $(h_i^t, intenh1_i^t, intenh2_i^t, intenh3_i^t)$, 作为输入于当前网络, 经过交流模块, 最终得到每个智能体的 $Q_i^j(o_i^t, a_i^t)$, 目标网络计算用来更新的目标值 $y_i^j = r^j + \gamma \max_{a'} Q^{target}(o_i^{t+1}, a'; \theta')$. 将 $l = \sum_{j=1}^{nbatch}(\sum_{i=1}^{n} y_i^j - \sum_{i=1}^{n} Q_i^j(o_i^t, a_i^t))^2$ 放入总 $loss$ 中;

16) end for;

17) 根据损失函数 $loss$ 计算得到策略梯度, 并对网络参数 θ 进行更新;

18) 在更新周期, 目标网络参数 θ' 替换为当前网络的参数 θ;

19) end for.

附录B　SMAC 环境下算法参数设置

为了能够更好地分析本文提出的多智能体注意力意图交流学习算法的性能, 本文在 SMAC 多个不同复杂度场景下进行实验. 本文的多智能体注意力意图学习算法的网络结构参数见表 B1.

首先, 环境初始化后会构建 N 个智能体的控制器, 其中对于本文算法, N 个控制器的网络参数是共享的, 区别只是在于自身的观测输入和编号信息是不一样的. 在控制器网络中, 每个智能体将自身的观测输入和过往的动作信息等经过 1 个 FC 层的编码和 ReLU 函数激活, 然后经过 GRU 循环神经网络的编码, 得到当前对于观测的信息输出, 其中在增加了 3 个意图网络后的维度输出为 $(4, N, rnn\_hidden\_dim)$, 这部分包括现实信息和三部分意图信息, 对这部分信息进行注意力提取, 最后得到 $(N, attention\_dim1)$ 信息, 代表每个智能体各自针对观测和意图信息的融合所生成的更高层次特征.

接着, 进行多智能体间的交流, 其中本文采用 8 头注意力机制处理交流结构信息. 交流的信息经过多头注意力的提取为 $(N, attention\_dim2)$, 通过 GRU 循环神经网络经过 k 时间步交流, 得到最终信息为 (N, rnn_hidden_dim). 最后, 这部分信息经过全连接层, 输入动作空间维度大小的 q 值, 大小为 $(N, n\_actions)$.

本文算法训练参数设置见表 B2. 智能体策略从控制器选取最大 Q 值对应的动作. 多个智能体利用自身策略同环境进行交互, 收集每集游戏的样本数据放到内存池中, 从而通过从内存池当中选取一批数据, 利用这些数据来训练智能体. 在每段游戏中, 智能体生成 $n\_episodes$ 样本数据, 放入内存池中. 根据 DQN 的模式计算的当前 Q 值和下一时刻的目标 Q 值, 首先, 对所有的智能体将两个 Q 值分别求和; 然后, 求得损失 $l = (Q - (r + \max_a Q^{target}))^2$, 将所有时刻、所有游戏集数的损失求和, 得到总损失; 最后, 按照均方根传递 (Root mean square prop, RMSProp) 优化方法对损失进行优化求解.

附录C　修改版追捕者环境下的算法参数设置

表 C1 给出了修改版追捕者环境下训练的参数设置. 每个独立的策略网络包含 3 个隐藏层. 所有隐藏层维度为 64, 激活函数为 ReLU. 其中智能体探索策略是保证游戏开始阶段, 智能体更多地进行探索, 随着训练持续进行, 智能体学到了更多知识, 智能体的探索会逐渐减少, 最后维持在一个最小的探测概率 $\epsilon \in [0.1, 1]$, 从而更有效地利用已学到的知识保证训练的稳定性和速度. 最后, 实验使用前馈策略且使用 Adam 优化器进行训练.

表 B1　MAAIC 算法网络参数

Table B1　Network parameters of MAAIC algorithm

参数名	设置值	说明
rnn_hidden_dim	64	对于局部观测的全连接特征编码维度, 循环网络的隐藏层维度
$attention\_dim1$	64	意图信息的注意力编码维度
$attention\_dim2$	64×8	多头注意力机制的编码维度
$n\_intention$	3	意图网络的个数

表 B2 SMAC 环境下 MAAIC 算法训练参数
Table B2 Training parameters of MAAIC algorithm in SMAC

参数名	设置值	说明
Lr	0.0005	损失函数的学习率
Optim_eps	0.00001	RMSProp 加到分母提升数值稳定性
Epsilon	1	探索的概率值
Min_epsilon	0.05	最低探测概率值
Anneal_steps	50000	模拟退火的步数
Epsilon_anneal_scale	step	探索概率值的退火方式
N_epoch	20000	训练的总轮数
N_episodes	1	每轮的游戏局数目
Evaluate_cycle	100	评估周期间隔
Evaluate_epoch	20	评估次数
Batch_size	32	训练的批数据大小
Buffer_size	5000	内存池大小
Target_update_cycle	200	目标网络更新间隔
Grad_norm_clip	10	梯度裁剪, 防止梯度爆炸

表 C1 MPP 环境下 MAAIC 算法训练参数
Table C1 Training parameters of MAAIC algorithm in MPP

参数名	设置值	说明
Training step	3000000	训练最大步数
Learning rate	0.0005	Adam 优化的学习率
Replay buffer size	600000	最大的样本存储数量
Mini-batch size_epsilon	32	更新参数所用到的样本数量
Anneal_steps	500000	模拟退火的步数
α	1	外在奖励系数
β	0.5	内在奖励系数

References

1 Kurach K, Raichuk A, Stańczyk P, Zajac M, Bachem O, Espeholt L, et al. Google research football: A novel reinforcement learning environment. In: Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA: 2020. 4501−4510

2 Ye D, Liu Z, Sun M, Sun M, Shi B, Zhao P, et al. Mastering complex control in MOBA games with deep reinforcement learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA: 2020. 6672−6679

3 Ohmer X, Marino M, Franke M, König P. Why and how to study the impact of perception on language emergence in artificial agents. In: Proceedings of the Annual Meeting of the Cognitive Science Society. Virtual Event: 2021.

4 Yao Hong-Ge, Zhang Wei, Yang Hao-Qi, Yu Jun. Joint regression object localization based on deep reinforcement learning. *Acta Automatica Sinica*, 2023, **49**(5): 1089−1098
(姚红革, 张玮, 杨浩琪, 喻钧. 深度强化学习联合回归目标定位. 自动化学报, 2023, **49**(5): 1089−1098)

5 Wu Xiao-Guang, Liu Shao-Wei, Yang Lei, Deng Wen-Qiang, Jia Zhe-Heng. A gait control method for biped robot on slope based on deep reinforcement learning. *Acta Automatica Sinica*, 2021, **47**(8): 1976−1987
(吴晓光, 刘绍维, 杨磊, 邓文强, 贾哲恒. 基于深度强化学习的双足机器人斜坡步态控制方法. 自动化学报, 2021, **47**(8): 1976−1987)

6 Sun Chang-Yin, Mu Chao-Xu. Important scientific problems of multi-agent deep reinforcement learning. *Acta Automatica Sinica*, 2020, **46**(7): 1301−1312
(孙长银, 穆朝絮. 多智能体深度强化学习的若干关键科学问题. 自动化学报, 2020, **46**(7): 1301−1312)

7 Lillicrap T P, Hunt J J, Pritzel A, Heess N, Erez T, Tassa Y, et al. Continuous control with deep reinforcement learning. In: Proceedings of the International Conference on Learning Representations. San Juan, Puerto Rico: 2016.

8 Tampuu A, Matiisen T, Kodelja D, Kuzovkin I, Korjus K, Aru J, et al. Multi-agent cooperation and competition with deep reinforcement learning. *Plos One*, 2017, **12**(4): Article No. e0172395

9 Sunehag P, Lever G, Gruslys A, Czarnecki M W, Zambaldi V, Jaderberg M, et al. Value-decomposition networks for cooperative multiagent learning. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. Stockholm, Sweden: 2017. 2085−2087

10 Rashid T, Samvelyan M, Schroeder C, Farquhar G, Foerster J, Whiteson S. QMIX: Monotonic value function factorization for deep multi-agent reinforcement learning. In: Proceedings of the International Conference on Machine Learning. Stockholm, Sweden: PMLR, 2018. 4295−4304

11 Rashid T, Farquhar G, Peng B, Whiteson S. Weighted QMIX: Expanding monotonic value function factorization for deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 2020, **33**: 10199−10210

12 Wang J H, Ren Z, Liu T, Yu Y, Zhang C. Qplex: Duplex dueling multi-agent Q-learning. In: Proceedings of the International Conference on Learning Representations. Virtual Event: 2021.

13 Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv: 1406.1078, 2014.

14 Gupta J K, Egorov M, Kochenderfer M. Cooperative multi-agent control using deep reinforcement learning. In: Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems. São Paulo, Brazil: Springer, 2017. 66−83

15 Busoniu L, Babuska R, De Schutter B. Multi-agent reinforcement learning: A survey. In: Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision. Singapore: IEEE, 2006. 1−6

16 Hernandez-Leal P, Kartal B, Taylor M E. A survey and critique of multi-agent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 2019, **33**(6): 750−797

17 Tan M. Multi-agent reinforcement learning: Independent vs. cooperative agents. In: Proceedings of the 10th International Conference on Machine Learning. Amherst, USA: 1993. 330−337

18 Hernandez-Leal P, Kartal B, Taylor M E. Is multi-agent deep reinforcement learning the answer or the question? Abrief survey. *Learning*, 2018, **21**: 22

19 Oroojlooyjadid A, Hajinezhad D. A review of cooperative multi-agent deep reinforcement learning. arXiv preprint arXiv: 1810.05587, 2018.

20 Lowe R, Wu Y, Tamar A, Harb J, Abbeel P, Mordatch I. Multi-agent actor-critic for mixed cooperative-competitive environments. arXiv preprint arXiv: 1706.02275, 2017.

21 Pesce E, Montana G. Improving coordination in small-scale multi-agent deep reinforcement learning through memory-driven communication. *Machine Learning*, 2020: 1−21

22 Kim W, Cho M, Sung Y. Message-dropout: An efficient training method for multi-agent deep reinforcement learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. Hawaii, USA: 2019. 6079−6086

23 Foerster J, Farquhar G, Afouras T, Nardelli N, Whiteson S. Counterfactual multi-agent policy gradients. In: Proceedings of the AAAI Conference on Artificial Intelligence. New Orleans, USA: 2018. 2974−2982

24 Son K, Kim D, Kang W J, Hostallero D E, Yi Y. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In: Proceedings of the International Conference on Machine Learning. Long Beach, USA: PMLR, 2019. 5887−5896

25 Yang Y, Hao J, Liao B, Shao K, Chen G, Liu W, et al. Qatten: A general framework for cooperative multi-agent reinforcement

learning. *CoRR*, 2020: Article No. 03939

26 Yang Y, Hao J, Chen G, Tang H, Chen Y, Hu Y, et al. Q-value path decomposition for deep multi-agent reinforcement learning. In: Proceedings of the International Conference on Machine Learning. Virtual Event: PMLR, 2020. 10706−10715

27 Foerster J N, Assael Y M, De Freitas N, Whiteson S. Learning to communicate with deep multi-agent reinforcement learning. arXiv preprint arXiv: 1605.06676, 2016.

28 Sukhbaatar S, Szlam A, Fergus R. Learning multi-agent communication with back-propagation. In: Proceedings of the Annual Conference on Neural Information Processing Systems. Barcelona, Spain: 2016. 2244−2252

29 Peng P, Wen Y, Yang Y, Yuan Q, Tang Z, Long H, et al. Multi-agent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play star-craft combat games. arXiv preprint arXiv: 1703.10069, 2017.

30 Singh A, Jain T, Sukhbaatar S. Learning when to communicate at scale in multi-agent cooperative and competitive tasks. arXiv preprint arXiv: 1812.09755, 2018.

31 Fu J, Li W, Du J, Huang Y. A multi-scale residual pyramid attention network for medical image fusion. *Biomedical Signal Processing and Control*, 2021, **66**: Article No. 102488

32 Locatello F, Weissenborn D, Unterthiner T, Mahendran A, Heigold G, Uszkoreit J, et al. Object-centric learning with slot attention. arXiv preprint arXiv: 2006.15055, 2020.

33 Jiang J, Lu Z. Learning attentional communication for multi-agent cooperation. arXiv preprint arXiv: 1805.07733, 2018.

34 Das A, Gervet T, Romoff J, Batra D, Parikh D, Rabbat M, et al. TarMAC: Targeted multi-agent communication. In: Proceedings of the International Conference on Machine Learning. Long Beach, USA: PMLR, 2019. 1538−1546

35 Liu Y, Wang W, Hu Y, Hao J, Chen X, Gao Y. Multi-agent game abstraction via graph attention neural network. In: Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA: 2020, 34. 7211−7218

36 Raileanu R, Denton E, Szlam A, Fergus R. Modeling others using oneself in multi-agent reinforcement learning. In: Proceedings of the International Conference on Machine Learning. Stockholm, Sweden: 2018. 4257−4266

37 Jaques N, Lazaridou A, Hughes E, Gulcehre C, Ortega P, Strouse D, et al. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In: Proceedings of the International Conference on Machine Learning. Long Beach, USA: PMLR, 2019. 3040−3049

38 Littman M L. Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the Machine Learning Proceedings. New Brunswick, USA: 1994. 157−163

39 Samvelyan M, Rashid T, De Witt C S, Farquhar G, Nardelli N, Rudner T G, et al. The star-craft multi-agent challenge. In: Proceedings of the Autonomous Agents and Multi-agent Systems. Montreal, Canada: 2019. 2186−2188

40 Yu W W, Wang R, Li R Y, Gao J, Hu X H. Historical best *Q*-networks for deep reinforcement learning. In: Proceedings of the IEEE 30th International Conference on Tools With Artificial Intelligence. Volos, Greece: IEEE, 2018. 6−11

41 Anschel O, Baram N, Shimkin N. Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning. In: Proceedings of the International Conference on Machine Learning. Sydney, Australia: PMLR, 2017. 176−185

俞文武 中国科学院软件研究所博士研究生. 2016 年获得湖南大学学士学位. 主要研究方向为深度强化学习.
E-mail: wenwu2016@iscas.ac.cn
(**YU Wen-Wu** Ph.D. candidate at the Institute of Software, Chinese Academy of Sciences. He received his bachelor degree from Hunan University in 2016. His main research interest is deep reinforcement learning.)

杨晓亚 中国科学院软件研究所硕士研究生. 2017 年获得吉林大学学士学位. 主要研究方向为强化学习.
E-mail: yangxiaoya17@mails.ucas.ac.cn
(**YANG Xiao-Ya** Master student at the Institute of Software, Chinese Academy of Sciences. She received her bachelor degree from Jilin University in 2017. Her main research interest is reinforcement learning.)

李海昌 中国科学院软件研究所副研究员. 2016 年获得中国科学院自动化研究所博士学位. 主要研究方向为计算机视觉, 模式识别和深度学习. 本文通信作者.
E-mail: haichang@iscas.ac.cn
(**LI Hai-Chang** Associate professor at the Institute of Software, Chinese Academy of Sciences. He received his Ph.D. degree from Institute of Automation, Chinese Academy of Sciences in 2016. His research interest covers computer vision, pattern recognition, and deep learning. Corresponding author of this paper.)

王 瑞 中国科学院软件研究所工程师. 2012 年获得山东大学硕士学位. 主要研究方向为智能信息处理.
E-mail: wangrui@iscas.ac.cn
(**WANG Rui** Engineer at the Institute of Software, Chinese Academy of Sciences. She received her master degree from Shandong University in 2012. Her main research interest is intelligent information processing.)

胡晓惠 中国科学院软件研究所研究员. 2003 年获得北京航空航天大学博士学位. 主要研究方向为智能信息处理与系统集成.
E-mail: hxh@iscas.ac.cn
(**HU Xiao-Hui** Professor at the Institute of Software, Chinese Academy of Sciences. He received his Ph.D. degree from Beihang University in 2003. His research interest covers intelligent information processing and system integration.)