中国科学: 信息科学 2022 年 第 52 卷 第 5 期: 742–764
SCIENTIA SINICA Informationis
《中国科学》杂志社
SCIENCE CHINA PRESS
评述

基于通信的多智能体强化学习进展综述

王涵, 俞扬*, 姜远

南京大学计算机软件新技术国家重点实验室, 南京 210023
* 通信作者. E-mail: yuy@lamda.nju.edu.cn

收稿日期: 2020–06–17; 修回日期: 2020–08–20; 接受日期: 2020–10–30; 网络出版日期: 2022–05–12

摘要 强化学习 (reinforcement learning, RL) 技术经历了数十年的发展, 已经被成功地应用于连续决策的环境中. 如今强化学习技术受到越来越多的关注, 甚至被冠以最接近通用人工智能的方法之一. 但是, 客观环境中往往不仅包含一个决策智能体. 因此, 我们更倾向于以多智能体强化学习 (multi-agent reinforcement learning, MARL) 为技术手段, 应对现实的复杂系统. 近十年来, 多智能体系统 (multi-agent system, MAS) 和强化学习的结合日渐紧密, 逐渐形成并丰富了多智能体强化学习这个研究方向. 回顾 MARL 的相关工作, 我们发现研究者们大致从学习框架的讨论、联合动作学习, 以及基于通信的 MARL 这 3 个角度解决 MARL 的问题. 而本文重点调研基于通信的 MARL 的工作. 首先介绍选取基于通信的 MARL 进行讨论的原因, 然后列举出不同性质的多智能体系统下的代表性工作. 希望本文能够为 MARL 的研究者提供参考, 进而提出能够解决实际问题的 MAS 方法.

关键词 强化学习, 多智能体系统, 部分可观测环境, 多智能体通信, 协同控制

1 引言

如今, 强化学习 (reinforcement learning, RL) 作为人工智能领域中的热门话题之一, 吸引了很多不同专业领域学者的关注. 强化学习的本质 [1] 是让智能体在与环境的不断交互中, 通过尝试和犯错, 学习如何在特定的时间段中作出合适的序列性决策以解决社会和工程中遇到的问题.

强化学习的发展过程有着鲜明的特征. 在 20 世纪 50 ∼ 60 年代以前, 关于 RL 的探索都局限于反复的试错. 而后, 贝尔曼提出贝尔曼方程 (Bellman equation) 以及离散的动态系统中的最优控制理论并且将其建模为马尔可夫决策过程 (Markov decision process, MDP). 然而最优控制的潜在前提是我们知道系统相关的所有特性, 实际上这个前提往往是无法满足的. 这一点恰恰是强化学习的独特研究背景之一. 在 20 世纪 60 年代, "Reinforcement Learning" 第一次出现在了工程领域的试错方法总结中. 其中影响最深远的就是 Minsky 的工作 [2], 其中提到了试错和信任分配 (credit assignment) 的问

引用格式: 王涵, 俞扬, 姜远. 基于通信的多智能体强化学习进展综述. 中国科学: 信息科学, 2022, 52: 742–764, doi: 10.1360/SSI-2020-0180
Wang H, Yu Y, Jiang Y. Review of the progress of communication-based multi-agent reinforcement learning (in Chinese). Sci Sin Inform, 2022, 52: 742–764, doi: 10.1360/SSI-2020-0180

题, 这些都是强化学习的起源. 此后研究者们从未知环境中试错的出发点提出了基于时序差分的方法 (temporal differences, TD) [3]、Q- 学习 [4] 和 SARSA [5].

当时的 RL 技术还处于比较朴素的阶段, 主要针对的是规模较小的离散状态离散动作的场景. 当状态或者动作空间连续时, 便无法得到准确的值函数. 这时就需要对值函数进行近似, 从而产生了基于值函数 (value based) 的强化学习方法. 此外, 如果直接对策略进行近似, 学习的目标就可以直接定义为最优策略搜索 (policy search) 的性能. 如果在策略近似的同时还引入了值函数的近似, 并且策略是基于值函数的评价而更新的, 这类方法属于策略近似的一种特殊形式, 称为 Actor-Critic 方法, 其中的 Actor 指的是策略, Critic 指的是值函数.

自从 2015 年, Mnih 等 [6] 在 Atari 环境中利用深度 Q- 学习取得了突破性进展之后, 深度强化学习 (deep reinforcement learning, DRL) 便开始在机器学习、人工智能领域掀起了一阵热潮. 研究者们不断发现 DRL 的巨大潜力, 不论是机器人控制 [7]、优化与调度 [8], 或者是游戏和博弈 [6,9] 等方面都能够借助于 DRL 来解决. 而当 DRL 在解决现实问题的时候, 研究者们往往高估了它的能力, 低估了实现它的难度 [10].

事实上, 现实世界中的问题是十分复杂的. 本文总结, 现实世界的复杂性很大程度上体现在: 多数任务所涉及的系统规模较为庞大, 并且根据一些规则或者常识可以分解为多个完成不同子任务的个体. 为了完成某个任务, 系统需要多个智能体同时参与, 它们会在各自所处的子空间分散执行任务, 但从任务层面来看, 这些智能体需要互相配合并且子决策的结果会互相影响. 这样的系统可以被称为多智能体系统 (multi-agent system, MAS). 在多智能体系统中, 各个智能体需要在环境不完全可知的情况下互相关联进而完成任务. 简而言之, 它们可以互相协同, 或者互相竞争, 也可以有竞争有合作. 如果将强化学习技术用于上述场景中, 相异于传统强化学习场景的是, 在这种系统中, (1) 至少有两个智能体; (2) 智能体之间存在着一定的关系, 如合作关系、竞争关系, 或者同时存在竞争与合作的关系; (3) 每个智能体最终所获得的奖赏会受到其余智能体的影响. 通常, 我们将这种场景下的强化学习技术称为多智能体强化学习 (multi-agent RL, MARL). MARL 场景中的环境是复杂的、动态的. 这些特性给学习过程带来很大的困难, 例如, 随着智能体数量的增长, 联合状态及动作空间的规模会呈现出指数扩大, 带来较大的计算开销; 多个智能体是同时学习的, 当某个智能体的策略改变时, 其余智能体的最优策略也可能会变化, 这将对算法的收敛性和稳定性带来不利的影响.

针对上述 MARL 的困难, 研究者们提出智能体可以在动态的环境中借助于一些辅助信息弥补其不可见的信息, 从而高效学得各自的策略. 为了达到这个目的, 研究者们提出了一些方法, 可以大致被分为以下几类: (1) 学习框架的讨论, 这类工作意在探索一种可行的学习框架, 因此这类工作更多地偏向于将已有的机器学习 (machine learning, ML) 研究背景或者 RL 技术向 MAS 的场景中作融合; (2) 联合动作学习, 这类方法基于单智能体的视角, 即将多个智能体合并为一个整体, 而原本各个智能体的动作则被视为系统 “子部件” 的动作, 但是这类方法在状态动作空间维数较高时会面临学习效率不高的问题; (3) 智能体之间的通信, 即智能体通过发送和接收抽象的通信信息来分析环境中其他智能体的情况从而协调各自的策略. 学习框架和联合的多动作学习算法主要依赖于集中式的训练学习或者直接共享某些局部信息等条件. 不难发现, 更容易适应于现实系统的是基于通信的这类方法: 集中各个智能体, 并使各个智能体分享的局部信息的训练模式在实际应用中很难满足. 因此, 我们希望智能体之间可以不依赖于集中式的训练学习方式, 依旧能够在不完全可知的环境中分析感知其他智能体的信息, 从而完成任务. 所以, 通过通信信息来补充环境的缺失信息的这种思路更容易被泛化. 近期, 更为迫切的实际需求是参与任务的多个智能体不愿意进行诸如策略参数等信息的共享. 这就是联邦学习 (federated learning, FL) 的要求. 在这种情况下, 算法更需要保证智能体之间只有有限的抽象信息用来

传输, 从而满足各个智能体对于隐私的需求.

在多智能体系统中, 如果对智能体的保护程度较高, 即智能体不会直接分享重要的内部信息, 智能体则需要一些辅助的信息来补充这一部分缺失的不可观测状态. 最直观的做法就是互相传递有意义的通信信息, 这种信息可以在一定程度上帮助智能体对环境进行理解. 但是, 在满足严格的互相不可见, 且有限信息共享的要求的前提下, 智能体之间要做到完全的独立学习与通信是十分困难的事情. 即便是在基于通信的 MARL 的工作中, 也有很大一部分工作依赖于集中式的训练学习或者依赖于智能体之间重要信息的共享 (例如智能体的动作). 而这样的学习方式有悖于实际的需求. 因此, 智能体需要能够自主地在更新策略的同时自行调整通信信息, 从而做到完全的不依赖于集中式的或基于局部信息共享的学习.

本文重点回顾基于通信的 MARL 的工作. 我们总结了基于通信的 MARL 的发展历程, 以及不同性质的多智能体系统场景下的代表性工作, 进一步给出不同工作的分析以及适用条件. 最后, 我们总结并展望未来可能进行的探索方向. 我们由衷希望本文能够为对研究 MARL 的读者提供帮助.

2 单智能体强化学习

本节主要介绍单智能体 DRL 的基础知识. 首先, 回顾传统的强化学习, 即单智能体 (single-agent RL, SARL) 的相关概念, 然后, 介绍深度强化学习的兴起、前沿的算法和现存的问题以及挑战. 方便后续章节为大家引入多智能体 RL 的问题设定、前沿研究的大致分类和框架.

2.1 强化学习: RL

通常, 强化学习中智能体与环境的交互过程可以被建模为 MDP. MDP 可以被记为一个五元组 $(\mathcal{S}, \mathcal{A}, R, T, \gamma)$, 其中与智能体本身紧密相关的包括 $\mathcal{S}$, 代表智能体所处的有限状态空间, $\mathcal{A}$, 表示智能体的有限动作集合. 智能体在环境中的状态改变主要体现在转移函数 T 上, T 函数是这样一种映射: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$. 它代表了从当前状态 $s \in \mathcal{S}$ 执行了动作 $a \in \mathcal{A}$ 之后到达 $s' \in \mathcal{S}$ 的概率, 这个概率是由环境本身的不确定性决定的. 此外, 智能体还会在执行动作到达下一状态之后得到环境反馈给它的奖赏 (reward). 在 MDP 中奖赏函数 R 被定义为 $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$. 奖赏 R 是一个立即的数值信号, 如果智能体在环境中作了长期决策, 它将在立即奖赏和长期的累计奖赏之间作出权衡 (trade-off). MDP 中的 $\gamma \in [0, 1]$ 就是为了平衡立即奖赏和长期累计奖赏的折扣因子.

在环境完全可观测的条件下, 最优决策可以由 MDP 模型通过动态规划的方法直接求解. 而求解 MDP 其实是希望找到一个最优策略 $\pi : \mathcal{S} \to \mathcal{A}$, 也就是找到从状态到动作的映射函数. 最优指的是, 执行这个策略之后, 智能体能够得到最大的累计期望奖赏. 有很多方法可以得到这样的策略 π. 之前提到的策略迭代的方法, 在能够准确获得状态奖赏以及转移函数信息的条件下, 求解这样的 MDP 是可行的. 但是实际情况是这些信息不能准确获得, 这时就需要用 RL 方法来求解 MDP.

基于值函数的方法是一类经典的 RL 方法, 以 Q- 学习为代表. 如果单智能体在稳态环境中, 并且对环境完全可观测, 其离散动作对应的策略可以通过 Q- 学习的方法解出. 采用 Q- 学习方法的智能体需要在学习过程中记录并更新关于每一个状态 – 动作对的 $\hat{Q}$- 值, 记为 $Q^{\pi}(s, a) = \mathbb{E}\{\sum_{k=0}^{\infty} \gamma^k r_{k+1} | s_0 = s, a_0 = a, \pi\}$. Q- 值本身表示的是智能体所处的当前状态在执行某动作之后对于当前策略的累计奖赏的预测. 每一次对 $\hat{Q}$- 值的更新实际上都是对最优策略 π^* 对应的最优值函数 Q^* 的近似. 智能体从状态 s 执行动作 a 而到达状态 s' 之后, Q- 值的更新方式如下:

$$\hat{Q}(s, a) \leftarrow \hat{Q}(s, a) + \alpha \left[\left(r + \gamma \max_{a'} \hat{Q}(s', a') \right) - \hat{Q}(s, a) \right], \tag{1}$$

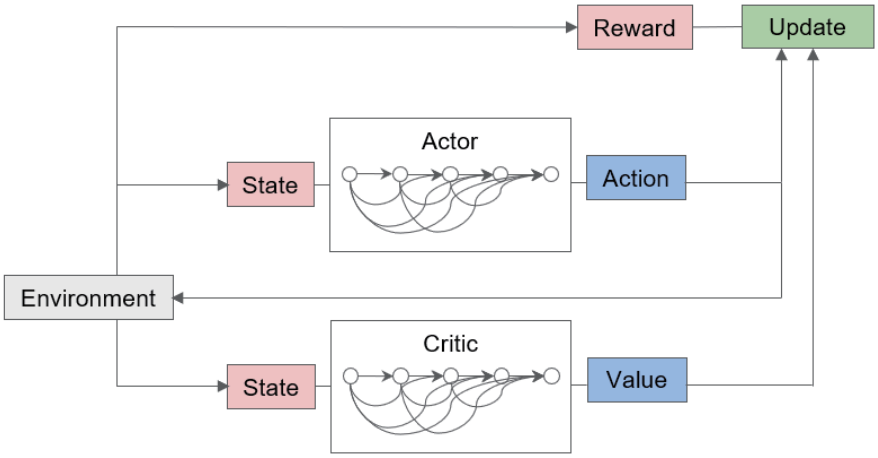

图 1 (网络版彩图) Actor-Critic 方法

Figure 1 (Color online) The framework of the Actor-Critic methods

其中 α 是更新步长, 取值范围为 $[0,1]$. 当状态和动作空间为离散并且有限时, Q- 学习往往能够收敛到最优 Q^*. 此外, SARSA 算法作为一种同策略 (on-policy) 的值函数 RL 算法, 它鼓励智能体在行动中学习, 而且自始至终只有一个策略 π.

另一类经典算法是基于直接策略优化的 RL 算法. 这类方法能够直接对策略的参数进行梯度更新. REINFORCE 是这类方法中比较具有代表性的经典算法. REINFORCE 利用蒙特卡洛方法 (Monte Carlo, MC) 来估计未来的累计回报 (return), 即累计奖赏之和. 假设策略参数是 θ, 其更新方式如下:

$$\theta_{t+1} \leftarrow \theta_t + \alpha R_t \frac{\nabla \pi(A_t; S_t, \theta_t)}{\pi(A_t; S_t, \theta_t)}, \tag{2}$$

其中, R_t 是 MC 得到的累计奖赏, α 是学习速率, A_t 是由策略 π 产生的动作. 由于直接进行梯度更新通常会带来较大的方差 (variance), 因此在用梯度进行直接策略优化的时候我们需要引入一个基准函数 (baseline) $b(s)$, 并且函数 $b(s)$ 不以动作 a 为变量, 例如 V- 值函数.

基于值函数和直接策略优化的方法各有优势, Actor-Critic 方法就是一类希望融合以上两类方法优势的 RL 学习框架. 这种方法包含两个主要元素, 一个是 Actor, 它表示策略, 主要通过策略梯度来进行优化求解; 另一个是 Critic, 它是一个值函数. Actor-Critic 是策略梯度和 Q- 学习的结合. 考虑到贝尔曼方程:

$$Q(s_t, a_t) = \mathbb{E}[r + \gamma V(s_{t+1})], \tag{3}$$

通常, 策略梯度的优化函数可以写成如下形式:

$$L = \sum \log \pi_\theta(s_t, a_t)(r + \gamma V(s_{t+1}) - V(s_t)). \tag{4}$$

这样 Critic 只需要一个 V- 值网络近似 V- 值, 其优化过程完全可以参考 Q- 学习的做法. 由此, 我们把策略梯度和 Q- 学习进行了融合, 得到了 Actor-Critic 方法, 如图 1 所示.

2.2 深度强化学习: DRL

DRL 真正崭露头角是在 2015 年文献 [6] 这项工作中. 作者将深度学习与传统 RL 的 Q- 学习融合起来, 提出了深度 Q- 网络 (deep Q-network, DQN), 并将其用于 Atari 游戏上, 展现了超越人类的

水平. 这项工作引入了两项技术. (1) 经验回放 (experience replay): DQN 有一个记忆库用于存储之前的轨迹, Q- 学习是一种异策略 (off policy) 的学习方法, 它可以学习过去经历过的, 甚至是学习别的智能体的轨迹. 所以每次 DQN 更新时, 都可以随机抽取一些之前的轨迹进行学习. 随机抽取这种做法打乱了轨迹之间的相关性, 也使得神经网络更新更有效率. (2) 目标 Q- 网络 (fixed Q-targets): 这是一种打乱相关性的机理, 如果使用这种目标 Q- 网络, DQN 会使用两个结构相同但参数不同的神经网络, 其中的评估 Q- 网络进行正常的更新, 随时具备最新的参数, 而目标 Q- 网络的更新频率较慢, 在评估 Q- 网络更新若干次之后才会依据评估 Q- 网络参数进行调整. 这样的方式通过更新的时间差打乱采样轨迹的相关性. 在 DQN 出现之后, 不少学者尝试提升 DQN 的性能, 并取得了一些成果. 例如, 在更新 Q- 值时取最大值的操作虽然可以快速让 Q- 值向可能的优化目标靠拢, 但是很容易过犹不及, 导致过度估计 (over estimation), 即最终得到的模型有很大的偏差 (bias). 而文献 [11] 提出了 Double DQN 结构, 通过解耦目标 Q- 值动作的选择和目标 Q- 值的计算这两步, 消除由于贪心而导致的过度估计. Schaul 等 [12] 注意到, 在经验回放池里面的不同样本由于 TD 误差的不同, 对反向传播的作用是不一样的: TD 误差越大, 对反向传播的作用越大; 反之则越小. 如果 TD- 误差的绝对值较大的样本更容易被采样, 算法收敛速度较快. 此后, 文献 [13] 从网络结构上将 DQN 进行了分解, 提出的 Dueling DQN 将 Q- 网络分成两部分, 第 1 部分仅与状态有关, 与具体的动作无关, 这部分是价值函数部分, 记作 V; 第 2 部分同时与状态和动作有关, 这部分是优势函数 (advantage function) 部分, 记为 A. 最终 Q- 网络的输出由价值函数网络的输出和优势函数网络的输出线性组合得到. 算法其实是对优势函数部分做了中心化的处理, 并且维持原本的 DQN 算法流程. 以上这些算法各自都可以提升 DQN 性能的某个方面. 因为它们都构建在同一个算法基础上, 所以能够被整合使用. 在文献 [14] 中, 研究人员讨论了综合上述所有方法的整合性方案, 并提出了单智能体系统的 Rainbow 算法. 研究人员展示了整合后的表现, 证明了它们很大程度上是互补的, 同时也提供了分类测试结果, 显示了每种改进对于提升性能的贡献.

由于值函数近似的方法无法有效处理连续动作空间的问题, 因此需要借助直接策略近似来增强 DRL 处理这类问题的能力. 直接策略近似大多依赖策略梯度对策略进行优化, 这类方法起初是在确定性策略上提出的, 它直接对轨迹的价值函数求导. Lillicrap 等 [15] 提出的 DDPG 就是用了确定性策略, 在文献 [16] 中出现的 DPG 基础上结合 DQN 的特点产生出来的算法. 首先 DDPG 算法是异策略的, 所以凭借着行为策略和评估策略的不同可以增加探索; 另外, 在 DDPG 中, 通过在行为策略上添加噪声使算法在环境中进行高效探索. Schulman 等 [17] 指出, 策略梯度算法的弱点在于固定地更新步长, 当步长不合适时, 更新了参数后可能不会得到更优的策略. 因此作者提出 TRPO, 旨在保证策略质量的稳定性. 而后, Schulman 等 [18] 认为 TRPO 依然有一个超参数并且其值难以确定, 造成了很多使用限制, 因此提出了 PPO 用来避免超参数的选择. 而文献 [19] 提出了 A3C 算法, 该算法利用多线程技术, 使智能体在多个线程里分别和环境进行交互学习, 每个线程都把学习的成果汇总起来, 指导在环境中后续的交互. 通过这种方法, A3C 避免了经验回放相关性过强的问题, 同时做到了异步并发学习. 这些 DRL 技术在文献 [20] 中被集中调查, 并且在经典 RL 环境, 以及仿真机器人环境等场景下进行了对比测试.

基于上述 DRL 技术的提升, 研究者们逐渐发现 RL 的潜能, Silver 等 [21] 指出, 在博弈的环境中学习自我博弈 (self-play) 使得围棋智能体 AlphaGo 成功击败了世界级的棋手; 此外, DRL 还可以被用来模拟仿生机器人的行动 [20]; OpenAI 也凭借 DRL 技术, 在 Dota 2 游戏中将顶级玩家击败; 直到 2019 年星际争霸游戏被攻克 [22]. DRL 技术逐渐被广为人知的企业所重视, 例如谷歌、特斯拉利用 DRL 研发无人驾驶汽车, 希望 DRL 能够全方位多角度地服务于现实世界 [23].

然而, 在实际应用 DRL 解决问题时, 诸如样本利用率低以及不稳定性等弊端逐渐暴露出来. 这让我们认识到: 现实世界中的问题是十分复杂的, 诸多场景涉及多智能体系统, 直接利用 DRL 来处理这些问题还无法达到令人满意的程度. 但正因为此, 研究者们开始总结单智能体 RL 的技术问题, 并开启 MARL 的研究进程.

3 多智能体强化学习: MARL

DRL 面对的问题的复杂性很大程度上体现在: 多数任务所涉及的系统结构较为繁杂, 往往根据一些规则或者常识可以分解为多个完成不同子任务的个体. 也就是说, 为了完成某个任务, 系统需要多个智能体同时参与, 它们会在各自所处的子空间分散执行任务, 但从任务层面来看, 它们需要互相配合并且这些智能体各自的子决策结果会互相影响.

在这样的多智能体系统中, 各个智能体需要在环境不完全可知的情况下互相关联, 进而完成任务. 它们需要互相配合. "配合" 没有限定一定要合作, 可以互相竞争也可以有竞争有合作, 依据任务本身来定. 对于 MAS 的场景, 同样需要对这类问题进行建模然后探索解决问题的方法.

3.1 多智能体强化学习建模

多智能体强化学习通常将部分可观测的马尔科夫过程 (partially observable MDP, POMDP), 记为 Dec-POMDP(S,T,A,R,γ) 作为数学模型. 假设有 N 个智能体, 每个智能体有自己的局部状态、动作空间, 并且会接收各自的奖赏. 因此, 单智能体系统中的 MDP 可以扩展到 MAS 中, 状态空间可以分解为 $S=S_1\times S_2\times\cdots\times S_N$, 动作空间可以表示为 $A=A_1\times A_2\times\cdots\times A_N$, 奖赏函数可以表示为 $R=R_1\times R_2\times\cdots\times R_N$, 而转移函数变为 $T=S\times A_1\times\cdots\times A_N$. 在每个时间步 t, 每个智能体 k 都会基于观测的状态 o_t^k 选择执行相应的动作 a_t^k, S 和 A 是全局的状态和联合动作. 在这种设定下, 由于智能体之间是不可见的, 它们以信息共享的方式补充它们不可见的缺失信息. 由于 DRL 的技术日渐成熟, 学者们尝试以 DRL 为基础, 在 MAS 中作 DRL 的扩展, 提出了一系列多智能体深度强化学习 (multi-agent DRL, MADRL) 算法, 并在一些经典的 MAS 实验环境中进行了有效性的验证.

3.2 多智能体强化学习进展: MARL, MADRL

对于 MARL 的研究并不是近期才出现的. 早期, 研究者们主要试图寻找一种能够将 RL 技术转接到 MAS 设定上的一种学习框架. 文献 [24] 尝试了在各个智能体独立地进行局部的 Q- 函数更新, 而不使用信息共享的方法下, 智能体能否完成任务的实验. 实验说明忽视智能体之间的互相影响, 仅仅将环境视为静态时, 独立的 RL 并不能够解决多智能体系统下的问题. 近期, 文献 [25] 将 DQN 用于 MARL 问题中, 并说明了在得到合适的奖赏的前提下, 这种 MA-DQN 的学习框架对合作和竞争的学习环境均可使用. 此外, 有两项工作 [26, 27] 将 Actor-Critic 的学习框架迁移到了 MARL 中, 同样说明这种方法的使用场景包括合作或者竞争以及混合二者的环境. 另外, Leibo 等 [28] 将自我博弈 (self-play) 的技术扩展到了群体性的 MAS 环境中. 除此之外, 近期出现的很多方法 [29~31] 尝试将较为成熟的 DRL 算法用在 MAS 的环境中, 观测这些 DRL 算法在 MAS 环境中的表现. 在 MAS 场景下, 由于每个智能体都是环境的一部分, 各个智能体在学习的同时对环境也产生了一定的影响. 这就导致了每个智能体所处的环境是变化的, 也就产生了 MARL 中所谓的不稳定的问题 (non-stationary). Foerster 等 [32] 对经验回放池 (replay buffer) 进行了改进, 提出了指纹标记方法 (fingerprints). 所谓的指纹, 就是一种可以标识采样数据新旧程度的值函数. 也就是说可以把对其他智能体策略的估计添加到经验回放池中,

同时利用 MAS 框架下的重要性采样收集样本, 进而对每个智能体的 Q- 值函数进行更新. 这些算法均可用在 MAS 中, 无论系统中的智能体是合作关系、竞争关系还是兼有合作和竞争.

随着研究者们对 MARL 研究的不断深入, 我们发现对于 MARL 的研究不仅仅局限于寻找一个通用的学习框架, 很多工作力图在某个特定的场景中进行探索. 例如, 针对合作场景的工作 [33~36] 出现在了我们的视野中. 事实上, 确实存在大量的需要智能体之间合作完成任务的例子, 例如, 文献 [37, 38] 将多个智能体的动作整合到一个 Q- 网络中, 这样就可以从整体的角度看待各个智能体的表现. 简而言之, 这种方法将分散的智能体合并成一个智能体, 将多智能体参与下的 MARL 简化为 RL. 但是这样的方法面对大规模场景时就难以奏效了. 类似地, 文献 [39] 提出在多智能体合作的场景中, 令同质的智能体共享策略参数来同步更新所有智能体的策略的方法. 也就是说我们可以维护一个全局的值函数, 并且可以将其应用于 DQN, DDPG 以及 TRPO, 从而得到了以上算法的 MAS 版本: PS-DQN, PS-DDPG 和 PD-TRPO. 但是这些算法均过多地依赖于智能体内部信息的共享.

除了上述方法外, 基于通信的方法近年来颇受研究者的关注. 因为通信本身更符合实际的应用场景的需要. 完全的去中心化学习依赖于智能体之间有效的信息传递, 这种信息传递不是简单的私密的信息共享, 而是通过智能体不断地跟环境交互后和收到其余智能体的通信信息后, 所给出的有意义的反馈. 接下来的章节将重点介绍通信相关的工作, 然后分析现存的问题以及未来可能的工作方向.

4 基于通信的多智能体强化学习

在实际系统中, 参与任务的各个智能体往往会考虑安全或者隐私, 不希望过多地依赖于直接共享各自领域的局部信息来完成任务. 这些关键的局部信息可能包括: 各个智能体的动作, 或者直接共享同样的策略网络结构, 甚至是集中起来共享经验池以更新各个智能体的策略, 也就是中心化的学习 (centralized learning) 的概念. 下面我们简要地将现有的基于通信的 MARL 或者 MADRL 算法归类, 然后列举现在每一类的研究进展. 依据算法利用的 DRL 技术, 现有的基于通信的多智能体深度强化学习算法 (communication-based multi-agent deep reinforcement learning, CB-MADRL) 大致可以分为以下几类:

(1) 基于值函数的 CB-MADRL. 这种方法依靠对值函数 (以 Q- 值函数为主) 进行重构使之适用于 MA 系统, 这部分工作在表 1 中总结.

(2) 包含直接策略搜索的 CB-MADRL. 由于表现不够稳定, 单纯使用直接策略搜索作 MAS 决策的工作十分少见. 现在大多学者都倾向于选择基于 Actor-Critic 框架作 CB-MADRL 的研究, Actor 是各个智能体的局部子策略, 通信的过程和效果主要依靠 Critic 来判定, 这部分算法在表 2 中总结.

(3) 提升通信效率的突破. 我们发现在以上两类方法逐渐发展的过程中, 学者们对这些算法也尝试了改进, 意在提升通信的效率进而提升算法的学习性能, 相关工作总结于表 3.

(4) 关于应急通信的研究. 如今研究领域间的交叉已经极为常见, 很多语言研究领域的研究者们开始尝试从通信语言如何产生, 以及通信信息的质量度量等方向进行研究, 从而丰富了多智能体通信的研究方向, 相关工作总结于表 4.

如果要求智能体通过通信的方式彼此协同完成一项任务, 智能体就需要通过将自己的信息, 例如状态和动作等, 编码成一条有限长的信息, 传递给其余智能体, 同时也接受来自其余智能体的信息. 其目的就是希望智能体能够将收到的信息作为观测的补充, 尽可能地还原不可见状态的信息, 进而得到近似全局状态下的最优动作. 上述过程中, 通信的问题主要集中在如何传递高质量的通信信息, 具体来说主要考虑: 通信信息需要包含哪些内容, 以及如何及时地更新通信信息. 在接下来的几个小节中, 我

表 1 基于值函数的 CB-MADRL 算法统计

Table 1 The algorithms based on value function approximation

算法名称	算法总结	智能体关系	实验环境
DIAL/RIAL	依据 Q- 网络进行动作的选取和通信信息的收发	合作	Switch Riddle, MNIST Games
DDRQN	智能体通过深度循环 Q- 网络进行动作和通信信息的评估	合作	Multi-Agent Riddles

表 2 包含直接策略搜索的 CB-MADRL 算法统计

Table 2 The algorithms principled on policy search

算法名称	算法总结	智能体关系	实验环境
CommNet	在中心化学习中心化执行的框架下使智能体进行连续通信	合作	Switch Riddle, MNIST Games
BiCNet	引入多智能体双向协调网络, 保证沟通方式的有效性和可扩展性	合作	StarCraft, Combat Games
MD-MADDPG	智能体使用共享内存作为通信信道, 在采取动作之前, 智能体先读取内存再写入反馈	合作	Swapping, Cooperative Navigation, Waterworld
Intrinsic A3C	对具有高度的互信息的动作进行额外奖励以加强协同	混合	Cleanup, Harvest
MACC	使用反事实推理来训练动作策略和通信策略	合作	MNIST, Traffic Management

表 3 提升通信效率的突破算法统计

Table 3 The algorithms designed to improve communication flexibility and learning efficiency

算法名称	算法总结	智能体关系	实验环境
ATOC	使用注意力通信模型学习何时需要通信, 以及如何整合共享信息以进行合作决策	合作	Navigation, Pushball, Predator-Prey
TarMAC	允许每个智能体通过简单的基于签名的软注意力机制, 主动选择将消息发送到哪些智能体	合作	Shapes, Traffic Junction, House3D
IC3Net	每个智能体接受各自的奖励进行训练，并从门机制中学习何时进行通信	合作, 竞争, 混合	BroodWars, Predator-prey, Traffic, Cooperative Navigation
I2C	提出独立推断通信, 使智能体能够学习智能体与智能体通信的先验知识	合作	Cooperative navigation, Predator prey, Traffic junction

们将主要从以上两点为大家介绍并分析现有算法的特性.

4.1 基于值函数的 CB-MADRL

之前提到的针对合作型的场景 MADRL 算法在近几年是一个比较热门的研究方向. 这个方向得到发展背后的主要推动力便是基于值函数的 CB-MADRL 的兴起.

谈到 CB-MADRL, 首先要提到的是文献 [40]. 这篇文章最先在基于值函数的 MADRL 中引入通信学习机制. 面对合作的多智能体的 POMDP 问题, 作者提出让智能体共享一个全局奖赏函数 R, 但是每个智能体只能观测到局部信息以及接收离散的有限长通信信息, 文中以 one-hot 向量来表示通信信息. 由于依赖于中心化的学习框架, 在训练时智能体可以传送连续信息; 在训练结束后, 智能体在

表 4 关于应急通信的研究统计

Table 4 The algorithms devoted to exploring the generation and utility of communication language in emergency communication

算法名称	文章总结	智能体关系	实验环境
LLM	提供学习交流的一般框架, 以及给出智能体可以最大化联合预期效用的条件	合作	Gather, Pump
Learn to Communicate	利用语言学, 认知科学和博弈论中关于语言出现的工作思想, 使智能体的语言与人类直觉保持一致	合作	Referential games
ST-GS	通过优化协作任务中的奖励可以学习如何与符号序列进行沟通	合作	Referential games
Negotiation	具有固有语义的通信通道使自私的智能体在特定任务中学会公平协商	非合作	Bargaining problem
Measuring Communication	提出沟通的因果影响 (CIC) 以直接度量正向监听的质量	合作, 竞争, 混合	Matrix communication games
Shaping Losses	为正向信号和正向监听引入归纳偏差, 以缓解联合探索问题	合作	MNIST, Treasure Hunt
Ease-of-Teaching	重置训练制度使语言更易于传授, 并使语言随着时间的推移变得更加结构化	合作	Referential games
LWM	引入语言世界模型 (LWM), 通过预测未来观测的潜在编码来解释自然语言信息	合作	2D navigation
Networked-MARL	进行紧急通信的详细分析, 并说明其与底层网络拓扑的关系	合作	Traffic management

分散执行时必须使用离散的信道进行通信. 同时作者在文章中提到了两种不同的通信结构: RIAL 和 DIAL, 图 2 所示. RIAL 结构其实是深度循环 Q- 网络 (deep recurrent Q-network, DRQN) 算法 [41] 与独立 Q- 学习 (independent Q-learning, IQL) 算法 [24] 的结合. 通信信息可以作为非稳定性环境的一种补偿, 同时 RIAL 使用两个 Q- 网络分别评价原始的动作以及离散的通信信息. 因此, RIAL 结构中 Q- 网络的输入不仅仅包含各个智能体的局部观察, 还包括上一时间步其余智能体传递的通信信息. 而 DIAL 是在 RIAL 的基础上提出的学习模式. 在训练时, DIAL 允许梯度信息在通信信道从信息接收方传回到信息发送方. 具体来说, 在中心化训练时, 通信信息的发送者直接将通信信息的输出层连接到接收方的 Q- 网络输入端. 在训练结束后, 通信实值信息被离散化. 同时这个实值信息从高斯分布中采样产生, 保证了一定的鲁棒性. 文中通过对实验结果的展示表明, DIAL 和 RIAL 均表现出远胜于不通信的实验效果.

与此同时, 文献 [42] 同样尝试了基于 Q- 学习的分布式深度循环 Q- 网络 (deep distributed recurrent Q-networks, DDRQN) 架构, 如图 3 所示. DDRQN 算法为每个智能体单独分配 DRQN 网络 [41], 然后组合这些子网络, 构建一个完整的多智能体训练系统. 为了提高训练效率, DDRQN 提出了一些改进, 在任何时刻, 每个智能体的 Q- 网络的输入不仅仅包含观测值、通信信息, 同时还要包含历史动作序列. 这样便于智能体把握历史状态, 从而在一定程度上保持稳定性; 另外, 智能体之间可以共享网络参数. 通过这种参数共享的方式, 网络中可学习参数的数目会大幅度减少, 从而加快学习的速度.

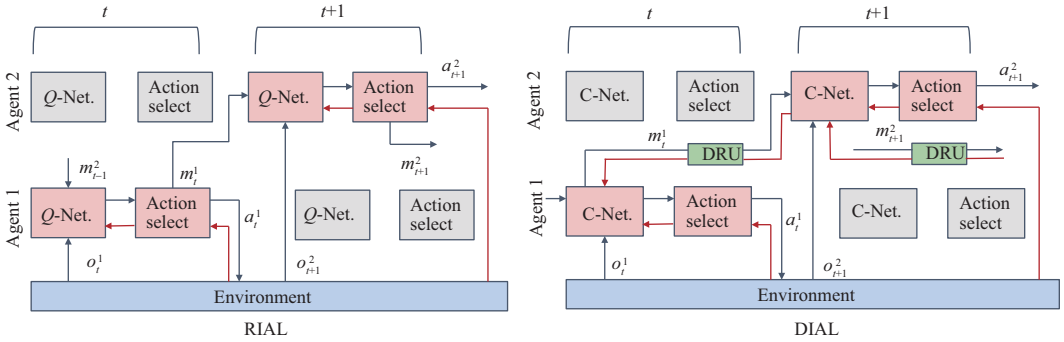

图 2 (网络版彩图) 基于 Q- 网络的通信模型 (修改自文献 [40]). DIAL 能够使梯度信息直接通过 Q- 网络在智能体间传递

Figure 2 (Color online) Q-Network based model (modified from [40]). DIAL enables communication information to be transmitted between agents by gradients through Q-nets

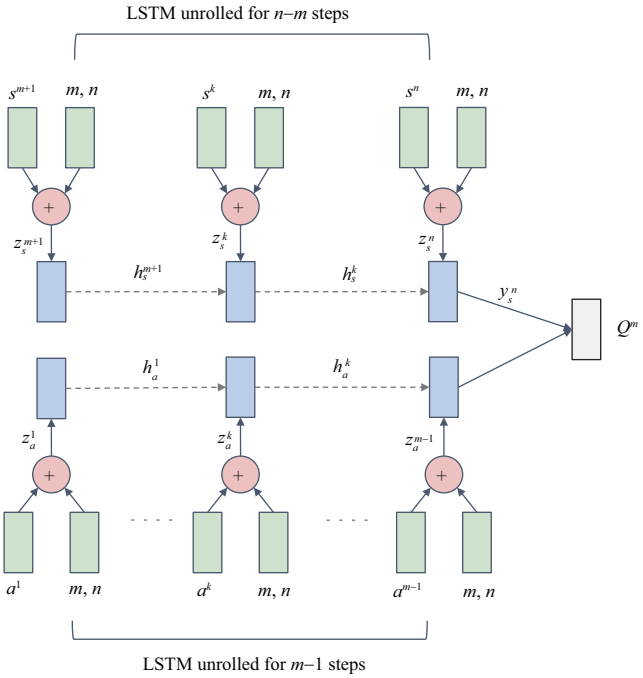

图 3 (网络版彩图) DDRQN (修改自文献 [41]) 的网络输入不仅包含局部观测和通信信息, 同时还包含动作信息以得到更稳健的决策

Figure 3 (Color online) The input of the Q-networks (modified from [41]) includes not only the observation and communication information, but also the historical action sequence, so as to make more stable decisions

4.2 包含直接策略搜索的 CB-MADRL

基于直接策略搜索的 CB-MADRL 算法最早出现在文献 [43] 中. 文献 [43] 同样是针对多智能体的 POMDP 问题, 提出了 CommNet 结构, 如图 4 所示, 并假设智能体之间传递的消息是连续变量. 但需要注意的是, 文献 [43] 遵循的是中心化训练中心化执行 (centralized training centralized execution, CTCE) 框架, 因而在大规模的多智能体环境下, 网络的结构会相对复杂, 并且需要处理的数据维数很

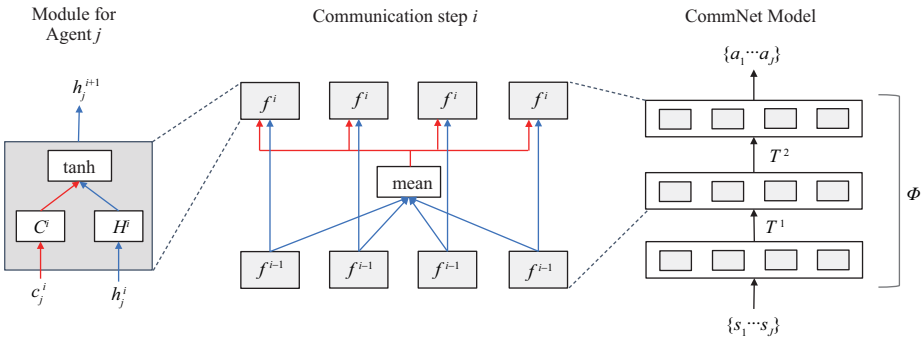

图 4 (网络版彩图) CommNet 是一个全局网络 (修改自文献 [43]), 同时接收所有智能体的观测从而输出各个智能体的动作

Figure 4 (Color online) The idea behind the algorithm (modified from [43]) is to build a global network that simultaneously receives the local observations of all agents as input and outputs the decisions of all agents

大, 强化学习算法训练较为困难并且表现不稳定.

由于 Actor-Critic 框架能够维持策略梯度和基于值函数的学习方法的优势, 因此学者们倾向于以 Actor-Critic 框架为基础探索新的解决 MAS 问题的算法. 首先, 文献 [44] 提出了 BiCNet 结构, 并且假设智能体之间传递的信息是离散的. 不同之处在于, 该文针对的不是完全合作的 MAS 问题, 而是零和随机博弈问题 (zero-sum stochastic game). 在小规模 MARL 问题中, 常见的解决方案是采用 Minimax Q-learning. 但对于星际这样的大规模游戏, Minimax Q-learning 就无法求解了. BiCNet 使用的是 DDPG 算法. 该算法在中心化的学习去中心化的执行 (centralized training decentralized execution, CTDE) 的框架下进行学习. 此外, 考虑到算法在大规模 MAS 环境下的可扩展性问题, 算法允许智能体之间共享模型参数. 但是, 为了能够提高训练效率, BiCNet 假设每个智能体都维护着同样的全局观测; 为了简单起见, BiCNet 假设对手的策略是固定的, 方便对对手建模. 遗憾的是这些假设尽管在一定程度上简化了算法设计, 但是也限制了 BiCNET 的可泛化性. 内存驱动 (memory driven, MD) 的合作型通信多智能体 DDPG 算法 (MD-MADDPG) [45] 是在多智能体深度确定性策略梯度 (multi-agent deep deterministic policy gradient, MADDPG) 的基础上提出的. 在 MD-MADDPG 中 (如图 5 所示), 智能体使用共享内存作为通信信道. 也就是说智能体能够对内存的信息先有一个初步的认识和判断. 因此, 在智能体决定动作之前, 它首先会读取内存数据, 然后在内存中写下它对当前内存信息的反馈 (response). 显然, 这种情境下, 智能体的策略是基于它的观测, 以及所有智能体对内存内容的反馈. 作者通过实验发现了智能体对内存反馈的变化跟任务也是有关系的: 复杂的任务往往会使得智能体对内存的反馈呈现较大的差异.

现在的多智能体通信渐渐地出现了较为统一的框架, 即 Actor 产生原始动作以及通信信息, Critic 会在训练时对当前的所有可获得的观测信息, 以及 Actor 产生原始动作和通信信息进行评价. 最近出现的基于 A3C 框架的 MADRL 算法便是一个典型的例子 [46]: 在图 6 中, 智能体传递出去的通信信息应该依据智能体的观测、收到的通信信息, 以及将要执行的动作而产生. 当然也可基于这个框架作调整, 例如, 以共享的底层网络来对输入的观测和通信状态进行特征提取, 而高层的网络分出两个 heads, 每个 head 对应了实际动作的产生和新的通信信息的产生. Actor 对应的策略的输入大致包含智能体的观测、接收到的通信信息. 此外, 很多工作会将部分历史序列同样送入策略网络. 这样, 一方面能够在一定程度上稳定智能体的行为, 在 Actor 更新时不会出现过大的波动; 另一方面能够指导 Critic 给出较为可靠的评价. 本文则利用了因果推断的机制去计算其动作或者通信信息对其他智能体的影响

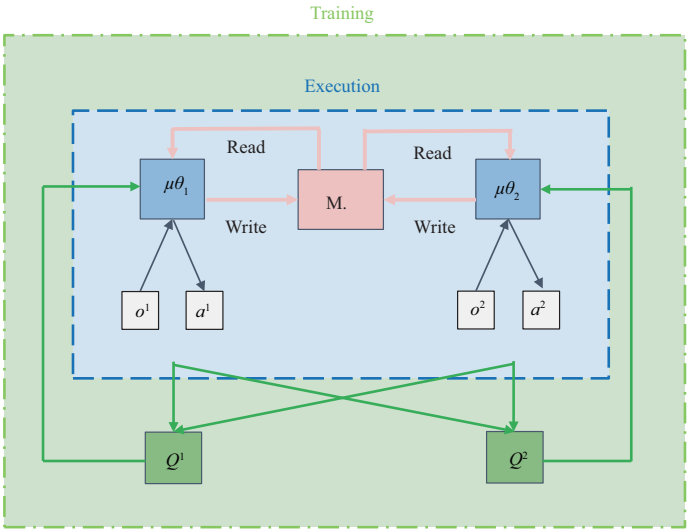

图 5 (网络版彩图) 在训练和测试期间, MD-MADDPG (修改自文献 [45]) 会基于本地观测和共享内存的内容生成新的动作并更新通信信息

Figure 5 (Color online) During training and testing, MD-MADDPG (modified from [45]) generates new actions and updates communication information based on local observations and the contents of shared memory

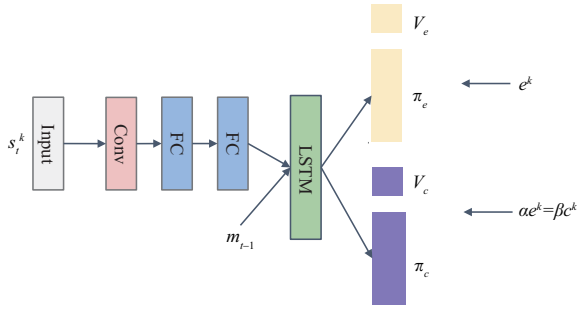

图 6 (网络版彩图) 因果通信网络 (修改自文献 [46]) 首先从观测和接收到的通信信息中提取特征, 然后分别用于产生新动作以及通信信息

Figure 6 (Color online) The network (modified from [46]) performs feature extraction on the observation s_t^k and the communication state m_{t-1}, followed by two heads π_e and π_m to generate the action and the communication information, respectively

力. 这种智能体影响力数值上相当于智能体动作或者通信信息之间的互信息; 并且影响力的奖励是使用分布式的方式来计算的, 能够有效解决突发通信的问题. 近期的文献 [47] 同样立足于因果推断, 提出了多智能体反事实通信学习. 作者发现, 现在的多智能体系统中的智能体并没有观察到环境的全部状态. 由于缺乏对整个状态的了解, 各智能体必须要共享观测信息或学习如何交流. 作者提出了 MACC 算法, 它解决了智能体的部分可观察性问题: MACC 让智能体同时学习动作策略和通信策略, 并具有联合可观测性. 这意味着可以通过对所有智能体的观察确定环境的完整状态. MACC 使用反事实推理来学习动作和通信策略. 这允许智能体预测其他智能体将如何对某些消息作出反应, 以及环境将如何对某些动作作出反应, 从而提高学习效率. MACC 使用 Actor-Critic 学习框架: 一个集中训练的 Critic 和分散的 Actors. 在 simple reference 环境中与通信和非通信算法进行了比较, 结果表明 MACC 能够通过有效的通信策略辅助智能体决策.

4.3 提升通信效率的突破

近年来的 CB-MADRL 的工作主要都是依赖于 Actor-Critic 框架的, 并且 Actor 会维护原始动作策略以及通信信息策略. 在整个学习过程中, 通信策略的学习很关键, 因为通信的目的就是为了尽可能地还原 POMDP 中不可见的部分. 另外, 通信策略的优化过程还需尽量高效. 在 Actor-Critic 的框架已经被大家认可之后, 越来越多的旨在提高通信质量和通信效率的工作出现在了各类顶级会议和顶级刊物上.

首先, 一个较为直观的问题是: 前面介绍的 CB-MADRL 算法均假设每个时间步所有的智能体都参与通信. 这种假设在一些场景中不够灵活, 会影响智能体的学习效率. Jiang 等 [48] 提出了一种能够让智能体在任一时间步, 自行决定是否需要与其他智能体进行通信, 进一步选择与哪些智能体进行通信的算法, 构建了基于注意力机制 (attention) 的通信模型 ATOC, 如图 7 所示. 该模型基于智能体的局部观察, 可同时适用于协作环境 (共享一个全局的奖赏函数或者拥有各自的奖赏函数) 以及竞争环境 (实质也是协作环境, 因为算法只控制一方). 其基本思想是, 通过各个智能体的局部观测与动作的编码, 决定是否需要与其视野可见范围内的其他智能体进行通信, 并且决定与哪些智能体进行通信. 对于决定进行通信的智能体, 我们称之为发起者 (initiator). 发起者从其视野范围内选择协作者共同形成一个通信群组, 这个通信群组在整个 episode 中动态变化, 并且只在需要时存在. 与 BiCNet 类似, 该文采用一个双向的 LSTM 网络作为通信群组之间的通信信道, 并且通信网络以通信群组中各个智能体的局部观测与动作的编码作为输入, 输出高层编码信息作为各智能体策略网络的额外输入, 指导协作策略的生成. 研究者将 ATOC 实现为端到端训练的 Actor-Critic 模型的扩展, 在测试阶段, 所有智能体共享策略网络、注意力单元和信道. 因此 ATOC 在大量智能体的情况下具备很好的扩展性. 而与 ATOC 类似, Das 等 [49] 也关注这个问题, 但是提出的 TarMAC 框架的通信对象的选择过程基于完全的学习而不依赖于任何预先设置的参数. 具体来说, 参照图 8, 每次传递通信信息时, 发送者同时会广播一个密钥, 该密钥对通信信息接收方智能体的属性进行编码, 而后由接收方智能体评价通信信息的准确性. 整个过程基于签名的软注意力机制 (soft attention) 来实现, 不依赖于额外的人为的参数控制. 另外, 文献 [50] 主要考虑在 MAS 场景中赋予智能体选择何时进行通信的能力. 这项工作提出的网络结构 IC3Net (individualized controlled continuous communication network) 可以用于混合 MA 场景而非仅仅局限于完全合作的 MAS 场景中. IC3Net 是基于 CommNet 的一种直接策略搜索的全局网络结构, 通过利用 gating mechanism 让智能体能够决定通信的时机, 从而提升学习效率. 但如图 9 所示, 因为 IC3Net 是基于 CommNet 的结构的, 也就是说多个智能体的策略网络会集成为一个全局的策略网络. 显然这里的训练和执行都是要智能体中心化地执行.

在实际应用中, 客观条件是通信信道的带宽往往是有限的, 不足以维持所有智能体同时都在有限带宽的信道中送入通信信息. 如果通信信息超出带宽限制, 就会出现信息丢失或者阻塞等情况. Kim 等 [51] 考虑到了通信带宽的问题, 并针对 MAS 场景设计了遵循 CTDE 模式的新算法. 与 ATOC 不同的是, 文献 [51] 使用了通信领域的 MAC (medium access control) 技术进而提出了 SchedNet 算法. 如图 10 所示, SchedNet 能够使得每个智能体生成各自的权重, 然后在需要通信时选择权重较大的前 K 个智能体使用信道传送信息. Ding 等 [52] 同样注意到: 现有的工作侧重于广播通信, 不仅不切实际而且会导致信息冗余, 甚至会影响学习过程. 为了解决这些困难, 作者提出了独立推断通信 (I2C), 如图 11 所示. 这种简单而有效的模型可以帮助智能体学习通信的先验. 先验知识通过因果推理来学习, 并通过前馈神经网络实现. 通过多智能体强化学习中的联合动作 – 价值函数推断一个智能体对另一个智能体的影响, 以说明智能体间通信的必要性. 该文还对智能体策略进行了规范化, 以便更好地利

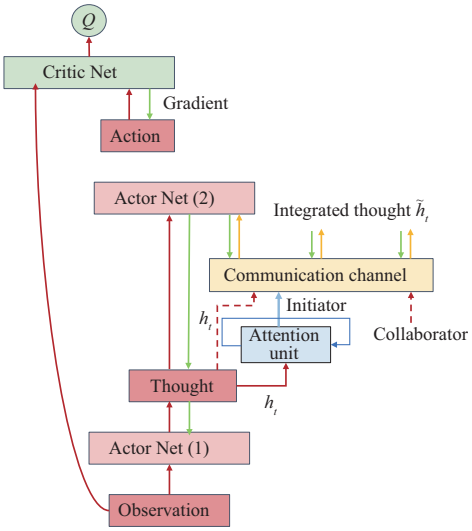

图 7 (网络版彩图) 基于注意力机制的 ATOC (修改自文献 [48])

Figure 7 (Color online) ATOC is principled on the soft attention mechanism (modified from [48])

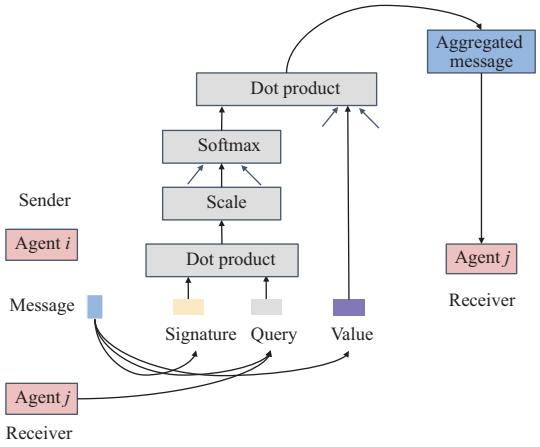

图 8 (网络版彩图) 基于注意力机制的 TarMAC (修改自文献 [49])

Figure 8 (Color online) TarMAC is built on the soft attention mechanism (modified from [49])

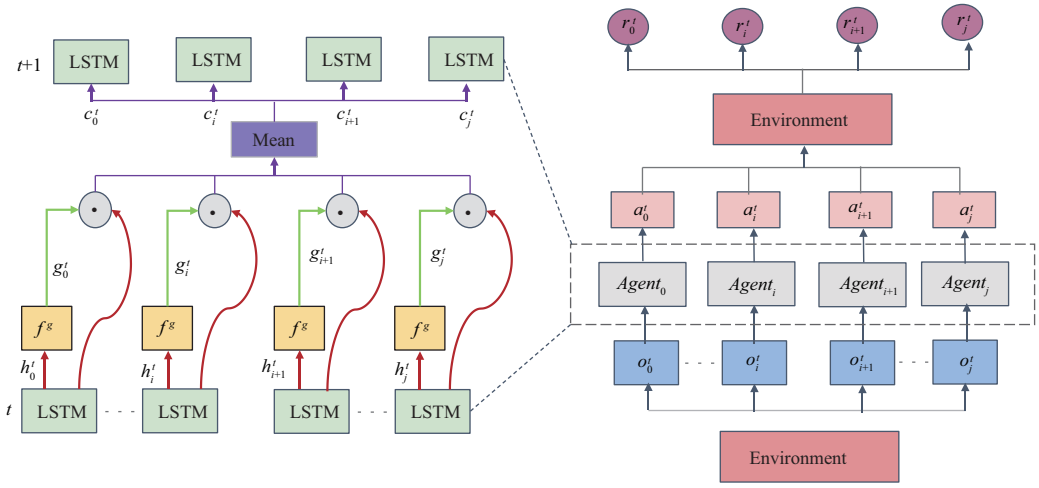

图 9 (网络版彩图) 以 CommNet 为基础的 IC3Net (修改自文献 [50])

Figure 9 (Color online) IC3Net is based on CommNet structure (modified from [50])

用已传递的消息. 实验结果表明, I2C 不仅可以降低通信开销而且在多种多智能体协作场景下可以提高性能.

4.4 关于应急通信的研究

数十年前, MAS 领域的研究者们已经着眼于分布式多智能体的通信研究 [53]. 很多语言研究领域的学者开始尝试从通信语言如何产生, 以及通信信息的质量度量等角度丰富多智能体通信的研究方向.

Claudia 等早在文献 [53] 中提出语言学习中行动和交流的一般框架 (language-learning model,

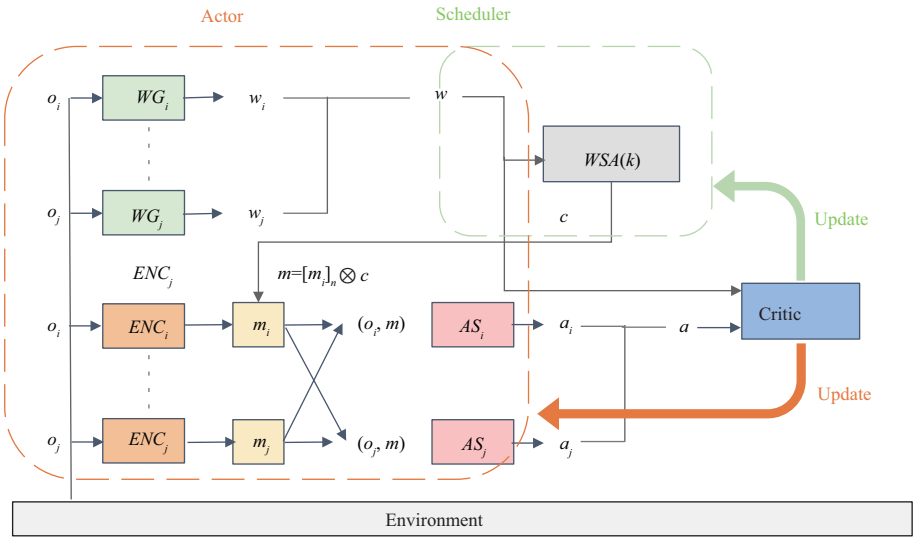

图 10　(网络版彩图) 使用 MAC 的 SchedNet (修改自文献 [51])

Figure 10　(Color online) SchedNet uses the MAC technology in the field of communication (modified from [51])

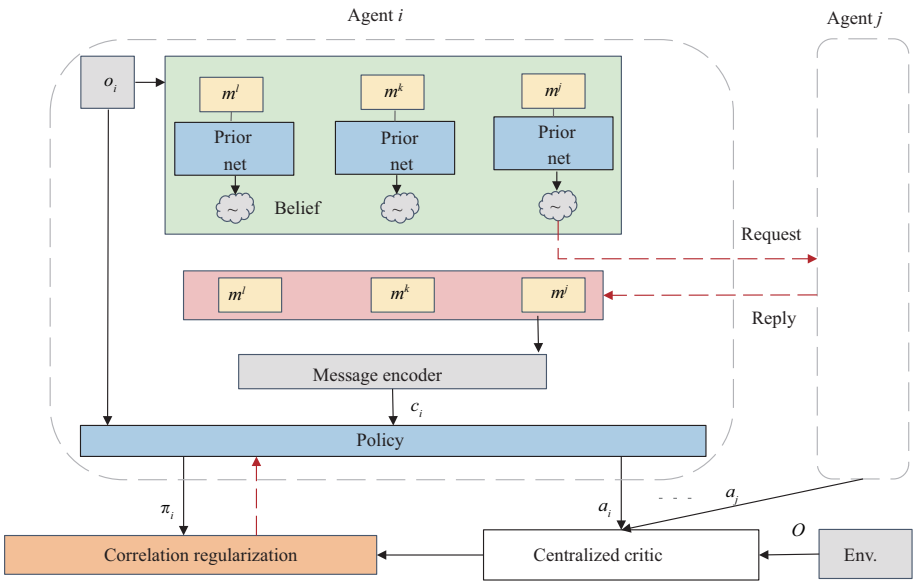

图 11　(网络版彩图) I2C 依据 CTDE 学习方式, 包含信息编码器、先验和策略网络以及中心化的评价网络 (修改自文献 [52])

Figure 11　(Color online) I2C can be instantiated by any framework of CTDE with a joint action-value function. I2C consists of a prior network, a message encoder, a policy network, and a centralized critic (modified from [52])

LLM), 如图 12 所示. 文章指出, 当更新置信状态的过程涉及到对其他智能体通信信息的解释时, 对语言内涵的理解尤其重要. 虽然许多通信工作的一般假设是通信语言是通信双方所共享的, 但该文认为智能体对通信语言不完美的理解同样可以作为置信状态的一部分从而得到可以泛化的策略.

近年来, 有很多同时期的工作都发现多个智能体可以通过交流完成一些指定任务, 并且其自创的交流 “语言” 和人类语言非常相似. 研究者们在通信的基础上尝试探索语言究竟是如何产生的. 例如,

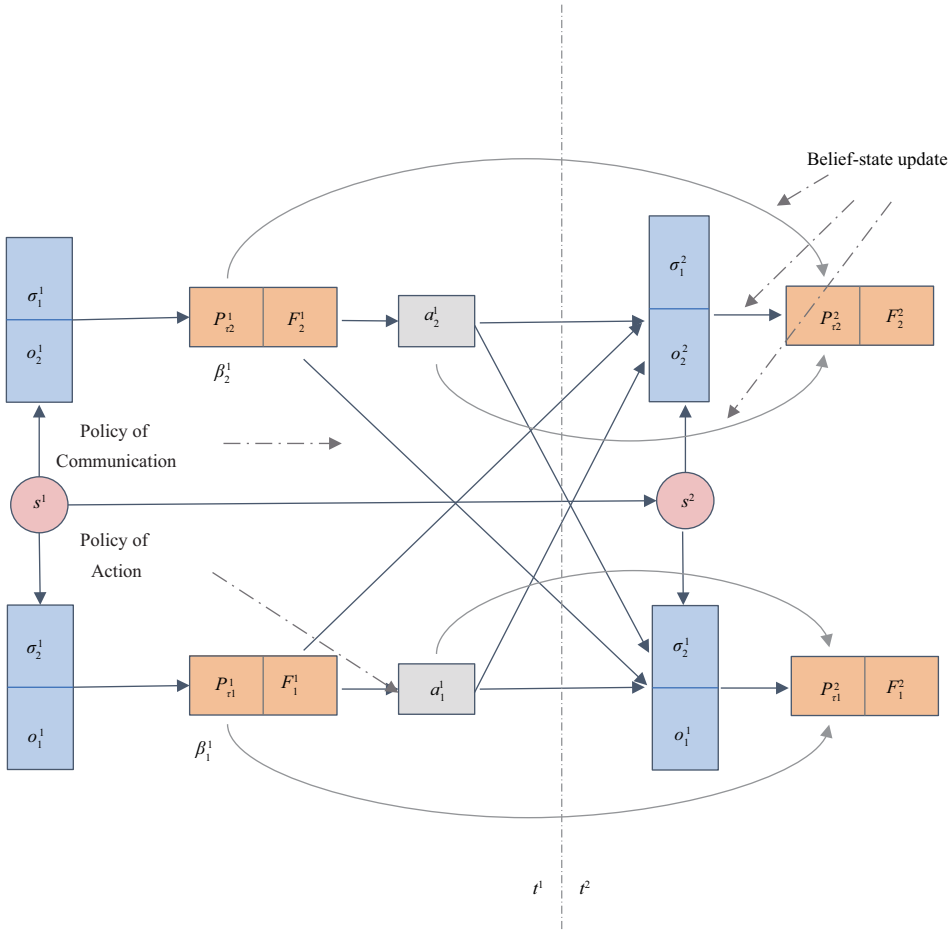

图 12 通用语言学习模型图 (修改自文献 [53]) 指出了交互双方进行语言学习时各要素的关系

Figure 12 The general language-learning model (modified from [53]) gives a graphical overview of the interrelations between the various components of the language-learning process

文献 [54] 提供了描述者和猜题者两个角色, 并让它们共同参与一个游戏, 如图 13 所示. 描述者看到两张相异的图片 (分别是目标图片和干扰图片), 并得知哪张图是目标图片. 游戏的规则是让描述者提供一个词 (词库是由两个智能体提前商量好的暗号), 使猜题者可以通过这个词成功识别正确图片. 作者建立了两种简单的图像嵌入方式, 一种直接将 VGG 的 softmax 层输出, 称为 agnostic 方法; 另一种先通过卷积层降维, 把目标图和干扰图的输出混合起来, 最后再 embedding, 称为 informed 方法. 简而言之, informed 发送者能够得到更多的信息. 文中显示, 所有的实验均可成功收敛, 且答对率接近 100%. 其中, informed 方法相比于 agnostic 方法由于使用了更多的信息, 可以收敛得更快且更准.

Serhii 等 [55] 同样使用了描述者和猜题者的游戏 (referential game) 场景. 不同的是该文使用的是通信序列而不是单一的符号, 并且描述者不能获得非目标图像. 从学习的角度来看, 这种设置更加现实也更加具有挑战性. 描述者和猜题者均使用 LSTM 网络作为策略, 图 14 是模型架构示意图, 其中菱形箭头、虚线箭头和实线箭头分别代表采样、复制和确定性函数. 发送方的输入是目标图像和表示消息开始的特殊令牌. 发送方从分类分布中抽样采样, 依次生成下一个词 w_i. 消息 m_t 是通过顺序采样获得的, 直到达到最大可能长度 L 或生成特殊词条为止. 在 RL 任务中, 智能体不能获得全部的环境信

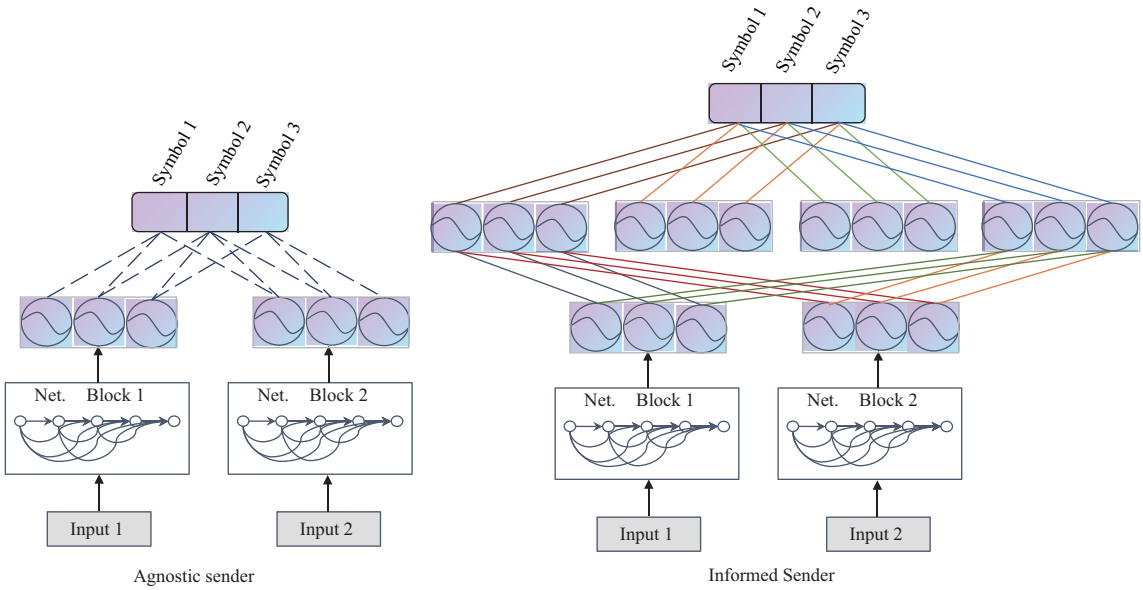

图 13 (网络版彩图) 发送者将目标和干扰表示作为输入; 接收者则是以随机顺序看到两个图像同时接收发送者的信号 (修改自文献 [54])

Figure 13 (Color online) The sender takes as input the target and distractor representations with a pointer to the target. The receiver takes as input the target and distractor image vectors in randomized order and the symbol produced by the sender (modified from [54])

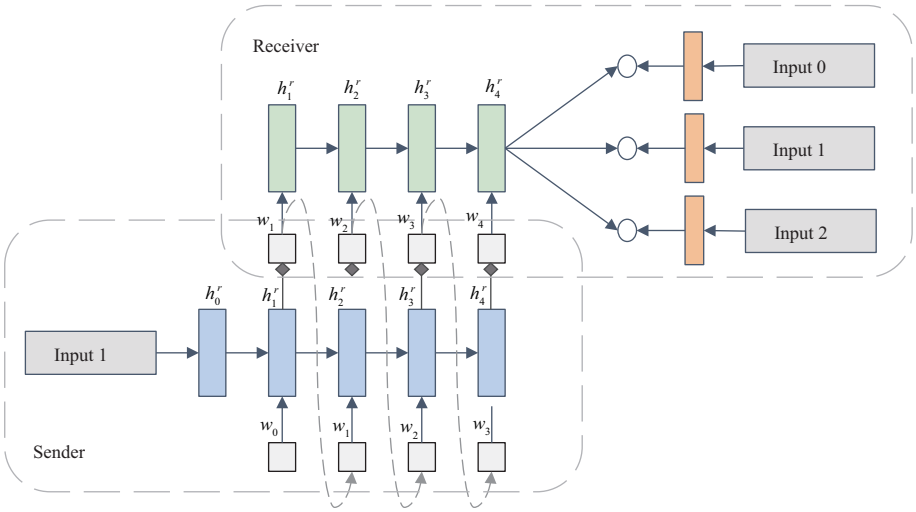

图 14 (网络版彩图) 发送者以目标图片和消息起始符为输入, 然后从分类分布中依序采样出符号输出 (修改自文献 [55])

Figure 14 (Color online) The inputs to the sender are target image and the special token that denotes the start of a message. Given these inputs, the sender generates next token in a sequence by sampling from the categorical distribution (modified from [55])

息, 即使可以获得, 该环境信息也是不可导的. 为了尽可能获得更多信息, 作者提出从 Gumbel-softmax 分布中获取词向量, 从而保证通信信息的可导性, 记作 GS-ST 方法. 从文中实验结果可以看出, 最大消息长度数值越大, 越能得到较高的学习成功率.

同样, 文献 [56] 的动机在于探索交流 (communication) 是如何产生的. 该文指出很多情况下人类的沟通是非完全合作性质的, 因此重点研究的是在这些非完全合作情境中语言的出现. 文章采用一个经典的谈判问题 (the bargaining problem) 作为研究的环境. 作者认为有效的交流在该问题中是非常关键的, 因为智能体需要通过交流传达出自己的需求, 以及通过交流推断对方的意图. 文章针对 bargaining 问题设置了两种通信通道, 第 1 种通道是 proposal, 第 2 种通道是 cheap talks. 在一对一的实验中, 发现 selfish agent 可以通过 proposal 通道和对方智能体达成相对合理的分割, 但是 selfish agent 却没有办法利用 cheap talks 通道; 相反 proposal 智能体却为了更好地和对方沟通达成相同目标, 通过 cheap talks 通道生成了语言. 近期关于应急通信的绝大多数研究表明, 增加通信信道有助于任务的成功完成. 但 Lowe 等 [57] 提出了新的问题: 我们如何确定通信在多智能体系统中的存在性? 尽管奖赏和任务完成率的提升是衡量通信效用的指标, 但也只能粗略地衡量智能体的学习交流能力. 当环境更复杂时, 对通信进行定性和定量的度量就变得十分关键, 无论是用于故障检测、性能评估, 还是建立信任. 该文将紧急通信度量的要素分为两大类: 正向发信 (positive signaling), 它表明智能体发送的信息与它的观测或动作相关; 正向监听 (positive listening), 用来表明接收到的信息在某种程度上确实影响着收信智能体的行为. 基于此, 该文检查了一些直观的度量通信的现有指标, 并指出它们在某些情况下可能会误导我们对通信的效用作出错误判断. 例如, 通过训练 DRL 智能体玩带有通信信道的简单矩阵游戏, 其结果是智能体似乎在交流 (发送的通信信息包含了其后续的动作信息), 然而这些信息并不以任何方式影响环境或其他智能体. 该文还调查了一些用于度量紧急通信的常用标准, 并就应该在何时使用这些度量标准提供了建议. 2019 年, 文献 [58] 研究在时间扩展的强化学习领域, 如果没有对智能体进行集中训练, 学习如何通信便会十分困难, 部分原因是联合探索的低效. 该文将文献 [57] 中的正向发信和正向监听的诱导偏差同时引入到奖赏函数中, 从而缓解了联合探索的低效. 该文演示了在一个简单的一步决策环境中, 这些偏差是如何提升学习效率的; 实验表明在更复杂的多步决策环境中, 具有这些诱导偏差的智能体可以获得更好的策略, 并能够分析产生通信协议.

在通信方法不断在多智能体合作场景中发挥作用的同时, 研究者们也开始在特殊的应用情境中研究其作用. 例如, 文献 [59] 在已学习的通信协议中发现了某种程度的语言结构, 通过周期性地在一个群体中依次引入新的智能体来取代旧的智能体, 探索了这种更新方式的便捷性, 并展示了它对最终语言结构的影响. Alexander 等 [60] 介绍了语言世界模型 (language world model, LWM), 一种语言条件生成模型. 它通过预测未来观测的潜在编码解释自然语言信息, 这样就为信息的理解提供了视觉基础, 类似于增强了对世界的观测能力, 其中可能包括监听者视野之外的物体. Alexander 等将这种观测合并到一个长期的内存状态中, 并允许监听者的策略对其设置访问条件, 类似于世界模型中的内存和控制器之间的关系. 实验证明这种做法提高了在 2D gridworld speaker listener 巡回任务中的有效通信和任务成功率. 近期, 文献 [61] 同样希望回答: 在学习执行一项共同任务的同时, 智能体能否开发一种语言? 为此, 文献 [61] 制定并研究了协作智能体通过一个固定的底层网络相互连接的 MARL 策略网络框架 Concept-Clustering (CC). 这些智能体可以通过交换离散的符号进行通信. 这些符号的语义并不是预定义的, 在训练过程中, 智能体需要开发一种帮助他们实现目标的语言, 作者提出了一种利用紧急通信训练这些智能体的方法: 通过将提出的框架应用于交通管制的问题证明其适用性. 该文对紧急通信进行了分析: 智能体所开发的语言是有语义基础的, 并且与底层网络拓扑结构有关.

5 归纳与展望

通过前面的回顾不难发现, 随着 DRL 技术的发展, MAS 场景的问题越来越多地可以利用 DRL

技术的迁移得到解决. 并且在各种 MAS 环境中都得到了测试, 甚至在星际这样的游戏上也取得了胜利. MADRL 的技术和突破是值得肯定的, 并且 MADRL 大背景下的现有工作已经有学者进行了总结 [62]. 我们更加希望各个智能体通过互相必要的沟通, 就能在不完全可知的环境中分析感知环境中其他智能体的信息, 从而完成既定的任务. 本节主要对现存的 CB-MADRL 算法进行归纳, 然后进一步探讨未来可能需要解决的问题和工作方向.

5.1 广为接受的技术

本小节内容主要是从 CB-MADRL 的算法中总结出广为学者们所认可的技术或者框架. 由于 CB-MADRL 归根结底还是解决 MAS 场景下 POMDP 这样一类问题, 因此其技术手段和面临的主要问题与文献 [62] 总结的大致类似. Mohamed 等 [63] 也总结了基于通信的多智能体系统的现状. 但是本文在回顾现有工作的基础上进行了分类和总结, 并在本小节主要提出可能的改进方向, 同时针对 CD-MADRL 的优势技术进行总结.

(1) 降低学习过程中的偏差. 越来越多的 MADRL 算法, 包括 CB-MADRL 算法在内, 选择了基于 Actor-Critic 的学习框架进行算法的设计. Actor-Critic 的学习框架是 DRL 技术在 MADRL 技术上较为成功的迁移, MADRL 的工作验证了 Actor-Critic 学习框架的可用性, 并且依旧在 MAS 场景下维持其综合策略梯度以及基于值函数学习的优势. 并且我们发现仅依靠直接策略搜索的算法, 往往需要维持全局的策略网络. 所以, 在提高学习效率并保持学习稳定性方面, Actor-Critic 的学习框架都是大家所认可的一种 MADRL 技术基础.

(2) 必要的信息共享. 无论是在 MADRL 的大背景还是 CB-MADRL 这个分支背景下, 大部分工作都必须依靠必要的信息共享才能进行有效的学习. 而所谓信息共享, 主要包含两个方面的共享. 第一, 参数共享. 这种方式是 MADRL 的重要学习方式, 当然 CB-MADRL 也可以在同质智能体环境中完成合作型任务时使用这种方式来提高学习效率. 第二, 中心化的学习去中心化的执行模式. 其中的中心化的学习和训练几乎是所有 CB-MADRL 甚至是 MADRL 算法遵循的学习模式. 在中心化的学习和训练过程中, 智能体能够利用其余智能体的观测、动作以及通信信息等重要信息来对未知环境进行必要的信息补充.

(3) 提升学习速率. 在设计 MARL 算法时, 向智能体提供领域知识可以极大地帮助它们学习. 因为大规模的状态和动作空间中, 延时奖赏且没有任何先验知识的 MARL 算法训练速度非常慢. 领域知识可以以多种形式提供. 例如, 设计有意义的奖赏函数, 奖励有潜力的动作; 或者在 CA-MARL 算法中预先让通信的各个智能体建立语料库等. 这些方法均可在一定程度上帮助 MARL 算法提高学习效率.

5.2 未来的工作方向

本小节的内容是我们对 CB-MADRL 这类算法发展方向的展望. 我们的出发点是希望 CB-MADRL 能够在更大程度发挥其优势, 去解决实际应用中的 MAS 的问题.

(1) 让 MARL 走向实际应用. 尽管很多 MARL 算法都在较为复杂的游戏环境中展示出了不错的能力, 但是很少有算法能够在真实环境中保证满足满足信息不完全可见的客观需求. 在这方面, 之前介绍的提升通信效率的突破, 以及关于应急通信的研究都是走向实际应用的尝试. 尤其以应急通信为例, 为一般的动态任务定义一个合适的 MARL 目标是一个困难的问题. 现有的大多数实验场景中, MARL 目标通常是在静态游戏中制定的. 这些目标对动态任务的扩展并不总是清晰的, 甚至是不可能的. 因此, 如何在实际场景中描述任务目标尤其重要: 既要保证任务能够完成, 也要对智能体的诸如学习时间、计算资源等学习成本进行惩罚.

(2) 摆脱中心化的学习和训练. 尽管中心化的学习和训练是一种主流的选择, 但是基于通信的方法理想的状态下应该做到: 严格满足不完全可见的前提, 保证智能体之间只有有限的抽象信息用来传输. 实际应用的场景往往更加苛刻: 参与任务的多个智能体不愿意共享本地信息的同时, 还可能会面临部分智能体信息缺失的问题. 例如, 文献 [64] 考虑的联邦多智能体强化学习的问题. 作者指出在实际应用中, 更一般的问题是: 算法需要提供对任务的多个参与者的局部信息的保护, 以及它们能够完成任务的保障. 如果智能体能够完全依赖于通信信息进行策略的调整, 而不需要任何敏感信息的共享, 在面对不同任务时, 智能体就可以根据不同的任务需求, 自适应地调整通信信息. 其实这种完全的去中心化的训练和执行已经有学者开始关注了 [65].

(3) 提高鲁棒性和泛化性. 如今测试 MADRL 算法的实验平台日渐丰富, 并且 Hernandez 在文献 [62] 中列举了十分全面的 MARL 的平台以供参考. 针对某一类问题设计算法时, 需要考虑如何提高算法的鲁棒性以及泛化性是十分重要的. 同样以文献 [64] 为例, 如何在信息不平衡的环境中通过通信的方式进行 MADRL 的学习过程? 文中提出了基于 Q- 学习的联邦学习框架, 这样的框架需要中心智能体完成: (i) 中心智能体需要在任务到来时决定任务的参与者; (ii) 中心智能体还需要对 Q- 值进行整合. 而实际情况是不同的任务会接踵而来, 有时甚至需要智能体同时完成多个任务, 并且不同的任务对应不同的参与者. 这种情况下较为合理的运作方式是使中心智能体仅仅完成任务 (i), 而任务 (ii) 则交由参与的智能体充分调用本地资源, 例如针对不同任务可以建立多个通信信道, 自主进行学习. 当然上述只是提高泛化性的一个举例, 不仅仅是 CB-MADRL, 甚至是 MADRL 都需要我们在鲁棒性和泛化性的提升上进行深入的思考.

6 结束语

多智能体强化学习的发展离不开深度强化学习的突破性进展. 而从多智能体强化学习这个层面来说, 在看到已有的成绩的同时, 提高学习效率、提高鲁棒性和泛化性的困难依旧存在. 这种困难是多智能体系统本身固有的性质, 例如环境的非稳定性、奖赏的延迟性和稀疏性、奖赏分配的困难性等. 尽管这些困难依旧是牵制这个领域发展的因素, 但多智能体强化学习服务于现实系统解决现实问题是学界的目标.

选择基于通信的多智能体强化学习算法进行介绍的主要原因是通信本身更迎合实际的应用场景的需求. 通信信息能够很自然地使得智能体摆脱中心化的学习的框架. 智能体之间的有效的信息传递不是简单的私密的信息共享, 而是智能体在不断地跟环境交互中所给出的有意义的反馈. 这种反馈通常是抽象的, 是需要协同的智能体互相理解的.

通过对现有的基于通信的多智能体深度强化学习算法的分析, 不难发现能用于现实多智能体系统中的基于通信的多智能体强化学习算法需要尽可能摆脱其对信息共享的依赖, 也就是尽可能保证较少的信息共享, 做到完全基于通信. 完全基于通信的隐含意义是智能体在互相不可知的情况下仅仅依靠通信信息实现缺失信息的补充, 进而摆脱过多的内部信息交流以及中心化学习的需求. 从而有如下的结果.

- 智能体的隐私需求得到保障: 智能体可以根据自身状态及接收的信息自行调整传送信息.
- 算法的泛化性得到提升: 如果智能体可以仅通过通信信息互相理解进而协同完成任务, 在面对不同任务时智能体可以根据不同的任务需求, 自适应地调整通信信息.

最后, 希望通过我们的介绍能够对多智能体强化学习, 特别是基于通信手段的多智能体强化学习方向有所关注的学者们提供一些帮助; 希望通过广大学者们的努力使得多智能体强化学习技术更快更

好地服务于现实世界中的系统.

参考文献

1 Richard S S, Andrew G B. Reinforcement learning—an introduction. In: Adaptive Computation and Machine Learning. Cambridge: MIT Press, 1998
2 Minsky M. Steps toward artificial intelligence. Proc IRE, 1961, 49: 8–30
3 Sutton R S. Learning to predict by the methods of temporal differences. Mach Learn, 1988, 3: 9–44
4 Watkins C J C H. Learning from delayed rewards. Dissertation for Ph.D. Degree. Cambridge: University of Cambridge, 1989
5 Rummery G A, Niranjan M. On-Line Q-Learning Using Connectionist Systems. Technical Report. Cambridge: University of Cambridge, 1994
6 Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518: 529–533
7 Abbeel P, Quigley M, Andrew Y N. Using inaccurate models in reinforcement learning. In: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, 2006. 1–8
8 Cheng G, Hyon S H, Morimoto J, et al. CB: a humanoid research platform for exploring neuroscience. In: Proceedings of the 6th International Conference on Humanoid Robots, Genova, 2006. 182–187
9 Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Computat, 1997, 1: 53–66
10 Irpan A. Deep reinforcement learning doesn't work yet. 2018. https://www.alexirpan.com/2018/02/14/rl-hard.html
11 Hado V H, Guez A, Silver D. Deep reinforcement learning with double Q-learning. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix, 2016. 2094–2100
12 Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay. In: Proceedings of the 4th International Conference on Learning Representations, San Juan, 2016
13 Wang Z Y, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning. In: Proceedings of the 33rd International Conference on Machine Learning, New York City, 2016. 1995–2003
14 Hessel M, Modayil M, Hasselt V H, et al. Rainbow: combining improvements in deep reinforcement learning. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, Orleans, 2018. 3215–3222
15 Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning. In: Proceedings of the 4th International Conference on Learning Representations, San Juan, 2016
16 Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms. In: Proceedings of the 31st International Conference on Machine Learning, Beijing, 2014. 387–395
17 Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization. In: Proceedings of the 32nd International Conference on Machine Learning, Lille, 2015. 1889–1897
18 Schulman J, Wolski F, Dhariwal P, at al. Proximal policy optimization algorithms. 2017. ArXiv:1707.0634
19 Mnih V, Puigdomènech B A, Mirza M, et al. Asynchronous methods for deep reinforcement learning. In: Proceedings of the 33rd International Conference on Machine Learning, New York City, 2016. 1928–1937
20 Duan Y, Chen X, Houthooft R, et al. Benchmarking deep reinforcement learning for continuous control. In: Proceedings of the 33rd International Conference on Machine Learning, New York City, 2016. 1329–1338
21 Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search. Nature, 2016, 529: 484–489
22 Vinyals O, Babuschkin I, Czarnecki W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature, 2019, 575: 350–354
23 Liu Q, Zhai J W, Zhang Z Z, et al. A survey on deep reinforcement learning. 2017, 40: 1–27
24 Tan M. Multi-agent reinforcement learning: independent vs. cooperative agents. In: Proceedings of the 10th International Conference on Machine Learning. San Francisco: Morgan Kaufmann, 1993. 330–337
25 Tampuu A, Matiisen T, Kodelja D, et al. Multiagent cooperation and competition with deep reinforcement learning. 2015. ArXiv:1511.08779
26 Lowe R, Wu Y, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments. In: Proceedings of Advances in Neural Information Processing Systems, Long Beach, 2017. 6379–6390
27 Iqbal S, Sha F. Actor-attention-critic for multi-agent reinforcement learning. In: Proceedings of the 36th International Conference on Machine Learning, Long Beach, 2019. 2961–2970
28 Leibo J Z, Pérolat J, Hughes E, et al. Malthusian reinforcement learning. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, Montreal, 2019. 1099–1107
29 Bansal T, Pachocki J, Sidor S, et al. Emergent complexity via multi-agent competition. In: Proceedings of the 6th

International Conference on Learning Representations, Vancouver, 2018
30 Leibo Z J, Zambaldi F V, Lanctot M, et al. Multi-agent reinforcement learning in sequential social dilemmas. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent, São Paulo, 2017. 464–473
31 Raghu M, Irpan A, Andreas J, et al. Can deep reinforcement learning solve erdos-selfridge-spencer games? In: Proceedings of the 35th International Conference on Machine Learning, Stockholm, 2018. 4235–4243
32 Foerster N J, Nardelli J, Farquhar G, et al. Stabilising experience replay for deep multi-agent reinforcement learning. In: Proceedings of the 34th International Conference on Machine Learning, Sydney, 2017. 1146–1155
33 Panait L, Luke S. Cooperative multi-agent learning: the state of the art. Auton Agent Multi-Agent Syst, 2005, 11: 387–434
34 Matignon L, Laurent G J, Fort-Piat N L. Independent reinforcement learners in cooperative Markov games: a survey regarding coordination problems. Knowledge Eng Rev, 2012, 27: 1–31
35 Palmer G, Tuyls K, Bloembergen D, et al. Lenient multiagent deep reinforcement learning. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, Stockholm, 2018. 443–451
36 Shayegan O, Jason P, Christopher A, et al. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In: Proceedings of the 34th International Conference on Machine Learning, Sydney, 2017. 2681–2690
37 Tesauro G. Extending Q-learning to general adaptive multi-agent systems. In: Proceedings of Annual Conference on Neural Information Processing Systems, 2003
38 Conitzer V, Sandholm T. AWESOME: a general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. Mach Learn, 2007, 67: 23–43
39 Gupta K J, Egorov M, Kochenderfer J M. Cooperative multiagent control using deep reinforcement learning. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, São Paulo, 2017. 66–83
40 Foerster N J, Assael M Y, Freitas D N, et al. Learning to communicate with deep multi-agent reinforcement learning. In: Proceedings of Advances in Neural Information Processing Systems 29, Barcelona, 2016. 2137–2145
41 Matthew J H, Stone P. Deep recurrent Q-learning for partially observable MDPs. In: Proceedings of AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents, Arlington, 2015. 29–37
42 Foerster N J, Assael M J, Freitas D N, et al. Learning to communicate to solve riddles with deep distributed recurrent Q-networks. 2016. ArXiv:1602.02672
43 Sukhbaatar S, Szlam A, Fergus R. Learning multiagent communication with backpropagation. In: Proceedings of Annual Conference on Neural Information Processing Systems, Barcelona, 2016. 2244–2252
44 Peng P, Yuan Q, Wen Y, et al. Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. 2017. ArXiv:1703.10069
45 Emanuele P, Giovanni M. Improving coordination in multi-agent deep reinforcement learning through memory-driven communication. 2019. ArXiv:1901.03887
46 Jaques N, Lazaridou A, Hughes E, et al. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In: Proceedings of the 36th International Conference on Machine Learning, Long Beach, 2019. 3040–3049
47 Simon V, Astrid V, Siegfried M, et al. Learning to communicate using counterfactual reasoning. 2020. ArXiv:2006.07200
48 Jiang J C, Lu Z Q. Learning attentional communication for multiagent cooperation. In: Proceedings of Advances in Neural Information Processing Systems, Montréal, 2018. 7265–7275
49 Das A, Gervet T, Romoff T, et al. TarMAC: targeted multi-agent communication. In: Proceedings of the 36th International Conference on Machine Learning, Long Beach, 2019. 1538–1546
50 Singh A, Jain T, Sukhbaatar S. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In: Proceedings of the 7th International Conference on Learning Representations, New Orleans, 2019
51 Kim D, Moon S, Hostallero D, et al. Learning to schedule communication in multi-agent reinforcement learning. In: Proceedings of the 7th International Conference on Learning Representations, New Orleans, 2019
52 Ding Z L, Huang T J, Lu Z Q. Learning individually inferred communication for multi-agent cooperation. 2020. ArXiv:2006.06455
53 Goldman C V, Allen M, Zilberstein S. Learning to communicate in a decentralized environment. Auton Agent Multi--Agent Syst, 2007, 15: 47–90
54 Angeliki L, Alexander P, Marco B. Multi-agent cooperation and the emergence of (nature) language. In: Proceedings of the 7th International Conference on Learning Representations, Toulon, 2017
55 Serhii H, Ivan T. Emergence of language with multi-agent games: learning to communicate with sequences of symbols. In: Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, 2017
56 Kris C, Angeliki L, Marc L, et al. Emergent communication through negotiation. In: Proceedings of the 8th Interna-

tional Conference on Learning Representations, Vancouver, 2018
57 Lowe R, Foerster J, Boureau Y, et al. On the pitfalls of measuring emergent communication. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, Montreal, 2019
58 Tom E. Biases for emergent communication in multi-agent reinforcement learning. In: Proceedings of the 33rd Conference on Neural Information Processing Systems, Montreal, 2019
59 Fushan L, Michael B. Ease-of-teaching and language structure from emergent communication. In: Proceedings of the 33rd Conference on Neural Information Processing Systems, Montreal, 2019
60 Alexander I C, Jason N. Emergent communication with world models. In: Proceedings of the 33rd Conference on Neural Information Processing Systems, Montreal, 2019
61 Shubham G, Rishi H, Ambedkar D. Networked multi-agent reinforcement learning with emergent communication. In: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, Auckland, 2020
62 Hernandez-Leal P, Kartal B, Taylor M E. A survey and critique of multiagent deep reinforcement learning. Auton Agent Multi-Agent Syst, 2019, 33: 750–797
63 Mohamed S Z, Etienne B. Learning to communicate in multi-agent reinforcement learning: a review. 2019. ArXiv:1911.05438
64 Zhuo H H, Feng W F, Xu Q, et al. Federated reinforcement learning. 2019. ArXiv:1901.08277
65 Zhang K Q, Yang Z R, Liu H, et al. Fully decentralized multi-agent reinforcement learning with networked agents. In: Proceedings of the 35th International Conference on Machine Learning, Stockholm, 2018. 5867–5876

Review of the progress of communication-based multi-agent reinforcement learning

Han WANG, Yang YU* & Yuan JIANG

National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023*, China*
* Corresponding author. E-mail: yuy@lamda.nju.edu.cn

Abstract Reinforcement learning (RL) technology has been successfully applied to various continuous decision environments in decades of development. Nowadays, RL is attracting more attention, even being touted as one of the closest approaches to general artificial intelligence. However, real-world problems often involve multiple intelligent agents interacting with each other. Thus, we focus on multi-agent reinforcement learning (MARL) to deal with such multi-agent systems in practice. In the past decade, the combination of multi-agent system and RL has become increasingly close, gradually forming and enriching the research field of MARL. Reviewing the studies on MARL, we found that researchers mainly solve MARL problems from three perspectives: learning framework, joint action learning, and communication-based MARL. In this paper, we focus from the studies on the communication perspective. We first state the reasons for choosing communication-based MARL and then list the president studies falling into the MARL category but different in nature. We hope that this article can provide a reference for developing MARL methods that can solve practical problems for the national welfare.

Keywords reinforcement learning, multi-agent system, partially observable environment, multi-agent communication, coordinated control