基于两阶段意图共享的多智能体强化学习方法

吴俊锋[1)] 王 文[1)] 汪 亮[1)] 陶先平[1)] 胡 昊[1)] 吴海军[1),2)]

[1)](南京大学计算机软件新技术国家重点实验室 南京 210023)

[2)](南京大学计算机科学技术与软件工程实验教学中心 南京 210023)

摘 要 近年来，强化学习技术在连续决策问题上展现出了强大的能力，成为机器学习领域的一个重要分支. 通过强化学习技术在多智能体系统下的发展和研究，多智能体强化学习技术有望成为群体智能行为涌现的关键技术手段，但在现阶段仍有诸多科学问题亟待解决. 在多智能体强化学习领域，如何提高智能体在协作场景下的合作能力一直是一个热门研究话题. 通信被认为是实现多智能体高水平协作的重要元素，因此有不少研究尝试从通信的角度入手，让智能体通过交流来实现更好的协作. 现有的大部分与通信有关的多智能体强化学习领域的工作关注于部分可观测问题，在这些工作中智能体通过通信信道共享了部分局部观测. 最新的一些研究开始关注如何让智能体通过共享意图来实现更好的协作. 然而，在不加限制的意图共享框架下，若智能体的最终行为与原先的意图不符，则可能会对其它智能体产生误导，此时引入通信反而产生了负作用. 因此需要一个新的多智能体意图共享框架，在有效利用意图信息的同时避免出现智能体间的意图误导. 针对上述问题，本文基于交流意图的思想，提出了一个新的多智能体强化学习意图通信框架2SIS. 在2SIS框架下，智能体在决策前需要进行两次通信，第一次通信传播意图信息，第二次通信传播意图依赖关系. 两次通信结束后每个智能体各自建立起意图依赖关系图，为了避免出现意图误导，对于意图依赖关系图上被依赖的智能体，2SIS禁止其基于其它智能体的意图进行重新决策，其最终决策即为其初始意图，仅有不被依赖的智能体被允许基于意图信息重新决策. 2SIS可以与任意基于值函数的强化学习算法结合实现训练. 在2SIS框架下训练的智能体能够学会如何正确地建立意图依赖关系从而实现单向的意图传播，并且不存在意图误导问题. 我们选用较具代表性的Double DQN算法作为基算法，在两个多智能体场景下验证了所提出方法的有效性. 有效性实验结果表明，相比于无通信以及广播式通信意图方式训练的智能体，2SIS框架下训练的智能体在收敛速度以及最终累积奖赏上有明显提升. 为了验证性能的提升来自于本文提出的方法，我们额外组织了消融实验，对方法的关键部分进行了控制变量，消融实验的结果说明2SIS框架下训练的智能体能够正确选择依赖对象是性能提升的关键. 最后我们组织了参数实验来说明本文引入的超参数会对训练过程产生怎样的影响以及如何为该参数选取一个合适的值.

关键词 多智能体系统；深度强化学习；深度多智能体强化学习；通信；意图共享；协作

中图法分类号 TP311 **DOI号** 10.11897/SP.J.1016.2023.01820

Multi-Agent Reinforcement Learning with Two Step Intention Sharing

WU Jun-Feng[1)] WANG Wen[1)] WANG Liang[1)] TAO Xian-Ping[1)] HU Hao[1)] WU Hai-Jun[1),2)]

[1)](*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210023)

[2)](*National Experimental Teaching Demonstration Center of Computer Science Technology and Software Engineering, Nanjing University, Nanjing* 210023)

Abstract In recent years, reinforcement learning has demonstrated its power in continuous decision-making problems and has become an important branch of machine learning study. As the development of reinforcement learning in multi-agent systems, multi-agent reinforcement learning

收稿日期：2022-04-09；在线发布日期：2023-01-16. 本课题得到2018年度科技创新2030—“新一代人工智能”重大项目（批准号：2018AAA0102302）资助. **吴俊锋**，硕士研究生，主要研究领域为强化学习、多智能体系统和群体智能. E-mail：wjf@smail.nju.edu.cn. **王 文**，博士研究生，主要研究领域为强化学习、多智能体系统和群体智能. **汪 亮（通信作者）**，博士，副教授，中国计算机学会（CCF）会员，主要研究领域为群体智能、群智软件、人工智能. E-mail：wl@nju.edu.cn. **陶先平**，博士，教授，中国计算机学会（CCF）会员，主要研究领域为软件方法学、群体智能. **胡 昊**，博士，副教授，中国计算机学会（CCF）会员（10740M），主要研究领域为面向无人系统的区块链软件体系、边缘智能、群体博弈等. **吴海军**，硕士，高级工程师，中国计算机学会（CCF）会员，主要研究领域为多媒体信息处理、体系结构等.

is expected to become a key technology for the emergence of swarm intelligent behavior, but there are still many scientific problems to be solved at the present stage. Cooperation problem is a popular research topic in the field of multi-agent reinforcement learning. Communication is considered a key element to achieve high-level cooperation among multi-agents. Therefore, some existing approaches try to combine communication with multiagent reinforcement learning, in order to achieve better cooperation among agents. Most of these approaches focus on partial observation problems. In these approaches, agents share their local observations with others through communication channels. In recent work, researchers attempt to let agents share intention to enhance cooperation among agents. However, under unrestricted intention sharing, if the final action of an agent is different with its original intention, it may mislead other agents, which make intention sharing harmful to train. Therefore, a new multi-agent intention sharing scheme is needed to avoid misleading intentions between agents while effectively utilizing intention information. To solve this problem, this paper proposes a multi-agent reinforcement learning intention sharing scheme—2SIS, based on the idea of intention sharing. Under the 2SIS scheme, an agent needs to communicate twice before making a decision. The first communication broadcast intention information, and the second communication broadcast intention dependency relationship. After the two communications, each agent establishes the intention dependency graph separately. In order to avoid intention misleading, 2SIS prohibits the agent that is dependent on other agents on the intention dependency graph from re-decision, and its final decision is exactly the same as its initial intention. Only the agent that is not dependent on any agent is allowed to make a new decision based on the intention information from others. 2SIS can be combined with any value-based reinforcement learning algorithm to perform training. Agents trained by 2SIS scheme can learn how to correctly establish intention dependencies to achieve one-side intention propagation, avoiding the problem of intention misleading. We select the representative Double DQN algorithm as the basic algorithm to verify the effectiveness of the proposed method in two multi-agent scenarios. The experimental results show that the agent trained by 2SIS scheme performs better in convergence speed and final cumulative reward than the agent trained without communication and the agent trained with unrestricted intention sharing. In order to demonstrate that the performance improvement comes from the method proposed in this paper, we organized an additional ablation experiment and conducted control variables for the key parts of the method. The results of the ablation experiment show that the selection of dependent targets for the agents trained under the 2SIS scheme makes key contributions to the performance improvement. Finally, we organize parameter experiments to illustrate how the hyperparameter introduced in this paper affect the training and how to choose an appropriate value for this hyperparameter for training.

Keywords　multi-agent system; deep reinforcement learning; deep multi-agent reinforcement learning; communication; intention sharing; cooperation

1　引　言

强化学习（Reinforcement Learning, RL）是机器学习领域的一个重要分支，强调智能体在不断与环境交互的过程中根据环境给予的反馈来优化自身策略，目标是最大化累计收益[1]. 强化学习在博弈论、控制理论、多智能体系统等多个领域都有发展和研究. 深度强化学习[2-3]是指结合了深度神经网络技术的强化学习方法，是人工智能领域的研究热点之一，目前深度强化学习技术已经在游戏[4-5]、控制系统[6-7]、数据库技术[8]、自动驾驶[9]、自然语言处理[10-11]、集群资源调度[12]等多个领域有所应用. 多智能体强化学习（Multi-agent Reinforce-

ment Learning, MARL)[13]是强化学习领域的一个重要研究方向，也是强化学习在多智能体系统领域下的发展和研究. 现实世界中有许多任务场景都可以使用多智能体系统来建模，比如无人机集群控制、交通信号灯控制等. 若采用中心化的单智能体强化学习方法来解决这类问题会面临可扩展性差、各种资源和条件的限制等问题. 多智能体强化学习方法则在这些场景上展现出了巨大的潜力[14-16]，成为开发具有群体智能的多智能体系统的重要方法. 多智能体强化学习逐渐成为人工智能领域的研究热点之一. 然而，场景中存在多个智能体导致了系统的复杂程度上升，多智能体强化学习相比于经典的单智能体强化学习要面临着更多困难和挑战[17]. 许多现实的多智能体场景需要智能体之间实现高度配合来完成任务目标，而如何让智能体学会协作配合则是多智能体强化学习领域的难点之一[18].

为了解决上述难题，有许多工作尝试利用通信来加强多智能体间的协作. 按照通信内容以及通信目标来分类，现有的工作可以分成两类. 第一类工作如 DIAL[19]、CommNet[20]、TarMar[21]、ATOC[22]等，在这些工作中，各个智能体的价值/策略网络经由通信信道连接起来，智能体在决策时可以互相传输信息. 此外，在训练过程中，梯度也可以经由通信信道在各个智能体之间流通，进而实现对群体策略的优化. 在这些早期对通信的研究工作中，智能体通过通信主要解决的问题是局部观测问题，智能体经由通信信道共享了一部分自身的观测[23].

第二类工作考虑让智能体互相交流自身的行为意图，进而实现更好的配合. 在这一方向上，IS (Intention Sharing)[23]首先给出了一种基于观测预测、动作预测的方法，预测未来一段时间内的观测以及动作意图，并对其编码后进行发送. IS 的通信结构是较为直接的广播式通信，即每个智能体都会向其它所有智能体广播自己未来一段时间的意图.

在不受限制的意图共享框架下，若某个智能体在接收到其它智能体的通信消息后改变了自身意图，则其原本传播的意图与实际将要执行的动作不一致. 这种不一致可能会误导群体中的其它智能体做出错误的动作，产生意图误导问题. 例如图 1 所示，两辆汽车在一个交叉路口相遇，且假设控制车辆的两个智能体的意图动作都是停车让行. 在智能体与彼此共享了自身的意图后，它们都认为对方会停车让行并且改变意图继续向前行驶，进而发生碰撞引发交通事故. 此时智能体共享的意图与实际执行的动作不一致，对智能体的决策产生了负面作用. 产生意图误导的根本原因是智能体在共享意图后又重新进行了决策，但利用意图进行决策又是意图共享的目的所在. 若让智能体按序决策并广播意图，即可在避免出现意图误导的条件下又能利用意图信息进行决策. 具体而言，各个智能体按照某个顺序依次决策，智能体在决策后广播自己的意图，供未决策的智能体进行决策，如此一来每个智能体在决策时接收到的意图都是准确的，从而避免出现意图误导问题. LFF (Leader Follower Forest)[24]首先给出了一种可以根据各个智能体的观测信息动态生成通信序关系的方法. 但是，LFF 中生成通信序关系需要一个中心化的控制器来完成，该控制器需要实时获得所有智能体的观测来生成通信序关系. 这种需要全局控制器的设定在多智能体强化学习中会面临难以扩展等问题. 此外，按序广播决策的模式对多智能体系统的同步控制提出了很高的要求.

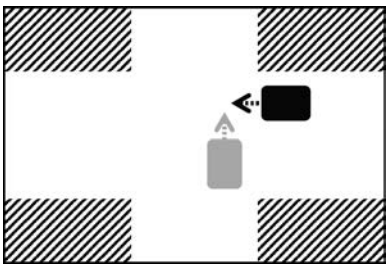

图 1 两辆汽车在交叉路口相遇的场景

注:灰色/黑色长方形代表汽车，车上的箭头表示汽车行驶方向，空白部分为车道.

针对上述问题，本文给出了一种基于两步广播的去中心化多智能体强化学习意图通信框架 2SIS (2 Step Intension Sharing). 在 2SIS 框架下，智能体仅需通过两次广播即可独立完成意图的共享和依赖关系的建立，并且不存在意图误导问题. 具体而言，在每个决策时刻，智能体首先根据自身局部观测生成动作意图并广播，然后根据接收到的其它智能体的意图以及自身的观测来决定要依赖哪些智能体的动作意图，并广播依赖信息. 在经过两轮广播后，智能体获取了其它智能体的意图信息以及依赖关系图. 为避免出现“误导”现象，若一个智能体被依赖了，那么该智能体将不能改变自己的意图，不被任何智能体依赖的智能体则可以根据意图信息重新决策. 此外，2SIS 框架可以与任意基于值函数的强化学习方法相结合，从而吸收它们的优势. 本文采用较具代表性的 Double DQN[25]算法作为基础算

法进行了实验. 实验结果表明,我们的方法在收敛速度、最终结果上均优于对比算法. 此外我们还设置了消融实验,与基于朴素规则建立依赖关系的算法版本进行了比较,消融实验的结果表明我们提出的方法确实让智能体学会了如何构建意图依赖关系.

总结上述内容,本文的主要贡献如下:

(1) 我们给出了一个基于两步广播的去中心化多智能体强化学习意图通信框架,该通信框架不需要任何全局的控制器,智能体在决策时只需要参与两次广播即可建立起多智能体意图通信的顺序结构.

(2) 我们给出了一种基于学习的意图重要性度量方法,智能体可以通过训练学会正确度量其它智能体意图的重要性并据此构建意图依赖关系.

(3) 本文提出的意图通信框架是通用的通信框架,可以结合任意基于值函数的强化学习算法实现训练,我们给出了基于 Double DQN 算法的实现.

(4) 我们在两个经典的多智能体协作场景下验证了本文所提出方法的有效性,与基准算法相比,本文提出的方法在收敛速度以及最终累积奖赏上均有明显优势.

本文剩余内容组织如下:第 2 节介绍了通信方面的多智能体强化学习相关工作;第 3 节简要介绍了强化学习相关的背景知识以及我们采用的基础算法;第 4 节介绍了我们提出的多智能体强化学习意图通信框架;第 5 节进行实验验证;第 6 节总结全文并讨论未来研究的方向.

2　相关工作

许多现实世界中的问题都可以采用一个协作多智能体系统进行建模. 直接使用单智能体的强化学习算法来解决多智能体场景下的问题,面临着若干挑战. 首先,在执行阶段,智能体通常只能获得局部观测,这意味着智能体的策略必须是分布式的. 但是强化学习的训练阶段往往都是在模拟器中进行,这意味着我们在训练阶段能轻松地获取全局的状态信息,因此便诞生了集中式训练-分布式执行这种多智能体强化学习训练框架,但是如何最大化利用训练阶段的全局状态信息仍是一个开放问题. 其次,在多智能体强化学习领域,奖赏分配也是一个棘手的问题. 在多智能体合作场景中,所有智能体在每一步获得的奖赏都是一致的,因而难以衡量每个智能体各自的贡献大小. 为了解决上述挑战,Foerster 等人[26]给出了一种集中式训练-分布式执行多智能体强化学习算法 COMA(Counterfactual Multi-Agent). COMA 基于 actor-critic 框架,其中 critic 网络是集中式的,可以获取环境的全局状态信息;actor 网络是分布式的,每个智能体都有一个 actor 网络. 为了衡量单个智能体在某一步动作执行中的贡献,COMA 提出了反事实基线这一概念. 反事实基线的计算类似于单智能体强化学习中的值函数,不同之处在于,反事实基线将所有其余智能体的动作当成观测的一部分,在此基础上计算 Q 值的期望作为基线. 智能体在该步的贡献便可以用当前 Q 值减去反事实基线来衡量. QMIX[27]是另一种基于值函数学习的集中式训练-分布式执行算法. QMIX 认为,在多智能体场景下,所有智能体的联合价值函数是难以学习的,即使能学习也难以衡量各个智能体各自的贡献. 针对该问题,QMIX 中的集中式 Q 函数不是直接采用一个网络来拟合所有智能体的联合价值函数,而是由各个智能体各自的 Q 函数经由一个混合网络组合而成,只要组合之后的 Q_{tot} 函数与每个智能体各自的 Q 函数的单调性保持一致,就可以保证组合前后的最优策略是一致的. 不同于 QMIX 算法需要引入 Q_{tot} 函数与每个智能体的 Q 函数之间的约束,Q 值路径分解方法[28](Q-value Path Decomposition,QPD)使用积分梯度来衡量各个智能体对全局 Q_{tot} 函数的贡献. QPD 利用一轮中状态-动作转移轨迹上的积分梯度将全局的 Q_{tot} 函数分解为局部 Q 函数,再对各个智能体的 Q 网络进行优化.

在多智能体强化学习领域,最早的与通信有关的工作在局部观测设置下的表格环境中展开[29]. 在深度多智能体强化学习出现后,通信相关的研究呈现出一个增长的趋势. Foerster 等人[19]首先提出了一种端到端的通信强化学习方法 DIAL. DIAL 中每个智能体维护一个名为 C-Net 的网络,用于生成通信消息,同时估计智能体各个动作的 Q 值. C-Net 输出的信息将直接和其它智能体的 C-Net 的输入相连接. 因此在训练阶段,梯度可以经由通信信道从消息的接收方流向消息的发送方,这种反馈能力最终使得智能体学会了如何与其它智能体分享自己的观测. 与 DIAL 中的离散通信信道不同,CommNet[20]使用连续的矢量信道将所有智能体连接起来,每个智能体接收到的消息是所有其它智能体发送的消息之和,并且智能体可以在一轮

决策中多次通信. CommNet 采用集中式的方法训练场景中所有的智能体，从接收观测作为输入、进行多轮通信到最终的决策环节全部由一个集中式的网络来完成.

DIAL 和 CommNet 中的通信都是广播式的通信，通信实现了所有智能体之间信息的共享. 当智能体数量很大的时候，在所有智能体之间共享信息将导致智能体很难找出有价值的信息. 针对该问题，Jiang 等人[22]提出了一种基于注意力机制的通信模型 ATOC. ATOC 模型中包含一个注意力单元，借助于注意力单元，智能体可以依次根据编码后的观测判断是否要发起通信，并基于一个确定性的规则选取离自己最近的对象组成通信小组，小组之间的成员可以共享彼此的观测编码，并通过一个 LSTM 模型来对信息进行集成. 不同于 ATOC 中基于确定性的规则选取通信目标，TarMAC[21]使用软注意力机制，智能体可以学会向谁发送信息以及发送什么样的信息. 在 TarMAC 框架下，每轮每个智能体广播的信息可以拆分为两个部分，签名（signature）中指出了该信息的接收对象，值（value）中则是具体的信息内容. 消息的接收方则会根据信息的签名部分来决定值部分的权重. Mao 等人[30]从认知一致性的角度入手，将邻域认知一致性（Neighborhood Cognitive Consistency，NCC）引入多智能体强化学习，提出了离散动作空间下的 NCC-Q 算法和连续动作空间下的 NCC-AC 算法. NCC-Q/NCC-AC 将多智能体环境建模为图，图中的结点为环境中参与交互的智能体，图中的边代表智能体之间的通信信道. 通过该信道，智能体可以与邻居结点互相分享自身编码后的局部观测. 为了实现智能体间的邻域认知一致性，NCC-Q/NCC-AC 使用图卷积网络从邻居结点的联合观测中提取出高层次认知向量，然后将该认知向量分解为针对智能体自身的认知以及针对邻居的认知两个分支分别进行计算，最后将两个分支计算的结果逐元素相加，得到的和作为价值函数的输入.

上述方法关注的都是多智能体强化学习中的部分可观测问题，智能体通过通信信道和其它智能体共享了自身的局部观测（或者是局部观测的编码）. 在最近的研究中，有部分工作把关注点放在了如何让智能体通过交流意图来加强合作. Kim 等人[23]提出了一种意图共享（IS）的方案，IS 允许智能体在共享局部观测的同时还能共享它们的未来一段时间内的意图. 在每一步，每个智能体根据自己的局部观测以及在上一步接收到的信息，预测自身未来一段时间内的动作和观测序列，然后通过注意力模块对该序列进行编码，从中提取出重要的部分，编码后的消息将被广播给其它所有智能体. 但智能体在共享意图后有可能会改变自身的动作，造成实际动作与意图不符的情况，此时意图信息有可能会误导其它智能体. 针对该问题，Liu 等人[24]提出了 LFF 方法，可以动态地生成智能体间的层次关系. 在决策时，处于高层次的智能体先进行决策并将决策的动作信息广播给所有低层次的智能体，处于低层次的智能体则可以获取来自所有高层次智能体的意向动作信息. 智能体严格按照层次关系逐层进行决策，进而实现了单向的意图共享，避免出现误导现象. LFF 将智能体间层次关系的构建问题也建模成了强化学习问题，使用一个集中式的 DDPG[31]智能体来估计智能体之间的依赖程度，在训练时需要依赖环境中的全局状态.

3 背景知识

本节对强化学习以及多智能体强化学习的一些背景知识进行了介绍.

3.1 马尔可夫决策过程

强化学习问题通常建模为马尔可夫决策过程（Markov Decision Process，MDP）[32-33]. 一个 MDP 可以用一个五元组$\langle \mathcal{S},\mathcal{A},P,R,\gamma\rangle$来形式化地定义. S 为状态空间，是环境状态的集合. $\mathcal{A}$为动作空间，是智能体所能采取的动作的集合. $P:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\mapsto[0,1]$为状态转移函数，$P(s'|s,a)$表示在环境状态为 s 时，智能体采取动作 a 后环境转移到状态 s'的概率. $R:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\mapsto\mathbb{R}$为奖赏函数，$R(s,a,s')$表示在环境状态为 s、智能体采取动作为 a 且环境转移到状态 s'后智能体所能获得的奖赏. $\gamma\in(0,1]$为折扣因子，代表着在累计奖赏中后续奖赏的权重，γ 越大意味着智能体的后续奖赏在累积奖赏中的权重越大. 在马尔可夫决策过程的基础上定义智能体的策略 $\pi:\mathcal{S}\mapsto\mathcal{A}$，即 π 是从状态空间到动作空间的一个映射. 强化学习的目标是通过不断地更新 π 来最大化累积奖赏，使得智能体从初始状态 s_0 开始按照策略 π 来执行动作能够获得尽可能高的累计奖赏，这个过程可以用下面的式子来表示：

$$\underset{\pi}{\text{maximize}} \quad \mathbb{E}\left[\sum_{t=0}^{T}\gamma^t R(s_t,\pi(s_t),s_{t+1})\right] \tag{1}$$

但在许多场景下智能体无法直接观察到环境的

状态，而是只能获得一个不完整的对环境状态的部分观测 o，此时称为部分可观察的马尔可夫决策过程（Partially Observable Markov Decision Process，POMDP）[34]. POMDP 需要一个七元组$\langle \mathcal{S},\mathcal{A},P,R,\Omega,O,\gamma\rangle$来定义. 除了与 MDP 相同的五元组外，观测空间 Ω 表示智能体所能获得的观测集合，$O:\mathcal{S}\times\mathcal{A}\times\Omega\mapsto[0,1]$是观测概率函数，$O(o|s',a)$表示智能体采取动作为 a 且环境转移到状态 s'后智能体获得的观测为 o 的概率. 相应的智能体的策略 π 变为从观测空间 Ω 到动作空间$\mathcal{A}$的映射，式 (1) 改写为

$$\underset{\pi}{\text{maximize}}\quad \mathbb{E}\left[\sum_{t=0}^{T}\gamma^t R(o_t,\pi(o_t),o_{t+1})\right] \tag{2}$$

3.2　多智能体强化学习建模

一个完全合作的多智能体任务可以用一个随机博弈（Stochastic Game）来表示. 一个随机博弈 G 由以下八元组定义：$\langle S,A,P,R,\Omega,O,N,\gamma\rangle$. 其中 S,A,Ω,O,γ 的含义与 POMDP 中一致. 不同之处在于，状态转移函数 $P:S\times\boldsymbol{A}\times S\to[0,1]$，其中的 $\boldsymbol{A}$ 表示所有智能体的联合状态空间. 在随机博弈中，状态的转移依赖于所有智能体的联合动作. 类似的，奖赏函数 $R:S\times\boldsymbol{A}\to\mathbb{R}$，奖赏函数的输出也依赖于所有智能体的联合动作，并且所有智能体都会获得相同的奖赏. N 为智能体的数目. 随机博弈的目标和 MDP/POMDP 一致，均为最大化累计奖赏的期望.

在多智能体场景下，另一种常见的建模方式是：每个智能体将其它智能体看作环境的一部分，那么每个智能体与环境交互的过程都能使用 POMDP 来进行建模. 因而智能体可以采用单智能体强化学习的方法进行训练，这种训练模式称为独立学习[35]. 相较集中式训练的模式，独立学习具有良好的可扩展性，这也是本文所采用的训练模式.

3.3　Q-Learning、DQN 和 Double DQN

动作价值函数是强化学习中的核心概念，定义如下：

$$Q_\pi(o_t,a)=\mathbb{E}\left[R(o_t,a,o_{t+1})+\sum_{i=t+1}^{T}\gamma^{i-t}R(o_i,\pi(o_i),o_{i+1})\right] \tag{3}$$

$Q_\pi(o_t,a)$的含义是，智能体在观测 o_t 下执行动作 a，后续按照策略 π 采取动作所能获得的累积奖赏的期望. 若能够获得对动作价值函数的准确估计，我们便可以度量在某个给定观测下执行不同动作的优劣，因此强化学习算法中有一类算法专门研究如何获得对动作价值函数的准确估计. Q-Learning 算法[36]是最基础的基于值函数的强化学习算法之一. 智能体与环境的一次交互可以用四元组$\langle o,a,o',r\rangle$来描述，其含义是：智能体在观测 o 下执行了动作 a，获得了对环境的下一个观测 o'和奖赏 r. Q-Learning 使用表格来存储每个 o,a 对应的动作价值函数，在每次交互之后，采取如下公式更新表格中的值：

$$Q(o,a)\leftarrow(1-\alpha)Q(o,a)+\alpha[r+\gamma\max_{a'}Q(o',a')] \tag{4}$$

上式中的 α 为更新步长. 若问题可以使用 MDP 建模，在经过足够多的交互后，Q 表格将会收敛成为最优动作价值函数[37].

采用表格形式模拟动作价值函数存在着训练效率低、无法应用于连续状态空间等问题，针对这些问题，DQN 算法[38-39]采用深度神经网络来拟合动作价值函数，称为 Q 网络，记为 $Q(o,a)$. 为了提高数据利用率以及消除相邻两次交互之间的相关性，DQN 算法使用经验回放池来存储智能体与环境的交互信息$\langle o,a,o',r\rangle$，在每次更新时从经验池中随机抽取一批样本进行更新. 为了提高训练过程的稳定性，DQN 算法额外维护了一个 Q 网络的副本用于构造更新目标，称为目标 Q 网络，记为 $\hat{Q}(o,a)$，Q 网络的更新目标为

$$y=r+\gamma\max_{a'}Q(o',a') \tag{5}$$

采用均方误差函数作为损失函数，则损失为

$$L(\theta)=[Q(o,a)-y]^2 \tag{6}$$

θ 为 Q 网络中的参数权值. 每隔若干次交互将目标 Q 网络的参数赋值为当前 Q 网络的参数，记目标 Q 网络的参数为 θ^-，即令 $\theta^-=\theta$.

DQN 算法中的最大化操作容易造成对 Q 值的过度估计[40]，针对过度估计问题，Hasselt 等人[25]在 2016 年提出了 Double DQN 算法，将动作的选择和评估解耦，采用当前 Q 网络选择最优动作，评估最优动作对应的 Q 值则由目标 Q 网络来完成，Double DQN 算法中 Q 网络的更新目标为

$$y=r+\gamma\hat{Q}(o',\arg\max_{a'}Q(o',a')) \tag{7}$$

其余过程与 DQN 算法相同. Double DQN 算法在很大程度上缓解了 DQN 算法的过度估计问题，是目前基于值函数的强化学习算法中较具代表性的算法之一，因此本文采用 Double DQN 算法作为基础算法.

4　基于两步广播的多智能体强化学习意图共享框架

本文给出了一个多智能体强化学习意图通信框

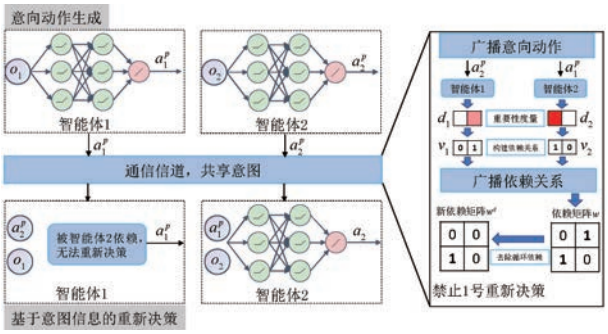

图 2 智能体数目为 2 时，在 2SIS 通信框架下智能体决策流程示意图

架，该框架可以与任意基于值函数的强化学习算法相结合进行训练，并且在执行阶段不依赖于任何全局控制器，每个智能体可以独立地完成各自的通信步骤. 该方法背后的思想是，在多智能体场景下，当某个智能体需要做出决策时，若能够知晓其它智能体的动作意图，将有助于智能体做出更好的决策来实现团队的协作配合[41-42]. 图 2 展示了以 2 个智能体为例的智能体在 2SIS 通信框架下的决策流程. 具体而言，在每个决策时刻 t，每个智能体首先根据自身的局部观测 o_i 生成意向动作 a_i^p，然后进行第一轮广播，将该意向动作发送给其它所有智能体. 记智能体总数为 N，在第一轮广播结束后，智能体 i 根据接收到的意向动作信息 $(a_1^p,\cdots,a_{i-1}^p,a_{i+1}^p,\cdots,a_N^p)$ 以及自身的局部观测 o_i 分别计算其它每个智能体的意向动作信息对于本轮自身决策的重要程度 d_i，若重要程度超过我们预定义的阈值 α，则将对应的智能体纳入依赖对象集合 v_i 中. 然后进行第二轮广播，每个智能体都将自己的依赖对象集合 v_i 发送给所有其它智能体. 在第二轮广播结束后，每个智能体根据接收到的智能体依赖信息构建一个依赖关系图. 依赖关系图是一个有向图，最初构建的依赖关系图可能存在有向回路（即存在循环依赖），因此我们设计了一个确定性算法以去掉依赖图中的循环依赖，确保最终得到的依赖关系图是一个有向无环图. 在得到无环的依赖关系图后，若智能体被其它智能体所依赖，那么该智能体将不能重新决策，其最终动作即为其意向动作；若智能体不被任何智能体所依赖，那么该智能体可以根据所依赖的智能体的意向动作信息重新进行决策. 本节的剩余部分详细描述了上述流程中的各个步骤.

4.1 意向动作生成

在每轮决策开始时，智能体首先需要根据自身的局部观测 o_i 生成意向动作 a_i^p. 每个智能体各自维护了一个值网络 Q_i^p，用于估计在当前观测 o_i 下执行各个动作最终能收获的累计奖赏的期望. 智能体 i 的意向动作由如下公式给出

$$a_i^p=\underset{a}{\operatorname{argmax}}\,Q_i^p(o_i,a) \tag{8}$$

随后在第一轮通信时每个智能体都将各自的意向动作广播给其它所有智能体.

4.2 构建依赖关系

在第一轮广播结束后，记智能体 i 接收到的信息 m_i 为

$$m_i=(a_1^p,\cdots,a_{i-1}^p,a_{i+1}^p,\cdots,a_N^p) \tag{9}$$

智能体需要根据观测 o_i 以及意向动作信息 m_i 分别计算其它智能体的动作在本轮决策中对于自己的重要程度，再依据重要程度来决定是否要依赖对应的智能体的意向动作.

4.2.1 计算其它智能体意图的重要程度

受到差分奖赏[43-44]方法的启发，我们通过计算其它智能体在执行不同动作下对自身累计奖赏的影响程度来衡量其它智能体的重要程度. 具体而言，每个智能体各自维护一个值网络 Q_i^e，其输入由三部分组成，第一部分是自身观测 o_i，第二部分为所有其它智能体的动作 $a_i^-=(a_1,\cdots,a_{i-1},a_{i+1},\cdots,a_N)$，第三部分为自身动作 a_i. $Q_i^e(o_i,a_i^-,a_i)$ 估计了在状态 o_i 下，其余智能体的动作为 a_i^-、自身动作为 a_i 时，智能体 i 后续的累计奖赏的期望值. 在决策时，智能体可以用接收到的意向动作信息 m_i 代替 a_i^- 来估计累计奖赏的期望. 若智能体 i 需要计算智能体 j 的动作对自身重要的程度，首先计算当智能体 j 执行不同动作时，在 Q_i^e 估计下智能体 i 的最优动作. 记 $m_{i,j,k}$ 表示将 m_i 中智能体 j 的动作 a_j^p 改为动作 k 后得到的结果，即

$$m_{i,j,k}=(a_1^p,\cdots,a_{j-1}^p,k,a_{j+1}^p,\cdots,a_N^p) \tag{10}$$

其它部分保持与 m_i 相同. 那么智能体 i 在 Q_i^e 估计下的最优动作 $u_{i,j,k}$ 为

$$u_{i,j,k}=\underset{a}{\operatorname{argmax}}Q_i^e(o_i,m_{i,j,k},a) \tag{11}$$

$u_{i,j,k}$ 表示在观测 o_i 下，智能体 i、j 以外的智能体执行的动作集合为 $m_i\backslash a_j^p$，智能体 j 执行的动作为 k 时，智能体 i 的最优动作. 记智能体 i 关于智能体 j 的最优动作集合为 $u_{i,j}$，那么

$$u_{i,j}=\{u_{i,j,k}\mid\forall k\in\mathcal{A}\} \tag{12}$$

$u_{i,j}$ 表示当智能体 j 执行不同动作时智能体 i 对应的最优动作. 我们使用记号 $d_{i,j}$ 来表示智能体 j 的动作信息相对于智能体 i 的重要程度. 首先，我们约定智能体不依赖于自身的动作意图，即对于任意的

i，都有 $d_{i,i}=0$. 若 $u_{i,j}$ 仅包含一个元素，即

$$u_{i,j,1}=u_{i,j,2}=\cdots=u_{i,j,|\mathcal{A}|} \tag{13}$$

此时意味着智能体 j 执行任何动作都不会改变智能体 i 的最优动作. 在这种情况下，我们认为智能体 j 的动作信息对智能体 i 是完全不重要的，即此时 $d_{i,j}=0$.

若 $u_{i,j}$ 中包含超过一个元素，那么说明在智能体 j 执行不同动作时智能体 i 的最优动作会发生改变，此时给智能体 i 提供智能体 j 的动作信息将有助于智能体 i 做出更好的决策. 但如果不同最优动作对应的 Q 值相差并不大，意味着智能体 j 的动作对智能体 i 的后续累计奖赏影响较小，对应的 $d_{i,j}$ 也应该比较小. 只有当 $d_{i,j}$ 大于一定值时，才有必要建立依赖关系. 受到 VBC[45] 的启发，我们计算不同 Q 值的标准差来衡量不同最优动作的 Q 值之间的差距. 具体而言，对于一个给定的动作 k，我们可以计算出当智能体 i 采取不同最优动作时，Q_i^e 的最大值 $\eta_{i,j,k}$ 和最小值 $\mu_{i,j,k}$：

$$\eta_{i,j,k}=\max_{u\in u_{i,j}} Q_i^e(o_i,m_{i,j,k},u) \tag{14}$$

$$\mu_{i,j,k}=\min_{u\in u_{i,j}} Q_i^e(o_i,m_{i,j,k},u) \tag{15}$$

由于 Q 值大小会受到具体环境奖赏设置的影响，为了建立一个统一的量化标准，需要对上述最大、最小值进行归一化，我们采用 softmax 函数进行归一化：

$$\phi_{i,j,k}=\mathrm{softmax}[\mu_{i,j,k},\eta_{i,j,k}] \tag{16}$$

选用 softmax 函数的理由是：在训练初期，各智能体尚处于探索阶段，此时 Q 值都比较小并且变化较为剧烈，按照这时的 Q 值大小来区分各个智能体的重要程度是不准确的. 在训练的末期，Q 值较大并且趋于稳定，用于区分各个智能体的重要程度相对更准确. 而 softmax 函数的特性可以保证，在原始数据标准差保持不变的情况下，softmax 结果的标准差会随着原始数据平均值的增大而增大. 因此我们就能得到，在执行动作 k 时，智能体 j 相对于智能体 i 的重要程度 $\delta_{i,j,k}$

$$\delta_{i,j,k}=\mathrm{std}(\phi_{i,j,k}) \tag{17}$$

取最大的 $\delta_{i,j,k}$ 作为最终的 $d_{i,j}$

$$d_{i,j}=\max_k \delta_{i,j,k} \tag{18}$$

4.2.2　根据重要程度决定是否建立依赖关系

我们引入了一个新的超参数 α 作为智能体建立依赖关系的重要程度阈值，智能体 i 的依赖对象集合 v_i 可以表示为

$$v_i=\{j \mid \forall j\leqslant N \wedge d_{i,j}>\alpha\} \tag{19}$$

其中 j 代表智能体 i 依赖的智能体的编号. 在第二轮通信中智能体会将该依赖对象集合广播给所有其它智能体.

4.3　去除循环依赖

在第二轮通信结束后，每个智能体都可以根据接收到的信息构建一个依赖关系图，可以用一个 $N\times N$ 的邻接矩阵 w 来表示，$w_{i,j}=1$ 表示智能体 i 依赖于智能体 j. w 中可能存在有向回路，有向回路意味着智能体之间出现了循环依赖，那么处在有向回路上的智能体都无法重新决策，也就失去了通信的意义，因此我们设计了一个贪心算法去除 w 中的有向回路. 具体步骤见算法 1.

算法 1.　循环依赖去除算法.

输入：有向图 w，w 中的每个结点都代表一个智能体，并且每个结点都被分配了一个唯一的整数编号

输出：有向无环图 w^d

1. 初始化 w^d 为 w
2. WHILE w 非空时
3.　选出 w 中出度为 0 的结点集合 v
4.　IF v 为空 THEN
5.　　找出 w 中出度最小且编号最小的结点 $v_{\min}$
6.　　将结点 $v_{\min}$ 的出边从 w_d 中删除
7.　　将 $v_{\min}$ 加入 v
8.　END IF
9.　将 v 中所有结点从 w 中删除
10. END WHILE

从有向图中删去最少的边使得原图不存在有向回路，被称为反馈边集（Feedback Arc Set，FAS）问题，FAS 问题是 NP 完全问题[46]，GR 算法[47]是 FAS 问题的一个多项式时间下的近似算法. 相比本文所给出的算法 1，采用 GR 算法来去除循环依赖的训练效果并不好. 可能的原因是，一味地追求删除最少的边可能会导致仍有大量智能体被依赖无法重新决策. 我们把如何找到最优循环依赖去除算法作为下个阶段的研究方向之一.

4.4　根据意向动作信息重新决策

为了避免出现意图误导现象，我们采用了较为严格的约束：对于任意的智能体 i，若存在智能体 j 依赖于智能体 i，即 $w_{j,i}^d=1$，那么该智能体 i 将不能重新决策，其最终动作为 a_i^p；若智能体 i 不被任何其它智能体所依赖，即 $\sum_{j\leqslant N} w_{j,i}^d=0$，智能体 i 将根据 m_i 和 w^d 重新决策：

$$a_i=\arg\max_a Q_i^\pi(o_i,m_i^d,a) \tag{20}$$

上式中的 m_i^d 为智能体 i 依赖的智能体的动作的集合，即

$$m_i^d=\{a_j^p \mid \forall j\leqslant N \wedge w_{i,j}^d=1\} \tag{21}$$

重新决策后可能会导致依赖关系发生变化，使得当前决策的依据具有一定的误导性. 但是我们通过实验发现该现象发生的频率较低，因此在目前的工作中暂未进行考虑，具体情况见附录 A.

4.5 基于 Double DQN 算法实现的 Double DQN-2SIS 算法

本文使用 Double DQN 算法进行对上述 Q_i^p、Q_i^e、Q_i^π 进行更新，也可以采用其它值函数更新算法来更新，本文提出的方法并不指定更新 Q 函数使用的算法. 基于 Double DQN 算法实现的 Double DQN-2SIS 算法伪代码见算法 2.

算法 2. Double DQN-2SIS 算法.

输入：环境 E，探索因子 ε，学习率 lr，折扣因子 γ，网络更新间隔 τ，重要程度阈值 α，总的训练步数 T

输出：所有智能体三个 Q 网络参数的权值，即 $\{Q_1^p, Q_2^p, \cdots, Q_N^p\}$、$\{Q_1^e, Q_2^e, \cdots, Q_N^e\}$ 和 $\{Q_1^\pi, Q_2^\pi, \cdots, Q_N^\pi\}$ 的参数权值

1. FOR 智能体编号 i, $1 \leqslant i \leqslant N$
2. 随机初始化 Q_i^p、Q_i^e 和 Q_i^π 以及对应的目标网络 $\hat{Q}_i^p$、$\hat{Q}_i^e$ 和 $\hat{Q}_i^\pi$
3. 初始化经验池 D_i^p、D_i^e、D_i^π 为空
4. END FOR
5. FOR 训练步数 t, $1 \leqslant t \leqslant T$
6. FOR 智能体编号 i, $1 \leqslant i \leqslant N$
7. 根据 Q_i^p 和当前观测 o_i，ε 贪心地选择意向动作 a_i^p 并广播
8. 接收到来自其他智能体的意向动作信息 m_i
9. 基于 Q_i^e 和 m_i 计算需要依赖的智能体对象集合 u_i 并广播
10. 根据接收到的依赖关系信息构建依赖关系图
11. 使用 4.3 节的方法去掉循环依赖得到依赖关系矩阵 w_i
12. 按照公式 (21) 构造 m_i^d
13. IF $\sum_{j \leqslant N} w_{j,i}^d \neq 0$ THEN
14. 最终动作 $a_i^\pi = a_i^p$
15. ELSE THEN
16. 根据 Q_i^π、o_i 和 m_i^d，ε 贪心地选择最终动作 a_i^π
17. END IF
18. 在环境 E 中执行动作 a_i^π，获得观测 o'_i 和奖赏 r_i
19. 重复 7-12 行的过程获取下一步的 m'_i 和 $m_i^{d'}$
20. 将 o_i, a_i^π, r_i, o'_i 存入 D_i^p
21. 将 $(o_i, m_i), a_i^\pi, r_i, (o'_i, m'_i)$ 存入 D_i^e
22. 将 $(o_i, m_i^d), a_i^\pi, r_i, (o'_i, m_i^{d'})$ 存入 D_i^π
23. 根据式 (6) 分别计算损失函数更新 Q_i^p、Q_i^e 和 Q_i^π 网络
24. 每隔 τ 步更新 $\hat{Q}_i^p$、$\hat{Q}_i^e$ 和 $\hat{Q}_i^\pi$ 的参数
25. END FOR
26. END FOR

4.6 Double DQN-2SIS 算法的计算和存储开销

在执行时刻，深度强化学习算法主要的计算和存储开销都来自于神经网络，因此我们关注决策过程中神经网络前向计算的次数及其所占的存储空间大小，结论如下：

1. 神经网络前向计算次数：2SIS:DDQN = $N \times |\mathcal{A}| + 2$:1；
2. 空间资源：2SIS:DDQN = 3:1.

在原始的 2SIS 方法下神经网络的前向计算次数会随着智能体数目的增加线性增长，由于维护了 3 个 Q 网络，因此 2SIS 的空间资源变为原来的 3 倍. 在本次实验的运行平台（AMD Ryzen 9 5900HS、Nvidia RTX 3070）下，对于本文中使用的包含 $67 \times 128 + 128 \times 32 + 32 \times 4$ 个参数的神经网络规模，完成一次前向计算的实际运行时间在 0.25 ms 左右，在本文的实验场景下 2SIS 方法的实际运行开销仍在毫秒级别. 在智能体数目非常多时，可以考虑限制智能体仅能与周围特定个数的智能体进行通信，从而限制 2SIS 方法的计算复杂度上限.

5 实验验证

为了验证本文方法的有效性，我们在两个多智能体场景下进行了实验.

5.1 实验组织

本文选取的两个多智能体任务场景分别为多智能体协作目标运输任务[48-49]和交叉路口通行[20,23]. 我们在这两个场景上分别使用以下三种算法进行了训练：

(1)DDQN：智能体间无通信的、基于独立学习的 Double DQN 算法；

(2)DDQN-BC：基于广播意图和独立学习的 Double DQN 算法，在决策时刻，所有智能体都会广播意图，然后根据接收到的意图重新决策；

(3)DDQN-2SIS：基于 2SIS 通信框架和独立学习的 Double DQN 算法.

在更多场景下的实验详见附录 B. 对于多智能体协作目标运输场景，我们还训练了同样使用该场景的 Lenient-DQN[49]算法进行对比；类似的，对于交叉路口通行场景，我们还训练了同样使用该场景的 Commnet[20]算法进行对比.

为了验证我们提出的方法能有效地选择值得依

赖的对象，我们进行了消融实验，与另外两种朴素的选择依赖对象的方法进行了比较.

2SIS 引入了一个额外的超参数 α，代表重要程度阈值，只有超过该阈值的智能体会被选为依赖对象. 为了探究超参数 α 对实验结果的影响程度，以及如何选取该参数，我们为超参数 α 设计了参数实验进行说明.

为了保证公平性，我们为所有 DDQN 系列的训练使用了相同的超参数：学习率 $lr=0.0001$，折扣因子 $\gamma=0.95$，探索率 $\varepsilon=0.1$，目标 Q 网络更新间隔 $\tau=50$. 对于 DDQN-2SIS 独有的参数 α，在多智能体协作目标运输场景下我们设置 $\alpha=0$，在交叉路口通行场景下我们设置 $\alpha=0.125$. 对于 Lenient-DQN 算法和 Commnet 算法，我们也在神经网络规模、学习率等参数的设置上尽可能地保证公平性. 此外，为了保证结果的准确性，所有实验均独立地重复了 5 次，实验结果图中的阴影部分为 95%置信区间.

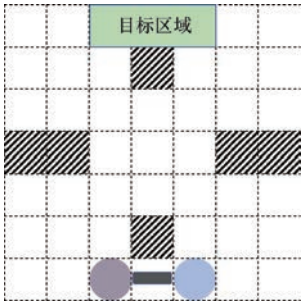

图 3　多智能体协作目标运输场景示意图

注:其中的两个圆圈分别表示两个智能体，两个智能体中间的长条为目标物体，阴影区域为障碍物，最上方占据 3 个网格的绿色区域为目标区域.

5.2　多智能体实验场景

在本小节我们对使用到的两个多智能体协作实验场景进行了详细的描述.

5.2.1　多智能体协作目标运输

在多智能体协作目标运输任务[48-49]下，智能体需要合作将目标物品运送到指定区域. 如图 3 所示，在一个 7×7 格大小的离散网格环境内，存在 2 个智能体、若干障碍物以及一个目标物体，智能体分布在目标物体的周围. 智能体和目标物体均占据一个网格，一个网格同一时刻只能存在一个实体(智能体、目标物体或障碍物). 一个智能体可选择的动作包括向上、向下、向左、向右移动. 在我们的设定下，目标物体非常重，需要所有智能体一起才能搬动，即所有智能体都选择向相同方向移动、且不会和障碍物冲突时目标物体才会随着智能体一起移动. 智能体的任务是将目标物体运输到地图上指定的目标区域. 当智能体的动作成功使得目标物体向上移动一格或者到达目标区域时，所有智能体会获得正的奖赏；反之若智能体的动作使得目标物体向下移动了一格或者未能移动目标物体时，所有智能体会获得一个负的奖赏. 具体奖赏设计如表 1 所示. 所有智能体的观测都是一致的，是一个包含 7×7 个像素的图片，其中像素值为 1 代表智能体，像素值为 2 代表障碍物，像素值为 3 代表目标区域的中心. 每轮任务的行动次数上限为 30.

表 1　多智能体协作目标运输任务奖赏设计

行为	奖赏
将目标向上移动一格	+1
将目标向下移动一格	−1
未能移动目标	−0.5
将目标送达目标区域	+5

5.2.2　交叉路口通行

在交叉路口通行场景[20,23]下，各个智能体需要在尽量避免碰撞的基础上尽快通过交叉路口. 如图 4 所示，在一个 6×6 格大小的离散网格环境内有一个双车道（单向单车道）的十字交叉路口，每辆车由一个单独的智能体来控制. 场景中总共存在 N 辆车，每辆车占据环境中的一个网格，每辆车在进入该路段时都会被预分配一条路线（直行、左转或者右转）. 控制车辆的智能体在每一步只能采取以下两个动作之一：沿预分配路线前进一格，或者停车等待. 若有多辆车同时出现在了一个网格上，那么认为这些车辆发生了一次碰撞. 为了加速训练，发生碰撞后相关车辆不会被移出环境或者挡住道路，可以继续沿原路线行驶，但是所有控制相关车辆的智能体都会获得一个负的奖赏. 当车辆沿预分配路线驶离该路段时智能体会获得一个正的奖赏，同时该智能体会随机从一条车道再次进入该路段. 为了鼓励车辆尽可能快地驶离路段，以提高该交叉路口的吞吐量，每过一步场景中所有的智能体都会获得一个较小的负的奖赏. 具体奖赏设计如表 2 所示. 智能体的观测包括自身位置、其它车辆位置、所有车辆的朝向、自身预分配路线，我们将上述信息编码为独热向量作为每个智能体的观测. 每轮智能体可以采取的行动次数上限为 40.

表 2　交叉路口通行场景奖赏设计

行为	奖赏
驶出路口	+10
每与一辆车发生碰撞	−5
每过一步	−1

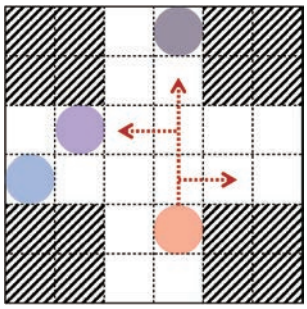

图 4 交叉路口通行场景

注:不同颜色的圆圈表示不同智能体,红色箭头的虚线表示智能体可选的路线(直行、右转或左转).

5.3 实验评价指标

采取的实验评价指标为强化学习中常用的平均累积奖赏指标:

$$R=\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}r_i^t \tag{22}$$

针对两个多智能体任务场景的特征,我们还为两个场景分别设计了额外的评价指标:

(1)多智能体目标运输场景

与目标区域的最小垂直距离:记录每一轮中智能体离目标区域的最短垂直距离,取最近 10 轮训练的结果的均值绘制曲线. 该指标刻画了智能体学会将目标运送到目标区域的程度;

(2)交通路口通行场景

平均发生碰撞次数:记录每一轮中智能体发生碰撞的次数,取最近 100 轮的结果的均值绘制曲线. 该指标衡量了智能体在通行时避撞的能力.

5.4 实验结果及分析

本节展示了在两个场景下不同算法训练的结果,并对实验结果进行了分析.

5.4.1 多智能体协作目标运输场景

多智能体协作目标运输场景需要两个智能体高度配合,在该场景下,我们认为在每一步队友的动作都至关重要,因此我们将该场景下的 DDQN-2SIS 算法的重要程度阈值 α 设为 0,代表着每一步两个智能体都会互相依赖. 去掉循环依赖后,实际上是在每一步 1 号智能体都会向 2 号智能体发送自己的意向动作信息,然后 2 号智能体根据自身局部观测以及 1 号智能体的意向动作信息重新进行决策. 我们分别使用原始的 DDQN 算法、基于广播式通信意图的 DDQN-BC 算法、LDQN (Lenient-DQN)算法以及我们提出的 DDQN-2SIS 算法在多智能体协作目标运输场景下训练了 2000 轮,实验结果如图 5 所示. 可以看到,DDQN、DDQN-BC 和 DDQN-2SIS 算法的平均累计奖赏均在训练的前 100 轮迅速增长,上述三种算法此时的表现无明显区别,而 LDQN 算法的平均累计奖赏则是处于缓慢增加状态. 在 100 轮以后,DDQN 算法以及 DDQN-BC 算法的平均累计奖赏稳定在 −2.5 左右,陷入了局部最优. 而 DDQN-2SIS 算法和 LDQN 算法的平均累计奖赏仍能逐渐上升,两者最终的平均累计奖赏都到达了 9 左右,要远高于其余两种算法. 纵观整个累计奖赏曲线,DDQN-2SIS 算法在收敛速度和最终结果上要优于 LDQN 算法. 通过观察离目标区域最短垂直距离发现,DDQN 算法以及 DDQN-BC 算法仅学会了执行相同动作,而未能学会将目标向目标区域运输. 在训练的前半程,LDQN 运输目标的能力要强于 DDQN-2SIS 算法,在训练的后半程则是 DDQN-2SIS 算法运输目标的能力更强,平均离目标区域的最短垂直距离更小. 在 DDQN-2SIS 算法的几次独立训练过程中,其累积奖赏曲线的差异比较大(阴影部分面积较大). 这是因为随机性对该场景的训练有较大影响,训练过程中智能体通过随机探索到达目标区域的次数会对训练的效率产生较大影响. 但纵观整个训练过程,我们可以得出结论,在多智能体目标运输场景下,我们提出的 DDQN-2SIS 算法表现要好于另外三种对比算法.

5.4.2 交叉路口通行场景

在交叉路口通行场景下,我们令智能体数目 $N=3,4,5$,并在每种智能体数目设置下分别使用四种算法进行了训练,实验结果如图 6 所示. 可以看到,在训练初期,DDQN、DDQN-BC 和 DDQN-2SIS 算法的累计奖赏曲线和碰撞次数曲线并无明显区别. 这是因为在训练初期,智能体均处于探索试错阶段,策略普遍表现较差. 对于 DDQN-2SIS 算法来说,在训练初期 Q_i^e 网络估计的 Q 值普遍偏小,经过 softmax 函数处理后的 Q 值之间相差很小,智能体之间尚处于一个几乎无通信的状态. 随着训练的进行,到了训练中期,上述三种算法的平均累计奖赏增长速度产生了区别,其中 DDQN-2SIS 算法的增长速度最快,DDQN-BC 算法次之,原始的 DDQN 算法增长速度最慢,平均碰撞次数曲线也出现了类似的模式,说明此时采用 DDQN-2SIS 算法训练的智能体已经能初步判断其它智能体动作对自己的重要程度. 在训练后期,三个算法的训练均趋于收敛,在三种智能体数目设置下均为 DDQN-2SIS 算法训练的智能体表现最好、平均累计

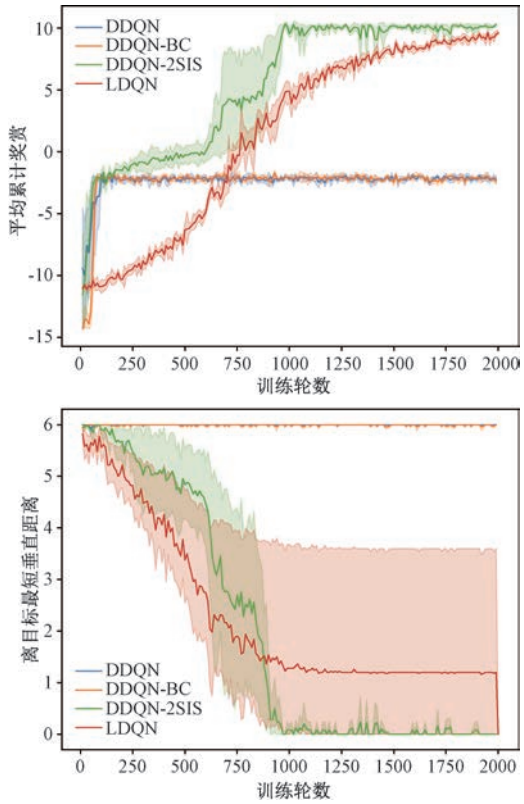

图 5　多智能体协作目标运输场景下的实验结果曲线

奖赏最高、平均碰撞次数最少. DDQN-BC 算法训练的智能体在智能体数目 N 为 3、4 时表现与原始的 DDQN 算法差不多，在 $N=5$ 时表现略好于原始的 DDQN 算法. 在训练的稳定性方面，原始 DDQN 算法的稳定性最差，阴影面积最大，DDQN-2SIS 与 DDQN-BC 算法的稳定性相差无几. CommNet 算法在该场景上的表现不佳，可能的原因是，CommNet 使用了一个集中式的神经网络将所有智能体的观测映射到动作. 在通信过程中，每个智能体都将自己的隐藏层状态（hidden state）作为通信内容广播给其它所有智能体，并将接收到的信息的平均作为下一层的输入. 上述过程中，每个智能体的发出的消息都被视作同等重要，而在我们的交通路口通行场景下，两辆相距较远的汽车发出的信息对于彼此而言相对没有那么重要，过多的冗余信息可能会干扰智能体的决策，增大学习难度. 另一方面，CommNet 交换的隐藏层信息可以看作是智能体对于自身局部观测的编码，该信息是智能体决策的重要依据，但仍不是其最终意图. 若智能体从环境中获取的观测是局部观测，那么 CommNet 可以在一定程度上帮助各个智能体共享彼此的观测，从而作出更好的决策. 在本文设计的实验中，我们更关注智能体在策略上的协作配合，因而我们约定每个智能体都能获得对环境的全局观测，CommNet 无法在共享观测这一方面获益. 纵观整个训练过程，我们可以得出结论，在交叉路口通行场景下，DDQN-2SIS 算法的表现要优于另外三种对比算法.

5.5　消融实验

为了说明 2SIS 的性能提升主要来自于依赖关系的建立方法，我们在交叉路口通行场景下额外设置了实验，将依赖关系的建立方法作为变量进行控制，得到了下述两种对比算法：

1. DDQN-ALL：在每一步，每个智能体都依赖其它所有智能体；

2. DDQN-RAN：在每一步，对于其它每个智能体，智能体都以 50%的概率依赖之.

除了上述建立依赖关系的部分，DDQN-ALL 算法与 DDQN-RAN 算法的其余部分和本文提出的 DDQN-2SIS 算法保持一致. 消融实验只在交叉路口通行场景下进行，与前面的实验设置一样，我们令智能体数目 $N=3,4,5$，并在每种智能体数目设置下分别使用三种算法进行了训练，实验结果如图 7 所示. 在智能体数目 $N=3$ 时，DDQN-2SIS 算法与 DDQN-ALL 算法的表现相差不多，在智能体数目 $N=4,N=5$ 时 DDQN-2SIS 算法的表现要好于 DDQN-ALL 算法. 而 DDQN-RAN 算法在三种智能体数目设置下的表现均差于另外两种，都是最差的. DDQN-ALL 算法训练的智能体在需要依赖其它智能体时总能正确发送依赖信息（因为在每一步都依赖所有其它智能体），并且智能体数目较少时，在观测中引入动作信息导致的观测空间增大问题没有那么明显，因此 DDQN-ALL 算法在智能体数目较少时表现较好. 随着智能体数目的增多，观测空间增大的问题逐渐明显，并且这种依赖方式会导致场景中的大部分智能体（N-1 个）因为被依赖而失去重新决策的能力，因此随着场景中智能体数目的增加，DDQN-2SIS 算法与 DDQN-ALL 算法的差距逐渐增大. 而 DDQN-RAN 算法在每一步都随机决定是否依赖某个智能体，在随机到正确依赖对象的情况下还需要不被其它智能体所依赖，否则无法利用意图信息重新进行决策. 并且引入意图信息导致的状态空间增大问题仍然存在. 这些困难导致了 DDQN-RAN 算法的表现在三种算法中是最差的. 上述实验能够在一方面说明我们提出的 DDQN-2SIS 算法是能够在通信阶段正确建立依赖关系的.

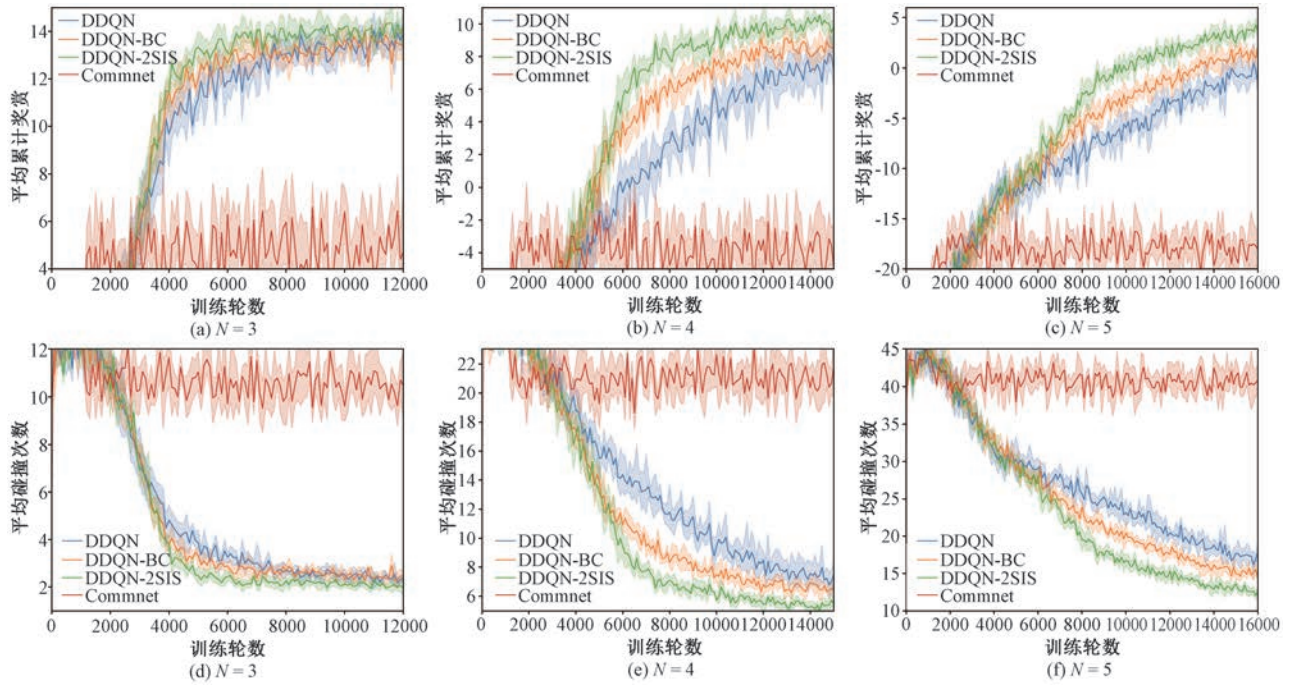

图 6 交叉路口通行场景下，智能体数目 $N=3,4,5$ 时的有效性实验结果曲线

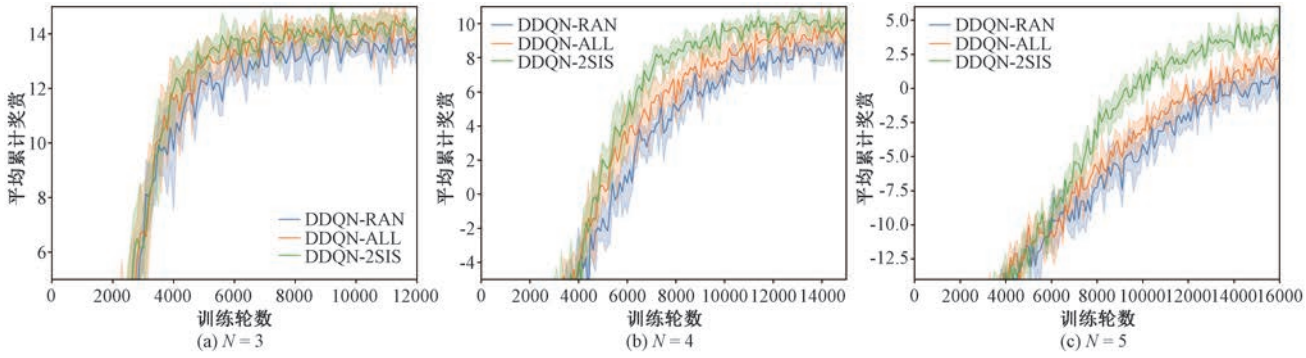

图 7 交叉路口通行场景下，智能体数目 $N=3,4,5$ 时的消融实验结果曲线

为了进一步说明 DDQN-2SIS 算法训练的智能体能够学会依赖正确的对象，我们把在交叉路口通行场景下(智能体数目 $N=4$)训练好的 1 号智能体的 Q_1^e 网络的参数保存了下来，并编写了测试程序加载 Q_1^e 来对智能体通信行为进行观察. 图 8 展示了 1 号智能体(绿色圆圈)在几种典型情景下选取依赖对象的情况. 图中的圆圈表示智能体，圆圈上的箭头表示智能体当前朝向. 箭头的颜色代表了智能体意向动作，灰色箭头表示停在原地，红色箭头表示朝前移动一格. 智能体的颜色表示是否被 1 号智能体所依赖. 红色表示智能体被 1 号智能体所依赖，灰色则表示智能体不被 1 号智能体所依赖. 在图 8(a)展示的空旷场景下，1 号智能体离其它智能体都有一定距离，此时其它智能体的动作不会对 1 号智能体的累计奖赏产生太大的影响，因此 1 号智能体的依赖对象集合为空集，即不依赖任何其它智能体. 在图 8(b)展示的相遇场景下，若红色的智能体选择反悔，将自己的动作改为前进一格，则有可能会和 1 号智能体相撞(在 1 号智能体不改变意向动作的情况下)，因此 1 号智能体需要依赖红色智能体的动作来决定是否要继续前进. 在图 8(c)展示的碰撞场景下，1 号智能体与红色的智能体已经发生了碰撞，处于同一个网格. 在下一个时刻，1 号智能体需要避免与红色的智能体执行相同的动作，否则会发生二次碰撞，因此在该情景下 1 号智能体也需要

依赖红色智能体的动作来决定自己下一时刻的动作. 智能体在这三个典型情景下的通信行为都比较符合直觉，说明 DDQN-2SIS 算法训练的智能体确实学会了如何选取依赖对象.

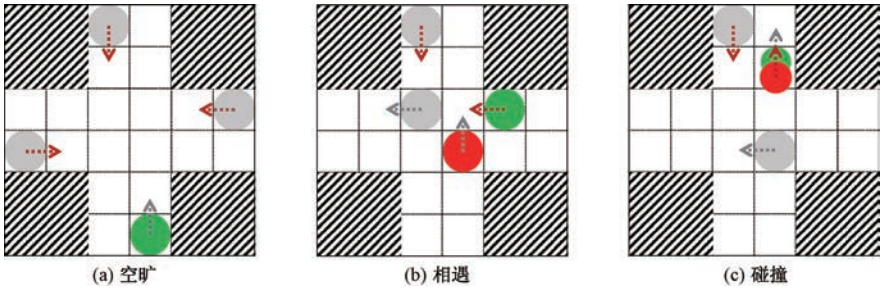

图 8　交叉路口通行场景下 DDQN-2SIS 智能体选取依赖对象的行为实例

5.6　参数实验

为了探究超参数 α 对实验结果的影响程度，以及如何选取该参数，我们为超参数 α 设计了参数实验. 根据我们的定义，重要程度阈值 α 的取值范围为[0，1]，重要程度超过该阈值的智能体将被选作依赖对象. 当 α 的取值趋于 0 时，在第二轮广播式通信后各个智能体获得的依赖关系图会包含更多的边；当 α 的取值趋于 1 时，则类似于不进行第二轮通信，第二轮通信后各个智能体获得的依赖关系图会包含更少的边. 为了探究该参数的设置对于实验结果的影响程度. 我们在智能体数目为 4 的交叉路口通行场景下补充了针对该超参数的参数实验，具体而言，我们在区间[0，1]内，以 0.05 为步长设置 α 进行训练，并分别统计了训练过程中（训练轮数 EP＝6000）和训练结束后（训练轮数 EP＝15000）智能体的平均累计奖赏，前者用于衡量算法的收敛速度，后者用于衡量算法的最终效果. 实验结果如图 9 所示. 从图 9 可以看出，随着重要程度阈值 α 的增大，算法的收敛速度有减慢的趋势；而在 $\alpha \leqslant 0.45$ 时，算法的最终的收敛效果均相差不大. 上述实验说明了 2SIS 的最终效果对于该超参数不是非常敏感，对于该超参数的选取，若只关注训练最终收敛效果，可以通过类似的参数扫描方法，在区间[0，1]内进行一个步长比较大的搜索即可，若同时关注收敛速度，则可以在此基础上进行更细粒度的参数搜索.

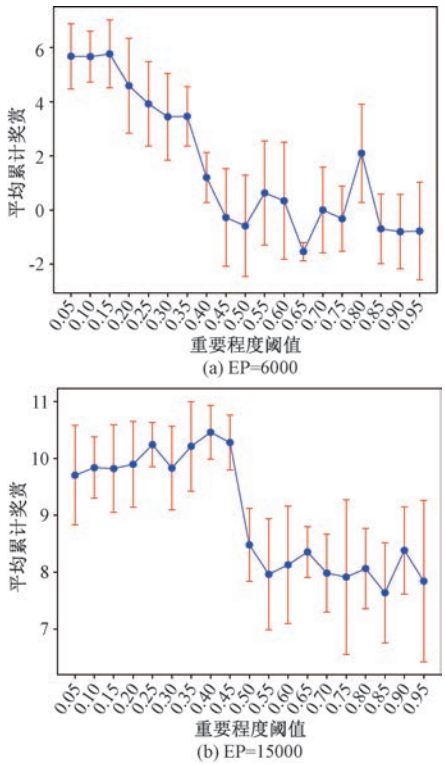

图 9　交叉路口通行场景下 DDQN-2SIS 重要程度阈值参数实验结果

6　总结与展望

得益于深度学习的迅速发展，深度强化学习的学习能力也大幅上升. 作为强化学习领域的重要分支. 多智能体强化学习领域面临更多的挑战和困难，如何提高智能体间的协作配合便是其中一个热门研究方向. 不少研究试图通过在多智能体强化学习中引入通信来加强智能体之间的配合，早期的相关研究关注如何通过通信来缓解多智能体强化学习中的局部观测现象，通过通信信道共享一部分观测，在一定程度上提高了智能体间的协作能力. 在最近的相关研究中，研究人员引入了通信意图的概念，让智能体互相交流自身的动作意图，以实现更好的配合. 但通信意图相关研究尚不充分，已有工作也存在意图误导、需要全局控制器等缺陷.

本文基于通信意图的思想，针对已有工作的不足提出了一种新的通信意图的方法. 按照本文提出的方法，智能体在决策前要先进行两轮通信，第一

轮通信所有智能体广播自己的意向动作，第二轮通信所有智能体广播自己的依赖对象集合，最后根据意向动作信息和依赖关系进行决策. 为了生成意向动作，每个智能体需要维护一个 Q^p 网络，以估计在当前观测下执行各个动作后续会获得的累计奖赏的期望. 受到差分奖赏技术的启发，我们让每个智能体维护一个 Q^e 网络，用于估计其它智能体执行不同动作时自身后续累计奖赏的期望，然后对可能出现的最大和最小 Q^e 值使用 softmax 函数进行归一化，最后取归一化结果的标准差作为对其它智能体重要程度的估计，依赖对象集合即为重要程度大于预设阈值的智能体集合. 第二轮通信结束后，智能体根据依赖关系信息构建依赖关系图，并使用相同的算法去掉其中的循环依赖. 对于被依赖的智能体，其最终动作即为其意向动作，而不被依赖的智能体则可以根据其它智能体的意向动作信息重新决策.

为了验证 DDQN-2SIS 算法的有效性，本文在多智能体协作目标运输和交叉路口通行两个多智能体协作场景下进行了训练. 实验结果显示，相较于原始的 DDQN 算法、广播式通信意图的 DDQN-BC 算法、Lenient-DQN 算法和 Commnet 算法，本文提出的 DDQN-2SIS 算法在两个场景下均展现出了明显的优势，验证了 DDQN-2SIS 算法的有效性. 此外，为了说明 DDQN-2SIS 算法训练的智能体能够学会依赖正确的对象，本文在交叉路口通行场景下额外设置了消融实验，与其它基于规则选取依赖对象的方法进行对比. 结果显示，DDQN-2SIS 算法在不同智能体数目设置下的表现均优于基于规则的方法. 最后，我们针对超参数 α 设计了额外的参数实验以说明该参数对算法训练的影响.

在接下来的工作中，我们将考虑如何在邻近的智能体之间构建局部依赖关系，以避免在大规模智能体场景下构建全局依赖关系开销较大的问题. 此外，将 2SIS 的单步意图共享框架扩展成多步意图共享也是我们未来工作的核心目标之一.

参 考 文 献

[1] Sutton R S, Barto A G. Reinforcement learning: an introduction. Cambridge, USA: MIT press, 2018

[2] Liu Quan, Zhai Jian-Wei, Zhang Zong-Zhang, et al. A survey on deep reinforcement learning. Chinese Journal of Computers, 2018, 41(1): 1-27 (in Chinese)
(刘全，翟建伟，章宗长，等. 深度强化学习综述. 计算机学报，2018，41(1)：1-27)

[3] Liu Jian-Wei, Gao Feng, Luo Xiong-Lin. Survey of deep reinforcement learning based on value function and policy gradient. Chinese Journal of Computers, 2019, 42(6): 1406-1438 (in Chinese)
(刘建伟，高峰，罗雄麟. 基于值函数和策略梯度的深度强化学习综述. 计算机学报，2019，42(6)：1406-1438)

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

[5] Berner C, Brockman G, Chan B, et al. Dota 2 with large scale deep reinforcement learning. arXiv preprint arXiv:1912.06680, 2019

[6] Rahmatizadeh R, Abolghasemi P, Behal A, et al. Learning real manipulation tasks from virtual demonstrations using LSTM. arXiv preprint arXiv:1603.03833, 2016

[7] Kalashnikov D, Irpan A, Pastor P, et al. Scalable deep reinforcement learning for vision-based robotic manipulation//Proceedings of the Conference on Robot Learning. Zürich, Switzerland, 2018: 651-673

[8] Li Guo-Liang, Zhou Xuan-He, Sun Ji, et al. A survey of machine learning based database techniques. Chinese Journal of Computers, 2020, 43(11): 2019-2049 (in Chinese)
(李国良，周煊赫，孙佶，等. 基于机器学习的数据库技术综述. 计算机学报，2020，43(11)：2019-2049)

[9] Chen J, Yuan B, Tomizuka M. Model-free deep reinforcement learning for urban autonomous driving//Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC). New Zealand, 2019: 2765-2771

[10] Gu J, Wang Y, Chen Y, et al. Meta-learning for low-resource neural machine translation. arXiv preprint arXiv: 1808.08437, 2018

[11] Skerry-Ryan R J, Battenberg E, Xiao Y, et al. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron//Proceedings of the International Conference on Machine Learning. Stockholm, Sweden, 2018: 4693-4702

[12] Mao H, Alizadeh M, Menache I, et al. Resource management with deep reinforcement learning//Proceedings of the 15th ACM Workshop on Hot Topics in Networks. Atlanta, USA, 2016: 50-56

[13] Du Wei, Ding Shi-Fei. Overview on multi-agent reinforcement learning. Computer Science, 2019, 46(8): 1-8 (in Chinese)
(杜威，丁世飞. 多智能体强化学习综述. 计算机科学，2019，46(8)：1-8)

[14] Pham H X, La H M, Feil-Seifer D, et al. Cooperative and distributed reinforcement learning of drones for field coverage. arXiv preprint arXiv:1803.07250, 2018

[15] Qie H, Shi D, Shen T, et al. Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning. IEEE access, 2019, 7: 146264-146272

[16] Calvo J A, Dusparic I. Heterogeneous multi-agent deep rein-

forcement learning for traffic lights control//Proceedings of Artificial Intelligence and Computer Science. Dublin, Ireland, 2018: 2-13

[17] Nguyen T T, Nguyen N D, Nahavandi S. Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. IEEE transactions on cybernetics, 2020, 50(9): 3826-3839

[18] Choi Y C, Ahn H S. A survey on multi-agent reinforcement learning: coordination problems//Proceedings of 2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications. QingDao, China, 2010: 81-86

[19] Foerster J, Assael I A, De Freitas N, et al. Learning to communicate with deep multi-agent reinforcement learning//Proceedings of the Advances in Neural Information Processing Systems. Barcelona, Spain, 2016: 2137-2145

[20] Sukhbaatar S, Fergus R. Learning multiagent communication with backpropagation//Proceedings of the Advances in Neural Information Processing Systems. Barcelona, Spain, 2016: 2244-2252

[21] Das A, Gervet T, Romoff J, et al. Tarmac: Targeted multi-agent communication//Proceedings of the International Conference on Machine Learning. Long Beach, USA, 2019: 1538-1546

[22] Jiang J, Lu Z. Learning attentional communication for multi-agent cooperation//Proceedings of the Advances in Neural Information Processing Systems. Montréal, Canada, 2018: 7254-7264

[23] Kim W, Park J, Sung Y. Communication in multi-agent reinforcement learning: intention sharing//Proceedings of the International Conference on Learning Representations. Vienna, Austria, 2021: 1-15

[24] Liu Z, Wan L, Sun K, et al. Multi-Agent intention sharing via leader-Follower forest. arXiv preprint arXiv:2112.01078, 2021

[25] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double q-learning//Proceedings of the AAAI Conference on Artificial Intelligence. Phoenix, USA, 2016: 2094-2100

[26] Foerster J, Farquhar G, Afouras T, et al. Counterfactual multi-agent policy gradients//Proceedings of the AAAI Conference on Artificial Intelligence. New Orleans, USA, 2018: 2974-2982

[27] Rashid T, Samvelyan M, Schroeder C, et al. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning//Proceedings of the International Conference on Machine Learning. Stockholm, Sweden, 2018: 4295-4304

[28] Yang Y, Hao J, Chen G, et al. Q-value path decomposition for deep multiagent reinforcement learning//Proceedings of the International Conference on Machine Learning. Vienna, Austria, 2020: 10706-10715

[29] Kirby S. Natural language from artificial Life. Artificial life, 2002, 8(2): 185-215

[30] Mao H, Liu W, Hao J, et al. Neighborhood cognition consistent multiagent reinforcement learning//Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA, 2020, 34(05): 7219-7226

[31] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning. arXiv preprint arXiv: 1509.02971, 2015

[32] Bellman R. A Markovian decision process. Journal of mathematics and mechanics, 1957, 6(5): 679-684

[33] Howard R A. Dynamic programming and markov processes. New York, USA: John Wiley, 1960

[34] Monahan G E. State of the art—A survey of partially observable Markov decision processes: theory, models, and algorithms. Management Science, 1982, 28(1): 1-16

[35] Tampuu A, Matiisen T, Kodelja D, et al. Multiagent cooperation and competition with deep reinforcement learning. PloS One, 2017, 12(4): e0172395

[36] Watkins C J C H, Dayan P. Q-learning. Machine Learning, 1992, 8(3): 279-292

[37] Regehr M T, Ayoub A. An elementary proof that Q-learning converges almost surely. arXiv preprint arXiv:2108.02827, 2021

[38] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013

[39] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529-533

[40] Thrun S, Schwartz A. Proceedings of the 1993 connectionist models summer school: Issues in using function approximation for reinforcement learning. New York, USA: Psychology Press, 1994

[41] Leslie A M. Pretense and representation: The origins of "theory of mind". Psychological Review, 1987, 94(4): 412

[42] Carmel D, Markovitch S. Opponent modeling in multi-agent systems//Proceedings of the International Joint Conference on Artificial Intelligence. Montréal , Canada, 1995: 40-52

[43] Agogino A K, Tumer K. Unifying temporal and structural credit assignment problems//Proceedings of the Autonomous Agents and Multi-Agent Systems Conference. New York, USA, 2004: 980-987

[44] Agogino A K, Tumer K. Analyzing and visualizing multiagent rewards in dynamic and stochastic domains//Proceedings of the Autonomous Agents and Multi-Agent Systems. Estoril, Portugal, 2008, 17(2): 320-338

[45] Zhang S Q, Zhang Q, Lin J. Efficient communication in multi-agent reinforcement learning via variance based control//Proceedings of the Advances in Neural Information Processing Systems, Vancouver, Canada, 2019: 3230-3239

[46] Karp R M. Complexity of computer computations: Reducibility among combinatorial problems. New York, USA: Springer, 1972

[47] Eades P, Lin X, Smyth W F. A fast and effective heuristic for the feedback arc set problem. Information Processing Let-

ters, 1993, 47(6): 319-323

[48] Busoniu L, Babuska R, Schutter B D. Innovations in multi-agent systems and applications-1: Multi-agent reinforcement learning: an overview. Berlin, Germany: Springer, 2010

[49] Palmer G, Tuyls K, Bloembergen D, et al. Lenient multi-agent deep reinforcement learning. arXiv preprint arXiv: 1707.04402, 2017

[50] Samvelyan M, Rashid T, De Witt C S, et al. The starcraft multi-agent challenge. arXiv preprint arXiv:1902.04043, 2019

附录 A 重新决策是否会导致依赖关系发生变化

在重新决策后，由于智能体的执行动作与第一轮广播里传播的意图动作有可能不同，这可能会导致重新决策前后智能体建立的依赖关系发生变化，从而使得当前决策的依据具有一定的误导性。因此我们在交通路口通行场景下额外设计了实验来统计重新决策后依赖关系的发生变化的概率，实验结果如表 3：

表 3 重新决策后依赖关系变化概率

智能体数目	依赖关系变化概率
3	0.97%±0.17%
4	3.30%±0.58%
5	5.9%±0.53%

可以看到，在我们的实验设置下，2SIS 在重新决策后依赖关系发生变化的概率比较小，目前我们认为该问题对算法训练的影响尚且不是很显著，因此可以暂时不考虑对依赖关系进行迭代计算。但重新决策后导致依赖关系发生变化的现象确实存在，并且随着智能体数目的增多依赖关系发生变化的概率有着上升的趋势，因此这是个值得关注的现象，我们将这一现象及其可能带来的误导性纳入了我们未来工作的考虑范围之中。

附录 B 在 SMAC[50] 场景下的实验结果

我们还在更复杂的 SMAC:3m 场景下采用 DDQN、DDQN-BC 和 DDQN-2SIS 方法进行了实验。实验结果如图 10：

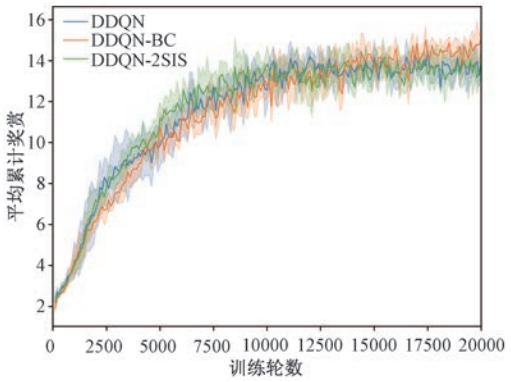

图 10 在 SMAC 3m 场景下 DDQN、DDQN-BC 和 DDQN-2SIS 的平均累计奖赏曲线

实验结果表明，在 SMAC:3m 场景下，2SIS 未能体现出明显优势，但另一方面，2SIS 也未展示出明显劣势。这说明该场景不处在 2SIS 的适用范围内，2SIS 关注的是当前决策时刻队友的下一步动作意图信息，对于有些场景来说，仅关注下一步的意图可能是不够的，可能需要接下来一段时间内的动作意图方才能体现出作用，而这一部分也是 2SIS 接下来的拓展目标之一。但即使在上述场景中使用 2SIS 方法进行训练，2SIS 仍能保证在引入通信后观测空间增大的影响下训练的结果不会变差。

WU Jun-Feng, M. S. candidate. His research interests include reinforcement learning, multi-agent system and swarm intelligence.

WANG Wen, Ph. D. candidate. His research interests include reinforcement learning, multi-agent system and swarm intelligence.

WANG Liang, Ph. D., associate professor. His research interests include swarm intelligence, swarm intelligence software and artificial intelligence.

TAO Xian-Ping, Ph. D., professor. His research interests include software methodology and swarm intelligence.

HU Hao, Ph. D., associate professor. His research interests include blockchain, edge intelligence and group game.

WU Hai-Jun, M. S., senior engineer. His research interests include multimedia information processing and computer architecture.

Background

How to make agents learn to cooperate is an important challenge in multi-agent reinforcement learning. Introducing communication to MARL is a natural way to achieve better cooperation among agents. Most of the existing work related to communication focuses on partial observation problems. In these work, the communication actually enables agents to share observa-

tions with others. Let agents share intention with each other is another way to combine communication with MARL. With teammates' intention, agents are supposed to perform better on cooperative tasks. However, MARL with intention sharing is still under studying. The few existing work faces the problems of "intention misleading" and "centralized controller".

Aiming at the above problems, this paper proposes a new multi-agent reinforcement learning intention sharing scheme—2SIS (2 Step Intension Sharing). In 2SIS, agents complete intention sharing and dependency relationship establishment in two step communication. At each step, the agent firstly generates action intention according to its own observation and broadcasts it. Then, according to the intentions of other agents and its own observation, the agent decides which agents' action intention to rely on and broadcasts dependency information. After two rounds of broadcasting, the agent gets the intention information and dependency graph of other agents. In order to avoid "intention misleading", 2SIS specifies that if an agent is relied on, it cannot change its intention. Only agent that is not dependent on by any agent can re-make decisions with intention information. Experimental results show that proposed method outperforms comparison algorithm in both convergence speed and result. And the ablation experiment results show that the proposed method can really make agents learn how to establish the dependency relation.