计算机科学
COMPUTER SCIENCE

基于观测重构的多智能体强化学习方法

史殿习，胡浩萌，宋林娜，杨焕焕，欧阳倩滢，谭杰夫，陈莹

引用本文

史殿习，胡浩萌，宋林娜，杨焕焕，欧阳倩滢，谭杰夫，陈莹. 基于观测重构的多智能体强化学习方法[J]. 计算机科学，2024，51(4)：280-290.

SHI Dianxi, HU Haomeng, SONG Linna, YANG Huanhuan, OUYANG Qianying, TAN Jiefu , CHEN Ying. Multi-agent Reinforcement Learning Method Based on Observation Reconstruction[J]. Computer Science, 2024, 51(4): 280-290.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230600055

基于观测重构的多智能体强化学习方法

史殿习[1,3]　胡浩萌[2,3]　宋林娜[2,3]　杨焕焕[2,3]　欧阳倩滢[1,3]　谭杰夫[3]　陈　莹[4]
1 智能博弈与决策实验室　北京 100091
2 国防科技大学计算机学院　长沙 410073
3 天津(滨海)人工智能创新中心　天津 300457
4 国防科技创新研究院　北京 100071
(dxshi@nudt.edu.cn)

摘　要　共同知识是多智能体系统内众所周知的知识集。如何充分利用共同知识进行策略学习,是多智能体独立学习系统中的一个挑战性问题。针对这一问题,围绕共同知识提取和独立学习网络设计,提出了一种基于观测重构的多智能体强化学习方法 IPPO-CKOR。首先,对智能体的观测信息进行共同知识特征的计算与融合,得到融合共同知识特征的观测信息;其次,采用基于共同知识的智能体选择算法,选择关系密切的智能体,并使用重构特征生成机制构建它们的特征信息,其与融合共同知识特征的观测信息组成重构观测信息,用于智能体策略的学习与执行;最后,设计了一个基于观测重构的独立学习网络,使用多头自注意力机制对重构观测信息进行处理,使用一维卷积和 GRU 层处理观测信息序列,使得智能体能够从观测信息序列中提取出更有效的特征,有效缓解了环境非平稳与部分可观测问题带来的影响。实验结果表明,相较于现有典型的采用独立学习的多智能体强化学习方法,所提方法在性能上有显著提升。

关键词:观测重构;多智能体协作策略;多智能体强化学习;独立学习
中图分类号　TP391

Multi-agent Reinforcement Learning Method Based on Observation Reconstruction

SHI Dianxi[1,3],HU Haomeng[2,3],SONG Linna[2,3],YANG Huanhuan[2,3],OUYANG Qianying[1,3],TAN Jiefu[3] and CHEN Ying[4]
1 Intelligent Game and Decision Lab(IGDL),Beijing 100091,China
2 College of Computer,National University of Defense Technology,Changsha 410073,China
3 Tianjin Artificial Intelligence Innovation Center,Tianjin 300457,China
4 National Innovation Institute of Defense Technology,Beijing 100071,China

Abstract　Common knowledge is a well-known knowledge set within a multi-agent system. How to make full use of common knowledge for strategic learning is a challenging problem in multi-agent independent learning systems. In addressing this problem,this paper proposes a multi-agent reinforcement learning method called IPPO-CKOR based on observation reconstruction,focusing on common knowledge extraction and independent learning network design. Firstly,the common knowledge features of agents' observation information are computed and fused to obtain fused observation information with common knowledge features. Secondly,an agent selection algorithm based on common knowledge is used to select closely related agents,and a feature generation mechanism based on reconstruction is employed to construct their feature information. The reconstructed observation information,composed of the fused observation information with common knowledge features,is utilized for learning and executing agent policies. Thirdly,a network structure based on observation reconstruction is designed,which employs multi-head self-attention mechanism to process the reconstructed observation information and uses one-dimensional convolution and GRU layers to handle observation information sequences. This enables the agents to extract more effective features from the observation information sequences,effectively alleviating the impact of non-stationary environments and partially observable problems. Experimental results demonstrate that the proposed method outperforms existing typical multi-agent reinforcement learning methods that employ independent learning in terms of performance.

Keywords　Observation reconstruction,Multi-agent cooperative strategy,Multi-agent reinforcement learning,Independent learning

到稿日期:2023-06-06　返修日期:2023-11-07
基金项目:科技部科技创新 2030-重大项目(2020AAA0104802);国家自然科学基金(91948303)
This work was supported by the Science and Technology Innovation 2030 Major Project(2020AAA0104802) and National Natural Science Foundation of China(91948303).
通信作者:陈莹(selina.ychen@foxmail.com)

1 引言

多智能体系统[1]是由多个智能体组成的计算系统，可以更好地适应大规模复杂场景，在多机器人、智慧交通和物流运输等许多领域具有广泛的应用。多智能体系统中的智能体间存在完全协作、完全对抗以及协作与对抗混合3类关系。其中，完全协作的多智能体系统通过使多个智能体分工协作完成某一任务，弥补单个智能体能力不足的问题，具有广泛的应用场景。多智能体深度强化学习是一种解决多智能体协作策略问题的有效方法，在电子游戏、编队控制、能源控制等领域得到了广泛应用，并取得了良好效果。然而，多智能体深度强化学习目前还面临着在真实世界中应用困难、环境非平稳性，以及部分可观测下的信息不完整性等问题，需要进一步的研究与改进。因此，开展多智能体深度强化学习方法研究，具有十分重要的理论价值和现实意义。

多智能体深度强化学习主要包括集中学习[2]、独立学习[3]和集中式训练分布式执行[4]3种学习范式。其中，集中学习将多个智能体联合起来作为一个整体，解决多智能体控制问题，但面临维度爆炸和通信稳定性等问题。MFMARL算法[5]通过平均场论有效应对大规模智能体交互、维度爆炸和算法设计复杂度等问题，但带来了较高的计算量。集中式训练分布式执行是指在训练过程中使用集中式的训练方式，而在决策过程中让智能体只依赖本地信息，从而实现分布式执行。MADDPG算法[4]是DDPG算法的一种变体，它使用集中式评论家和分布式演员进行训练。VDN算法[6]使用DQN算法学习联合Q值，并将其视为每个智能体本地状态-动作价值函数的线性加和。QMIX算法[7]则使用参数化混合网络计算联合Q值，并强制实施单调性约束。HATRPO和HAPPO算法[8]将置信域学习应用于多智能体强化学习当中，取得了最佳性能。集中式训练分布式执行虽然解决了执行阶段的问题，但仍然存在维度爆炸和通信依赖等问题。采用集中式训练的方法在训练过程中需要大量通信，且需要全局信息，对训练环境有较高的要求，而这些要求在现实环境中往往难以满足，因此这类方法的应用主要局限于模拟环境或实验室环境。

独立学习将多智能体强化学习看作多个智能体独立地学习自己的策略。IA2C，IQL和IPPO算法[9-10]分别是AC，Q-Learning和PPO算法的变体，这些算法采用独立学习的方法，无需全局信息，对智能体间通信的需求较小，具有良好的灵活性和可扩展性，可广泛应用于多智能体系统当中。然而，由于环境非平稳性问题和部分可观测性问题的限制，其虽然在简单环境中具有良好的效果，但在复杂环境中难以学习到好的策略。以此为动机，本文针对基于独立学习的多智能体强化学习方法展开研究，旨在提高此类方法的性能表现，扩展其应用场景。

经过多年的研究和发展，基于独立学习的多智能体强化学习方法因其应用的灵活性和低成本，取得了良好的进展；但受到环境非平稳性和部分可观测性等问题的影响，与集中式训练的方法相比依然存在较大的性能差距[11]。共同知识(Common Knowledge)是智能体组内众所周知的知识集[12]，被广泛应用于辅助决策[13]等诸多任务当中。共同知识越丰富，智能体之间的联系就越紧密。基于共同知识的丰富程度，我们可以选择出与智能体关系密切的智能体，这类智能体的特征信息对于决策更具有参考价值。如果能够显式地利用关系密切的智能体的特征信息，将能有效缓解环境非平稳性等问题。同时，通过更好地处理智能体观测信息序列，可以从观测信息序列中提取出尽可能多的特征信息。为此，本文围绕共同知识提取和独立学习网络设计，提出了一种基于观测重构的多智能体强化学习方法IPPO-CKOR(Independent Proximal Policy Optimization Using Common Knowledge Observation Reconstruction)。首先，通过计算智能体之间共同知识的丰富程度，选择关系密切的智能体，使得智能体能够分辨出关系密切的智能体并与之协作；其次，构建关系密切的智能体的特征信息，并与融合共同知识特征的观测信息组成重构观测信息，使得智能体能够显式地利用关系密切的智能体的特征信息进行决策，以缓解环境非平稳性问题；然后，设计了一个基于观测重构的独立学习网络结构，在提取各智能体特征信息的同时，更好地处理智能体观测信息序列，以有效应对部分可观测性问题；最后，实验结果表明，在星际争霸多智能体挑战环境(The StarCraft Multi-Agent Challenge，SMAC)[14]中，本文所提出的IPPO-CKOR方法与现有典型的独立学习方法相比，具有显著的性能提升。

本文第2章对相关工作进行了重点描述；第3章具体描述了本文所提出基于观测重构的多智能体强化学习方法；第4章描述了仿真实验的设置以及实验结果；最后总结全文并展望未来。

2 相关工作

2.1 多智能体强化学习

2.1.1 分布式部分可观测马尔可夫决策过程

完全协作的多智能体强化学习常被建模为分布式部分可观测马尔可夫决策过程(Decentralized Partially Observable Markov Decision Processes，Dec-POMDP)[15]，它描述了合作智能体团队在部分可观测和随机性的环境下选择序列动作的多智能体任务。Dec-POMDP是一个元组$\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \Omega, O, r, \rho, \gamma \rangle$，其中，$\mathcal{N}:=\{1,\cdots,N\}$表示环境中的智能体，$s\in\mathcal{S}$表示一个状态。在开始时，从环境中获得初始状态$s_0\sim\rho$。在每个时间步$t$，所有智能体$i\in\mathcal{N}$从观测函数$O(s_t,i)$中获得局部观测$o_t^i=O(s_t,i)\in\Omega$，然后同步选择动作$a_t^i\in\mathcal{A}$，形成联合动作$\boldsymbol{a}_t:=\{a_t^i\}_{i=1}^N\in\mathcal{A}^N$。所有智能体在状态$s_t$下执行联合动作$\boldsymbol{a}_t$后，由转移函数$P$生成下一个状态$s_{t+1}\sim P(s_t,\boldsymbol{a}_t)$，同时，智能体获得团队奖励$r_t=r(s_t,\boldsymbol{a}_t)$。

智能体的观测和动作历史表示为$\tau_t^i\in\mathcal{T}_t:=(\Omega\times\mathcal{A})^t\times\Omega$，而所有智能体的历史表示为$\boldsymbol{\tau}_t:=\{\tau_t^i\}_{i=1}^N$。每个智能体只根据局部观测从分布式策略中选择其动作$a_t^i\sim\pi^i(\cdot\mid\tau_t^i)$。智能体的目标是学习分布式的联合策略$\pi(\boldsymbol{a}\mid\boldsymbol{\tau}_t):=\prod_{i=1}^N\pi^i(a^i\mid\tau_t^i)$，该策略能使折扣回报期望$J(\pi)\doteq\mathbb{E}[\sum_{t=0}^{\infty}\gamma^t r_t]$最大化，其中，$\gamma\in[0,1)$是一个折扣因子。$\pi(\boldsymbol{a}\mid\boldsymbol{\tau}_t)$可以推导出用于估计联合动作$\boldsymbol{a}_t$在$s_t$状态下的折扣回报的期望价值函数$Q^\pi$。然后，

智能体可以基于$Q^{\pi}(s_t,\boldsymbol{\tau}_t,\boldsymbol{a}_t):=\mathbb{E}[\sum_{i=0}^{\infty}\gamma^i r_{t+i}]$使用联合策略$\pi$。

2.1.2 独立多智能体强化学习

目前，采用独立学习方式的多智能体强化学习方法主要是将单智能体强化学习算法扩展到多智能体环境当中，为每个智能体单独使用一个单智能体的强化学习算法。IQL算法[9]将单智能体的DQN算法扩展到分布式的多智能体环境当中，为每个智能体执行一个Q-learning算法。IAC算法[11]将演员-评论家框架扩展到多智能体的场景中，每个智能体拥有各自独立的演员和评论家，独立地学习协作的策略。SNAC算法[11]在IAC的基础上实现了参数的共享，即所有策略共享一套演员和评论家，只训练一个策略用于团队的决策；SEAC算法[11]保持每个智能体具有独立的参数，但是智能体在更新时，通过使用所有智能体的数据的方式进行更新，从而实现经验共享。独立近端策略优化算法(Independent Proximal Policy Optimization，IPPO)[10]为每个智能体使用一个PPO算法，在一些简单环境中获得了很好的效果。IPPO算法中的智能体基于策略剪切学习以θ为参数的分布式策略π_θ，独立地进行策略更新。算法使用了优势函数基于独立学习的变体，其中每个智能体通过以$\gamma=0.99$和$\lambda=0.95$为折扣因子的广义优势估计(Generalized Advantage Estimation，GAE)，学习基于本地观测信息的以ϕ为参数的价值函数$V_\phi(o_t^i)$。为了鼓励智能体探索，避免过拟合，IPPO算法在损失计算中加入了正则化的熵损失。具体地，智能体i的优势估计如式(1)所示：

$$A_t^i=\sum_{l=0}^{h}(\gamma\lambda)^l\delta_{t+l}^i \tag{1}$$

其中，$\delta_t^i=r_t(o_t^i,a_t^i)+\gamma V_\phi(o_{t+1}^i)-V_\phi(o_t^i)$是$t$时间步下的TD误差。算法使用团队奖励值$r_t(s_t,\boldsymbol{a}_t)$来近似个体奖励值$r_t(o_t^i,a_t^i)$。式(2)为智能体$i$策略损失的计算公式：

$$\mathscr{L}^i(\theta)=E_{o_t^i,a_t^i}\left[\min\left(\frac{\pi_\theta(a_t^i\mid o_t^i)}{\pi_{\theta_{\text{old}}}(a_t^i\mid o_t^i)}A_t^i,\ clip\left(\frac{\pi_\theta(a_t^i\mid o_t^i)}{\pi_{\theta_{\text{old}}}(a_t^i\mid o_t^i)},1-\epsilon,1+\epsilon\right)A_t^i\right)\right] \tag{2}$$

除了对策略更新进行剪切之外，IPPO算法还通过值剪切将每个智能体的价值函数的更新限制在小于ε[16]的范围内。价值函数损失的计算式如式(3)所示：

$$\mathscr{L}^i(\phi)=\mathbb{E}_{o_t^i}[\min\{(V_\phi(o_t^i)-\hat{V}_t^i)^2,(V_{\phi_{\text{old}}}(o_t^i)+clip(V_\phi(o_t^i)-V_{\phi_{\text{old}}}(o_t^i),-\epsilon,+\epsilon)-\hat{V}_t^i)^2\}] \tag{3}$$

其中，ϕ_{old}为参数更新前的旧参数，而$\hat{V}_t^i=A_t^i+V_\phi(o_t^i)$。该更新公式能够将价值函数的更新限制在信任域内，避免函数过拟合至最近的数据批次。IPPO算法的总损失值计算如式(4)所示：

$$\mathscr{L}(\theta,\phi)=\sum_{i=1}^{n}(\mathscr{L}^i(\theta)+\lambda_{\text{critic}}\mathscr{L}^i(\varphi)+\lambda_{\text{entropy}}\mathscr{H}(\pi^i)) \tag{4}$$

其中，$\mathscr{L}(\pi^i)$是策略的熵，λ_{critic}和λ_{entropy}分别是价值函数损失与熵损失的权重。

上述算法总体上都是简单地将对应的单智能体算法扩展到多智能体的场景中，没有针对独立学习的特点进行改进。在多智能体环境中，状态转移函数P和奖励函数r都是以联合动作$\boldsymbol{a}_t$为条件，而在独立学习中，每个智能体将其他所有智能体视作环境的一部分，仅依赖个体观测信息进行分布式的联合决策。但是，环境中的所有智能体都在学习，它们的策略不断变化，所以对独立学习的智能体来说，转移函数和奖励函数是非稳定的。因此，对智能体来说，环境是非平稳的，这导致算法难以学习到一个比较好的策略。

2.2 共同知识

共同知识是一个博弈论中的概念，指的是一组信息，智能体组中所有智能体都拥有这组信息，所有智能体知道所有智能体拥有，且所有智能体知道所有智能体知道所有智能体拥有，以此类推[12]。换而言之，共同知识在其所在智能体组内是众所周知的，且大家都知道彼此拥有这组信息。

视野共同知识(Field-of-view Common Knowledge)是完全历史共同知识(Complete-history Common Knowledge)的一种形式[17]。在多智能体系统当中，当智能体能够通过本地的观测推测出其他智能体的部分观测时，则会出现视野共同知识，因此，其广泛存在于多智能体的系统当中。对一个智能体而言，组内的视野共同知识则是各个智能体能够互相重构的观测结果的交集。

目前，共同知识已经开始应用于多智能体强化学习当中。Nayyar等[18]使用共同知识将分布式的规划问题重新建模为POMDP，然后使用动态编程与集中的协作器来解决这些问题。但是，他们提供的方法并没有实现对高维环境的支持。Guestrin等[19]将智能体的价值函数表示为基于固定先验共同知识的特定背景下的单智能体价值之和。MACKRL算法[13]通过在演员-评论家架构中使用分级策略利用共同知识，实现了分布式的联合动作选择。然而，MACKRL的实现较为复杂，当智能体的数量很大时，分层策略的构建变得非常复杂。IPPO-CK[20]则是将共同知识整合到观测中，并在训练过程中动态地学习如何利用共同知识，显著降低了利用共同知识的复杂程度。

2.3 共同知识特征计算与融合

IPPO-CK算法[20]在IPPO算法基础上增加了共同知识特征的计算与融合，即计算观测信息内共同知识的特征信息，通过将这些特征信息与观测信息融合，得到融合共同知识特征的观测信息(简称为融合观测信息)，显式地利用共同知识。共同知识特征信息计算与融合的具体过程是：智能体i得到观测信息o^i后，分别计算智能体i与其中各个智能体的共同知识值(Common Knowledge Value，CKV)以及共同知识协作指数(Common Knowledge Cooperation Index，CKCI)，最后将共同知识值和协作指数与原始观测信息o^i结合，得到融合观测信息m_t。

智能体i观测信息内智能体j的共同知识值CKV_j反映了两个智能体视野共同知识的丰富程度，它是两者共同观测到的盟友、敌方智能体数量的加权和除以智能体i观测范围内盟友、敌方智能体数量的加权和，计算公式如式(5)所示：

$$CKV_j=\frac{\omega\times N_{\text{al\_ck}}^j+N_{\text{en\_ck}}^j}{\omega\times N_{\text{al\_seen}}+N_{\text{en\_seen}}} \tag{5}$$

其中，ω是共同知识盟友信息权重，作用是调整计算过程中对盟友共同知识的侧重程度；$N_{\text{al_seen}}$和$N_{\text{en_seen}}$分别是智能体在观测范围内看到的盟友和敌方智能体的数量；$N_{\text{al_ck}}^j$和$N_{\text{en_ck}}^j$分别

是智能体 i 和智能体 j 两个智能体都能观测到的盟友、敌方智能体的数量。

智能体的共同知识协作指数是对观测范围内所有智能体共同知识丰富程度的总结，通过观测内各个盟友、敌方智能体的共同知识值的加权和除以盟友与敌方智能体数量的加权和计算。式(6)为共同知识协作指数的计算公式：

$$CKCI=\frac{\omega * \sum_{i_{al}=1}^{N_{al}} CKV_{i_{al}}+\sum_{i_{en}=1}^{N_{en}} CKV_{i_{en}}}{\omega \times N_{al}+N_{en}} \tag{6}$$

在计算出智能体 i 观测范围内各个智能体的共同知识值和共同知识协作指数后，结合原始观测信息，得到融合共同知识特征的观测信息。具体的融合方式为：将观测信息内智能体的 CKV 插入对应特征信息的后方，同时，将共同知识协作指数插入到智能体 i 自身特征信息的后方，由此得到融合信息 m_t。

3 基于观测重构的多智能体强化学习方法

3.1 概述

为充分发掘和利用智能体之间的共同知识，选择出与智能体关系密切的其他智能体，提升智能体的策略学习效率，我们提出了一种基于观测重构的多智能体强化学习方法 IPPO-CKOR，方法框架如图 1 所示。其核心是观测信息重构机制和基于观测重构的独立学习网络两个部分。观测信息重构机制的作用是构建出关系密切的智能体的特征信息并与本地观测信息结合，形成重构观测信息；在此基础上，使用基于观测重构的独立学习网络，对观测重构机制得到的重构观测信息进行进一步处理。首先，观测重构机制通过基于共同知识特征的计算与融合方法，对观测信息内的共同知识特征进行提取，得到融合共同知识特征的观测信息；其次，基于二阶共同知识丰富度（Two Stage Common Knowledge Richness，TSCKR）来衡量智能体间的关系密切程度，并通过排序算法选择出关系密切的智能体，构建出它们的特征信息，将重构出的信息与融合观测信息相结合得到重构观测信息，并替代原始观测信息用于智能体后续的决策与策略学习。观测信息重构使得智能体具备了分辨出观测范围内与自身关系密切的智能体的能力，并通过重构观测信息更好地利用了这部分关系密切智能体的特征信息。这样一来，智能体能够在决策过程中更多地关注关系密切智能体的特征信息，减轻其他智能体特征信息在决策中的权重，进而提高决策质量，缓解其他智能体带来的环境非平稳性问题。

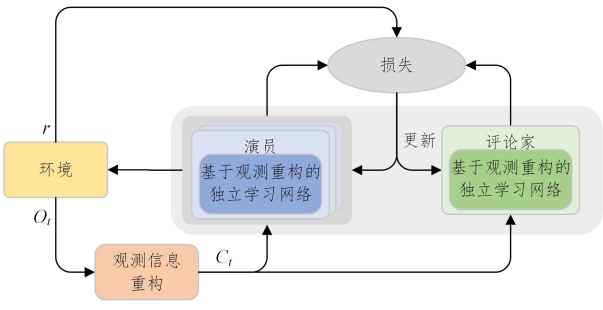


图 1 基于观测重构的多智能体强化学习方法的框架

Fig. 1 Framework of multi-agent reinforcement learning method based on observation reconstruction

在 IPPO-CKOR 方法中，演员与评论家内部的策略函数和价值函数均使用基于观测重构的独立学习网络近似。该网络包含了多头自注意力机制[21]、一维卷积和 GRU 层等模块。首先，网络将输入的重构观测信息进行分离，得到重构特征信息矩阵和融合观测信息。一方面，使用多头自注意力机制对重构特征信息矩阵进行处理，得到智能体特征信息的重要性以及各个智能体特征之间的关系；另一方面，使用一维卷积对多个连续时间步的融合观测信息进行处理，将信息降维的同时，得到多个连续时间步融合观测信息序列的特征。然后，使用 GRU 层对信息进行处理，更好地提取融合观测信息序列的特征信息。最后，将这两部分信息连结并输入到多层感知机 MLP 中，得到网络的输出。基于观测重构的独立学习网络，一方面使智能体获得了自身与其观测信息内关系密切智能体特征信息间的关系与重要性，高效地利用了重构观测信息；另一方面，使得智能体更好地从观测信息序列中获得信息，有效地缓解了独立学习的信息不完整性所带来的部分可观测性问题。

IPPO-CKOR 方法中的智能体在与环境交互的过程中，对观测信息进行观测信息重构，得到重构观测信息并替代原始观测信息参与到策略的学习或执行中。执行阶段，演员中的策略网络使用重构观测信息作为输入，输出策略得到动作，并组成联合动作在环境中执行。策略学习阶段，演员和评论家中的策略网络和价值网络均以重构观测信息作为输入，使用输出信息和环境的反馈信息计算出网络的损失，然后通过反向传播对网络参数进行更新。IPPO-CKOR 方法中的策略网络和价值网络均使用了基于观测重构的独立学习网络结构，以实现对重构观测信息的高效处理。观测信息重构机制的加入，使得智能体能够更专注于关系密切智能体的特征信息，有效缓解了其他智能体带来的环境非平稳性问题。基于观测重构的独立学习网络实现了对重构观测信息的高效利用，同时通过对观测信息序列的深入处理，提取出更丰富的特征信息，从而有效缓解独立学习中的部分可观测性问题。

3.2 观测信息重构机制

如图 2 所示，观测信息重构机制包括共同知识特征计算与融合、智能体选择、重构特征生成和重构观测信息生成 4 个部分。共同知识特征计算与融合的作用是提取出观测信息 o_t 内共同知识信息的特征，得到融合共同知识特征的观测信息 m_t，将共同知识的特征信息融合进观测信息的同时，为后续的智能体选择提供了依据。智能体选择机制的作用是基于二阶共同知识丰富度和智能体间的距离对智能体排序，选择出与此智能体关系密切的其他智能体；重构特征生成的作用是构建被选中智能体的特征信息；重构观测信息生成的作用是将所构建的特征信息和融合共同知识特征的观测信息相结合，得到重构观测信息。

在进行观测信息重构时，首先，通过共同知识特征计算与融合方法，对原始观测信息中共同知识的特征信息进行提取与融合，得到融合观测信息；其次，基于在共同知识特征计算过程中得到的二阶共同知识丰富度，使用智能体选择机制对智能体排序，选择出关系密切的智能体；然后，使用重构特征生成机制为选择的智能体构建特征信息，组成重构特征信息

矩阵；最后，将重构特征信息矩阵与融合观测信息结合，得到重构观测信息，完成观测信息重构的全过程。

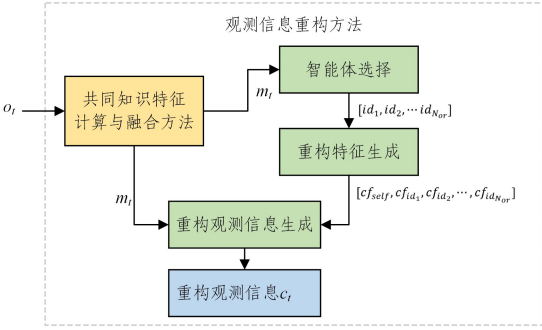


图 2 观测信息重构机制

Fig. 2 Observation information reconstruction mechanism

3.2.1 共同知识特征的计算与融合

为了有效地对智能体间共同知识特征进行计算和融合，基于前期工作中提出的 IPPO-CK 算法中的共同知识特征计算方法[20]，针对其共同知识值区分度不足且容易出现值相同的问题，在共同知识值的基础上，采用计算二阶共同知识丰富度的方式，更好地区分智能体之间共同知识的丰富程度，从而更好地反映智能体间关系的密切程度。二阶共同知识丰富度是盟友与敌方智能体共同知识值的加权和除以观测到的盟友和敌方智能体数量的加权和。具体地，智能体 i 观测信息内智能体 j 的二阶共同知识丰富度计算方式，如式(7)所示：

$$TSCKR_j=\frac{\omega\times\sum_{x=1}^{N^j_{al\_ck}}CKV_x+\sum_{y=1}^{N^j_{en\_ck}}CKV_y}{\omega\times N_{al\_seen}+N_{en\_seen}} \tag{7}$$

二阶共同知识丰富度值的范围为 $TSCKR_j\in[0,1]$，与共同知识值相同，值越高，则智能体 j 与智能体 i 的共同知识丰富度越高；值越低，则智能体 j 与智能体 i 的共同知识丰富度越低。在共同知识协作指数的计算方面，我们使用二阶共同知识丰富度替代原有的共同知识值，计算式如式(8)所示：

$$CKCI=\frac{\omega*\sum_{i_{al}=1}^{N_{al}}TSCKR_{i_{al}}+\sum_{i_{en}=1}^{N_{en}}TSCKR_{i_{en}}}{\omega\times N_{al}+N_{en}} \tag{8}$$

通过上述计算，在得到各个智能体的二阶共同知识丰富度和共同知识协作指数后，将它们与原始观测信息进行结合，进而得到融合共同知识特征的观测信息。融合方式与我们前期工作中提出的 IPPO-CK 算法[20]类似，即将二阶共同知识丰富度插入到观测信息内对应智能体的特征信息后方，而共同知识协作指数则插入到智能体自身特征信息的后方。同时，保存上述计算过程中得到的二阶共同知识丰富度用于后续的智能体选择。

3.2.2 智能体选择

智能体选择机制基于二阶共同知识丰富度对智能体关系密切程度进行区分，同时对智能体进行排序，选择出二阶共同知识丰富度高的智能体组$[id_1, id_2, \cdots, id_{N_{or}}]$作为关系密切的智能体，后续进行这些智能体特征信息的构建。当二阶共同知识丰富度值越高时，两个智能体观测信息内的共同知识越丰富，它们之间的关系也就越密切；同时，二阶共同知识丰富度拥有较好的区分度，因此能很好地区分智能体之间关系的密切程度。由于盟友、敌方智能体在决策中的角色不同，其关系密切程度的衡量标准也存在区别。为此，通过为盟友、敌方智能体的二阶共同知识丰富度设置权重，实现对盟友智能体和敌方智能体重要程度的区分。盟友智能体权重由重构盟友权重变量确定，敌方智能体权重设置为 1。在排序时，将加权的二阶共同知识丰富度作为排序的标准。为了应对少数可能出现的加权二阶共同知识丰富度值相同的情况，智能体选择机制使用智能体间的距离作为辅助，进行智能体间的排序。智能体选择机制对智能体进行关系密切程度排序的具体规则为：依据加权二阶共同知识丰富度值的大小从高到低将智能体排序，对于值相同的情况，依据智能体之间的距离从小到大排序。这样就能够在绝大多数情况下很好地区分智能体之间关系的密切程度，从排序的智能体序列头部开始选择智能体即可选出关系密切的智能体。对于少数该机制无法区分的情况，我们认为这些智能体的关系密切程度是几乎相同的，因此不再进行区分。

智能体选择机制算法 ASA(Agent Selection Algorithm)的伪代码描述如算法 1 所示，该算法基于二阶共同知识丰富度数组 *tsckrs*、智能体距离矩阵 ***dists***、重构数量 N_{or} 和观测信息 *obs*，计算得到需要构建特征的智能体的编号数组 *ids*。伪代码中，N_{ally} 为盟友智能体数量，N_{enemy} 为敌方智能体数量，*agent_id* 为进行观测重构智能体的编号，*rc_ally_weight* 为盟友重构权重。ASA 算法首先声明一个优先队列 q，该队列以一个大小为 2 的数组作为优先级，依据数组索引 0 位置上的值从小到大排序，当数组索引 0 位置上的值相等时，依据数组索引 1 位置上的值从小到大排序(行 1)；其次，遍历盟友和敌方智能体，为每个智能体创建一个大小为 2 的优先级数组，索引 0 位置的值是负的二阶共同知识丰富度与盟友或敌方重构权重的乘积，索引 1 位置的值是两个智能体之间的距离，以此作为优先级项，将该智能体的 *id* 添加到优先队列 q 中(第 2—7 行)；最后，从优先队列 q 中依次弹出 N_{or} 个智能体的编号，组成编号数组 *ids*(第 8—10 行)。

算法 1 智能体选择算法(Agent Selection Algorithm，ASA)

输入：tsckrs，dists，N_{or}，obs

输出：ids

```
1. 声明优先队列 q，空数组 ids
2. for i=0 to N_ally+N_enemy:
3.     if i!=agent_id
4.         if i<N_ally:
5. q.put([[tscks[i] * rc_ally_weight,dists[agent_id][i]],i])
6.         else:
7. q.put([[-tscks[i],dists[agent_id][i],i])
8. for rc_id=0 to N_or:
9. ids.append(q.pull())
10. return ids
```

3.2.3 重构特征生成

智能体重构特征生成的核心是分别为智能体自身和被选中的智能体构建特征信息，并堆叠为重构特征矩阵。智能体自身特征信息$cf_{self}=[feat_{self}, one_hot_{self}]$分为两部分：$feat_{self}$ 是观测信息中关于该智能体自身的部分，one_hot_{self} 是智能体自身编号的 one-hot 编码。被选中的智能体的特征信息

$cf_{id}=[feat_{id},one_hot_{id}]$同样分为两部分：$feat_{id}$是智能体特征信息，其具体信息与 SMAC 环境下的盟友与敌方智能体特征构成一致[14]；one_hot_{id}是该智能体编号的 one-hot 编码。在智能体特征信息中加入 one-hot 编码的作用是标志出该特征信息对应的智能体，用于网络识别。构建完各个智能体的特征后，将它们堆叠为重构特征信息矩阵 $\boldsymbol{cf}=[cf_{self\_id},cf_{id_1},cf_{id_2},\cdots,cf_{id_{N_{or}}}]$，其中自身特征信息位于第 0 行，其余智能体特征按照排序依次排列。

智能体重构特征生成算法 ARFGA（Agent Reconstruction Feature Generation Algorithm）的伪代码描述如算法 2 所示。该算法基于智能体编号 *id* 和智能体信息 *units*，计算得到智能体的重构特征 *feat*。首先，声明表示智能体特征信息的空数组 *feat*，然后在其尾部依次拼接可见或可攻击信息、距离信息、相对 X 坐标、相对 Y 坐标这 4 个信息（第 1－2 行）；其次，如果设定为可以看到智能体健康度，则在 *feat* 后方拼接健康度，当智能体有护甲时，再拼接护甲信息（第 3－6 行）；再次，如果表示智能体单元类型的位数大于 0，则在 *feat* 后拼接智能体的单元类型编码（第 7－8 行）；最后，拼接 onehot 编码后，返回重构完毕的智能体特征 *feat*（第 9－10 行）。

算法 2 智能体重构特征生成算法（ARFGA）

输入：id，units

输出：feat

1. 声明数组 feat
2. 在 feat 后依次拼接可见/可攻击信息、距离信息、相对 X 坐标、相对 Y 坐标 4 个信息
3. if obs_all_health：
4. 在 feats 后拼接智能体健康度信息
5. if shield_bits＞0
6. 在 feat 后拼接智能体护盾信息
7. if unit_type_bits＞ 0：
8. 在 feat 后拼接单元类型
9. 在 feat 后拼接智能体编号的 onehot 编码
10. return feat

3.2.4 观测信息重构

智能体观测信息重构的基础是共同知识特征的计算与融合，特别是计算过程中的二阶共同知识丰富度，为重构过程中关系密切智能体的选择提供了依据，同时，使用融合过程得到的融合观测信息代替原始观测信息。智能体观测信息重构的核心是依据二阶共同知识丰富度和智能体间的距离，选择 N_{or} 个智能体，为智能体自身和这些智能体构建特征信息，再由这些特征信息组成矩阵 ***agent_feat***，与融合共同知识特征的观测信息相结合，最终得到智能体的重构观测信息。

智能体观测信息重构算法 AORA（Agent Observation Reconstruction Algorithm）的伪代码描述如算法 3 所示。该算法基于智能体单元信息 *units*、智能体编号 *id*、重构数量 N_{or} 和观测信息 o_t，通过调用 ASA 和 ARFGA，完成观测信息的重构并得到重构观测信息 c_t。首先，调用共同知识特征计算与融合算法[20]得到融合观测信息 m_t，并声明空的重构特征矩阵 $\boldsymbol{cf}_t$（第 1－2 行）；其次，获取智能体之间的二阶共同知识丰富度并得到智能体间的距离矩阵（第 3 行）；然后，调用 ASA 得到被选中智能体的编号数组（第 4 行）；接着，调用 ARFGA 为智能体自身以及每个被选中智能体构建出特征信息，并拼接在一起得到 $\boldsymbol{cf}_t$（第 5－7 行）；最后，将融合共同知识特征的观测信息 o_t 与重构特征矩阵 $\boldsymbol{cf}_t$ 相结合，得到并返回重构观测信息 c_t（第 8－10 行）。

算法 3 智能体观测信息重构算法（AORA）

输入：units，id，N_{or}，o_t

输出：c_t

1. 调用共同知识特征信息计算与融合算法，获得融合共同知识特征的观测信息 m_t
2. 声明空矩阵 $\mathbf{cf}_t$
3. 从 m_t 中获得二阶共同知识丰富度 tsckrs、距离矩阵 **dists**
4. 调用 ASA 得到编号数组 ids
5. 调用 ARFGA 构建智能体 id 的特征信息并拼接到 $\mathbf{cf}_t$ 后
6. for i in ids：
7. 调用 ARFGA 重构智能体 i 的特征信息并拼接到 $\mathbf{cf}_t$ 后
8. $c_t=m_t$
9. 将 $\mathbf{cf}_t$ 展开为一维，拼接在 c_t 后方
10. return c_t

3.3 基于观测重构的独立学习网络

在获得智能体重构观测信息后，进一步使用 Conv1D、GRU、自注意力机制和全连接层等构建基于观测重构的独立学习网络。该网络以多个时间步的观测信息作为输入，通过 Conv1D 提取时序特征，以减少维度；使用 GRU 记忆对决策有参考价值的信息，保留当前时间步不存在的信息，提高决策的连续性、准确性和稳定性；使用多头自注意力机制处理智能体特征之间的关系，保留重要部分并得到这些特征信息之间的关联；最后，使用线性层进行融合与分类，得到网络的输出信息。

3.3.1 网络结构

图 3 给出了基于观测重构的独立学习网络的网络结构。基于观测重构的独立学习网络的输入信息是重构观测信息，由融合共同知识特征的观测信息和重构特征信息两部分组成，并且是当前及此前共 3 个时间步的信息。因此，在将信息输入到模型前，通过重构观测信息恢复模块将重构观测信息分离为两部分并进行处理：首先，将构建出的重构特征信息与融合观测信息 $\boldsymbol{m}_t$ 进行分离；其次，对于 $\boldsymbol{m}_t$，保留 3 个时间步的全部信息，得到 $[\boldsymbol{m}_{t-2},\boldsymbol{m}_{t-1},\boldsymbol{m}_t]$；最后，对于重构特征信息，仅保留当前时间步的信息，每个智能体特征信息为一行，恢复为重构特征信息矩阵 $\boldsymbol{cf}_t$。

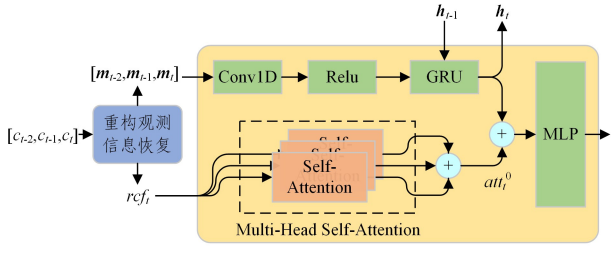


图 3 基于观测重构的独立学习网络结构图

Fig. 3 Schematic diagram of independent learning network based on observation reconstruction

基于观测重构的独立学习网络使用一个 Conv1D 层对

$[\boldsymbol{m}_{t-2}, \boldsymbol{m}_{t-1}, \boldsymbol{m}_t]$的时间步维度进行卷积处理，再使用 ReLU 激活函数对卷积层输出数据进行激活处理。Conv1D 的一维卷积核大小为 3，将时间步维度的信息降到一维，之后通过 squeeze 操作移除时间步维度。将 Conv1D 层经过 ReLU 进行处理后的输出数据与隐藏数据$\boldsymbol{h}_{t-1}$一起输入到 GRU 层，获得输出的隐藏信息$\boldsymbol{h}_t$。

在此基础上，将重构特征信息输入到多头自注意力机制中，对不同智能体的特征信息进行处理，得到 att_t；然后，仅保留输出信息中智能体自身的特征信息，即索引 0 上的信息 att_t^0，这部分信息包含了其他智能体特征信息对智能体自身的影响以及自身信息的重要性；最后，将 GRU 层的输出$\boldsymbol{h}_t$与多头自注意力的输出 att_t^0 连结，并将其作为 MLP 的输入数据，得到网络的最终输出。MLP 由若干全连接层构成，除最后一层外，每层的输出使用 ReLU 激活函数进行激活处理。

3.3.2 GRU 层

GRU 层的输入形式与一般的 RNN 一致[22-23]，以$\boldsymbol{x}_t$和上一节点的隐藏信息$\boldsymbol{h}_{t-1}$作为输入，输出$\boldsymbol{y}_t$和当前节点的隐藏信息$\boldsymbol{h}_t$，当前节点的隐藏信息$\boldsymbol{h}_t$会传递到下一节点作为输入的一部分。GRU 内部使用重置门控 $\boldsymbol{r}$ 和更新门控 $\boldsymbol{z}$ 实现信息的记忆与更新，它们的门控信号矩阵的计算方式如式(9)和式(10)所示：

$$\boldsymbol{z}_t=\sigma(\boldsymbol{W}^z\boldsymbol{x}_t+\boldsymbol{U}^z\boldsymbol{h}_{t-1}) \tag{9}$$

$$\boldsymbol{r}_t=\sigma(\boldsymbol{W}^r\boldsymbol{x}_t+\boldsymbol{U}^r\boldsymbol{h}_{t-1}) \tag{10}$$

依靠门控信号，先使用重置信号与$\boldsymbol{h}_{t-1}$做哈达玛积(Hadamard Product)，得到$\boldsymbol{h}'_{t-1}=\boldsymbol{h}_{t-1}\odot\boldsymbol{r}_t$，然后将$\boldsymbol{h}_{t-1}$与输入信息$\boldsymbol{x}_t$拼接，依据式(11)计算得到$\tilde{\boldsymbol{h}}_t$。

$$\tilde{\boldsymbol{h}}_t=\tanh(\boldsymbol{W}^h\boldsymbol{x}_t+\boldsymbol{U}^h\boldsymbol{h}'_{t-1}) \tag{11}$$

$\tilde{\boldsymbol{h}}_t$包括了当前的输入信息$\boldsymbol{x}_t$，将$\tilde{\boldsymbol{h}}_t$有选择性地加入到新隐藏状态，增加了当前时刻状态的记忆。最后，GRU 层利用更新门控进行记忆的更新，该更新融合了遗忘与记忆两个功能。式(12)给出了$\boldsymbol{h}_t$的具体计算式。

$$\boldsymbol{h}_t=(1-\boldsymbol{z}_t)\odot\boldsymbol{h}_{t-1}+\boldsymbol{z}_t\odot\tilde{\boldsymbol{h}}_t \tag{12}$$

其中，$(1-\boldsymbol{z}_t)\odot\boldsymbol{h}_{t-1}$部分实现了对上一节点隐藏信息的选择性遗忘，而$\boldsymbol{z}_t\odot\tilde{\boldsymbol{h}}_t$则实现了对当前节点中$\tilde{\boldsymbol{h}}_t$所包含信息的选择性记忆。将二者有机地结合起来，在一步操作中实现了记忆与遗忘两个功能，从而在基本保证与 LSTM 同等性能的前提下，有效减少了参数量与计算量。

3.3.3 多头自注意力机制

自注意力机制是注意力机制的变体，它减少了对外部信息的依赖，使得网络能够更好地捕捉不同输入向量之间的相互关联，了解向量之间的内在关联。在自注意力机制中，每一个输入到自注意力层的向量都会得到一个对应的输出向量，该输出向量包含了对应输入向量的信息，也包含了所有向量对该输入向量的影响。独立学习网络使用自注意力机制来处理重构之后的智能体特征信息，对智能体自身特征信息向量进行分析和重构，构建盟友和敌方智能体特征信息向量之间的内在关联，从而实现信息的有效融合。

多头自注意力机制(Multi-head Self-attention)是自注意力机制的进阶版本，由于信息之间的相关性具有不同形式，所以同时进行多个自注意力机制的计算，可以有效地适应这种不同形式的信息相关性，其计算过程如图 4 所示。首先，设置$\boldsymbol{W}_Q$，$\boldsymbol{W}_K$，$\boldsymbol{W}_V$ 3 个矩阵，将输入的特征矩阵 $\boldsymbol{F}$ 分别与 3 个矩阵相乘，得到 $\boldsymbol{Q}=\boldsymbol{W}_Q\boldsymbol{F}$，$\boldsymbol{K}=\boldsymbol{W}_K\boldsymbol{F}$ 和 $\boldsymbol{V}=\boldsymbol{W}_V\boldsymbol{F}$ 3 个矩阵；其次，由 $\boldsymbol{Q}$ 和$\boldsymbol{K}^{\mathrm{T}}$矩阵做点乘得到关系得分矩阵 $\boldsymbol{S}=\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}$，然后进行 softmax 操作，得到$\hat{\boldsymbol{S}}=\mathrm{softmax}(\boldsymbol{S})$；最后，由 $\boldsymbol{V}$ 矩阵和$\hat{\boldsymbol{S}}$点乘，得到 $\boldsymbol{V}\hat{\boldsymbol{S}}$。自注意力机制的计算式如式(13)所示：

$$Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\mathrm{softmax}(\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}})\boldsymbol{V} \tag{13}$$

在式(13)中，需要学习的是 $\boldsymbol{Q}$，$\boldsymbol{K}$ 和 $\boldsymbol{V}$ 这 3 个矩阵的参数。设计 h 组矩阵 $\boldsymbol{W}_{Q_i}$，$\boldsymbol{W}_{K_i}$ 和$\boldsymbol{W}_{V_i}$($i\in(1,\cdots,h)$)，对每一组矩阵，均采取上面的计算方式，将输入信息 $\boldsymbol{F}$ 与各组矩阵相乘得到h 组$\boldsymbol{Q}_i$，$\boldsymbol{K}_i$和$\boldsymbol{V}_i$矩阵。之后，各组矩阵分别计算各自的输出，将 $\boldsymbol{Q}_i$ 与 $\boldsymbol{K}_i^{\mathrm{T}}$ 做点，得到$\boldsymbol{S}_i=\boldsymbol{Q}_i\boldsymbol{K}_i^{\mathrm{T}}$，并使用 softmax 进行归一化，得到$\hat{\boldsymbol{S}}_i=\mathrm{softmax}(\boldsymbol{S}_i)$。其后，通过$\hat{\boldsymbol{S}}_i$和$\boldsymbol{V}_i$的点乘，得到输出 $\boldsymbol{head}_i=\hat{\boldsymbol{S}}_i\boldsymbol{V}_i$。最后，将得到的 h 个 $\boldsymbol{head}_i$连结起来，得到多头自注意力机制的输出信息，如式(14)所示：

$$MultiHead(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=Concat(\boldsymbol{head}_1,\cdots,\boldsymbol{head}_i) \tag{14}$$

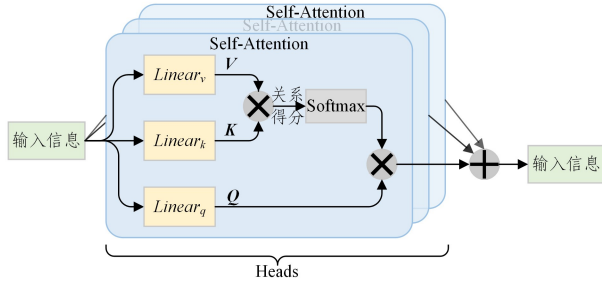


图 4 多头自注意力机制

Fig. 4 Schematic diagram of multi-head self-attention mechanism

3.4 算法描述

IPPO-CKOR 方法是在 2.1.2 小节 IPPO[20] 方法和 IPPO-CK 方法基础上进行改进优化，引入观测重构机制，并使用基于观测重构的独立学习网络进行实现的。与 IPPO 方法相比，IPPO-CKOR 方法具有如下两个方面的特点：一是采用基于共同知识的智能体观测重构方式，使独立学习的智能体更关注关系密切智能体的信息，缓解其他智能体带来的环境非平稳性问题；二是使用基于观测重构的独立学习网络来近似算法中的策略函数与价值函数，在高效利用重构观测信息的同时，提高了智能体对观测信息序列特征信息的提取能力，从而有效缓解了部分可观测性问题。与 IPPO 方法相比，在问题建模方面，IPPO-CKOR 方法在将问题建模为 Dec-POMDP 过程的基础上，增加了元组 C，智能体在获得观测信息后，调用 AROA 算法进行观测信息的重构，得到重构观测信息$c_t^i=C(m_t^i)$，并替代原始观测信息o_t^i作为观测信息参与到 Dec-POMDP 的过程中，用于智能体的策略的学习与执行。在损失计算方面，在 IPPO-CKOR 方法中，智能体 i 的优势估计与式(1)一致，但公式中δ_t^i的计算方式变为$\delta_t^i=r_t(c_t^i,a_t^i)+\gamma V_\phi(c_{t+1}^i)-V_\phi(c_t^i)$。式(15)给出了新的智能体 i 策略网络损失计算方式，式(16)为新的价值网络损失计算方式。

$$\mathcal{L}(\theta)=E_{c_t^i,a_t^i}\left[\begin{array}{l}\min\left(\dfrac{\pi_\theta(a_t^i\mid c_t^i)}{\pi_{\theta_{\text{old}}}(a_t^i\mid c_t^i)}A_t^i,\right.\\ \left.clip\left(\dfrac{\pi_\theta(a_t^i\mid c_t^i)}{\pi_{\theta_{\text{old}}}(a_t^i\mid c_t^i)},1-\epsilon,1+\epsilon\right)A_t^i\right)\end{array}\right] \tag{15}$$

$$\mathcal{L}^i(\phi)=E_{c_t^i}[\min\{(V_\phi(c_t^i)-\hat{V}_t^i)^2,(V_{\phi_{\text{old}}}(c_t^i)+clip(V_\varphi(c_t^i)-V_{\phi_{\text{old}}}(c_t^i),-\epsilon,+\epsilon)-\hat{V}_t^i)^2\}] \tag{16}$$

在式(16)中，$\hat{V}_t^i=A_t^iV_\phi+V_\phi(c_t^i)$。最终的总损失公式与式(4)相同。IPPO-CKOR 方法使用基于观测重构的独立学习网络作为方法中的策略函数和价值函数，保证网络在独立学习中的函数拟合能力。网络的具体结构由超参数网络结构与大小数组确定。网络结构与大小数组 0 位置上的数字规定了一维卷积和 GRU 层的神经元数量，而 0 位置之后的数字规定了 MLP 的层数与大小，0 位置之后的每一个数字对应了 MLP 中的一层。例如，当网络结构与大小参数为[128，128，128]时，一维卷积、GRU 层的大小为 128，而 MLP 层有两层全连接层，大小均为 128。因此，MLP 层的层数与大小可以根据地图而改变，以适应不同的地图难度。多头自注意力机制的头数和每一头自注意力机制的嵌入层大小分别由 *att_head* 和 *att_ebd* 两个参数规定。

IPPO-CKOR 方法的算法伪代码描述如算法 4 所示，该算法以经验库 D、最大环境时间步 MAX_STEP、采样数量 $BATCH$、环境信息 env_info 和环境交互回合数 N_{ep} 作为输入，学习并保存智能体的策略。算法的核心思想为通过循环进行环境交互和策略学习两个过程，逐渐学习到好的智能体策略。在环境交互阶段中，智能体从环境中获取观测信息并进行观测信息重构，然后将其输入到策略网络中，获取动作并在环境中执行，如此循环积累与环境的交互数据，用于后续的策略学习。每当累计进行一定数量的环境交互后，则算法执行策略学习阶段，从保存的交互数据中随机采样出一批数据进行策略学习，更新算法中的策略网络和价值网络。策略网络和价值网络均采用基于观测重构的独立学习网络结构构建。策略学习结束后，如尚未达到预设的环境步数，则继续执行环境交互阶段；如达到步数，则保存最终学习到的策略并终止算法。

IPPO-CKOR 算法的具体工作过程为：首先，进行一系列初始化操作，包含初始化采用基于观测重构的独立学习网络的策略网络、价值网络，以及计数器和环境等(第 1－2 行)；其次，当步计数器未达到最大时间步时，循环地进行环境交互和策略学习，逐步学习智能体的控制策略(第 3－24 行)；最后，存储学习到的策略并终止算法(第 25 行)。在第 3－24 行的循环体内，首先，智能体与环境交互 N_{ep} 个 episode，并将数据存入经验库中，在交互的过程中，智能体会调用 AORA 对从环境中获取的观测信息进行观测重构，用得到的重构观测信息代替原始观测信息(第 4－13 行)；其次，从经验库中随机采样一批数据用于策略网络和价值网络的参数学习(第 14 行)；最后，利用采样数据的每一步数据进行学习，依次计算出优势、策略网络损失、价值网络损失、熵损失，并基于式(4)计算出总损失，然后通过反向传播对价值网络和策略网络的参数进行更新(第 15－24 行)。

算法 4 IPPO-CKOR 算法

输入：D，MAX_STEP，BATCH，env_info，N_{ep}

输出：c_t

1. 初始化采用基于观测重构的独立学习网络结构的策略网络π_θ、价值网络V_φ，初始化步计数器 env_steps＝0、环境 env
2. n_ally ＝env_info. n_ally
3. while env_steps＜ MAX_STEP：
4. for e＝0 to N_{ep}：
5. 重置 env
6. t＝0
7. while not env. terminated()：
8. 从 env 获取智能体的观测$\{o_t^0,o_t^1,\cdots,o_t^{n_{ally}}\}$
9. 调用 AORA 进行观测信息重构，得到重构观测信息$\{c_t^0,c_t^1,\cdots,c_t^{n_{ally}}\}$
10. 基于策略网络π_θ，用ϵ-greedy 方式选择动作，组成联合动作并在环境中执行
11. t ＋＝1
12. 将回合数据存入 $\mathcal{D}$
13. env_steps ＋＝t
14. 从 $\mathcal{D}$ 中随机采样 BATCH 个的 episodes
15. for ep in episodes：
16. for t＝0 to ep. steps：
17. for i＝0 to n_ally：
18. 从 ep 中获取观测动作历史τ_t^i、策略π^i
19. 基于式(15)计算智能体的优势A_t^i
20. 基于式(16)计算策略网络损失 $\mathcal{L}(\theta)$
21. 基于式(17)计算价值网络损失$\mathcal{L}(\theta)$
22. 计算策略的熵 $\mathcal{H}(\pi^i)$
23. 基于式(4)计算总的损失 $\mathcal{L}(\theta,\phi)$
24. 进行反向传播，更新参数 θ 和 ϕ
25. 存储策略网络π_θ

4 实验验证

4.1 实验设置

为了验证本文提出的 IPPO-CKOR 方法的有效性，在星际争霸 SMAC 环境[14]中选择了多种不同类型的地图，设计了一系列的实验对 IPPO-CKOR 方法的有效性进行验证。

在实验过程中，每当算法运行累计达到 10 000 个时间步时，则暂停进行算法评估，智能体采用贪心方式，分布式地执行动作选择，运行 32 个测试回合。在测试回合中，以测试胜率作为算法的性能评价指标。

将智能体在时间限制内击败所有敌方单位的回合所占百分比记为测试胜率。在实验分析中，选取测试胜率的变化曲线图来体现不同算法的性能差异。每一次算法实验由多个独立训练重复进行，以保证获取到算法的平均性能表现。在实验结果图中，每个算法的学习曲线图展示的是测试胜率的平均数以及[25，75]百分位数。综合考虑计算代价与统计显著性，为每次实验运行 5 个独立的实验。除了对观测信息进行处理外，包括原始观测信息及奖励值在内的各项设定与 SMAC 环境原生设置保持一致，游戏的难度设置为最高难度 7。

4.2 对比实验

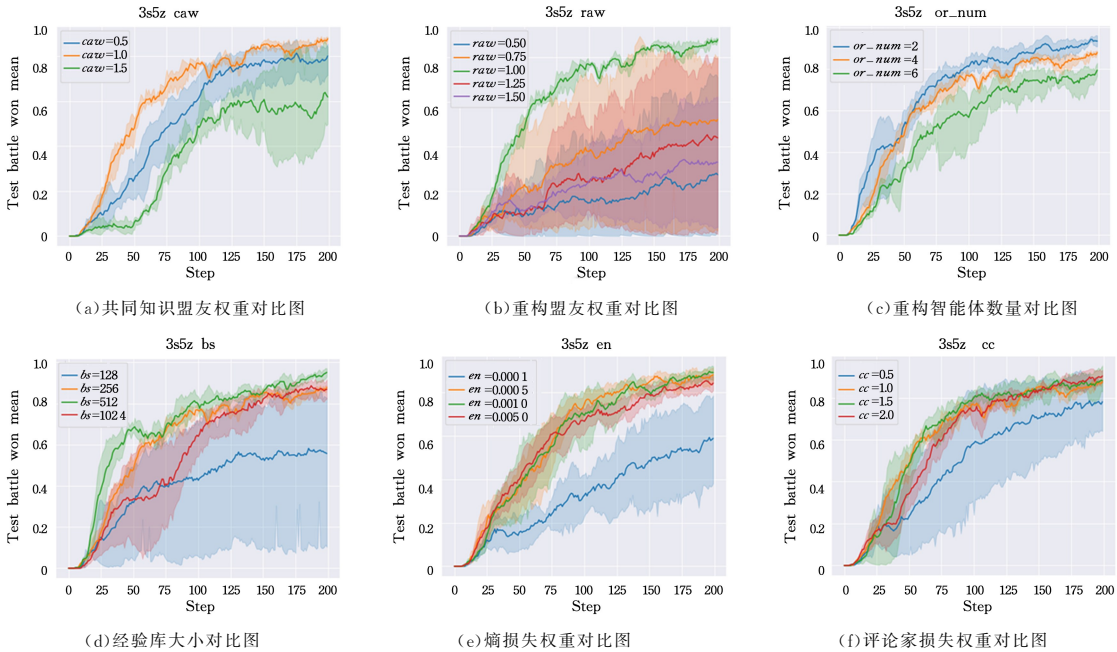
为了验证不同参数对 IPPO-CKOR 算法性能的影响，在 SMAC 环境中，在 3s5z 地图下进行不同参数的对比实验，探究主要超参数对 IPPO-CKOR 算法的性能影响。将需要对比的参数设定为：共同知识盟友权重 $caw\in[0.5,1,1.5]$，观测重构盟友权重 $raw\in[0.5,0.75,1,1.25,1.5]$，重构智能体数量 $or\_num\in[2,4,6]$，经验库大小 $buffer\_size(bs)\in[128,256,512,1024]$，熵损失的权重 $entropy(en)\in[0.0001,0.0005,0.001,0.005]$，以及价值网络损失权重 $critic\_coef(cc)\in[0.5,1,1.5,2]$。参数对比实验的目的是评估 IPPO-CKOR 算法在不同参数设置下的性能提升情况。

为了测试 IPPO-CKOR 算法的性能，将其与 IPPO，IQL 和 IPPO-CK 等目前具有代表性的算法进行了性能对比实验。在实验中，选择了 8m，3s5z，8m_vs_9m 和 10m_vs_11m 这 4 个地图，覆盖了具有 5～10 个己方智能体的同构对称、异构对称、同构非对称 3 种场景。同时，为了验证观测信息重构与其他算法的兼容性，还实现了 IQL-CKOR 算法，并进行算法的对比实验。该算法类似于 IPPO-CKOR 算法，为 IQL 算法增加了观测重构模块来处理观测信息，并使用基于观测重构的独立学习网络来近似 Q 函数。

为了验证 IPPO-CKOR 方法中观测信息重构部分的有效性，在 3s5z 地图上进行了消融实验。通过在 IPPO-CKOR 算法的基础上移除基于观测重构的独立学习网络，并使用 RNN 网络构成策略网络和价值网络，实现了 IPPO-CKOR-RNN 算法。该算法相较于 IPPO-CK 算法增加了观测信息重构，相较于 IPPO 算法增加了共同知识特征的计算与融合以及观测信息重构。

4.3 参数对比实验结果及分析

图 5 展示了 IPPO-CKOR 算法参数对比实验的学习曲线。共同知识盟友权重的作用是在共同知识特征计算与融合过程中，确定盟友智能体的重要性。从图 5(a)所示的共同知识盟友权重对比结果中可以看出，将共同知识盟友权重设置为 1 时，算法的性能表现最佳；而当该值设置为 0.5 时，算法性能表现略有下降；当值设置为 1.5 时，算法性能较差。由此可以得出，在 3s5z 地图中，共同知识盟友权重应设置为 1。重构盟友权重，是为了在观测重构过程中调整盟友智能体的重要性。从图 5(b)所示的重构盟友权重对比图中可以看出，权重的最优值为 1，此时算法性能表现最佳；而在其他值下，性能都明显下降。重构智能体数量规定了在观测信息重构过程中，选择并构建特征信息的智能体的数量。从图 5(c)所示的重构智能体数量对比图中可以看出，在值设置为 2 时，算法性能最佳；随着值上升到 4 和 6 时，算法的性能逐步下降。从图 5(d)所示的经验库大小对比图中可以看出，将经验库大小设置为 512 个 episode 时，算法的性能最佳；设置为 256 时，算法同样有较好的表现；而当设置为 128 和 1024 时，算法性能表现不佳。从图 5(e)所示的熵损失权重对比图中可以看出，在权重设置为 0.0005，0.001 和 0.005 时，算法都具有较好的表现；设置为 0.0001 时，性能不佳。从图 5(f)所示的评论家损失权重对比图中可以看出，权重值设置为 0.5 时，算法性能不佳；其他值下，算法均有较好的表现。



(a)共同知识盟友权重对比图 (b)重构盟友权重对比图 (c)重构智能体数量对比图

(d)经验库大小对比图 (e)熵损失权重对比图 (f)评论家损失权重对比图

图 5 参数对比实验结果图

Fig. 5 Diagram of parameter comparison experiment results

总的来看，从实验中我们可以发现，IPPO-CKOR 算法对 caw 和 raw 参数的要求较为严格，而对其他参数相对宽容，进行简单的参数尝试即可得到较好的性能表现。

4.4 性能对比实验结果及分析

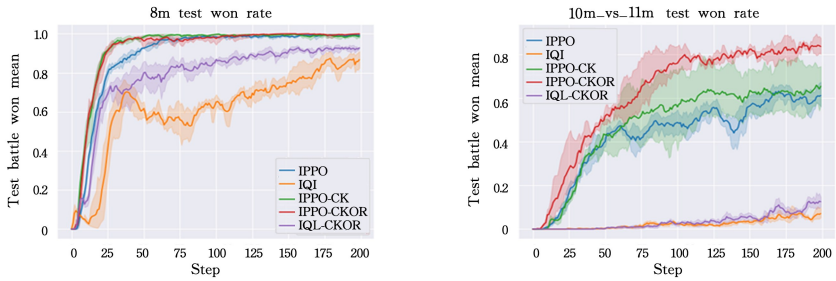
图 6 展示了本文提出的 IPPO-CKOR 算法与 IQL-CKOR，IPPO，IQL 和 IPPO-CK 等目前具有代表性的算法在 8m，10m_vs_11m，3s5z 和 8m_vs_9m 这 4 张地图下的对比实验结果，IPPO-CKOR 算法的参数如表 1 所列。图 6(a)展示了 8m 地图下的学习曲线，在这张简单地图中，IPPO 和 IPPO-CK 算法已经可以取得较好的效果。相较于 IPPO-CK 算法，

IPPO-CKOR 算法尽管在 100 万步之前没有展现出性能优势，但 100 万步之后，胜率曲线超过了其他所有算法，在训练结束时学习到了最好的策略，展现出了性能提升。同时，IQL-CKOR 算法相比于 IQL 算法有较大的性能提升，缩小了与 IPPO 算法的性能差距。图 6(b)和图 6(c)展示了 10m_vs_11m 和 3s5z 地图下的学习曲线。在这两个地图中，IPPO-CKOR 算法相较于 IPPO-CK 和 IPPO 算法都有显著的性能提升，从学习曲线上看，该算法从训练一开始便取得了性能优势，并且一直保持到训练的结束，最终收敛到了较高的胜率。同时，IQL-CKOR 算法相较于 IQL 算法有一定的性能提升，但是与 IPPO 等算法还存在较大的性能差距。图 6(d)展示了 8m_vs_9m 地图下的学习曲线。在该地图下，IPPO-CKOR 算法相较于 IPPO-CK 和 IPPO 算法均具有显著的性能提升，从训练一开始就在性能上取得领先，并最终收敛到了一个比较高的胜率上。IQL-CKOR 相较于 IQL 算法同样具有显著的性能提升，并且与 IPPO 算法的性能已经比较接近。

表 1 IPPO-CKOR 算法参数

Table 1 IPPO-CKOR parameters

地图名称	自注意力机制头数	学习率	评论家权重	熵损失权重	重构数量	共识盟友权重	重构盟友权重	经验库大小	网络结构与大小
8m	4	0.0001	1	0.0001	3	1	1	256	[128,128]
10m_vs_11m	2	0.0001	1	0.0001	5	1.5	1	256	[128,128,128]
3s5z	4	0.0001	1	0.0001	4	1.5	1	256	[128,128,128]
8m_vs_9m	2	0.0001	1	0.0005	4	1	1	256	[128,128]



(a)8m 地图 (b)10m_vs_11m 地图

(c)3s5z 地图 (d)8m_vs_9m 地图

图 6 算法对比实验结果

Fig. 6 Experiment results of algorithm comparison

综合来看，在上述实验结果中，IPPO-CKOR 算法相较于 IPPO-CK 和 IPPO 算法，在策略学习能力上均有普遍的提升，在一些地图中展现出了显著的性能提升；同时，IQL-CKOR 算法与 IQL 算法相比也展现出了性能的提升，这些结果证明了观测信息重构和基于观测重构的独立学习网络的有效性。

4.5 消融实验结果及分析

表 2 列出了消融实验算法构成。图 7 给出了 IPPO-CKOR 算法消融实验的结果。

表 2 消融实验算法构成

Table 2 Constituent elements of ablation experiment algorithms

算法	共同知识特征计算与融合	观测信息重构	基于观测重构的独立学习网络
IPPO			
IPPO-CK	√		
IPPO-CKOR-RNN	√	√	
IPPO-CKOR	√	√	√

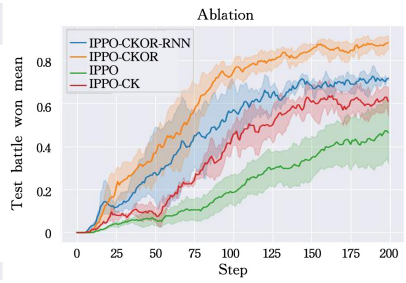


图 7 消融实验结果

Fig. 7 Ablation experiment results

我们在 IPPO-CKOR 算法的基础上，移除基于观测重构的独立学习网络，使用 IPPO 算法所用的 RNN 网络近似算法的策略网络和价值网络，得到了 IPPO-CKOR-RNN 算法。与 IPPO-CK 相比，该算法增加了观测信息重构，且网络结构更简单。与 IPPO 相比，该算法增加了共同知识特征计算与

融合以及观测信息重构。

从实验结果来看，相较于 IPPO-CK 和 IPPO 算法，IPPO-CKOR-RNN 均展现出了性能提升，这证明了观测信息重构部分的有效性。但是，由于重构观测信息带来的信息维度增大，智能体对重构观测信息的利用率较低，这种性能提升并不显著。而增加了基于观测重构的独立学习网络的 IPPO-CKOR 算法相较于 IPPO-CKOR-RNN 算法，在性能上有显著的提升，这证明了基于观测重构的独立学习网络的有效性，以及该网络对于重构观测信息的适用性，两者同时使用能够取得最好的效果。

结束语 针对独立学习的智能体缺少对关系密切的智能体的特征信息进行显式利用的问题，本文提出了一种基于观测重构的多智能体强化学习方法 IPPO-CKOR。一方面，通过基于二阶共同知识丰富度的智能体选择算法，选择关系密切的待重构智能体，然后构建出这些智能体的特征信息，与融合共同知识特征的观测信息一起组成重构观测信息，使得智能体能够基于关系密切的智能体组的特征信息进行决策，从而有效缓解了环境非平稳性问题。另一方面，基于观测重构的独立学习网络，使用多头自注意力机制对被选中的关系密切的智能体特征信息和自身特征信息进行处理，同时，使用一维卷积和 GRU 层对观测信息序列进行处理，实现对重构信息和观测信息序列的高效利用，有效缓解了部分可观测问题。实验结果表明，IPPO-CKOR 算法在所选地图的对比实验中展现出了显著的性能提升，在策略学习速度和最终学习到的策略两方面都有明显的提高。

参 考 文 献

[1] LI Y, XU F, XIE G Q, et al. Survey of development and application of multi-agent technology[J]. Computer Engineering and Applications, 2018, 54(9): 13-21.

[2] CLAUS C, BOUTILIER C. The dynamics of reinforcement learning in cooperative multiagent systems[C]// AAAI/IAAI. 1998.

[3] TAN M. Multi-agent reinforcement learning: Independent vs. cooperative agents[C]// Proceedings of theTenth International Conference on Machine Learning. 1993: 330-337.

[4] LOWE R, WU Y I, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[J]. Advances inNeural Information Processing Systems, 2017, 30: 6382-6393.

[5] YANG Y, LUO R, LI M, et al. Mean field multi-agent reinforcement learning[C]// International Conference on Machine Learning. PMLR, 2018: 5571-5580.

[6] SUNEHAG P, LE · VER G, GRUSLYS A, et al. Value-Decomposition Networks for Cooperative Multi-Agent Learning Based On Team Reward[C]// AAMAS. 2018.

[7] RASHID T, SAMVELYAN M, SCHROEDER C, et al. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning[C]// InternationalConference on Machine Learning. PMLR, 2018: 4295-4304.

[8] KUBA J G, CHEN R, WEN M, et al. Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning[C]// International Conference on Learning Representations. 2021.

[9] TAMPUU A, MATIISEN T, KODELJA D, et al. Multiagent cooperation and competition with deep reinforcement learning[J]. PloS one, 2017, 12(4): e0172395.

[10] DE WITT C S, GUPTA T, MAKOVIICHUK D, et al. Is independent learning all you need in the starcraft multi-agent challenge? [J]. arXiv: 2011.09533, 2020.

[11] CHRISTIANOS F, SCHÄFER L, ALBRECHT S. Shared experience actor-critic for multi-agent reinforcement learning[J]. Advances in Neural Information Processing Systems, 2020, 33: 10707-10717.

[12] OSBORNE M J, RUBINSTEIN A. A course in game theory[M]. MIT press, 1994.

[13] SCHROEDER DE WITT C, FOERSTER J, FARQUHAR G, et al. Multi-agent common knowledge reinforcement learning[C]// Neural Information Processing Systems. 2019.

[14] SAMVELYAN M, RASHID T, DE WITT C S, et al. The starcraft multi-agent challenge[J]. arXiv: 1902.04043, 2019.

[15] KAELBLING L P, LITTMAN M L, MOORE A W. Reinforcement learning: A survey[J]. Journal ofArtificial Intelligence Research, 1996, 4: 237-285.

[16] SCHULMAN J, MORITZ P, LEVINE S, et al. High-dimensional continuous control using generalized advantage estimation[J]. arXiv: 1506.02438, 2015.

[17] HALPERN J Y, MOSES Y. Knowledge and common knowledge in a distributed environment[J]. Journal of the ACM(JACM), 1990, 37(3): 549-587.

[18] NAYYAR A, MAHAJAN A, TENEKETZIS D. Decentralized stochastic control with partial history sharing: A common information approach[J]. IEEE Transactions on Automatic Control, 2013, 58(7): 1644-1658.

[19] GUESTRIN C, VENKATARAMAN S, KOLLER D. Context-specific multiagent coordination and planning with factored MDPs[C]// AAAI/IAAI. 2002: 253-259.

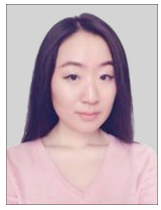
[20] HU H, SHI D, YANG H, et al. Independent Multi-agent Reinforcement Learning Using Common Knowledge[C]// 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2022: 2703-2708.

[21] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. arXiv: 1706.03762, 2017.

[22] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[23] CHUNG J, GULCEHRE C, CHO K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv: 1412.3555, 2014.

SHI Dianxi, born in 1966, Ph.D, professor, Ph.D supervisor. His main research interests include distributed object middleware technology, adaptive software technology, artificial intelligence and robot operation systems.



CHEN Ying, born in 1985, Ph.D, assistant research fellow. Her main research interests include artificial intelligence algorithm and framework design and optimization.

（责任编辑：柯颖）