面向 Ad-Hoc 协作的局部观测重建方法*

陈皓 [1,2] 杨立昆 [1,2] 尹奇跃 [1,2] 黄凯奇 [1,2,3†]

（1 中国科学院自动化研究所智能系统与工程研究中心 北京 100190；2 中国科学院大学人工智能学院 北京 100049；3 中国科学院脑科学与智能技术卓越创新中心 上海 200031）

（2022 年 3 月 2 日收稿；2022 年 4 月 1 日收修改稿）

陈皓，杨立昆，尹奇跃. 面向 Ad-Hoc 协作的局部观测重建方法[J]. 中国科学院大学学报，DOI:10. 7523/j.ucas. 2022. 028.

摘　要　近年来，多智能体强化学习得到了研究人员们的广泛关注。在多智能体强化学习的研究中，如何进行 Ad-Hoc 协作，也就是说如何适应种类和数量变化的队友，是一个关键问题。现有方法或者有很强的先验知识假设，或者使用硬编码的规则来进行合作，缺乏通用性，无法泛化到更一般的 Ad-Hoc 协作场景。为解决该问题，本文提出了一种面向 Ad-Hoc 协作的局部观测重建算法，利用注意力机制和采样网络对局部观测进行重建，使得算法认识到并充分利用不同局面中的高维状态表征，实现了在 Ad-Hoc 协作场景下的零样本泛化。本文在星际争霸微操环境和 Ad-Hoc 协作场景上与代表性算法的性能进行对比与分析，验证了算法的有效性。

关键词　多智能体；深度强化学习；信用分配；Ad-Hoc 协作

中图分类号：TP183　　**文献标志码：**A　　**DOI:**10. 7523/j.ucas. 2022. 028

Local Observation Reconstruction for Ad-Hoc Cooperation

CHEN Hao[1,2], YANG Likun[1,2], YIN Qiyue[1,2], HUANG Kaiqi[1,2,3]

（1 *CRISE, Institute of Automation, Chinese Academy of Sciences, Beijing* 100190, *China*；

2 *School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing* 100049, *China*；

3 *CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai* 200031, *China*）

Abstract　In recent years, multi-agent reinforcement learning has received a lot of attention from researchers. In the study of multi-agent reinforcement learning, the question of how to perform ad-hoc cooperation, i.e., how to adapt to a changing variety and number of teammates, is a key problem. Existing methods either have strong prior knowledge assumptions or use hard-coded protocols for cooperation, which lack generality and can not be generalized to more general ad-hoc cooperation scenarios. To address this problem, this paper proposes a local observation reconstruction algorithm for ad-hoc cooperation, which uses attention mechanisms and sampling networks to reconstruct local observations, enabling the algorithm to recognize and make full use of high-dimensional state representations in different situations and achieve zero-shot generalization in ad-hoc cooperation scenarios. In this paper, the performance of the algorithm is compared and analyzed with representative algorithms on the StarCraft micromanagement environment and ad-hoc cooperation scenarios to verify the effectiveness of the algorithm.

Keywords　Multi-Agent；Deep Reinforcement Learning；Credit Assignment；Ad-Hoc Cooperation

在现实生活中，多智能体系统(Multi-agent System, MAS)无处不在，比如电力分配网络[1]，自动驾驶车辆[2]，传感器网络[3]等。强化学习是处理这些多智能体问题的常用方法。然而相比单智能体强化学习，多智

* 国家自然科学基金（61876181），北京市科技创新计划（Z19110000119043），青年创新促进会、中国科学院和中国科学院项目（QYZDB-SSWJSC006）资助

† 通信作者, E-mail: kqhuang@nlpr.ia.ac.cn

能体强化学习面临着环境非平稳，信用分配，Ad-Hoc 协作等独特的挑战。

为了应对上述挑战，多智能体强化学习(Multi-agent Reinforcement Learning, MARL)在近年来取得了以中心化训练分布式执行(Centralized Training with Decentralized Execution, CTDE)框架[4-5]为代表的一系列突破性进展[6-12]。在 CTDE 训练范式中，智能体在中心化训练阶段可以使用全局信息，在分布式执行阶段只能依赖局部观测信息进行决策，也就是满足了部分可观测的限制。在 CTDE 框架下，以 QMIX[8]为代表的价值函数分解方法通过信用分配网络把每个智能体的本地 Q 值组合成全局 Q 值，更好地评价了每个智能体的贡献，实现了更好的合作，在很多极具挑战性的任务上取得了良好的效果。

然而在很多实际的应用场景中，智能体需要在测试的时候与从未见过的队友进行协作，比如不同种类，数量的队友。这被称为 Ad-Hoc 协作问题[13]。在 CTDE 框架下，现有的基于信用分配的方法往往不能利用由于不同队友设定带来的变化的输入信息，从而导致算法无法在 Ad-Hoc 协作场景下取得较好的性能。

针对上述问题，本文提出了一种面向 Ad-Hoc 协作的局部观测重建算法，首先把智能体的局部观测信息分解为三个部分，然后利用注意力机制处理长度变化的输入信息，使得算法对数量变化的智能体输入不敏感，最后利用采样网络实现了局部观测抽象，使得算法认识到并充分利用不同局面中的高维状态表征。最终实现了对每个智能体的局部观测信息的重建，使得算法可以在 Ad-Hoc 协作场景下进行零样本泛化。本文在星际争霸微操环境(StarCraft Multi-Agent Challenge, SMAC) [14]和 Ad-Hoc 协作场景上对算法性能进行了验证，实验结果表明，在 1c3s5z 等简单地图，5m_vs_6m 等困难地图，极度困难地图 MMM2，以及 Ad-Hoc 协作场景上，算法取得了超越现有代表性算法的性能，学到了更好的协作策略。

本文贡献包括如下两方面内容：(1)提出了一种面向 Ad-Hoc 协作的局部观测重建算法，通过注意力机制和采样网络对局部观测进行了重建，使得算法认识到并充分利用不同局面中的高维状态表征，有效提升了算法在 Ad-Hoc 协作场景下的零样本泛化能力；(2)在星际争霸微操环境和 Ad-Hoc 协作场景上与代表性算法的性能进行对比与分析，并进行消融实验，验证了算法的有效性。

本文内容安排如下：第 1 节介绍问题定义与相关工作，第 2 节详细介绍面向 Ad-Hoc 协作的局部观测重建算法，第 3 节介绍实验设计与结果分析，最后总结本文内容并展望未来的研究方向。

1 问题定义与相关工作

1.1 问题定义

本文研究对象为完全合作的多智能体任务。而分布式部分可观测马尔科夫决策过程(Decentralized Partially Observable Markov Decision Process, Dec-POMDP) [15]可以用于建模该问题。一个 Dec-POMDP 可以由一个元组$G = \langle S, U, P, r, Z, O, n, \gamma \rangle$定义的随机过程来描述，其中$s \in S$为环境的真实状态。$n$个智能体$i \in I \equiv \{1, \dots, n\}$在每个时间步与环境交互并同时选择一个动作$u^i \in U$。将这些动作联合起来形成所有智能体的联合动作$u \in U \equiv U^n$，同时也蕴含了环境的状态转移方程：$P(s'|s,u): S \times U \times S \to [0,1]$。各个智能体都拥有共同的奖励函数$r(s,u): S \times U \to R$，其中$\gamma \in [0,1)$是折扣因子。本文研究的是一个部分可观测情况，即各智能体得到自己的观测$z \in Z$，都需通过观测函数$O(s,i): S \times I \to Z$。每个智能体都拥有自己的动作-观测历史$\tau^i \in T \equiv (Z \times U)^*$，并利用它生成随机策略$\pi^i(u^i|\tau^i): T \times U \to [0,1]$。定义累积折扣回报为$R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+1}$。智能体的联合策略动作值函数与累积折扣回报函数相关，并定义为$Q^\pi(s_t, u_t) = E_{s_{t+1:\infty}, u_{t+1:\infty}}[R_t|s_t, u_t]$。

1.2 多智能体深度强化学习

深度强化学习在近几年取得了很大的进展[16-17]，

其中 DQN 算法[18-19]是一个具有代表性的成果。这些单智能体强化学习算法的突破，也激励着研究者们将深度强化学习应用在多智能体领域中[20-22]。 一个显而易见的想法是将单智能体算法直接迁移至多智能体问题上，但这种直接的迁移并没有收获良好的效果[23]。原因在于，在多智能体的设定下，智能体的决策环境不满足马尔科夫性质。即各个智能体的策略都在不断变化，导致单个智能体在决策过程中观测到的环境在不断变化。这种问题被称为环境非平稳问题。正是该问题使得单智能体强化学习算法在多智能体问题中失去了理论保证。除了环境非平稳问题，在多智能体强化学习领域中还存在着诸多具有挑战性的问题，比如部分可观测问题，信用分配问题，状态-动作空间指数爆炸问题等等。在多智能体领域中，学者们提出了诸多方法以应对上述问题，其中具有代表性的是行为分析[24-27]，学习通信[28-32]，学习合作[33-34]，对手建模[35-37]四类方法。这些方法有地效促进了多智能体深度强化学习的发展。

1.3 中心化训练分布式执行

在强化学习中，动作-值函数的优异程度很大程度上决定整个算法的效果。在多智能体系统中，学习最优的动作-值函数也是一个重要的问题。当智能体数目较少时，完全中心化的算法一般具有较好的性能，然而随着智能体数目的增多，完全中心化是算法引起了状态-动作空间指数爆炸的问题。而完全去中心化的算法，比如独立 Q 学习算法(Independent Q Learning, IQL) [23]，虽然可以解决上述问题，但是将其他智能体建模成环境的方法却导致了环境非平稳的问题，从而难以获得较好的性能。

综上所述，完全中心化和完全去中心化都不能得到良好的效果，所以学者们提出了中心化训练分布式执行框架，如图 1 所示。在 CTDE 框架的训练过程中，算法可以获取所有智能体的动作-观测历史τ和全局状态s，而在测试过程中，每个智能体只根据自己的动作-观测历史τ^i来进行决策。这种方法缓解了环境非平稳和状态-动作空间指数爆炸的问题并在多智能体深度强化学习中得到了广泛应用[6-9]，并成为了目前学界使用的主流框架。

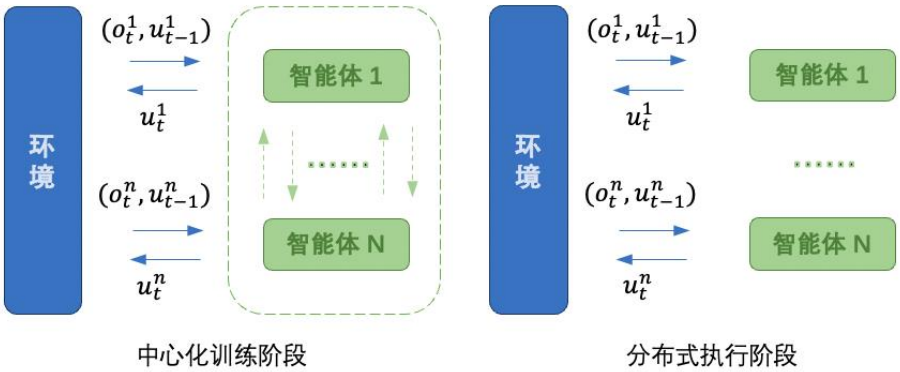


图 1 中心化训练分布式执行框架下多智能体与环境交互

Fig. 1 Multi-agent interaction with environment under CTDE framework

1.4 基于值分解的信用分配方法

基于值分解的方法是实现智能体之间信用分配的一种常用方法，也是多智能体深度强化学习研究中的一个热点。

在基于值分解的方法中，最优的联合动作等于最优个体动作的集合，是个体-全局最优条件(Individual-Global-Max, IGM) [9] 中定义的最优的个体动作和最优的联合动作之间应该满足的关系。使用

符合 IGM 条件的基于值分解的方法可以使算法较快收敛并取得更好的性能。其中，IGM 条件可以被表示为：

$$\arg\max_{u} Q_{\text{tot}}(\tau,u)=\begin{pmatrix}\arg\max_{u^1} Q_1(\tau^1,u^1)\\ \vdots \\ \arg\max_{u^n} Q_n(\tau^n,u^n)\end{pmatrix}. \quad (1)$$

VDN 算法[7]假设联合动作值函数$Q_{\text{tot}}(\tau,u)$是每个智能体的动作值函数$Q_i(\tau^i,u^i)$的简单加和，通过如下所示的加性约束满足了 IGM 条件。

$$Q_{\text{tot}}(\tau,u)=\sum_{i=1}^{n} Q_i(\tau^i,u^i). \quad (2)$$

QMIX 算法假设联合动作值函数$Q_{\text{tot}}(\tau,u)$满足单调性约束，进而满足 IGM 条件。因此实现了对联合动作-值函数的分解. 具体限制条件如下所示：

$$\frac{\partial Q_{\text{tot}}(\tau,u)}{\partial Q_i(\tau^i,u^i)}\geq 0,\ \forall i\in\{1,\dots,n\}. \quad (3)$$

QMIX 中的单调性约束是指随着某个智能体的$Q_i(\tau^i,u^i)$增大，$Q_{\text{tot}}(\tau,u)$也随之会增大，反之亦然。同时为了使值分解网络具有单调性以满足 IGM 条件，QMIX 限制了值分解网络的权重非负。

QTRAN 算法[9]把原有的联合动作-值函数转化为新的可分解的联合动作-值函数以实现更广泛的值分解。QTRAN 算法移除了 VDN 算法和 QMIX 算法中的结构化约束，并使转化前后的联合动作-值函数具有相同的最优动作。上述过程同样带来了算法复杂度增加的问题，使得算法整体性能略有降低。

1.5 Ad-Hoc 协作

Ad-Hoc 协作是多智能体领域中的一个极具挑战性的研究问题，在近年来获得了学界大量的关注[38-40]。在 Ad-Hoc 协作问题中，智能体需要学会如何跟变化的队友合作，包括种类上的变化，数量上的变化等。也就是说，在训练和测试的时候，智能体会面对完全不同，甚至从未见过的队友设置，并且需要在这种情况下取得较好的性能。前人的方法或者有很强的先验知识假设[41-42]，或者使用硬编码的规则来进行合作[43-44]，都具有很多的限制，从而无法泛化到更一般的 Ad-Hoc 协作场景。本文提出的算法可以实现在 Ad-Hoc 协作场景下的零样本泛化，在从未见过的队友设置上取得较好的效果。

2 面向 Ad-Hoc 协作的局部观测重建算法

本节详细介绍面向 Ad-Hoc 协作的局部观测重建算法 LOBRE(Local Observation Reconstruction for Ad-Hoc Cooperation)。传统的信用分配算法只能接受固定长度的输入数据，无法适应数量变化的队友，同时，传统的信用分配算法没有针对 Ad-Hoc 协作问题重建智能体的局部观测，导致无法充分利用局部观测信息，从而无法取得很好的 Ad-Hoc 协作能力。为解决该问题，本文提出的 LOBRE 算法通过局部观测重建网络对智能体接收到的局部观测进行了重构，从而实现了在 Ad-Hoc 协作场景下的零样本泛化。本节结构如下：首先介绍局部观测重建网络，之后介绍局部观测重建网络中的采样网络，最后介绍算法的总体优化目标。

2.1 局部观测重建网络

为了实现在 Ad-Hoc 协作场景下的零样本泛化，本文对智能体的局部观测网络进行重建，一方面可以适应数量变化的队友，另一方面使得算法认识到并充分利用不同局面中的高维状态表征，有利于提升算法在 Ad-Hoc 协作场景下的性能。

本文所提的 LOBRE 算法的网络结构如图 2 所示，在中心化训练的过程中，本文所提的 LOBRE 算法首先接收智能体的局部观测信息作为智能体决策网络的输入，输出每个智能体的 Q 值。然后通过信用分配网络处理每个智能体的 Q 值进而得到$Q_{\text{tot}}(\tau,u)$。智能体决策网络由局部观测重建网络，门控循环单元(Gated Recurrent Unit, GRU)和线性层组成。

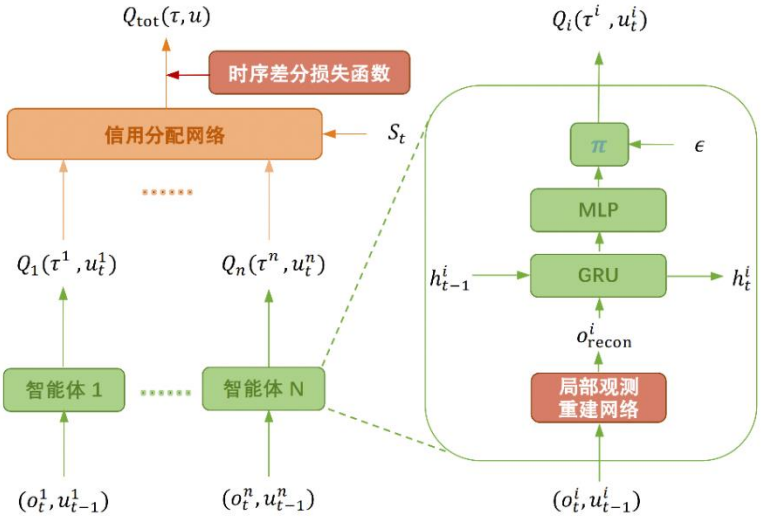


图 2 LOBRE 算法的网络结构

Fig. 2 Network structure of LOBRE algorithm

局部观测重建网络的具体网络结构如图 3 所示，在局部观测重建网络中，智能体的局部观测首先被分为了三个部分，分别为：该智能体自己的特征O_{own}，与智能体数目有关的特征O_{var}和与智能体数目无关的特征O_{inv}。然后，本文使用缩放点积注意力来处理该智能体自己的特征O_{own}和与智能体数目有关的特征O_{var}，如下式所示：

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V}. \tag{4}$$

其中，$\boldsymbol{Q}$，$\boldsymbol{K}$，$\boldsymbol{V}$分别对应O_{own}，O_{var}，O_{var}。

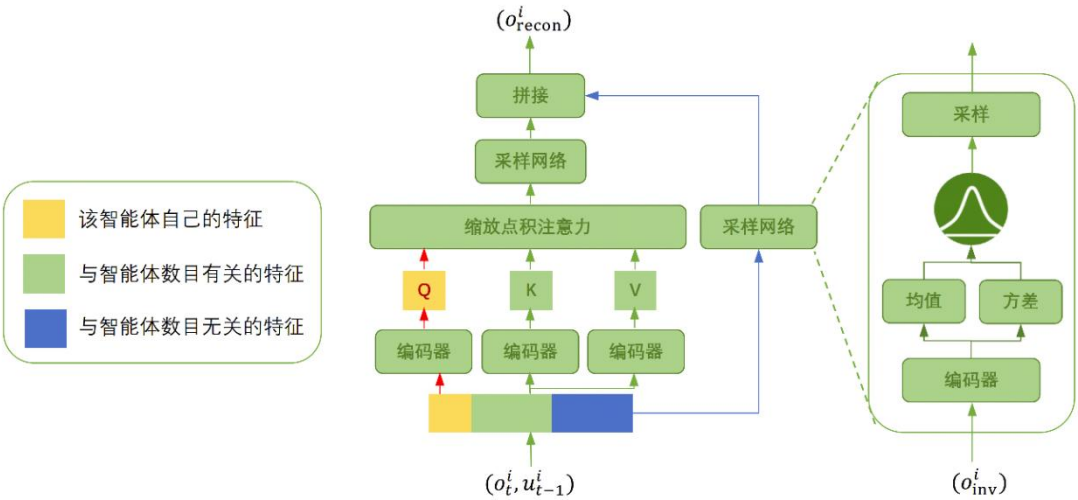


图 3 局部观测重建网络的结构

Fig. 3 The structure of local observation reconstruction network

然后，本文使用采样网络来处理缩放点积注意力的输出信息，以及与智能体数目无关的特征O_{inv}，本文会在下一节详细介绍关于采样网络的结构设计。最后，本文把从采样网络中输出的两个信息矩阵拼接起来得到经过重建的局部观测信息O_{recon}，并将其作为局部观测重建网络的输出。该输出O_{recon}会作为重建后的局部观测信息输入到智能体后续的决策网络中用于智能体的决策。

2.2 采样网络

本节详细介绍局部观测重建网络中的采样网络。

本文所提的 LOBRE 算法通过采样网络实现了局部观测抽象，使得算法认识到并充分利用不同局面中的高维状态表征。局部观测重建网络包含两个采样网络，结构相同，本文以图 3 右侧的采样网络为例进行阐述。如图 3 所示，采样网络首先通过编码器f把输入信息O_{inv}编码为均值μ和方差σ，如下式所示：

$$(\mu,\sigma)=f(O_{\mathrm{inv}}). \tag{5}$$

之后利用该均值和方差构建多维高斯分布$N(\mu,\sigma)$。进而从中采样得到输出信息：

$$O_{\mathrm{out}}\sim N(\mu,\sigma). \tag{6}$$

由于均值μ和方差σ都是通过编码器f得到的，因此该多维高斯分布$N(\mu,\sigma)$是可以被学习的。同时，为了保证从该多维高斯分布中采样的过程是可以被神经网络学习的，本文在对该多维高斯分布进行采样的过程中采用了重参数化技巧。

由于本文所提的 LOBRE 算法没有改变信用分配网络的结构，由(3)式可知，本文所提的 LOBRE 算法仍然满足 IGM 条件，也就是说最优的联合动作等于最优个体动作的集合。为了证明加入的两个采样网络的有效性，本文在实验部分进行了验证，证明了加入采样网络之后再进行拼接得到经过重建的局部观测信息O_{recon}相比直接拼接得到O_{recon}效果更好。

2.3 总体优化目标

本文算法框架的总体优化目标是 DQN 算法中的时序差分误差（TD-error），如下式所示：

$$L_{\mathrm{TD}}(\theta)=\sum_{i=1}^{\mathrm{b}}\left[\left(y_i^{\mathrm{tot}}-Q_{\mathrm{tot}}(\tau,u;\theta)\right)^2\right]. \tag{7}$$

$$y^{\mathrm{tot}}=r+\gamma\max_{u'}Q_{\mathrm{tot}}(\tau',u';\theta^-). \tag{8}$$

其中，b是从经验回放池中采样得到的一个批次样本的数量，θ是网络的参数，θ^-是目标网络的参数，目标网络参数θ^-通过周期性地拷贝θ来进行更新。更新周期为 200 个时间步。

本文所提的 LOBRE 算法使用中心化训练分布式执行框架进行端到端训练从而对上述总体优化目标进行优化。在中心化训练阶段，网络参数共享，同时信用分配网络可以获取全局状态信息s和联合动作-观测历史τ。在分布式执行阶段，智能体不能使用信用分配网络，只能使用本地的动作-观测历史τ^i进行决策。

3 实验设计与结果分析

本节结构如下：首先，介绍实验环境设置，之后，介绍基线算法和超参数设置，最后，在星际争霸微操环境和 Ad-Hoc 协作场景上与代表性算法的性能进行对比与分析，并进行消融实验，证明算法的有效性。

3.1 实验环境设置

本文所采用的算法性能测试环境是星际微操环境。星际微操环境是一个标准的用于检验多智能体决策能力的复杂环境，是当今验证多智能体算法性能的一个重要环境。它具有状态-动作空间大，部分可观测，环境非平稳等特点，复杂性强，挑战性高，是学术界常用的算法验证环境。图 4 展示了在地图 MMM2 上的对战画面。

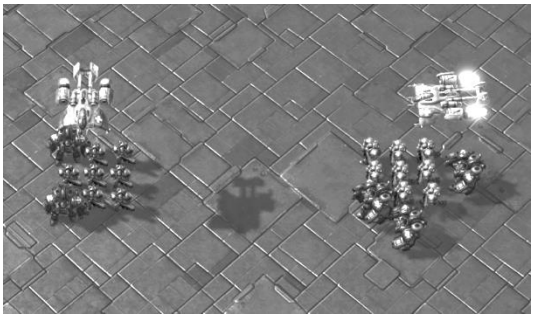


图 4 地图 MMM2 的对战画面
Fig. 4 Combat scenario of map MMM2

微操是指对每个智能体分别进行控制以达到击败对手的目的。随着智能体数目的增多，这种控制形式即可以被建模为多智能体问题。具体在星际微操环境中，智能体动作空间是离散的，具体包括四方向移动，攻击目标，停止和不执行任何动作。其中，不执行任何动作单独适用于阵亡的智能体。星际微操环境禁用了星际争霸原有的移动-攻击指令，彻底分离了移动和攻击指令，使得智能体必须在每个时间步都进行动作决策，增加了决策的复杂度。

每个智能体都有自身的局部观测，其观测范围局限于以之为中心的圆形范围内，并且无法得到观测范围以外的信息。具体地说，智能体的观测可以得到智能体当前位置，智能体之间的相对位置和观测范围内智能体的血量和护甲这三类信息。全局观测信息由各智能体的局部观测信息结合得到。为保证智能体在观测到敌人后需要移动一段距离才可以进行攻击，每个智能体的观测范围都被设置为九，攻击范围都被设置为六。

智能体在每个时间步都会接收到环境返回的全局奖励。全局奖励具体奖励设置为：在对敌方智能体造成的伤害的基础上减去我方智能体受到的伤害的一半，每杀死一个对手都会将十点奖励额外给予智能体，赢得游戏会将当前团队剩余生命值的总和外加两百点的奖励给予智能体。

算法运行过程中，每隔两千个时间步运行二十局对战作为测试，并统计当前时间点上的胜率，最终绘制胜率-时间步的曲线图。本文使用的是基于星际争霸 SC2.4.10(B75689) 游戏版本的 SMAC 环境。在 Ad-Hoc 协作场景的实验中，本文使用一张地图训练算法，使用另一张地图来测试算法性能，比如 5m-5ma 表示在 5m 地图上进行训练，在 5ma 地图上进行测试。

本文在算法训练和测试的过程中使用了多种星际争霸微操环境的地图，包括从简单，困难，到极度困难不同难度的地图，以及同构，异构，对称，非对称多种设置的地图，具体如表 1 所示。其中，狂热者，跳虫是近战单位。掠夺者，海军陆战队员，巨像，追猎者是远程单位。医疗运输机无法进行攻击，但是具有治疗友方单位的能力。

表 1　本文使用的地图

Table 1 Maps used in this paper

地图名称	友方单位	敌方单位	特点
1c3s5z	一个巨像，三个追猎者，五个狂热者	一个巨像，三个追猎者，五个狂热者	异构，对称
2s3z	两个追猎者,三个狂热者	两个追猎者,三个狂热者	异构，对称
3s2z	三个追猎者,两个狂热者	三个追猎者,两个狂热者	异构，对称
5m_vs_6m	五个海军陆战队员	六个海军陆战队员	同构，非对称
6m_vs_6m	六个海军陆战队员	六个海军陆战队员	同构，对称
5m	五个海军陆战队员	五个海军陆战队员	同构，对称
5ma	五个掠夺者	五个掠夺者	同构，对称
MMM2	一架医疗运输机，两个掠夺者，七个海军陆战队员	一架医疗运输机，三个掠夺者，八个海军陆战队员	异构，非对称

本文在上述每张星际争霸微操地图上都使用了五个不同的随机种子运行两百万个时间步进行实验并画出了算法的测试胜率-时间步曲线图，同时用阴影表示了 25%到 75%分位数的结果。本文实验所使用的显卡为 NVIDIA TITAN RTX GPU 24G。

3.2 基线算法和超参数设置

本文的基线算法为基于值分解的多智能体信用分配算法 VDN，QMIX，QTRAN 以及独立 Q 学习算法 IQL，如表 2 所示。上述算法均使用了星际争霸微操环境 SMAC 内置的代码。同时，为了在智能体数量变化的测试环境下运行上述基线算法，本文使用注意力机制模型增强了上述基线算法，从而使得上述基线算法可以接收数量变化的智能体输入信息。为了保证对比实验的公平，本文使用了 SMAC 环境提供的原始参数运行实验。

表 2　本文使用的算法

Table 2 Algorithms used in this paper

算法名称	描述
IQL	独立Q学习
VDN	加性假设
QMIX	单调性假设
QTRAN	转化联合动作-值函数
IQL_Attn	IQL 算法增加注意力模块
VDN_Attn	VDN 算法增加注意力模块
QMIX_Attn	QMIX 算法增加注意力模块
QTRAN_Attn	QTRAN 算法增加注意力模块

本文算法的信用分配网络使用了 QMIX 算法的信用分配网络，具体参数设置与 QMIX 相同。在智能体决策网络中，本文使用了具有 64 维隐藏状态的门控循环单元， 同时该门控循环单元的尾端连接了一个线性层。本文通过$\varepsilon-$贪心算法实现对智能体动作的探索。在训练过程中，本文对超参数ε进行线性退火，在五万个时间步内从 1 逐步减少到 0.05，并在训练结束前保持不变。本文所有的神经网络都使用了 RMSprop 优化器进行优化，学习率设置为5×10^{-4}。本文中智能体之间进行参数共享，所有的神经网络均进行了随机初始化并进行端到端训练。

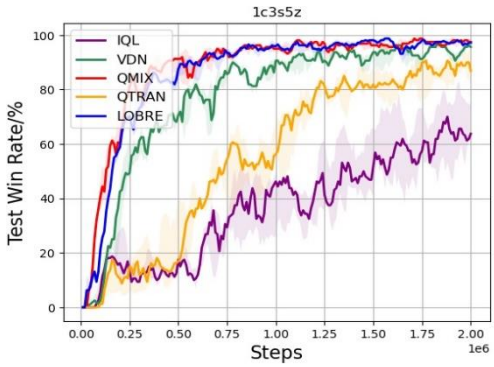
3.3 实验结果

本文将本文所提的 LOBRE 算法与 IQL, VDN, QMIX, QTRAN 算法在星际争霸微操环境和 Ad-Hoc 协作场景下进行对比。其中，5m_vs_6m-6m_vs_6m 地图表示在 5m_vs_6m 地图上进行训练，在 6m_vs_6m 地图上进行测试，也就是说在测试的时候增加了我方智能体的数目。3s2z-2s3z 地图表示在 3s2z 地图上进行训练，在 2s3z 地图上进行测试，也就是说在测试的时候同时改变了不同种类的我方智能体的数目。5m-5ma 地图表示在 5m 地图上进行训练，在 5ma 地图上进行测试。5ma-5m 地图表示在 5ma 地图上进行训练，在 5m 地图上进行测试。上述两个地图是在测试的时候改变了我方智能体的种类。

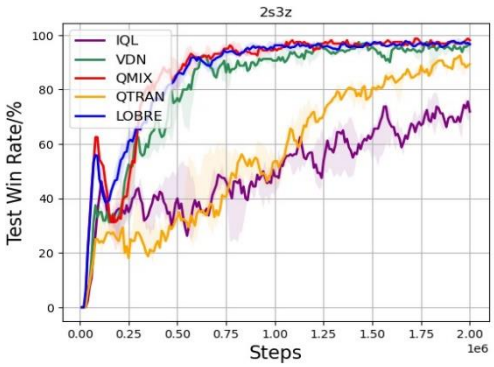
在本文所涉及到的地图中，1c3s5z，2s3z 是简单地图，5m_vs_6m 是困难地图，需要训练更长的时间才能有较好的效果，MMM2 是极度困难地图，需要学到集火等某些特定的微操技巧才有可能取得对局的胜利。5m_vs_6m-6m_vs_6m 地图由于在测试的时候新增了队友，因此相对容易，5m-5ma 地图由于在测试的时候更换了智能体的种类，因此相对复杂。

从图 5 和图 6 所示的实验结果中可以发现，本文所提的 LOBRE 算法在上述地图上都取得了超过现有代表性算法的性能。其中，在简单地图上本文所提的 LOBRE 算法性能略有提升，在复杂地图上本文所提的 LOBRE 算法性能提升更加明显，比如在 5m_vs_6m 地图，MMM2 地图上。这可能是因为针对局部观测的抽象和重建更有利于算法对于复杂策略的学习。同时，本文所提的 LOBRE 算法在 Ad-Hoc 协作场景下取得了较大的提升，这是因为本文所提的 LOBRE 算法通过对局部观测进行重建，使得算法可以有效利用不同局面中的高维状态表征，适应变化的队友，从而提升了算法在 Ad-Hoc 协作场景下的零样本泛化能力。

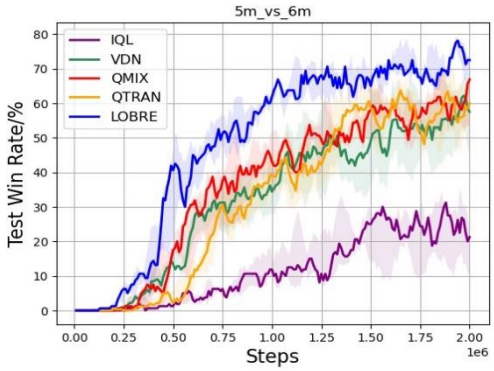
本文在 Ad-Hoc 协作场景 5m_vs_6m-6m_vs_6m 和 3s2z-2s3z 上进行消融实验，如图 6 所示。通过本文所提的 LOBRE 算法和 QMIX_Attn 算法的性能对比可以得知，采样网络的加入提升了算法的性能。通过本文所提的 LOBRE 算法和 QMIX 算法的性能对比可以得知，局部观测重建网络的加入提升了算法的性能。



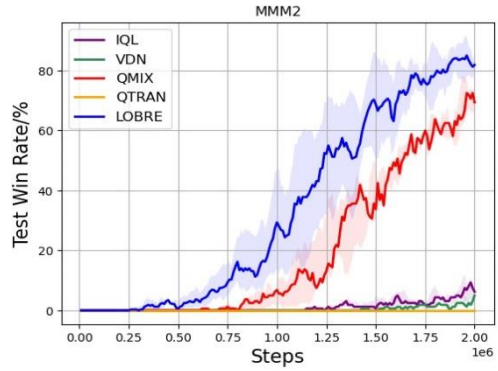
(a)1c3s5z 上的实验结果



(b)2s3z 上的实验结果



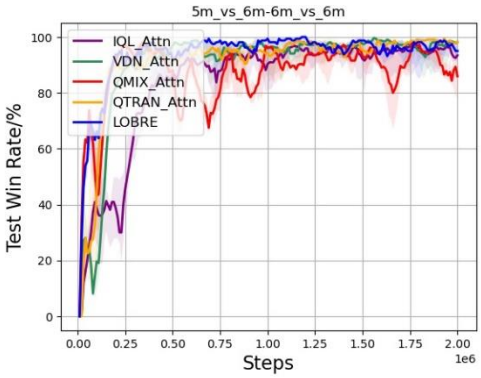
(c)5m_vs_6m 上的实验结果



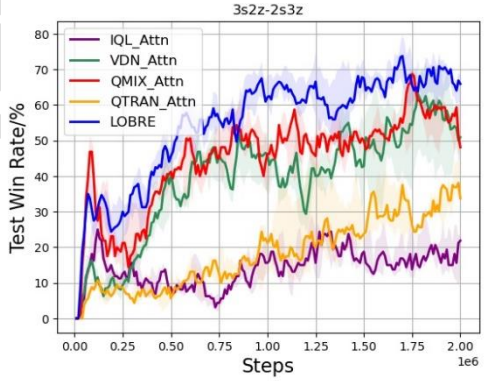
(d)MMM2 上的实验结果

图 5 LOBRE 算法在 SMAC 上的实验结果

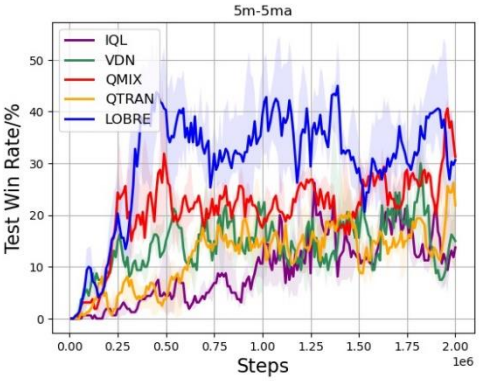
Fig. 5 Experimental results of LOBRE algorithm on SMAC



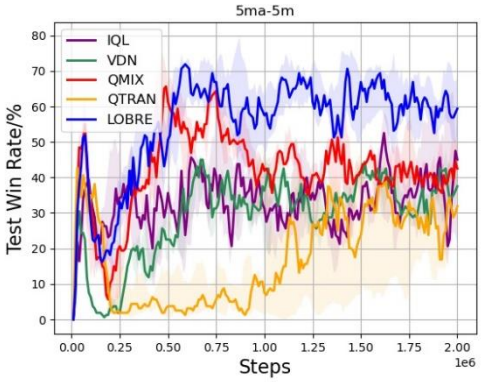
(a) 5m_vs_6m-6m_vs_6m 上的实验结



(b)3s2z-2s3z 上的实验结果



(c)5m-5ma 上的实验结果



(d)5ma-5m 上的实验结果

图 6 LOBRE 算法在 Ad-Hoc 协作场景的实验结果

Fig. 6 Experimental results of LOBRE algorithm on ad-hoc cooperation scenarios

4 总结与展望

本文提出了一种面向 Ad-Hoc 协作的局部观测重建算法。该算法通过局部观测重建网络对智能体接收到的局部观测进行了重构，通过采样网络实现了局部观测抽象，使得算法认识到并充分利用不同局面中的高维状态表征，从而实现了在 Ad-Hoc 协作场景下的零样本泛化。对比实验结果表明，本文所提的 LOBRE 算法在星际争霸微操环境和 Ad-Hoc 协作场景下均取得了超过现有代表性算法的性能，证明了本文所提的 LOBRE 算法的有效性。未来值得进一步探索的问题包括如何更有效地利用局部观测来增强算法的零样本泛化能力，如何更好地对局部观测进行抽象从而得到高维状态表征等。

参考文献

[1] Gao Y Q, Wang W, Yu N P. Consensus multi-agent reinforcement learning for volt-VAR control in power distribution networks[J]. IEEE Transactions on Smart Grid, 2021, 12(4): 3594-3604. DOI:10.1109/TSG.2021.3058996.

[2] Bhalla S, Ganapathi Subramanian S, Crowley M. Deep multi agent reinforcement learning for autonomous driving[C]//Advances in Artificial Intelligence, 2020: 67-78. DOI:10.1007/978-3-030-47358-7_7.

[3] Ye D Y, Zhang M J, Yang Y. A multi-agent framework for packet routing in wireless sensor networks[J]. Sensors, 2015, 15(5): 10026-10047. DOI:10.3390/s150510026.

[4] Oliehoek F A, Spaan M T J, Vlassis N. Optimal and approximate Q-value functions for decentralized POMDPs[J]. Journal of Artificial Intelligence Research, 2008, 32: 289-353. DOI:10.1613/jair.2447.

[5] Kraemer L, Banerjee B. Multi-agent reinforcement learning as a rehearsal for decentralized planning[J]. Neurocomputing, 2016, 190: 82-94. DOI:10.1016/j.neucom.2016.01.031.

[6] Foerster J, Farquhar G, Afouras T, et al. Counterfactual multi-agent policy gradients[EB/OL]. 2017: arXiv: 1705.08926[cs.AI] (2017-05-24) [2022-03-28]. https://arxiv.org/abs/1705.08926

[7] Sunehag P, Lever G, Gruslys A, et al. Value-decomposition networks for cooperative multi-agent learning[EB/OL]. 2017: arXiv: 1706.05296 (2017-06-16) [2022-03-28]. https://arxiv.org/abs/1706.05296

[8] Rashid T, Samvelyan M, Witt C S D, et al. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning[C]// International Conference on Machine Learning, 2018: 4295-4304. DOI: 10.48550/arXiv.1803.11485

[9] Son K, Kim D, Kang W J, et al. QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning[J]. CoRR, 2019, abs/1905.05408, 2019

[10] Vinyals O, Babuschkin I, Czarnecki W M;, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning[J]. Nature, 2019, 575(7782): 350-354. DOI:10.1038/s41586-019-1724-z

[11] Baker B, Kanitscheider I, Markov T, et al. Emergent tool use from multi-agent autocurricula[J]. CoRR, 2019, abs/1909.07528, 2019

[12] van der Vaart P, Mahajan A, Whiteson S. Model based multi-agent reinforcement learning with tensor decompositions [EB/OL]. arXiv:2110.14524 (2021-10-27) [2022-03-28]. https://arxiv.org/abs/2110.14524

[13] Stone P, Kaminka GA, Kraus S, et al. Ad hoc autonomous agent teams: Collaboration without pre-coordination[C/OL]// Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010. (2010-07-11) [2022-03-28] https://dl.acm.org/doi/10.5555/2898607.2898847

[14] Samvelyan M, Rashid T, De Witt CS, et al. The

starcraft multi-agent challenge [EB/OL]. arXiv:1902.04043 (2019-02-11) [2022-03-28]. https://arxiv.org/abs/1902.04043

[15] Oliehoek F A, Amato C. A Concise Introduction to Decentralized POMDPs[M]. Cham: Springer International Publishing, 2016. DOI:10.1007/978-3-319-28929-8.

[16] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature, 2016, 529(7587):484-489. DOI:10.1038/nature16961

[17] Moravčík M, Schmid M, Burch N, et al. DeepStack: Expert-level artificial intelligence in heads-up no-limit poker[J]. Science, 2017, 356(6337): 508-513. DOI:10.1126/science.aam6960.

[18] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J]. CoRR, 2013, abs/1312.5602, 2013

[19] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540):529-533. DOI:10.1038/nature14236

[20] Yang Y D, Wang J. An overview of multi-agent reinforcement learning from game theoretical perspective[EB/OL]. arXiv:2011.00583 (2020-11-01) [2022-03-28]. https://arxiv.org/abs/2011.00583

[21] Hernandez-Leal P, Kartal B, Taylor M E. A survey and critique of multiagent deep reinforcement learning[J]. Autonomous Agents and Multi-Agent Systems, 2019, 33(6): 750-797. DOI:10.1007/s10458-019-09421-1.

[22] Yang Y D, Luo J, Wen Y, et al. Diverse auto-curriculum is critical for successful real-world multiagent learning systems [EB/OL]. arXiv:2102.07659 (2021-02-15) [2022-03-28]. https://arxiv.org/abs/2102.07659

[23] Tan M. Multi-agent reinforcement learning: Independent vs. cooperative agents [EB/OL] //Proceedings of the tenth international conference on machine learning， 1993: 330-337. (1993-06-27) [2022-03-28] https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.8066&rep=rep1&type=pdf&ref=https://githubhelp.com

[24] Raghu M, Irpan A, Andreas J, et al. Can deep reinforcement learning solve Erdos-Selfridge-Spencer games [EB/OL]. arXiv:1711.02301 (2017-11-07) [2022-03-28]. https://arxiv.org/abs/1711.02301

[25] Bansal T, Pachocki J, Sidor S, et al. Emergent complexity via multi-agent competition[EB/OL]. arXiv:1710.03748 (2017-10-10) [2022-03-28]. https://arxiv.org/abs/1710.03748

[26] Leibo J Z, Perolat J, Hughes E, et al. Malthusian reinforcement learning [EB/OL]. arXiv:1812.07019 (2018-12-17) [2022-03-28]. https://arxiv.org/abs/1812.07019

[27] Leibo J Z, Zambaldi V, Lanctot M, et al. Multi-agent reinforcement learning in sequential social dilemmas [EB/OL]. arXiv:1702.03037 (2017-02-10) [2022-03-28]. https://arxiv.org/abs/1702.03037

[28] Mordatch I, Abbeel P. Emergence of grounded compositional language in multi-agent populations [EB/OL]. arXiv:1703.04908 (2017-03-15) [2022-03-28]. https://arxiv.org/abs/1703.04908

[29] Lazaridou A, Peysakhovich A, Baroni M. Multi-agent cooperation and the emergence of (natural) language[EB/OL] . arXiv:1612.07182 (2016-12-21) [2022-03-28]. https://arxiv.org/abs/1612.07182

[30] Foerster J N, Assael Y M, de Freitas N, et al. Learning to communicate with deep multi-agent reinforcement learning[J]. CoRR, 2016, abs/1605.06676, 2016

[31] Sukhbaatar S, Szlam A, Fergus R. Learning multiagent communication with backpropagation[J]. CoRR, 2016, abs/1605.07736, 2016

[32] Peng P, Wen Y, Yang Y D, et al. Multiagent bidirectionally-coordinated nets: Emergence of Topologyhuman-level coordination in learning to play StarCraft combat games[EB/OL].

arXiv:1703.10069 (2017-03-29) [2022-03-28]. https://arxiv.org/abs/1703.10069

[33] Palmer G, Tuyls K, Bloembergen D, et al. Lenient multi-agent deep reinforcement learning[EB/OL]. arXiv:1707.04402 (2017-07-14) [2022-03-28]. https://arxiv.org/abs/1707.04402

[34] Omidshafiei S, Pazis J, Amato C, et al. Deep decentralized multi-task multi-agent reinforcement learning under partial observability [EB/OL]. arXiv:1703.06182 (2017-03-17) [2022-03-28]. https://arxiv.org/abs/1703.06182

[35] Lanctot M, Zambaldi V, Gruslys A, et al. A unified game-theoretic approach to multiagent reinforcement learning[EB/OL]. arXiv:1711.00832 (2017-10-02) [2022-03-28]. https://arxiv.org/abs/1711.00832

[36] Hong ZW, Su SY, Shann TY, et al. A deep policy inference q-network for multi-agent systems[EB/OL]. arXiv:1712.07893 (2017-12-21) [2022-03-28]. https://arxiv.org/abs/1712.07893

[37] Heinrich J, Silver D. Deep reinforcement learning from self-play in imperfect-information games[J]. CoRR, 2016, abs/1603.01121, 2016

[38] Chen S, Andrejczuk E, Cao Z G, et al. AATEAM: achieving the ad hoc teamwork by employing the attention mechanism[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(5): 7095-7102. DOI:10.1609/aaai.v34i05.6196.

[39] Zhang T, Xu H, Wang X, et al. Multi-agent collaboration via reward attribution decomposition[EB/OL]. arXiv:2010.08531 (2020-10-16) [2022-03-28]. https://arxiv.org/abs/2010.08531

[40] Mahajan A, Samvelyan M, Gupta T, et al. Generalization in Cooperative Multi-Agent Systems[EB/OL]. arXiv:2202.00104 (2022-01-31) [2022-03-28]. https://arxiv.org/abs/2202.00104

[41] Agmon N, Stone P. Leading ad hoc agents in joint action settings with multiple teammates[C]//AAMAS, 2012: 341-348. DOI: 10.5555/2343576.2343625

[42] Stone P, Kaminka G A, Rosenschein J S. Leading a best-response teammate in an ad hoc team[C]//Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets, 2010: 132-146. DOI:10.1007/978-3-642-15117-0_10.

[43] Tambe M. Towards flexible teamwork[J]. Journal of Artificial Intelligence Research, 1997, 7: 83-124. DOI:10.1613/jair.433.

[44] Grosz B J, Kraus S. Collaborative plans for complex group action[J]. Artificial Intelligence, 1996, 86(2): 269-357. DOI:10.1016/0004-3702(95)00103-4.