arXiv:2302.06872v1 [cs.RO] 14 Feb 2023

# 基于贪心边际贡献的自适应价值分解：面向协作式多智能体强化学习的贪心边际贡献计算

Shanqi Liu 浙江大学 中国杭州
s
hanqiliu@zju.edu.cn

Yujing Hu* 网易伏羲人工智能实验室 中国杭州
huyujing@corp.netease.com

Runze Wu 网易伏羲人工智能实验室 中国杭州
wurunze1
@corp.netease.com

Dong Xing 浙江大学 中国杭州
dongxing
@zju.edu.cn

Yu Xiong 网易伏羲人工智能实验室 中国杭州
xiongyu1@corp.netease.com

Changjie Fan 网易伏羲人工智能实验室 中国杭州
fanchangjie@corp.netease.com

Kun Kuang 浙江大学 中国杭州
kunkuang
@zju.edu.cn

Yong Liu* 浙江大学 中国杭州
yongliu
@iipc.zju.edu.cn

## 摘要

现实世界中的合作通常需要智能体之间进行密集的协同。这项任务已在协作式多智能体强化学习（MARL）框架下得到广泛研究，而价值分解方法是其中最前沿的解决方案之一。然而，传统方法将价值函数学习为单智能体效用的单调混合，无法解决具有非单调回报的任务，这限制了它们在通用场景中的应用。近期方法从隐式信用分配的角度着手，通过学习具有完全表达能力的价值函数，或采用额外结构来改进合作，以解决这一问题。然而，由于联合动作空间庞大，这些方法要么难以学习，要么无法充分捕捉智能体之间复杂的交互，而这对于解决具有非单调回报的任务至关重要。此外，现实场景中的应用通常要求策略具有可解释性，但隐式信用分配方法的可解释性有限。为解决这些问题，我们提出了一种新颖的显式信用分配方法，以应对非单调问题。我们的方法——基于贪心边际贡献的自适应价值分解（AVGM）——基于一种自适应价值分解，能够学习动态变化的智能体群体的合作价值。我们首先说明，所提出的价值分解能够考虑智能体之间复杂的交互，并且适用于在大规模场景中学习。随后，我们的方法使用从价值分解中计算得出的贪心边际贡献作为个体信用，以激励智能体学习最优协作策略。我们进一步通过引入动作编码器扩展该模块，从而确保计算贪心边际贡献的时间复杂度为线性。实验结果表明，我们的方法取得了显著的性能提升在若干非单调领域中。此外，我们展示了模型具有良好的可解释性和合理性。这表明，我们的模型可应用于需求更贴近现实的场景。

## 关键词

多智能体协作；信用分配；非单调

**ACM 引用格式：**
Shanqi Liu、Yujing Hu*、Runze Wu、Dong Xing、Yu Xiong、Changjie Fan、Kun Kuang 和 Yong Liu*。2023。采用贪心边际贡献计算的合作式多智能体强化学习自适应价值分解。载于*第22届自主智能体与多智能体系统国际会议论文集（AAMAS 2023）*，伦敦，英国，*2023*年*5*月*29*日至*6*月*2*日，自主智能体与多智能体系统国际基金会，12页。

## 1 引言

许多现实任务，如双臂操作 [11],自动驾驶 [13] 和群体 [8]，都需要多个智能体相互合作。尤其是在许多场景中，智能体需要同时选择最优动作来完成共同目标，例如双手抬举任务要求双臂同时抬起物体 [1, 2]。在这些任务中学习协作策略仍然颇具挑战，因为联合动作空间会随着智能体数量增加而呈指数级增长；此外，由于智能体之间存在复杂交互，进行奖励分解（即信用分配）也十分困难。合作式多智能体强化学习（MARL）已被广泛用于从智能体的经验中学习此类任务的有效行为。目前，基于价值的合作式多智能体强化学习的一种流行范式是集中训练、分散执行（CTDE），代表性方法包括 VDN 和 QMIX [20, 23]。这些方法将集中式价值函数学习为各智能体效用函数的单调分解，并通过最大化每个智能体对应的效用函数实现分散式执行，这一原则称为

第*22*届自主智能体与多智能体系统国际会议论文集（*AAMAS 2023*），*A. Ricci*、*W. Yeoh*、*N. Agmon*、*B. An*（编），*2023*年*5*月*29*日至*6*月*2*日，伦敦，英国。© 2023 自主智能体与多智能体系统国际基金会（www.ifaamas.org）。保留所有权利。

*通讯作者。

个体全局最大值（IGM）原则 [6]。尽管这些方法已成功求解具有单调收益矩阵的任务并达到最优 [7], , 但在具有非单调收益矩阵的任务中却会失效。由于单调价值函数会导致在具有非单调收益的环境中只能得到次优的价值近似 [22, 25], , 因此这些方法无法表示智能体的最优动作取决于其他智能体动作的策略。这个问题称为相对过度泛化 [17], , 它使智能体无法解决双手抬举等任务。

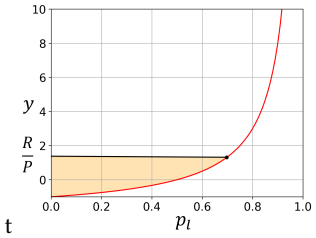
为了解决这一问题，近期研究要么旨在学习具有完整表达能力的价值函数 [25] , 要么采用辅助方法，例如更加重视较优的联合动作，以找到最优协作策略 [19]。然而，这些方法仍面临其他问题。一方面，由于联合动作空间的规模随智能体数量呈指数级增长，学习完整的表达能力十分困难。另一方面，采用辅助方法来提升合作，既无法捕捉智能体之间复杂的交互，也缺乏最优性的理论保证 [5] （第3节将详细说明这些问题的原因）。此外，当前方法通常属于隐式信用分配方法，通过混合网络学习信用分配。与基于 Shapley 值 [21] 或边际贡献的显式信用分配方法相比，隐式方法对分布式信用缺乏可解释性，这阻碍了它们在安全性至关重要的场景中的应用。

在这项工作中，我们提出了一种新颖的显式信用分配方法，通过利用每个智能体的贪心边际贡献来解决非单调问题。首先，我们提出一种自适应价值分解方法，将个体效用建模为其他可观测智能体的动作上的分布。价值分解使智能体能够学习个体效用，并将与其他智能体的交互纳入考量。我们说明，这种个体效用能够正确表示非单调收益矩阵，并在第3节进一步探究以特定值为效用建模的先前方法为何无法应对非单调情境。此外，由于效用仅对可观测智能体之间的交互进行建模，其学习复杂度不会像那些追求完全表达能力的方法那样随智能体数量呈指数级增长。然而，尽管这种个体效用能够正确描述收益矩阵，我们仍无法直接将其用作策略的价值函数，因为它需要其他智能体的动作。为解决这一问题，我们的方法通过从自适应价值分解中学习贪心边际贡献，优化每个智能体的分散式策略；该方法计算最优边际贡献，从而避免在计算策略时必须知道其他智能体动作的要求。我们说明，优化贪心边际贡献能够避免采取导致协调失误的动作，从而实现合作策略。这也通过增加采取合作动作的样本数量来增强探索。此外，由于最大化边际贡献的计算时间复杂度与可观测智能体数量呈指数关系，我们提出一种动作编码器，将动作映射到精简的潜在空间，以确保最大化过程的搜索时间呈线性增长。这降低了计算负担，并使学习过程切实可行。最后，由于我们的方法能够学习与动态变化的可观测智能体协作时各动作的价值，因此也提升了学习策略与可扩展智能体协作的适应能力。我们在需要智能体同时协调的非单调机器人任务中，将本方法与若干最先进的基线方法进行比较。结果表明，本方法显著提升了渐近性能和效率。此外，我们还展示了本方法所具备、而现有方法所缺乏的可解释性。

## 2 相关工作

许多通用的深度多智能体强化学习方法已应用于复杂的多智能体环境，包括 COMA [3], 、MADDPG [14], 等。目前，集中训练、分散执行已成为合作式多智能体强化学习的事实主流范式 [9, 14]。具体而言，VDN [23] 通过将每个智能体的效用相加来分解联合动作Q值。QMIX [20] 在 VDN 的基础上扩展，使联合动作Q值可以是各智能体效用的单调组合，并且该组合可随全局状态变化。然而，QMIX 及相关的基于QMIX的变体方法所引入的联合动作价值单调约束，会导致可证明的探索能力不足和相对过度泛化 [17]。为解决这一问题，QPLEX [25] 和 QTRAN [22] 旨在学习具有完整表达能力的价值函数。然而，据报道，它们在实际应用中表现不佳 [5, 24]。这是因为，在复杂的多智能体强化学习任务中，由于在大型联合动作空间中探索困难，学习完整的表达能力并不切实际。相比之下，我们将智能体交互价值的建模限制在可能合作的智能体对之间，从而避免学习复杂度随联合动作空间规模呈指数级增长。此外，我们的方法还通过提高采样协作动作的概率来增强探索。也有一些方法使用辅助结构来解决这一问题。例如，MAVEN [15] 通过引入用于分层控制的潜在空间，融合了基于价值和基于策略的方法，从而实现有承诺的、时间跨度更长的探索。Weighted QMIX [19] 基于 QMIX，通过引入权重来更加重视较优的联合动作，以纠正次优性。UneVEn [5]通过对通用后继特征进行线性分解，同时学习一组相关任务。然而，这些方法无法在考虑其他智能体动作变化的情况下估计动作价值，并且通常在具有极端奖励的环境中失效。相反，我们的方法能够根据效用，并考虑所有合作智能体之间的交互，估计每个智能体的贡献；研究表明，该方法适用于所有非单调环境。此外，上述所有方法都属于隐式信用分配方法，缺乏对分布式信用的可解释性。

我们的工作也与信用分配有关。与隐式信用分配方法相比，显式方法会将每个智能体的贡献归因于可证明的局部最优性 [10]。COMA [3] 利用反事实优势来学习价值函数。QPD [28] 设计了多通道混合器评论家网络，并利用积分梯度沿路径分配信用。然而，这些方法采用单调价值函数，忽略了复杂的




| | $C$ | $L$ |
|---|---|---|
| $C$ | $+R$ | $-P$ |
| $L$ | $-P$ | 0 |

t

**图 1：左：方程 (1) 的示意图。橙色区域表示可使用 $Q_i(\tau_i, a_i)$ 作为策略的方法能够求解的收益矩阵。右：双手抬升示例的收益矩阵。**

智能体间交互，而这些交互对于非单调任务中的合作至关重要。为解决这一问题，SQDDPG [26] 和 Shapley[12] 采用夏普利值 [21] 来估计智能体间复杂的交互。夏普利值源自合作博弈论，能够通过估计参与智能体的贡献，合理地分配收益。然而，SQDDPG [26]假设智能体按顺序采取动作，这在许多场景中并不成立。Shapley [12] 使用单个集中式评论家网络来估计夏普利值，当智能体数量较多时，这种做法并不可行。此外，由于计算夏普利值涉及指数时间复杂度 [26]，这些方法只能得到近似的夏普利值。

## 3 动机

本节以机器人学中的双手臂任务为例进行说明，尽管我们的方法可应用于具有非单调收益的一般任务。双手臂任务是指需要双臂机器人进行操作的任务。与单臂机器人相比，双臂机器人在完成复杂的现实世界任务方面具有显著优势，尤其是符合人类直觉的任务 [18, 27]。例如，简单任务如抬起一个沉重的物体，或复杂任务如外太空装配与维修或家务 [11]。然而，使用多智能体强化学习学习双手臂任务颇具挑战，因为多条手臂之间的协作动作通常相互依赖。此外，这些任务的收益矩阵是非单调的，这会阻碍智能体学习最优策略。为说明这一问题的原因，我们以双手抬举任务为例。该任务要求两个智能体同时采取协作动作 $C$ 来抬起物体，协作奖励为 $+R$。否则，智能体单独采取动作 $C$ 时会受到惩罚 $-P$，因为单独无法抬起物体，且会浪费能量。同时，动作 $L$ 表示动作空间中的懒惰动作（例如，保持静止），这类动作无法实现合作，但可以避免惩罚。相应的非单调收益矩阵见图 1。

我们通过分析个体效用的形式，探究这类任务中学习困难的原因。为了学习分散式策略，现有方法将每个智能体的个体效用建模为某个价值函数 $Q_i(\tau_i,\ a_i)$。然而，由于这类任务中的回报取决于所有智能体之间的交互，这种分解方式无法考虑其他智能体的动作，因此在某些情况下不足以表示最优策略。我们说明，更全面、能够考虑智能体之间交互的个体效用应为 $Q_c^i(\tau_i,\ u_i^-,\ a_i)$，其中 $u_i^-$ 是潜在协作智能体的联合动作（详见第 5.1 节）。根据这一分解，个体效用 $Q_i(\tau_i,\ a_i)$ 应被视为从关于 $u_i^-$ 的分布 $Q_c^i(\tau_i,\ u_i^-,\ a_i)$ 中采样的变量。优化 $Q_i(\tau_i,\ a_i)$ 而非 $Q_c^i(\tau_i,\ u_i^-,\ a_i)$，可能导致策略收敛到平均策略。我们说明，这解释了为什么现有方法在具有非单调收益矩阵的环境（如双手抬举任务）中会失效。例如，在双手抬举任务中，当

$$\frac{R}{P} < \frac{2p_l - 1}{1 - p_l}. \tag{1}$$

其中，$p_l$ 是每种策略采取动作 $L$ 的概率。详细推导见附录 6。图 1 展示了方程 (1) 的结果。

我们发现，直接使用 $Q_i(\tau_i,\ a_i)$ 作为策略的方法（如 QMIX），只能解决所有可能收益中的一小部分，因为即使很小的 $P$ 也可能导致策略失效，这是由于 $P$ 的权重会随着 $p_l$ 增大而呈指数级增长。然而，其他旨在解决相对过度泛化问题的方法可以降低 $p_l$，从而解决更多收益。例如，Weighted QMIX 通过更加重视较优的联合动作来降低 $p_l$。然而，由于它没有对智能体之间的实际交互进行建模，其重要性权重是经验性的。因此，它无法保证 $p_l$ 降至零，并且在 $P$ 较大时仍会失效。QPLEX 可以通过学习具有完全表达能力且收敛的价值函数，将 $p_l$ 降至零。然而，随着智能体数量增加，联合动作空间呈指数级增长，因此这类价值函数很难学习。此外，学习完全表达能力需要充分探索，而在示例环境中很难做到这一点，因为在均匀探索期间，$p_l$ 通常较大。这是因为协作动作（例如 lift）寥寥无几，而懒惰动作（例如 stay 和 move）在整个动作空间中占主导地位。然而，我们的方法通过学习能够考虑智能体间交互的可行价值分解，并利用贪心边际贡献确保 $p_l$ 可以降至零，从而解决了这些问题。因此，我们的方法在实践中兼具最优性和效率。

## 4 预备知识

### 4.1 去中心化部分可观测马尔可夫决策过程

一个完全合作的多智能体序贯决策任务可以描述为去中心化部分可观测马尔可夫决策过程（Dec-POMDP），其由一组可能的全局状态 $S$、动作 $A_1, ..., A_N$ 和观测 $\Omega_1, ..., \Omega_N$ 定义。在每个时间步，每个智能体 $i \in \{1, ..., N\}$ 选择一个动作 $a_i \in A_i$，它们共同构成一个联合动作 $\mathbf{u} \in U$。下一个状态由转移函数 $P$: $S \times U \to S$ 决定。每个智能体 $o_i \in \Omega_i$ 的下一个观测由观测函数 $O$: $S \to \Omega$ 更新。所有智能体共享相同的奖励 $r$: $S \times U \to \mathbf{r}$和联合价值函数 $Q_{tot} = E_{s_{t+1:\infty}, a_{t+1:\infty}}[R_t | s_t, \mathbf{ut}]$，其中 $R_t = \sum_{j=0}^{\infty} \gamma^j r_{t+j}$ 是折扣回报。每个智能体的观测也可以由动作与观测历史替代

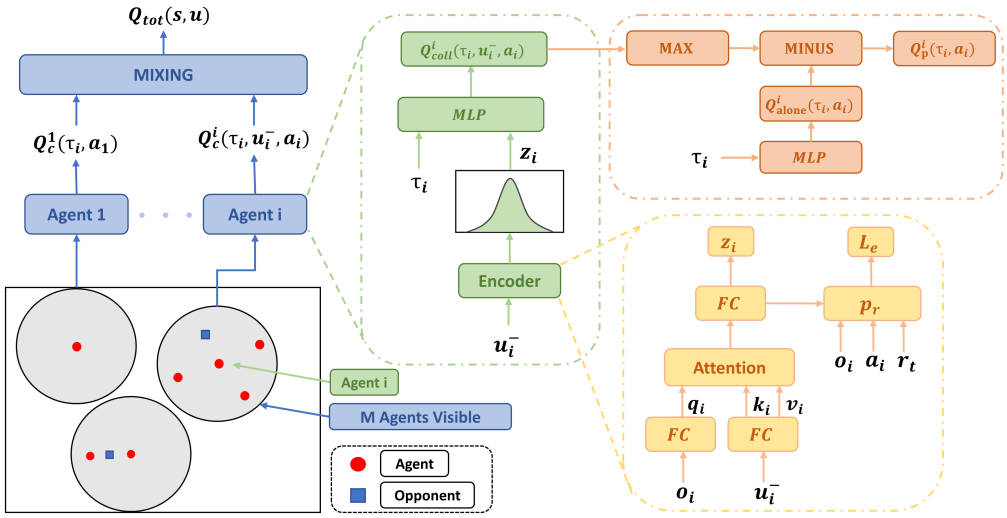



**图2：我们方法的架构。左侧（蓝色）：总Q值的架构。中间（绿色）：非单调评论家网络 $Q_c^i(\tau_i, u_i^-, a_i)$的框架。右上方（红色）：贪心边际贡献和 $Q_p^i(\tau_i, a_i)$的框架。右下方（黄色）：动作编码器和预测模型的框架。**

以此作为处理部分可观测性的代理 [20, 23]。智能体 $i$ 的动作与观测历史可表示为 $\tau_i$，即 $(o_i^0, a_i^0, ..., o_i^t)$ 。

## 4.2 边际贡献

在这项工作中，我们还引入了夏普利值的边际贡献。智能体 $i$ 在夏普利值中的边际贡献定义如下

$$\phi_i = v(C) - v(C/i) \tag{2}$$

其中， $C$ 是由相互合作以实现共同目标的智能体组成的团队，$C/i$ 表示缺少智能体 $i$时的集合。 $v(C)$表示用于估计一组智能体合作情况的价值函数。此外，我们用 $N, M_i$ 表示智能体总数，$M_i$ 表示智能体 $i$所观测到的智能体数量。

# 5 方法

在本节中，我们提出一种显式信用分配方法——基于贪心边际贡献的自适应价值分解（AVGM），用于在收益非单调的任务中学习最优协作策略。首先，我们提出一种能够捕捉智能体之间交互的价值分解方法。然后，我们提出一种基于该价值分解的演员-评论家结构，用于训练集中式评论家网络，并从评论家网络中计算贪心边际贡献，以训练分散式执行策略。

## 5.1 通过自适应价值分解学习非单调评论家网络

由于当前的个体效用 $Q_i(\tau_i, a_i)$ 无法捕捉智能体之间的交互，因此我们将智能体的个体效用修订为一种更全面的形式。首先，由于环境的总奖励由智能体之间各种可能的交互产生，

我们有

$$r_{tot}(s,u) = \sum_{i=1}^{N} \hat{r}_1(o_i, a_i) + \sum_{i<j=2}^{i<j=N} \hat{r}_2(o_i, o_j, a_i, a_j) + ... + \hat{r}_N(o_1, ..., o_N, a_1, ..., a_N). \tag{3}$$

值得注意的是，方程 (3) 中的每个 $\hat{r}_m$ 仅在所指智能体之间存在交互时才会出现。否则，由于奖励由其他项表示，因此应将其忽略。然而，组合数量庞大，使得这种分解难以在实践中应用。幸运的是，我们可以在多智能体分散式执行环境中简化分解。在分散式执行环境中，我们假设合作模式为：每个智能体只能与其可观测范围内的其他智能体交互。原因在于，分散式策略只能依据观测范围内的信息采取行动。如果智能体需要与观测范围之外的其他智能体合作，我们就必须引入一种通信方法来传递必要的消息。否则，智能体无法识别哪些智能体可能实现合作，这会导致从局部观测的角度来看，合作示例不规则，从而无法学习合作策略。此外，在大多数机器人场景中，实际交互范围都小于视野范围，这意味着，如果一个智能体处于另一个智能体的交互范围内，它也必然处于对方的视野范围内。

根据这一原则，我们将 team $i$ 定义为智能体 $i$视野范围内的所有 $M_i$ 智能体。智能体 $i$ 的总奖励可表示为 $r_{team}^i(o_i, u_i^-, a_i)$，其中 $u_i^-$表示 team $i$中其余智能体的联合动作。由于观测 $o_i$可作为团队的全局状态，其中已包含所有其他智能体的信息，因此智能体 $i$ 只需借助其他智能体的动作信息来确定总奖励。此外，考虑到智能体之间各种可能的交互

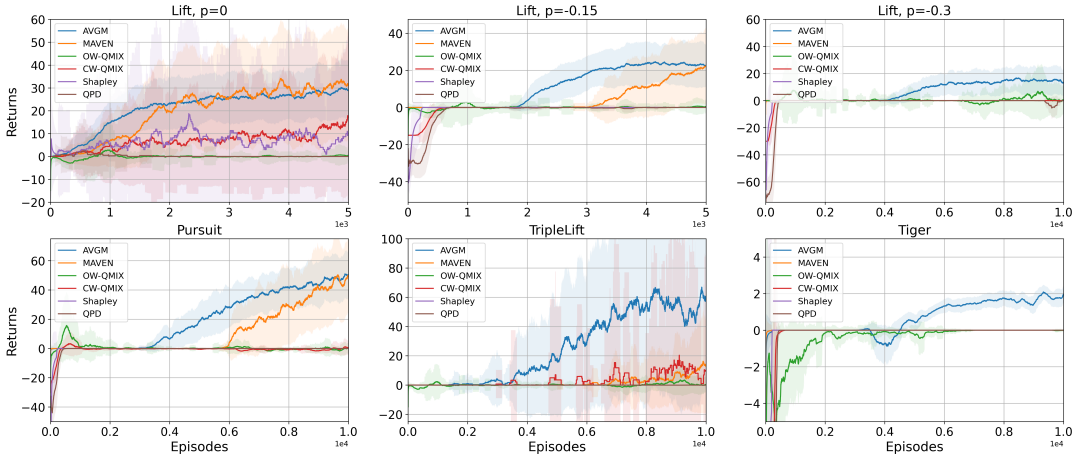



**图3：上：惩罚从 0 增至 -0.3 时，*lift* 中的结果。下：*pursuit*、*triplelift* 和 *tiger*中的结果。**

我们可以将 $r^i_{team}(o_i,\ u_i^-,\ a_i)$分解为方程 (3)：

$$r^i_{team}(o_i,u_i^-,a_i)=\hat{r}^i_1(o_i,a_i)+\sum_{j=1,j\neq i}^{M_i}\hat{r}^i_2(o_i,a_j,a_i) + \ldots+\hat{r}^i_{M_i}(o_i,a_1,\ldots,a_{M_i},a_i). \tag{4}$$

其中，只有当所描述的智能体与智能体 $i$之间存在交互时，$\hat{r}^i_m$ 才存在；否则它应等于零。结合方程 (4) 和方程 (3)，可得

$$r_{tot}(s,u)=\sum_{i=1}^{N}r^i_{team}(o_i,u_i^-,a_i). \tag{5}$$

详细推导见附录 5。基于该分解，我们可以推导出相应的自适应价值分解，其中每个智能体的个体效用为 $Q^i_c(\tau_i,\ u_i^-,\ a_i)$，它在每个时间步学习与动态变化的可观测智能体合作的价值。于是，我们有

**定理 1：** 对于任意 $r_{tot}(s,u)$，对应的 $Q_{tot}(s,u)=\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r_{tot}(s,u)\mid\pi\right]$以及每个智能体的效用 $Q^i_c(\tau_i,u_i^-,a_i)$满足

$$\arg\max_u(Q_{tot}(s,u))=\{\arg\max_{a_1}(Q^1_c(\tau_1,u_1^-,a_1)),\ldots, \arg\max_{a_N}(Q^N_c(\tau_N,u_N^-,a_N))\}. \tag{6}$$

详细证明见附录 5。定理 1 表明，使用效用 $Q^i_c(\tau_i,\ u_i^-,\ a_i)$ 的自适应价值分解能够在给定任意奖励函数的情况下表示无偏价值分解，并满足分散式执行的 IGM 原则。此外，由于该效用仅涉及可观测智能体，因此随着环境中智能体数量的增加，它不会使学习问题变得更加困难。

既然我们提出了 $Q^i_c(\tau_i,\ u_i^-,\ a_i;\ \theta)$ 作为智能体的效用，以解决非单调问题，我们希望从中学习一种能够考虑彼此动作并实现协作的策略。然而，策略 $a_i\sim Q^i_c(\tau_i,\ u_i^-)$ 要求先执行 $u_i^-$，而 $u_i^-$ 又要求先生成 $a_i$，这就陷入了死锁。为了解决这个问题，我们提出了一种演员-评论家结构，用于训练集中式评论家，并根据评论家网络计算贪心边际贡献，以训练分散式执行策略。

基于定理 1，我们构建了非单调集中式评论家网络。由于定理 1 表明，$Q^i_{coll}$能够在给定任意奖励函数的情况下表示无偏价值分解，并满足分散式执行的 IGM 原则。因此，我们可以使用所有$Q^i_{coll}$，通过类似于 QMIX 的单调混合网络计算 $Q_{tot}$，而定理 1 保证混合值是无偏的。因此，我们的集中式评论家网络由每个智能体的自适应效用 $Q^i_c(\tau_i,\ u_i^-,\ a_i)$ 以及用于生成全局 Q 值 $Q_{tot}(s,\ u)$ 的混合网络组成。优化目标是均方误差（MSE）

$$\mathcal{L}_{TD}(\theta)=\mathbb{E}_\pi[Q_{tot}(s_t,\mathbf{u}_t)-y_t]^2 \qquad y_t=r_t+\gamma\max_{\mathbf{u}_{t+1}}Q_{tot}\ (s_{t+1},\mathbf{u}_{t+1})\,. \tag{7}$$

然而，我们注意到，智能体并非总能观察到其他智能体；当智能体之间没有交互时，价值函数也应有所不同。因此，我们将 $Q^i_c(\tau_i,\ u_i^-,\ a_i)$构建为两个价值函数，

$$Q^i_c(\tau_i,u_i^-,a_i)=\begin{cases}Q^i_{coll}(\tau_i,u_i^-,a_i;\theta_1) & M_i>0\\ Q^i_{alone}(\tau_i,a_i;\theta_2) & M_i=0\end{cases} \tag{8}$$

其中，$M_i$是智能体 $i$所观察到的智能体数量。$Q^i_{coll}(\tau_i,\ u_i^-,\ a_i)$表示合作价值函数，当能够观察到任何其他智能体时选择该函数。$Q^i_{alone}(\tau_i,\ a_i)$学习单独行动的智能体的自私价值函数，并在周围没有其他智能体时选择该函数。在训练过程中，我们根据每个时间步是否能够观察到其他智能体，从这两个价值函数中选择一个作为 $Q^i_c(\tau_i,\ u_i^-,\ a_i)$，以生成 $Q_{tot}(s,\ u)$。

## 5.2 通过贪心边际贡献进行显式信用分配

尽管我们已经有了非单调评论家函数，但仍需学习分散式执行策略。幸运的是，我们可以通过优化每个智能体的边际贡献来训练分散式策略，从而无需依赖其他智能体的动作。原因是，边际贡献只能通过

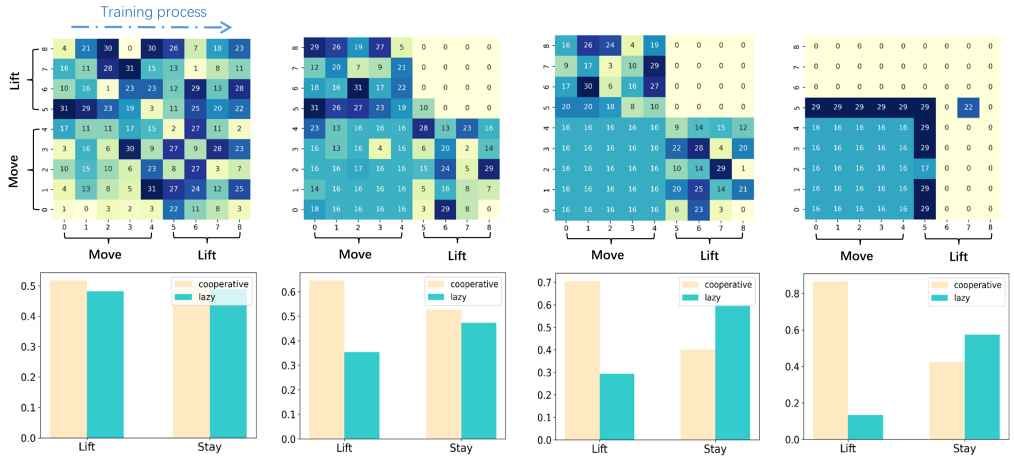



**图 4：上：训练过程从初始阶段（左）到最终阶段（右）的动作编码器输出结果。x 轴和 y 轴分别表示其他智能体的动作。不同的颜色和数字表示不同的动作表示。下：训练过程从初始阶段（左）到最终阶段（右）的信用分配值结果。所有结果均已归一化，以便更好地呈现。**

实现合作才能最大化。因此，如果某个动作能够最大化边际贡献，它就是合作动作。根据边际贡献的定义，如果智能体 $i$ 能够观察其他智能体，我们有

$$\phi_i(\tau_i, a_i) = v(T_i) - v(T_i/i) = Q^i_{coll}(\tau_i, u_i^-, a_i) - Q^i_{coll}(\tau_i, u_i^-, a_i^l) \tag{9}$$

其中，$T_i$ 是由智能体 $i$ 及其观测到的所有智能体组成的智能体集合，$a_i^l$ 是无法与任何智能体实现合作的懒惰动作。在实践中，我们使用

$$a_i^l = \arg\max_{a_i}(Q^i_{alone}(\tau_i, a_i)) \tag{10}$$

其理由在于，由 $Q_{alone}$ 表示的策略是一种只考虑独自行动的自私策略。在非单调环境中，这种策略应当采取懒惰动作，以避免收益矩阵中的惩罚。此外，即使在单调环境中，协作动作通常也不同于自私动作，因此方程 10 在各种环境中都成立。然而，仍然存在另一个问题：对于任意 $(\tau_i,\ a_i)$，都有多个 $\phi_i(\tau_i,\ a_i)$分别对应不同的 $u_i^-$。这可能导致我们的策略收敛到一种平均策略，就像我们在第3节中讨论的情况一样。为了解决这一问题，我们提出了贪心边际贡献，

$$\begin{aligned}\phi_i^*(\tau_i, a_i) &= Q^i_{coll}(\tau_i, u_i^{-*}, a_i) - Q^i_{coll}(\tau_i, u_i^{-*}, a_i^l)\\ (u_i^{-*}, a_i^*) &= \arg\max_{(u_i^-, a_i)}(Q^i_{coll}(\tau_i, u_i^-, a_i))\end{aligned} \tag{11}$$

其核心思想是，我们期望策略能够学习动作 $a_i$ 的潜在最优价值。当所有其他智能体都采取最优合作动作 $u_i^{-*}$与智能体 $i$ 合作时，即可达到最优价值。换言之，我们鼓励智能体基于乐观信念来学习每个动作的价值，即认为其他智能体采取非合作动作的可能性为零，因此所有其他智能体都会与其自身合作。由于非单调环境中的主要问题是智能体往往会采取懒惰动作，担心其他智能体不合作，因此这种乐观信念可以通过提高采样合作动作的概率来促进探索，并确保智能体摆脱次优策略，最终收敛到最优协作策略。这样，我们得到分散式策略的价值函数 $Q_p^i(\tau_i,\ a_i;\ \mu)$，以及分散式策略 $\pi(a_i|\tau_i) = \arg\max_{a_i}(Q_p^i(\tau_i,\ a_i))$。当智能体能够观察其他智能体时，我们使用 $\phi_i^*(\tau_i,\ a_i)$ 作为目标；独自行动时则使用 $Q_{alone}$。总体损失目标为

$$\mathcal{L}_p(\mu) = \begin{cases}\mathbb{E}_\pi[Q_p^i(\tau_i, a_i) - \phi_i^*(\tau_i, a_i)]^2 & M_i > 0\\ \mathbb{E}_\pi[Q_p^i(\tau_i, a_i) - Q_{alone}(\tau_i, a_i)]^2 & M_i = 0\end{cases} \tag{12}$$

## 5.3 学习动作类别表示

然而，$Q_c^i(\tau_i,\ u_i^-,\ a_i)$ 的结构在实践中不可行，因为 $u_i^-$ 的维度是不确定的，我们无法据此构建神经网络。更重要的是，在计算贪心边际贡献时，我们需要搜索 $Q_c^i(\tau_i,\ u_i^-,\ a_i)$ 的最大值。如果搜索所有可能的 $u_i^-$，时间复杂度将呈指数增长。为了解决这些问题，我们提出了一种动作编码器 $z_i \sim f(u_i^-;\phi)$，将动作信息映射到潜在空间中的不同类别，以降低维度。其背后的核心思想是，我们发现 $u_i^-$ 中的许多组合实际上对环境和智能体 $i$ 具有相同的影响，因此它们在潜在空间中应属于同一类别。例如，在要求四个智能体采取协作动作的场景中，无论其他三个智能体中的哪一个采取懒惰动作，智能体 $i$ 面临的情况都是相同的。

为处理 $u_i^-$的不确定维度，我们在编码器中使用注意力结构 [4, 16] 。此外，注意力结构可以帮助智能体更加关注更有可能合作的智能体。我们根据 $o_i$的信息，计算 $u_i^-$中每个$a_j$的权重。注意力的整体结构如图2所示。然而，由于缺少动作类别的标签，编码器必须以无监督方式进行学习。

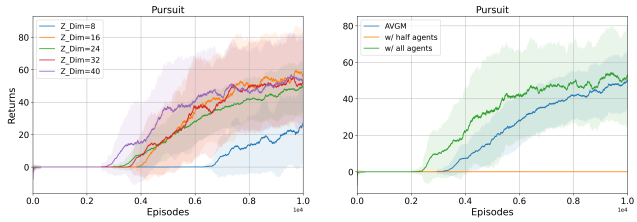



**图5：左：不同 $z$ 维度的比较。右：不同智能体选择方法的比较。**

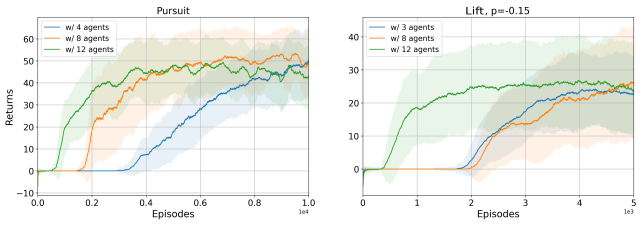



**图6：大规模场景下的结果。左：追捕中的结果。右：*lift*中的结果。**

因此，在实践中，我们使用预测模型 $p_r(o_i,\ f(u_i^-)\ a_i;\ \eta)$来学习编码器，因为学习到的动作类别应包含足够的信息，使得在给定智能体 $i$的观测和动作时，可以预测智能体 $i$ 的下一个奖励。通过最小化以下损失函数来训练整体模型，该损失函数由方程（5）推导得到，

$$\mathcal{L}_e(\phi,\eta) = (\sum_{i=0}^{N} p_r(o_i, f(u_i^-), a_i) - r_t)^2 \tag{13}$$

其中，$r_t$是全局奖励。最后，我们通过将 $u_i^-$替换为 $z_i \sim f(u_i^-)$，将$Q_c^i(\tau_i, u_i^-, a_i)$修改为$Q_c^i(\tau_i, z_i, a_i)$。该公式保证线性搜索时间为 $O(Z)$，其中 $Z$是预先定义的类别数量。

# 6 实验

## 6.1 环境与实验设置

我们设计了多个具有非单调收益的高难度协作机器人任务，以评估各方法。实验基于 MAgnet [29],开展，我们在其中实现了四个侧重点各异的任务：*lift*、*triplelift*、追捕 和*tiger*。所有任务都是非单调环境，要求所有智能体根据其他智能体的动作采取最优合作动作，否则智能体将受到惩罚。*lift* 和*triplelift* 要求两个或更多智能体、以及三个或更多智能体分别共同抬起一件货物。这些任务与双手抬举任务类似，但难度更高，因为智能体需要先移动到货物所在位置。追捕 是一种捕食者-猎物任务，需要两个或更多智能体共同攻击猎物并将其捕获。*tiger* 比追捕更困难，因为猎物会被智能体的攻击动作杀死，而*tiger* 中的猎物每个时间步都能恢复生命值，这促使智能体学会攻击生命值较高的猎物，并避免杀死猎物以获得更多奖励。我们将采用贪心动作选择的评估回合表现作为最终性能。性能通过每个智能体的平均回报进行评估。所有实验均使用五个随机种子进行。所有场景实现的更多细节见附录1。

在实验中，我们将方法 AVGM 与 QPD、Shapley、MAVEN 和 WQMIX 进行比较。为确保比较公平，所有方法采用相同的基本超参数和网络结构，并使用相近的参数设置。更多实验设置请参阅附录2。 *

*https://github.com/Locke637/AVGM

## 6.2 性能

我们在图3中展示了六种不同机器人场景的结果。在*lift*中，我们展示惩罚如何影响所有方法的学习过程。我们在*lift* 中评估所有方法，惩罚值从 0 逐渐增大到 -0.3。结果表明，只有 AVGM 能够解决所有场景，而随着惩罚增大，其他方法的性能均有所下降。尤其是，结果显示，随着惩罚增大，其他方法往往会学到一种懒惰策略，为了避免惩罚而从不采取协作动作。这一现象与我们在第3节中的先前分析一致。我们还注意到，在惩罚较大的场景中，MAVEN 和 WQMIX 的性能更好，因为它们的设计旨在缓解非单调性。然而，它们无法彻底解决非单调问题，因为在惩罚最大的一些任务中仍然失败。此外，QPD 性能不佳是由于未考虑智能体之间的交互；而 Shapley 在较难场景中失败，是因为集中式评论家网络不足以表示最优动作的真实边际贡献，从而导致次优策略。在*pursuit*中，结果与*lift* 中的情况类似，只有 AVGM 学到了最优策略。在*triplelift*中，我们发现，随着合作变得越来越难以实现，其他方法的性能逐渐下降。然而，AVGM 仍能通过考虑其他智能体的动作并搜索最优边际贡献来解决任务。最后，在*tiger* 中，其他所有方法均告失败，因为低效的探索会导致该任务中的所有猎物被杀死，使得找到合作模式变得困难。结果表明，AVGM 在非单调环境中的学习效率很高。

## 6.3 可解释性分析

本节分析我们的方法的若干特性，以展示 AVGM 的可解释性。为更好地进行说明，我们选取了*lift* 中一个具有代表性的情形，其惩罚值为 -0.3，此时所有智能体都处于正确位置，准备抬起货物。示意图见附录 3。该情形适合用于展示可解释性，因为智能体必须考虑彼此的动作，才能决定采取合作动作还是懒惰行动。

在图 4 上方，我们展示了动作编码器在训练过程中的输出变化。我们说明，动作编码器学会了对其他智能体的动作进行有意义的类别表示。该表示会根据其他智能体是否采取抬升动作或移动动作对其动作进行分类。我们注意到，编码器首先学会了两名智能体都在抬升或移动的简单情形。经过进一步训练，编码器学会了对其余动作组合进行分类。

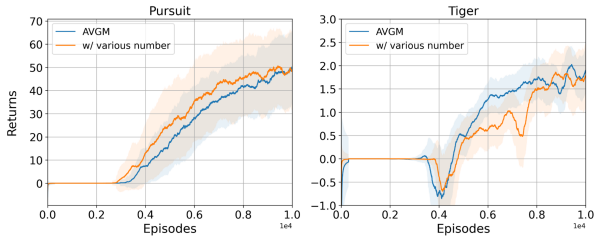



**图 7：可扩展性的结果。左：追捕中的结果。右：*tiger*中的结果。**

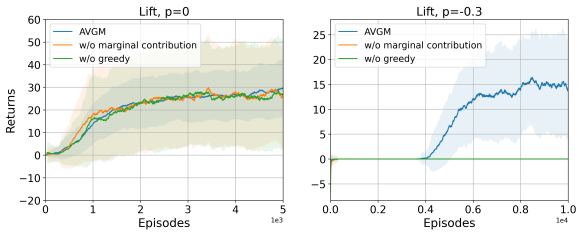



**图 8：边际贡献的消融研究结果。左：惩罚为 0 时的*lift* 结果。右：惩罚为 -0.3 时的*lift* 结果。**

此外，我们在下方展示了边际贡献的可解释性。我们比较了在其他智能体采取协作动作或懒惰动作时，采取最优升降动作和懒散停留动作的边际贡献。结果表明，随着训练的推进，在其他智能体采取抬升动作的情况下，采取升降动作的边际贡献相较于其他智能体采取懒惰动作时，优势越来越大。同时，在其他智能体采取懒惰动作时，采取停留动作的边际贡献大于其他智能体采取协作动作时的边际贡献。结果表明，边际贡献是合理的，因为它与基于先验知识的分析一致。因此，我们展示了方法的可解释性，这也为其在现实场景中的进一步应用提供了潜力，因为我们能够解释每个智能体的行为并降低不可预测的风险。最后，我们还展示了训练过程中所有智能体的 $Q_c^i(\tau_i,\ u_i^{-*},\ a_i^*)$ 方差；附录 4 中的结果表明，在考虑其他智能体动作的情况下，智能体能够学会对环境形成一致的感知。

## 6.4 消融实验

*6.4.1* 不同维度的比较 $z$。 我们比较了 AVGM 在追捕中使用不同 $z$ 8, 16, 维 24, 32, 40 5 度（ ）时的性能。结果如图 所示。结果表明，当维度数量超过 16 时，使用不同 $z$ 维度对最终性能的影响有限。然而，使用 8 维 $z$ 会显著降低性能，因为如此低维的潜在动作嵌入不足以表示所有必要的行为模式。这体现了 AVGM 在合理范围内对 $z$ 维度这一超参数的鲁棒性。

*6.4.2* 不同智能体选择方法的比较。我们比较了在追捕中使用所有智能体、仅使用可见智能体以及使用部分可见智能体时的性能。结果如图5所示。具体而言，使用部分可见智能体是指随机舍弃一半可见智能体。结果表明，使用所有智能体与仅使用可见智能体的性能相近。这说明可见智能体之间的信息足以表示最优策略，这与我们的分析一致。与此同时，使用部分可见智能体的性能显著下降。这表明，缺少可见智能体的联合动作 $u_i^-$会导致次优策略。

*6.4.3* 可扩展性。首先，我们比较了追捕场景中智能体数量为（4、8、12）以及*lift*场景中智能体数量为（3、8、12）时，大规模场景下的性能。结果如图6所示。结果表明，所有设置下的最终性能都相近。这说明增加智能体数量对 AVGM 的性能影响有限，因为与环境中智能体的总数相比，每个智能体视野内的智能体数量要少得多且有限。这表明，AVGM 能够在大规模场景中学习，而对于学习具有完整表达能力的价值函数的方法而言，这些场景是难以处理的。此外，使用更多智能体进行训练可以在训练早期学会合作，因为策略是在更多经验的基础上训练的，发现合作模式的可能性也会更高。另外，由于 AVGM 的结构能够处理可变的输入维度，且策略能够学会与可扩展智能体合作，我们研究了学习策略能否在智能体数量不同的场景之间进行零样本迁移。我们在追捕和*tiger* 场景中测试了我们方法的可扩展性，其中智能体数量在 3 至 5 之间变化。图 7 中的结果表明，我们的方法能够推广到智能体数量不同的场景，因为在测试环境中性能没有显著下降。

*6.4.4* 边际贡献的消融实验。 我们还针对边际贡献进行了消融研究，包括使用实际边际贡献而非贪心边际贡献，以及不使用边际贡献、直接通过使用 $Q_c^i(\tau_i, u_i^{-*}, a_i)$ 优化策略。消融研究在惩罚为 0 和 -0.3 的*lift*场景中进行。结果如图 8 所示。我们发现，缺少方法中的这些部分会导致非单调场景下的性能显著下降，而在单调场景中的影响有限。

## 7 结论

在这项工作中，我们提出了一种新颖的显式信用分配方法，用于解决具有非单调回报的机器人任务。我们的方法 AVGM 基于自适应价值分解，在考虑其他可观测智能体动作的基础上，利用贪心边际贡献来鼓励智能体学习最优协作策略。这使我们的方法能够在具有非单调回报的任务中学习合作，而这对现有的基于价值的集中训练、分散执行方法来说是一项挑战。此外，我们提出了一种动作编码器，以保证 AVGM 的时间复杂度为线性。实验表明，AVGM 在性能方面有显著提升，且具有可解释性。

## 8 致谢

本论文的出版获得了国家自然科学基金（NSFC）62088101“自主智能无人系统”项目以及中国国家自然科学基金委员会（编号：U21A20484）资助。

## 参考文献

[1] RC Bonitz 和 Tien C Hsia。1996。《面向协作机械臂的基于内力的阻抗控制》。*IEEE*机器人与自动化汇刊 12，1（1996），78–89。[2] Fabrizio Caccavale、Pasquale Chiacchio、Alessandro Marino 和 Luigi Vil-lani。2008。《双臂协作机械臂的六自由度阻抗控制》。*IEEE/ASME*机电一体化汇刊 13，5（2008），576–586。[3] Jakob Foerster、Gregory Farquhar、Triantafyllos Afouras、Nantas Nardelli 和 Shimon Whiteson。2018。反事实多智能体策略梯度。载于*AAAI*人工智能会议论文集，第32卷。[4] Alex Graves、Greg Wayne 和 Ivo Danihelka。2014。神经图灵机。*arXiv*预印本 *arXiv:1410.5401*（2014）。[5] Tarun Gupta、Anuj Mahajan、Bei Peng、Wendelin Böhmer 和 Shimon Whiteson。2021。Uneven：面向多智能体强化学习的通用价值探索。载于国际机器学习大会。PMLR，3930–3941。[6] Wan Ju Kang、David Earl Hostallero、Kyunghwan Son、Daewoo Kim 和 Yung Yi。2019。QTRAN：通过变换学习协作式多智能体强化学习的因子分解。载于第*31*届国际机器学习大会论文集，机器学习研究论文集。*PMLR*。[7] Jian Hu、Siyang Jiang、Seth Austin Harding、Haibin Wu 和 SW Liao。2021。重新审视协作式多智能体强化学习中的实现技巧与单调性约束。*arXiv*预印本 *arXiv:2102.03479*（2021）。[8] Maximilian Hüttenrauch、Adrian Šošić 和 Gerhard Neumann。2017。面向群体系统的引导式深度强化学习。*arXiv*预印本 *arXiv:1709.06011*（2017）。[9] Shariq Iqbal 和 Fei Sha。2019。用于多智能体强化学习的演员-注意力-评论家。载于国际机器学习大会。PMLR，2961–2970。[10] Kenneth E Kinnear、William B Langdon、Lee Spector、Peter J Angeline 和 Una-May O'Reilly。1994。遗传编程进展。第3卷。麻省理工学院出版社。[11] Jinoh Lee、Pyung Hun Chang 和 Rodrigo S Jamisola。2013。双臂机器人执行非对称双手任务时的相对阻抗控制。*IEEE*工业电子汇刊61，7（2013），3786–3796。[12] Jiahui Li、Kun Kuang、Baoxiang Wang、Furui Liu、Long Chen、Fei Wu 和 Jun Xiao。2021。用于多智能体强化学习的 Shapley反事实信用分配。*arXiv*预印本 *arXiv:2106.00285*（2021）。[13] Shanqi Liu、Licheng Wen、Jinhao Cui、Xuemeng Yang、Junjie Cao 和 Yong Liu。[无日期]。编队前进：一种多智能体协同移动的分散式分层学习方法。载于*2021 IEEE/RSJ*智能机器人与系统国际会议（*IROS*）。IEEE，4777–4784。[14] Ryan Lowe、Yi Wu、Aviv Tamar、Jean Harb、Pieter Abbeel 和 Igor Mordatch。2017。面向混合合作竞争环境的多智能体演员-评论家。*arXiv*预印本 *arXiv:1706.02275* (2017)。[15] Anuj Mahajan、Tabish Rashid、Mikayel Samvelyan 和 Shimon Whiteson。2019。Maven：多智能体变分探索。*arXiv*预印本 *arXiv:1910.07483* (2019)。[16] Junhyuk Oh、Valliappa Chockalingam、Honglak Lee 等。2016。Minecraft 中的记忆、主动感知与动作控制。载于国际机器学习大会。PMLR，2790–2799。[17] Liviu Panait、Sean Luke 和 R Paul Wiegand。2006。通过偏置协同进化搜索优化多智能体行为。*IEEE*进化计算汇刊 10，6 (2006)，629–645。[18] Daniel Rakita、Bilge Mutlu、Michael Gleicher 和 Laura M Hiatt。2019。基于共享控制的双臂机器人操作。*Science Robotics* 4，30 (2019)，eaaw0955。[19] Tabish Rashid、Gregory Farquhar、Bei Peng 和 Shimon Whiteson。2020。Weighted QMIX：扩展深度多智能体强化学习的单调价值函数分解。*arXiv*预印本 *arXiv:2006.10800* (2020)。[20] Tabish Rashid、Mikayel Samvelyan、Christian Schroeder、Gregory Farquhar、Jakob Foerster 和 Shimon Whiteson。2018。QMIX：深度多智能体强化学习的单调价值函数分解。载于国际机器学习大会。PMLR，4295–4304。[21] Lloyd S Shapley。2016。*17. n*人博弈的一个价值。普林斯顿大学出版社。[22] Kyunghwan Son、Daewoo Kim、Wan Ju Kang、David Earl Hostallero 和 Yung Yi。2019。QTRAN：通过变换学习分解合作式多智能体强化学习。载于国际机器学习大会。PMLR，5887–5896。[23] Peter Sunehag、Guy Lever、Audrunas Gruslys、Wojciech Marian Czarnecki、Vinicius Zambaldi、Max Jaderberg、Marc Lanctot、Nicolas Sonnerat、Joel Z Leibo、Karl Tuyls 等。2017。用于合作式多智能体学习的价值分解网络。*arXiv*预印本 *arXiv:1706.05296* (2017)。[24] Lipeng Wan、Zeyang Liu、Xingyu Chen、Han Wang 和 Xuguang Lan。2021。用于多智能体强化学习最优协调的基于贪心的价值表示。*arXiv*预印本 *arXiv:2112.04454* (2021)。[25] Jianhao Wang、Zhizhou Ren、Terry Liu、Yang Yu 和 Chongjie Zhang。2020。QPLEX：双重决斗多智能体Q学习。*arXiv*预印本 *arXiv:2008.01062* (2020)。[26] Jianhong Wang、Yuan Zhang、Tae-Kyun Kim 和 Yunjie Gu。2020。Shapley Q值：解决全局奖励博弈的局部奖励方法。载于*AAAI*人工智能会议论文集，第34卷。7285–7292。[27] Chenguang Yang、Yiming Jiang、Zhijun Li、Wei He 和 Chun-Yi Su。2016。具有全局稳定性保证和运动精度的双臂机器人神经控制。*IEEE*工业信息学汇刊 13，3 (2016)，1162–1171。[28] Yaodong Yang、Jianye Hao、Guangyong Chen、Hongyao Tang、Yingfeng Chen、Yujing Hu、Changjie Fan 和 Zhongyu Wei。2020。深度多智能体强化学习的Q值路径分解。载于国际机器学习大会。PMLR，10706–10715。[29] Lianmin Zheng、Jiacheng Yang、Han Cai、Ming Zhou、Weinan Zhang、Jun Wang 和 Yong Yu。2018。MAgent：面向人工群体智能的多智能体强化学习平台。载于*AAAI*人工智能会议论文集，第32卷。

## A 场景设置与训练详情

在 MAgent 中，每个智能体对应一个网格，并拥有局部观测，其中包含以该智能体为中心的方形视野，以及一个特征向量，包含附近智能体的坐标、生命值（HP）和 ID，以及该智能体的上一个动作。离散动作包括移动、停留和攻击。我们选择了四种不同的场景：*lift*、*triplelift*、追捕 和*tiger*。*lift* 和*triplelift* 要求两个或更多以及三个或更多智能体共同抬起一件货物。这与双手抬举任务类似，但难度更高，因为智能体需要先移动到货物旁。追捕 是一种捕食者-猎物任务，需要两个或更多智能体共同攻击猎物并将其捕获。*tiger* 难度更高，因为猎物会被智能体的攻击动作杀死，而*tiger* 中的猎物每个时间步都能恢复生命值，这促使智能体学会攻击生命值较高的猎物，并避免杀死猎物以获得更多奖励。这些场景的详细设置如表1所示。我们通过展示合作成功时返回的奖励 $R$ 来呈现收益矩阵，而 $P$ 表示采取合作动作但未能实现合作时受到的惩罚。MAgent 的全局状态是全局信息的小地图（$6 \times 6$）。实验中使用的对手策略是在追捕 和*tiger*场景中的随机逃逸策略。

| | Lift | 追捕 | triplelift | tiger |
|---|---|---|---|---|
| 智能体数量 | 3 | 4 | 3 | 4 |
| 敌人数量 | 3 | 4 | 3 | 4 |
| 地图尺寸 | $6 \times 6$ | $7 \times 7$ | $6 \times 6$ | $7 \times 7$ |
| Patoff | R=1,P=-0.3 | R=1.5 ,P=-0.3 | R=2,P=-0.005 | R=1.5 ,P=-0.3 |

**表1：MAgent场景设置。**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | A | 0 | 0 |
| 0 | A | C | A | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

**表2：示意图**

我们将折扣因子设为 0.99，并使用学习率为 5e-4 的 RMSprop优化器。使用 $\epsilon$-贪心进行探索，其中 $\epsilon$ 从 1.0 到 0.05 在 700k 步内线性退火。批量大小为 100，每 200 个回合更新一次目标网络。MAgent 中每个回合的长度限制为 100 步。我们使用不同的随机种子运行所有实验五次，并在所有图中绘制均值/标准差。所有实验均在同一台计算机上进行，配置为 Intel i 7-7700K、64GB RAM 和 NVIDIA GTX3090。系统为 Ubuntu 18.04，使用 PyTorch 框架。

## B 模型实现细节和超参数

所有对比方法的网络均使用相同的 LSTM网络，由一个包含 GRU 的循环层组成，其隐藏状态维度为 64，前面设有一个全连接层，后面设有两个全连接层。所有混合网络均使用一个隐藏状态维度为 32 的全连接层。我们的评论家网络和策略网络使用两个隐藏状态维度为 64 的全连接层，后接一个隐藏状态维度为 32 的全连接层。动作编码器网络使用一个隐藏状态维度为 64 的全连接层，并使用另一个隐藏状态维度为 64 的全连接层输出类别结果。所有实验中，类别潜在空间的大小均为 32。预测模型使用两个隐藏状态维度为 64 的全连接层，后接一个隐藏状态维度为 32 的全连接层。

## C 可解释性分析部分示意图

实验中使用的示例如表 2 所示，其中表格中的 $A$ 表示智能体，0 表示空白位置，$C$ 表示货物。这种情况下，所有智能体都处于正确位置，准备抬起货物。然而，智能体必须考虑彼此的动作，以决定是采取协作动作来完成协作任务，还是采取懒惰行动以避免惩罚。

## D 所有智能体个体效用的方差分析

图 9 展示了训练过程中所有智能体的 $Q_c^i(\tau_i,\ u_i^{-*},\ a_i^*)$ 方差。该数值表示在已知所有其他智能体都会采取协作动作的信念条件下，每个智能体采取最优协同行动时的 Q值。根据我们的方法，随着训练的进行，方差应当逐渐减小。这是因为智能体在考虑其他智能体的动作时，会学会对环境形成一致的认知。结果与我们的结论一致。值得注意的是，训练初期方差上升，是因为价值函数的学习可能会使方差高于初始随机生成的数值。该结果证明，自适应个体效用能够学会合作的正确价值，而这正是显式信用分配的基础。

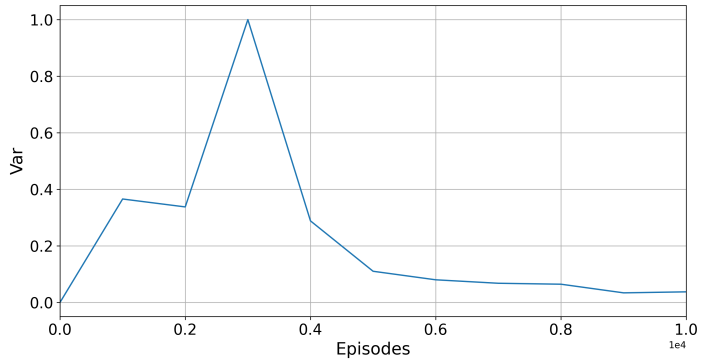



**图 9：训练过程中所有智能体个体效用方差的结果** **s.**

## E 价值分解证明

首先，由于 t 环境的总奖励由智能体之间各种可能的交互产生，我们有 e

$$r_{tot}(s,u) = \sum_{i=1}^{N} \hat{r}_1(o_i,a_i) + \sum_{i<j=2}^{i<j=N} \hat{r}_2(o_i,o_j,a_i,a_j) + ... + \hat{r}_N(o_1,...,o_N,a_1,...,a_N). \tag{14}$$

并且我们假设，合作模式是指在分散式执行设置中，每个智能体只能与自身能够观测到的智能体进行交互。

按照这一原则，我们定义 $r^i_{team}(o_i,\ u_i^-,\ a_i)$，并将其分解为：

$$r^i_{team}(o_i,u_i^-,a_i) = \hat{r}^i_1(o_i,a_i) + \sum_{j=1,j\neq i}^{M_i} \hat{r}^i_2(o_i,a_j,a_i) + ... + \hat{r}^i_{M_i}(o_i,a_1,...,a_{M_i},a_i). \tag{15}$$

我们注意到，对于 $\hat{r}^i_m = \hat{r}^i_m(o_i,\ a_j,\ ...a_m,\ a_i)$，如果 $\hat{r}^i_m$ 不为零，则有

$$\hat{r}^i_m(o_i,a_j,...a_m,a_i) = \frac{1}{m}\hat{r}_m(o_i,o_j,...,o_m,a_j,...a_m,a_i). \tag{16}$$

原因是，如果 $\hat{r}_m$ 存在，它表示智能体集合 $(i,\ j,\ ..,\ m;\ j<i<m)$ 达成的合作。由于集合中的所有智能体都采取了特定的联合动作，因此每个智能体都应平等分享合作奖励 $\hat{r}_m$。接着，我们可以将方程 (15) 修改为

$$\begin{aligned} r^i_{team}(o_i,u_i^-,a_i) = \hat{r}_1(o_i,a_i) + \tfrac{1}{2}\sum_{j=1,j\neq i}^{M_i}\hat{r}_2(o_i,o_j,a_j,a_i) + ... \\ + \tfrac{1}{M_i}\hat{r}_{M_i}(o_i,o_1,...,o_{M_i},a_1,...,a_{M_i},a_i). \end{aligned} \tag{17}$$

我们将所有 $r^i_{team}(o_i,\ u_i^-,\ a_i)$ 相加

$$\begin{aligned} \sum_{i=1}^{N} r^i_{team}(o_i,u_i^-,a_i) &= \sum_{i=1}^{N}\hat{r}_1(o_i,a_i) + \sum_{i=1}^{N}\tfrac{1}{2}\sum_{j=1,j\neq i}^{M_i}\hat{r}_2(o_i,o_j,a_j,a_i) + ... \\ &= \sum_{i=1}^{N}\hat{r}_1(o_i,a_i) + \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{M_i}\hat{r}_2(o_i,o_j,a_j,a_i) + \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N-M_i} 0 + ... \end{aligned} \tag{18}$$

其中，0 表示智能体 $i$ 视野之外的智能体，其对应的奖励为零。接着，我们有

$$\begin{aligned} &\sum_{i=1}^{N}\hat{r}_1(o_i,a_i) + \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{M_i}\hat{r}_2(o_i,o_j,a_j,a_i) + \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N-M_i} 0 + ... \\ &= \sum_{i=1}^{N}\hat{r}_1(o_i,a_i) + \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\hat{r}_2(o_i,o_j,a_j,a_i) + ... \\ &= \sum_{i=1}^{N}\hat{r}_1(o_i,a_i) + \tfrac{1}{2}\sum_{i<j=2}^{i<j=N} 2\times\hat{r}_2(o_i,o_j,a_j,a_i) + ... \\ &= \sum_{i=1}^{N}\hat{r}_1(o_i,a_i) + \sum_{i<j=2}^{i<j=N}\hat{r}_2(o_i,o_j,a_j,a_i) + ... \\ &= r_{tot}(s,u). \end{aligned} \tag{19}$$

因此，我们有

$$r_{tot}(s,u) = \sum_{i=1}^{N} r^i_{team}(o_i,u_i^-,a_i). \tag{20}$$

接着，我们定义价值分解 $Q^i_c$，用于对每个智能体的个体效用进行建模。根据方程 (20)，我们有

$$Q_{tot}(s,u) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r_{tot}\ (s,u) \mid \pi\right] = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t \sum_{i=1}^{N} r^i_{team}(o_i,u_i^-,a_i) \mid \pi\right] = \sum_{i=1}^{N} Q^i_c(s,u). \tag{21}$$

此外，我们有

$$\arg\max_{a_i}(Q_{tot}(s,u)) = \arg\max_{a_i}(Q^i_c(s,u)) = \arg\max_{a_i}(Q^i_c(\tau_i,u_i^-,a_i)). \tag{22}$$

| | C | L |
|---|---|---|
| C | +R | -P |
| L | -P | 0 |

**表 3：非单调收益矩阵**

第一部分是因为 $a_i$ 由项目 $Q_c^i$ 表示；第二部分的原因是， $Q_c^i$ 仅与智能体 $i$以及智能体 $i$能够观测到的所有智能体有关，而它们所有必要的信息都包含在 $(\tau_i,\ u_i^-,\ a_i)$ 中，因此我们可以得到给定 $(\tau_i,\ u_i^-,\ a_i)$ 时 $Q_c^i$ 的无偏估计值。因此，根据公式 21 和公式 22，我们有

$$\arg\max_u(Q_{tot}(s,u)) = \{\arg\max_{a_1}(Q_c^1(\tau_1,u_1^-,a_1)),...,\arg\max_{a_N}(Q_c^N(\tau_N,u_N^-,a_N))\}. \tag{23}$$

对公式 23 的直观理解是，每个智能体根据对其他合作智能体动作的感知采取动作，因此它们能够采取相应的合作动作，并且联合动作是最优的合作联合动作。总而言之，公式 23 表明，在给定真实的 $Q_{tot}(s,u)$ 的情况下，效用 $Q_c^i(\tau_i,u_i^-,a_i)$ 可以满足 IGM 原则，而不是像 QMIX 等其他方法所学习的单调集中式价值函数。这证明了，使用效用 $Q_c^i(\tau_i,u_i^-,a_i)$ 的自适应价值分解能够在给定任意奖励函数的情况下表示无偏的价值分解，并满足分散式执行的 IGM 原则。因此，我们证明了定理 1。

## F 个体效用局限性分析

我们分析了表 3 中非单调收益矩阵下个体效用 $Q_i(\tau_i,a_i)$ 的局限性。我们指出，个体效用 $Q_i(\tau_i,a_i)$ 应被视为从分布 $Q_c^i(\tau_i,u_i^-,a_i)$ 中采样的变量。根据这一结论，我们有 $Q_i(\tau_i,a_i)$ 的损失应为

$$\mathcal{L}_i = \sum_{k=1}^{K_i} p_k \cdot (\hat{Q}_c^i(\tau_i,u_i^{k-},a_i) - Q_i(\tau_i,a_i))^2. \tag{24}$$

其中，$\hat{}Q_c^i$表示真实的价值函数， $u_i^{k-}$表示 $u_i^-$和 $p_k$的某种组合发生的可能性。因此，$Q_i(\tau_i,\ a_i)$通过优化 $L_i$学习收敛后的价值；当 $L_i$最小时，我们得到收敛后的$\hat{}\ Q_i(\tau_i,\ a_i)$，

$$\hat{Q}_i(\tau_i,a_i) = \sum_{k=1}^{K_i} p_k \cdot \hat{Q}_c^i(\tau_i,u_i^{k-},a_i). \tag{25}$$

在非单调环境中，通常只有一个特定的联合动作能够促成合作，而任何智能体采取懒惰动作都可能导致任务失败。由此，我们得到合作动作 $a_i^*$的价值为

$$\hat{Q}_i(\tau_i,a_i^*) = \sum_{c=1}^{C_i} p_c \cdot \hat{Q}_c^i(\tau_i,u_i^{c-*},a_i^*) + \sum_{l=1}^{L_i} p_l \cdot \hat{Q}_c^i(\tau_i,u_i^{l-},a_i^*). \tag{26}$$

其中， $p_c$ 表示其他智能体采取协作动作 $u_i^{c-*}$的可能性， $p_l$ 表示其他智能体采取懒惰动作$u_i^{l-}$的可能性。此外，我们有

$$\sum_{c=1}^{C_i} p_c + \sum_{l=1}^{L_i} p_l = 1 \tag{27}$$

同理，我们得到懒惰动作 $a_i^-$的价值为

$$\hat{Q}_i(\tau_i,a_i^-) = \sum_{c=1}^{C_i} p_c \cdot \hat{Q}_c^i(\tau_i,u_i^{c-*},a_i^-) + \sum_{l=1}^{L_i} p_l \cdot \hat{Q}_c^i(\tau_i,u_i^{l-},a_i^-). \tag{28}$$

我们知道，当$\hat{}\ Q_i(\tau_i,\ a_i^-)$大于$\hat{}\ Q_i(\tau_i,\ a_i^*)$时，由 $Q_i(\tau_i,\ a_i)$表示的策略会失效，即

$$\begin{aligned}\hat{Q}_i(\tau_i,a_i^-) - \hat{Q}_i(\tau_i,a_i^*) = \sum_{c=1}^{C_i} p_c \cdot (\hat{Q}_c^i(\tau_i,u_i^{c-*},a_i^-) - Q_c^i(\tau_i,u_i^{c-*},a_i^*)) \\ + \sum_{l=1}^{L_i} p_l \cdot (\hat{Q}_c^i(\tau_i,u_i^{l-},a_i^-) - Q_c^i(\tau_i,u_i^{l-},a_i^*)) > 0\end{aligned} \tag{29}$$

将示例收益矩阵代入方程（29），　　.

$$\hat{Q}_i(\tau_i,a_i^-) - \hat{Q}_i(\tau_i,a_i^*) = p_c \cdot (-P-R) + p_l \cdot (0-(-P)) = (p_l-1)\cdot(P+R) + p_l \cdot P > 0 \tag{30}$$

这意味着由 $Q_i(\tau_i,\ a_i)$ 表示的策略将在以下情况下失效

$$R \cdot (1-p_l) < (2p_l-1)\cdot P. \tag{31}$$

这等于

$$\frac{R}{P} < \frac{2p_l-1}{1-p_l}. \tag{32}$$