期末作业

合作博弈智能中的协作方法

502022370001 曹博文[1]，502022370047 王春力[1]，502022370071 庄镇华[1]

1. 南京大学

摘要 多智能体协作一直是人工智能的研究热点，有助于解决人类社会的许多问题。由于近些年来深度强化学习展现出的惊人决策能力，利用其解决多智能体协作问题成为热门，并诞生了多智能体深度强化学习这一研究方向。因此，本文针对基于 Dueling DQN 的多智能体值分解深度强化学习设计高效的多智能体协作算法这一任务，进行了以下工作：

引言部分主要介绍多智能体深度强化学习的研究背景、意义与目标，以及多智能体值分解深度强化学习的研究现状。相关工作部分主要梳理多智能体深度强化学习的发展分支与脉络，主要有分散式、集中式和值分解式方法。其中值分解是多智能体深度强化学习的主流方法，如 VDN、QMIX 方法，但已有的值分解方法要么限制值分解的表达能力，要么放松 IGM 一致性的约束条件。方法部分介绍了 VDN、QMIX 等模型的实现方法，QPLEX 将 Dueling DQN 分解 Q 值的设计融合到基于值的多智能体强化学习算法中，实现了对现有多智能体强化学习算法的扩展和性能提升。实验部分主要是 VDN、QMIX、QPLEX 实验复现部分，在矩阵游戏、两态多智能体马尔可夫游戏和星际争霸 II 三种实验场景下进行了实验复现，证明 QPLEX 的优势在于其新颖的 Duplex Dueling 模块。

关键词 深度强化学习, 多智能体协作, 值分解, Agent 技术

1 引言

1.1 研究背景与意义

智能体（Agent）是人工智能（Artificial Intelligence, AI）领域中的一个重要概念。任何可以感知所处环境并能独立做出决策影响环境的对象都可以抽象为智能体。随着科技的发展以及人工智能领域各项技术的逐渐成熟，多智能体系统成为了研究人工智能的主流方向之一。多智能体系统是多个自主的，相互作用的智能体组成的集合，是分布式人工智能的一个重要分支，其研究目的是解决多个智能体构成相互协作的系统如何进行协作，对抗，交互，策略学习等诸多问题。

多智能体协作是指多个智能体之间在有限时间和有限资源的情况下相互协作完成一项任务或者分别完成复杂任务的某项子任务。早期的工作中人们一般利用人类专家的经验人为设计一些策略让

期末作业: 曹博文, 王春力, 庄镇华. 期末作业. Agent 技术.
Bowen CAO, Chunli Wang, Zhenhua ZHUANG,. Final Assignment. Agent.

智能体执行。这些提前编写好的策略通常无法应对动态实时复杂多变的环境，其性能上限完全取决于人类对解决该任务的思考。随着 AI 技术的发展，研究者开始尝试设计 AI 方法去解决多智能体协作问题，希望智能体可以根据环境当前状态输出最优的策略，并且在不同的环境中具有很强的泛化能力。强化学习（Reinforcement Learning, RL）作为一种通过持续地与环境交互并根据得到的反馈改进自身策略以最大化期望收益的机器学习方法天然契合了这一需求，因此强化学习与多智能体系统交叉融合形成的多智能体强化学习（Multi-Agent Reinforcement Learning, MARL）成为了 AI 中的重要研究方向。

随着计算设备性能的提高和海量数据的产生，深度学习（Deep Learning）开始兴起，深度神经网络（Deep Neural Network, DNN）展现出了强大的表征（representation）能力。这种方法被称作深度强化学习（Deep Reinforcement Learning, DRL)。研究者开始将 DRL 与多智能体系统结合，形成了多智能体深度强化学习（Multi-Agent Deep Reinforcement Learning, MADRL）这一研究方向。DeepMind 开发的 AlphaStar、OpenAI 开发的 OpenAI Five、腾讯公司开发的“绝悟”等游戏 AI 都能够达到甚至超越人类顶尖玩家的水平，为多智能体深度强化学习在无人控制系统、智能决策等诸多领域的应用前景提供了广阔想象空间。

1.2 研究现状

多智能体深度强化学习是 DRL 与多智能体系统交叉结合而产生的新领域。相较于单智能体深度强化学习，多智能体深度强化学习还存在环境非平稳性（non-stationarity)、智能体数量引起的维数灾难（curse of dimensionality)、多智能体信用分配（multi-agent credit assignment）和相对过度泛化（relative overgeneralization）等问题。

多智能体深度强化学习目前的主要工作分为四类：行为分析、通信学习、智能体建模、值分解方法，接下来对这四类多智能体算法的相关研究进行简要的介绍。

（1）**行为分析**主要研究将单智能体强化学习算法直接扩展到多智能体环境中，观察是否会出现新的行为（比如学会了合作、竞争或者产生了交互)。最早的算法是完全独立 Q 学习（Independent Q-Learning, 简称 IQL）[1], 是将 Q 学习直接运用到多智能体环境中，每个智能体将其他智能体完全当作环境的一部分，但是 IQL 算法因为多智能体环境的非平稳导致算法效果极不稳定。

（2）**通信学习**一般是通过假设各个智能体之间存在信息通信，智能体需要根据自身状态来判断是否与其他智能体通信。通信是多智能体交互过程中出现的重要特征，特别是智能体之间需要合作，而每个智能体又只能获得部分观测，此时环境中智能体需要通过通信来更好的达成一个共同目标。Peng[27] 提出了一种基于行动者-评论家算法框架的算法，假设所有智能体共享通信，即所有智能体拥有同样的全局观察。但是这样的假设太过严格，导致该算法有很大的局限性。

（3）**智能体建模**这类方法主要通过分析其他智能体的策略或者通过对目标等信息建立决策模型来更好的了解队友或者对手智能体。Neil [2] 在多智能体深度强化学习领域引入行为和脑科学中的 Theory of Mind(简称 ToM) 理论，用其他智能体的过去的动作轨迹来预测其他智能体未来的动作。通过建立神经网络来对不同智能体未来的行为进行预测，以此来指导自身下一步的动作。

（4）**基于值分解的方法**可以算是目前多智能体深度强化学习的最主流的工作，其核心的方法是先分散式地学习每个智能体的个体动作值函数，然后集中式地利用个体动作值拟合联合动作值函数。

值分解既使得联合动作值函数的计算复杂度线性增长，又考虑了其他智能体的信息，在可拓展性和平稳性之间取得了平衡。

2 相关工作

近些年来，值分解式方法成为了主流，其先分散式地学习每个智能体的个体动作值函数，然后集中式地利用个体动作值拟合联合动作值函数。值分解既使得联合动作值函数的计算复杂度线性增长，又考虑了其他智能体的信息，在可拓展性和平稳性之间取得了平衡。

2.1 值分解方法分类

值分解式方法大致可以被分为三类。

第一类方法重点聚焦**联合动作值函数的近似**。VDN [3] 将联合动作值函数分解为每个智能体动作值函数的简单和，从而将一个复杂的学习问题分解为多个局部的、更易学习的子问题。QMIX [4] 认为 VDN 采用的线性分解模式过于简单，严重限制了集中式联合动作值函数的表征能力。QMIX 引入超网络来学习联合动作值函数与个体动作值函数之间的非线性关系，同时令联合动作值函数和个体动作值函数满足单调性约束。实验表明其表征能力超过了 VDN。QTRAN [5] 进一步放宽 VDN 和 QMIX 的限制，假定满足个体全局最大 (Individual-Global-Max, IGM) 条件的任务都是可分解的，而 VDN 的可加性和 QMIX 的单调性都只是可分解性的两个充分条件。QTRAN 认为联合动作值函数可以转换为所有智能体的个体动作值函数与全局状态值的和。

第二类方法聚焦**在值分解的基础上引入其他机制来改善智能体协作**。Mahajan 等认为 QMIX 为确保 IGM 条件而采用的单调性限制容易收敛到局部最优策略，因此提出一个混合了值函数和策略梯度的方法 MAVEN，引入一个层次化控制的潜在策略空间，并通过最大化潜在空间和轨迹的互信息提高探索能力。NDQ [6] 在值分解中加入通信机制，通过引入两个信息论正则项，一个最大化个体动作值函数和通信消息的互信息以提高信息量，一个最小化智能体之间通信消息的熵以减小通信代价，促进了智能体之间更有效的协作。

第三类方法聚焦**基于协作图的分解模式的探索**。协作图是一种可表示多智能体系统协调需求的方法。之前所述的值分解式方法都可以看作是完全断开的协作图。一些工作利用协作图探索了其他的值分解模式。Castellini 等 [7] 通过大量实验发现，在很多协作问题中，仅使用个体动作值拟合联合动作值是不够的。可以将联合动作值函数表示为协作图，通过协作图的局部结构来拟合联合动作值，即联合动作值可以分解为局部联合动作值的组合。DCG [8] 同时考虑个体动作值和局部联合动作值，并使用消息传递来协调协作图中所有智能体之间贪婪动作的选择。DICG [8] 则在 DCG 的基础上通过引入自注意力机制动态确定隐式协作图的结构，并使用图卷积网络学习推理联合动作值。

2.2 现有值分解方法

值分解式方法混合了分散式方法和集中式方法的特性，先分散式地学习每个智能体的个体动作价值函数，然后集中式地利用个体动作价值拟合联合动作价值函数。在该方法的框架下，联合动作

价值函数的计算复杂度随智能体数量线性增长，此外还考虑了其他智能体的信息，在可拓展性和平稳性中取得了平衡。值分解式方法也大多采用 CTDE 框架进行训练。

值分解的核心就是将联合动作价值函数 Q_{tot} 看作是由每个智能体的个体动作价值函数 Q^i 线性或非线性组合起来的，即

$$Q_{tot}(\tau, u_1, ..., u_n) \approx Q_{tot}(\tau, Q^1, ..., Q^n).$$

直接对联合动作价值函数进行优化，通过梯度传播端到端地更新个体动作价值函数。

在值分解的框架下，通常将个体全局最大 (Individual-Global-Max, IGM) 条件作为智能体执行分散策略的原则。这样的假设确保了对联合动作价值函数和个体动作价值函数的动作选择保持一致。

定义 对于一个联合动作价值函数 $Q_{tot}(\tau, u) : T^N \times U \to R$，如果存在个体动作价值函数满足以下公式：

$$arg\max_u Q_u(\tau, u) = \begin{pmatrix} arg\max_{u^1} Q^1(\tau^1, u^1) \\ \vdots \\ arg\max_{x^n} Q^n(\tau^n, u^n) \end{pmatrix}.$$

那么称在 τ 的条件下 $[Q_i]$ 对 Q_{tot} 满足 IGM 条件。这种情况下，也称 $Q_{tot}(\tau, u)$ 可以因式分解为 $[Q^i(\tau^i, u^i)]$ 或 $[Q^i(\tau^i, u^i)]$ 是 $Q_{tot}(\tau, u)$ 的因子。

VDN 认为可以将复杂的多智能体协作任务进行分解，由于直接学习联合动作价值函数 $Q_{tot}(\tau, u)$ 比较困难，所以可以将每个智能体的个体动作价值函数 $Q_{tot}(\tau, u)$ 相加的和作为联合动作价值函数的逼近，即

$$Q_{tot}(\tau, u) = \sum_{i=1}^{n} \omega_i Q^i(\tau^i, u^i), \forall i \in N.$$

其中 ω_i 表示权重系数。VDN 中简单地将所有 ω_i 都设为 1，Qatten [9] 则利用多头注意力机制动态学习 ω_i 。这种假设被称作联合动作价值函数的可加性，可加性是 IGM 条件的充分条件。这种简单的分解方式使得每个智能体可以从联合动作价值函数中提取和构造分散式的策略。

QMIX 认为 VDN 简单地将联合动作价值函数 $Q_{tot}(\tau, u)$ 分解为个体动作价值函数 $Q^i(\tau^i, u^i)$ 的线性和的做法严重限制了联合动作价值函数可以表征的复杂性，而且忽略了训练过程中所有额外的状态信息。为了解决上述问题，QMIX 假设 $Q_{tot}(\tau, u)$ 和 $Q^i(\tau^i, u^i)$ 之间是非线性关系，并要求 $Q_{tot}(\tau, u)$ 对每个 $Q^i(\tau^i, u^i)$ 是单调的以满足 IGM 条件，即

$$\frac{\partial Q_{tot}(\tau, u)}{\partial Q^i(\tau^i, u^i)} \geqslant 0, \forall i \in N.$$

单调性是满足 IGM 的充分不必要条件。为了确保满足约束，QMIX 利用超网络生成拟合 $Q_{tot}(\tau, u)$ 的混合网络的权重参数，以确保混合网络的权重参数都是非负的。

3 方法

3.1 VDN 方法

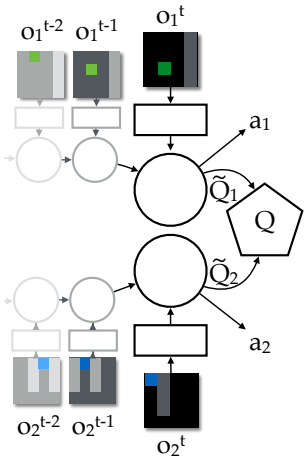

图 1 VDN 模型

VDN 方法主要的假设是，联合动作价值函数可以分解为单个智能体的价值函数

$$Q((h^1, h^2, ..., h^d), (a^1, a^2, ..., a^d)) \approx \sum_{i=1}^{d} \tilde{Q}_i(h^i, a^i),$$

其中 $\tilde{Q}_i$ 仅取决于每个智能体的局部观察。该方法通过使用联合奖励从利用 Q 学习规则梯度反向传播来学习 $\tilde{Q}_i$，即 $\tilde{Q}_i$ 是隐式学习的，而不是从特定于智能体 i 的任何奖励中学习的，并且不用假设 $\tilde{Q}_i$ 是任何特定奖励的动作价值函数。如图1所示，这种方法的一个特性是，尽管学习需要集中化训练，但可以独立地部署学习到的智能体，因为每个智能体的本地 $\tilde{Q}_i$ 贪婪行动就等价于中央决策器通过最大化总和 $\sum_{i=1}^{d} \tilde{Q}_i$ 来选择联合行动。

考虑具有 2 个智能体的情况（为了简化说明），并且其中奖励在智能体观察中进行累加分解，$r(\mathbf{s}, \mathbf{a}) = r_1(o^1, a^1) + r_2(o^2, a^2)$，其中 (o^1, a^1) 和 (o^2, a^2) 分别是智能体 1 和 2 的（观察，动作）。这可能是团队游戏的情况，例如，当智能体观察自己的目标，但不一定是队友的目标。如果 (o^1, a^1) 不足以完全建模 $\bar{Q}_1^{\pi}(s, a)$，则智能体 1 可以在其 LSTM 中存储来自历史观察的附加信息，或者在通信信道中从智能体 2 接收信息，在这种情况下，我们可以认为以下近似是有效的

$$Q^{\pi}(\mathbf{s}, \mathbf{a}) =: \bar{Q}_1^{\pi}(\mathbf{s}, \mathbf{a}) + \bar{Q}_2^{\pi}(\mathbf{s}, \mathbf{a}) \approx \tilde{Q}_1^{\pi}(h^1, a^1) + \tilde{Q}_2^{\pi}(h^2, a^2).$$

因此，VDN 架构鼓励这种简单的分解方法，并且为了减少可学习参数的数量，该方法在智能体之间共享某些网络权重，权重共享还产生了智能体不变性的概念，这对于避免懒惰智能体问题是有效的。

3.2 QMIX 方法

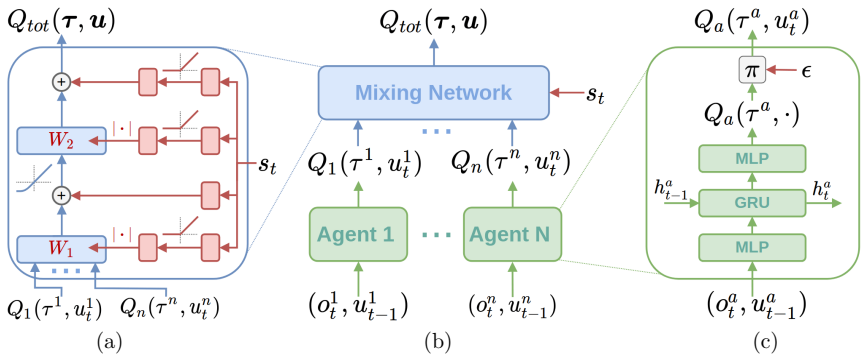

图 2 QMIX 模型

QMIX 方法的关键是认识到 VDN 为了提取与中央策略完全一致的分散策略，其完全累加分解是不必要的，在充分的动作观察历史条件下，存在一个确定性的最优策略。因此，只需要建立保证确定性贪婪分散策略和确定性贪婪集中策略之间一致性的最佳联合动作价值函数。

$$\operatorname{argmax}_{\mathbf{u}} Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \begin{pmatrix} \operatorname{argmax}_{u^1} Q_1(\tau^1, u^1) \\ \vdots \\ \operatorname{argmax}_{u^n} Q_n(\tau^n, u^n) \end{pmatrix}.$$

VDN 方法满足上式约束。然而，QMIX 提出一种宽松的约束，使得该表示可以推广到更大的单调函数簇。此时单调性被定义为对 Q_{tot} 与每个 Q_a 之间关系的约束：

$$\frac{\partial Q_{tot}}{\partial Q_a} \geqslant 0, \forall a \in A.$$

QMIX 的实现方式如下，即如图2，通过使用由智能体网络、混合网络和一组超网络组成的架构来表示 Q_{tot}。对于每个智能体 a，存在一个智能体网络表示其个体价值函数 $Q_a(\tau_a, u_a)$。它在每个时间步接收当前个体观察 o_t^a 和最后一个动作 u_{t-1}^a 作为输入。混合网络是前馈神经网络，其将智能体网络的输出作为输入并单调地混合它们，产生 Q_{tot} 的值。为了加强单调性约束，混合网络的权重（但不是偏置）被限制为非负值。

混合网络的每一层的权重由单独的超网络产生。每个超网络将状态 s 作为输入，并生成混合网络的一层的权重。每个超网络由两个具有 ReLU 非线性的全连接层组成，然后是绝对值激活函数，以确保混合网络权重是非负的。超网络的输出是一个向量，该向量被调整为适当大小的矩阵。偏置以相同的方式产生，第一个偏置由具有单个线性层的超网络产生，最终的偏置由具有 ReLU 非线性的双层超网络产生。

QMIX 最小化如下损失以实现端到端训练：

$$\mathcal{L}(\theta) = \sum_{i=1}^{b} \left[\left(y_i^{tot} - Q_{tot}(\tau, \mathbf{u}, s; \theta) \right)^2 \right].$$

3.3 QPLEX 方法

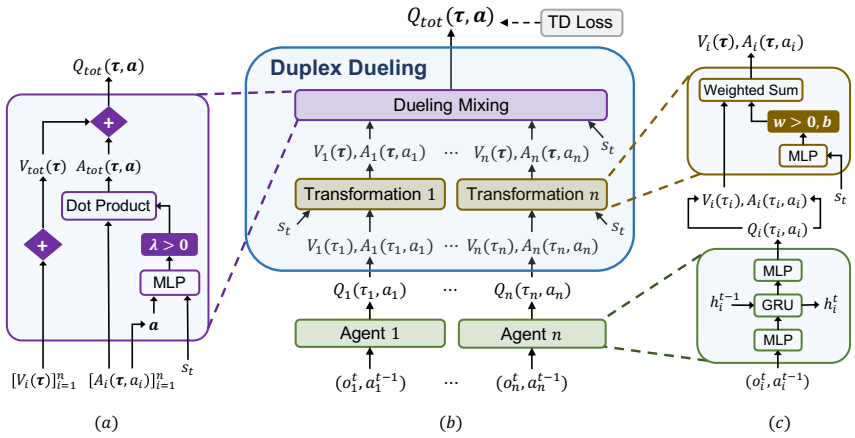

图 3 QPLEX 模型: (a) Dueling Mixing 网络 (b) QPLEX 整体架构 (c) 智能体网络结构 (底部) 以及 Transformation 网络结构 (顶部)

QPLEX 是 Dueling DQN 分解 Q 值思想在多智能体领域的完美应用。该算法属于值分解式多智能体深度强化学习方法，是集中训练分散执行 (CTDE) 的典型范式。集中训练分散执行式方法需要保持 Individual-Global-Max (IGM) 原则，即保持联合 Q 值取最优时的联合动作与单个智能体的 Q_i 值取最优的局部动作的一致。然而，现有的多智能体强化学习方法为了实现可扩展性，要么限制值分解的表达能力，要么放松 IGM 一致性的约束条件，而 QPLEX 是一种新颖的多智能体强化学习方法，它通过 Duplex Dueling 网络架构来分解联合 Q 值函数，完美实现了 IGM 原则，从而完成高效的联合 Q 值函数学习。

QPLEX 的整体架构如图3所示，主要由如下两个部分组成：(i) 每个智能体的动作价值函数以及 (ii) 在 Advantage-based IGM 约束条件下将独立动作价值函数转化成联合动作价值函数的 Duplex Dueling 结构。在集中训练阶段，整个网络以端到端的方式进行学习，优化目标为如下公式所示的时间差分损失。在分散执行阶段，Duplex Dueling 结构会被移除，每个智能体都依据自身的 Q 网络和局部动作观测历史来选择自身的最优动作。

$$\mathcal{L}(\theta) = E_{(\tau,a,\tau,\tau)\in D}[(r + \gamma V(\tau'; \theta^-)) - Q(\tau, a; \theta))^2].$$

Duplex Dueling 网络结构 Duplex Dueling 结构通过两个模块联系个体和联合动作价值函数：(i) 一个在训练过程中将全局状态和联合历史信息加入到个体动作价值函数的 Transformation 网络以及 (ii) 由个体动作价值函数生成联合动作价值函数的 Dueling Mixing 网络。Duplex Dueling 网络首先对每一个智能体推导出个体 Dueling 结构，即计算它的状态价值函数 $V_i(\tau_i) = max_{a_i} Q_i(\tau_i, a_i)$ 和优势函数 $A_i(\tau_i, a_i) = Q_i(\tau_i, a_i) - V_i(\tau_i)$，最后利用个体 Dueling 结构实现联合 Dueling 结构。

Transformation 网络结构 在联合动作观测历史的条件下，Transformation 网络利用全局信息将个体的 Dueling 结构 $[V_i(\tau_i), A_i(\tau_i, a_i)]_{i=1}^n$ 转化为 $[V_i(\tau), A_i(\tau, a_i)]_{i=1}^n$，如下式所示，对于任意智

能体 i，

$$Q_i(\tau, a_i) = \omega_i(\tau) Q_i(\tau_i, a_i) + b_i(\tau),$$
$$V_i(\tau) = \omega_i(\tau) V_i(\tau_i) + b_i(\tau),$$
$$A_i(\tau, a_i) = Q_i(\tau, a_i) - V_i(\tau) = \omega_i(\tau) A_i(\tau_i, a_i),$$

其中 $\omega_i(\tau) > 0$，这种正线性变换保持了贪婪动作选择的一致性，缓解了 Dec-POMDP 中部分可观察性所带来的弊端。如论文 QMIX、QTRAN、Qatten 所提到的，Transformation 网络利用到的全局信息可以是全局状态 s 或者联合动作观测历史 τ。

Dueling Mixing 网络结构 Dueling Mixing 网络将 Transformation 网络的输出 $[V_i, A_i]_{i=1}^n$ 当做输入，然后产生最终的输出 Q_{tot}。Dueling Mixing 网络使用个体 Dueling 结构来计算联合状态价值函数和联合优势函数，最后通过联合 Dueling 结构输出联合动作价值函数 $Q_{tot}(\tau, a) = V_{tot}(\tau) + A_{tot}(\tau, a)$，由于 Advantage-based IGM 对状态价值函数 V 没有约束，为了实现高效学习，文章使用简单的加和方式来定义联合状态价值函数：

$$V_{tot}(\tau) = \sum_{i=1}^{n} V_i(\tau),$$

为了保持联合优势函数和个体优势函数的一致性，QPLEX 通过如下方式计算联合优势函数：

$$A_{tot}(\tau, a) = \sum_{i=1}^{n} \lambda_i(\tau, a) A_i(\tau, a_i) \quad \lambda_i(\tau, a) > 0,$$

联合优势函数是个体优势函数与正权重的点积，正权重和联合历史动作有关，权重的正性会维持贪婪动作选择的一致性，权重的加性保证了值分解表达能力的完整性。为了实现对基于联合历史动作的权重的有效学习，QPLEX 使用了一种可扩展的多头注意力机制：

$$\lambda_i(\tau, a) = \sum_{k=1}^{K} \lambda_{i,k}(\tau, a) \phi_{i,k}(\tau) v_k(\tau),$$

K 是注意力机制的头数，$\lambda_{i,k}(\tau, a)$ 和 $\phi_{i,k}(\tau)$ 是经过 sigmoid 正则化后的注意力权重，$v_k(\tau) > 0$ 是每个注意力机制头的键值。λ_i 的 sigmoid 正则化激活联合动作价值函数的个体信用分配带来了稀疏性，这也使得多智能体能够更加高效的学习。最终的联合动作价值函数可以被化简成以下形式

$$Q_{tot}(\tau, a) = V_{tot}(\tau) + A_{tot}(\tau, a) = \sum_{i=1}^{n} Q_i(\tau, a_i) + \sum_{i=1}^{n} (\lambda_\tau(\tau, a) - 1) A_i(\tau, a_i),$$

可以看到，Q_{tot} 由两项构成，第一项是个体动作价值函数 $[Q_i]_{i=1}^n$ 的加和，也是 Qatten 论文中的联合动作价值函数 Q_{tot}^{Qatten}，第二项纠正了 Q_{tot}^{Qatten} 和真实 Q_{tot} 值之间的误差，这一项也是 QPLEX 对值分解方法完整表达能力的主要贡献。

4 实验比较

4.1 实验环境

共基于三种环境进行了实验验证，分别为：矩阵游戏（Matrix Games），两态多智能体马尔科夫游戏（Two-state MMDP）和星际争霸 II 微场景（StarCraft Multi-Agent Challenge, SMAC）。

4.1.1 矩阵游戏

游戏有两个智能体参与，每个智能体都有三种动作，根据不同智能体的不同动作，得到的不同回报值构成回报矩阵，并且游戏仅有一个最佳联合动作 (A, A)，此时获得的回报值最大。虽然这是一个简单的合作式多智能体任务，但也很容易陷入局部最优值，多智能体的目标就是经过不断的迭代尝试找出最佳联合动作组合使得回报值最大。

例如表1所示，当智能体 a_1、a_2 采取联合动作 $(A^{(1)}, A^{(1))}$ 时，可以获得最大的回报值 8，而当它们采取联合动作 $(A^{(2)}, A^{(2))}$ 或 $(A^{(3)}, A^{(3)})$ 时，就会陷入次优解 6。

$a_1 a_2$	$A^{(1)}$	$A^{(2)}$	$A^{(3)}$
$A^{(1)}$	**8**	-12	-12
$A^{(2)}$	-12	6	0
$A^{(3)}$	-12	0	6

表 1 矩阵游戏的回报值矩阵

4.1.2 两态多智能体马尔可夫游戏

多智能体马尔可夫决策过程 (MMDP) 是具有完全可观察性、完全合作式的多智能体任务。对于两态 MMDP，它包含两个智能体、两个动作和一个奖励（见图4）。两个智能体将从状态 $s2$ 开始，探索 100 个环境步骤并试图获得外部奖励。最优的 MMDP 的策略是两个智能体在状态 $s2$ 都执行动作 $A^{(1)}$，这也是唯一能够获得正奖励的联合动作。

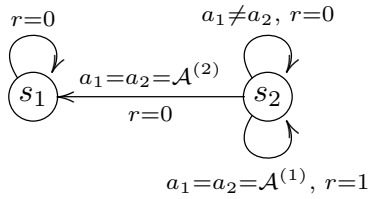

图 4 两态多智能体马尔可夫游戏

4.1.3 星际争霸 II 微场景

即时战略 (Real Time Strategy) 游戏是强化学习社区中具有挑战性的任务，星际争霸是其中的典型代表。SMAC 是一个聚焦微观管理（Micromanagement）的多智能体协作对抗场景，即对团队内每个智能体进行细粒度的控制对抗敌方团队。现在 SMAC 已经成为多智能体强化学习常用的一个基准测试环境。

图 5 星际争霸 II 微场景

状态和观察空间

在 SMAC 中，每个智能体需要根据自己视野范围内的局部观察信息决定自己的执行动作，因此对所有智能体而言这是一个部分可观察的环境。观察信息包括相对距离，坐标 x，坐标 y，健康值，防护值，单位类型，其中防护值并不是所有单位都具有的属性。此外，智能体只能在存活状态下才能观察其他单位，并且无法区别其他单位是死亡还是超出视野范围。

动作空间

SMAC 的动作空间都是离散的，每个智能体可以采取移动 [方向]、攻击 [单位编号]、停止和不操作这些动作，其中智能体只能朝东南西北四个方向移动，攻击 [单位编号] 也只能对智能体攻击范围内出现敌方单位使用，不操作当且仅当智能体死亡时才可使用。医疗运输机（Medivac）是一个辅助单位而非进攻单位，因此需要将其动作空间中的攻击 [单位编号] 替换为治疗 [单位编号]。视野范围的区域比攻击范围的区域更大，因此智能体需要先移动寻找敌方单位，然后靠近敌方单位到达攻击范围内才能攻击。

环境奖励

所有存活的智能体在每个时间步都会得到一个环境奖励，环境奖励等于对敌方单位造成的总伤害。此外，当智能体消灭一个敌方单位时，会得到 10 点的额外奖励；当赢得对战时，会得到 200 点的最终奖励。在所有场景中可获得的最大累积奖励都会等比例缩放到一个较小的值。在对抗中，如果一方消灭另一方的所有单位则会获得胜利。同时一个场景有其最大时间步限制，一旦双方对抗时间超过这个限制，环境便会判定双方平局。

本章选择在 2s3z、2s_vs_1sc、MMM、3s5z 和 1c3s5z 等 17 个场景中进行算法性能评估。

4.2 评价指标

4.2.1 矩阵游戏

矩阵游戏的评价指标较为简单，就是一定迭代次数后的智能体**估计的最优回报值**，估计的最优回报值与实际的最优回报值越接近则说明模型越优秀。

4.2.2 两态多智能体马尔可夫游戏

根据算法学习到的联合动作价值 Q_{tot} 是否**收敛**到真实最优奖励值 100 进行评价。

4.2.3 星际争霸 II 微场景

测试胜率

判断不同算法的性能指标为**在环境中对抗内置 AI 的胜率**。

各个算法在每训练 10000 个时间步后便会额外执行 32 个回合，在执行额外回合时多智能体算法不使用任何探索的手段，完全利用现有的策略。之后通过统计获胜的次数来计算出胜率。

由于平局的原因大多数是因为多智能算法由于部分观察的原因没有探测到敌人的位置导致的，并没有完成预定的任务，所以在处理中，所有的平局并没有按比例计入获胜的局数。在完成所有 32 个额外回合后，统计胜率并记录，作为算法的指标。

4.3 实验结果与分析

4.3.1 矩阵游戏

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	**8**	-12	-12
$\mathcal{A}^{(2)}$	-12	0	0
$\mathcal{A}^{(3)}$	-12	0	0

(a) Payoff of matrix game

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	**8.0**	-12.1	-12.1
$\mathcal{A}^{(2)}$	-12.2	-0.0	-0.0
$\mathcal{A}^{(3)}$	-12.1	-0.0	-0.0

(b) Q_{tot} of QPLEX

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	**8.0**	-12.0	-12.0
$\mathcal{A}^{(2)}$	-12.0	-0.0	0.0
$\mathcal{A}^{(3)}$	-12.0	0.0	0.0

(c) Q_{tot} of QTRAN

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-8.0	-8.0	-8.0
$\mathcal{A}^{(2)}$	-8.0	-0.0	**-0.0**
$\mathcal{A}^{(3)}$	-8.0	-0.0	-0.0

(d) Q_{tot} of QMIX

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-6.2	-4.9	-4.9
$\mathcal{A}^{(2)}$	-4.9	-3.6	-3.6
$\mathcal{A}^{(3)}$	-4.9	**-3.6**	-3.6

(e) Q_{tot} of VDN

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-6.2	-4.9	-4.9
$\mathcal{A}^{(2)}$	-4.9	-3.5	**-3.5**
$\mathcal{A}^{(3)}$	-4.9	-3.5	-3.5

(f) Q_{tot} of Qatten

图 6 矩阵游戏回报矩阵及模型估计 Q_{tot} 值

图6(a) 代表矩阵游戏的回报矩阵，粗体表示最优联合动作获得的最优奖励。图6(b-f) 分别代表 QPLEX、QTRAN、QMIX、VDN、Qatten 的联合动作价值函数 Q_{tot}，粗体表示根据联合动作-价值函数最终的联合动作选择获得的奖励，可以看到，只有 QPLEX 和 QTRAN 算法得到了最优解，其他算法都陷入了次优解，并且 Q_{tot} 估计也与真实值存在一定偏差。

图6代表 QPLEX 和其他基线算法在矩阵游戏上的学习曲线，可以发现只有 QPLEX 和 QTRAN 收敛到了最优值，这说明这两种算法相对其他算法具有更丰富的表现力。

4.3.2 两态多智能体马尔可夫游戏

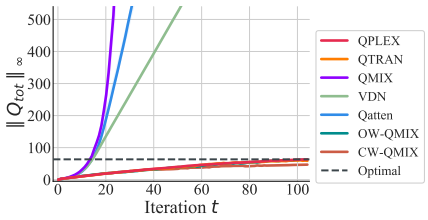

图 7 两态多智能体马尔可夫游戏学习曲线

图7横坐标代表迭代轮次，纵坐标代表 Q_{tot} 的无穷范数，可以发现只有 QPLEX 和 QTRAN 算法收敛到了真实最优奖励值 100，其他基线算法都发散到了正无穷，此实验证明了 QPLEX 具有很好的稳定性与可靠性。

4.3.3 星际争霸 II 微场景

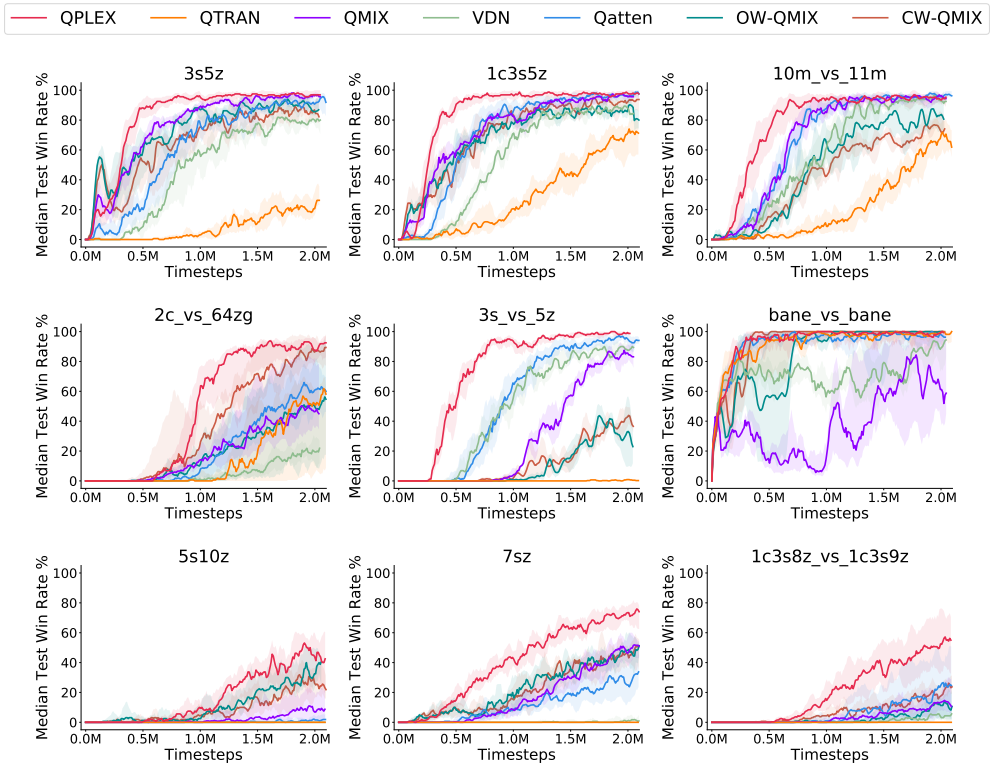

图 8 星际争霸 II 9 个微场景的学习曲线

本节在星际争霸 II 的 17 个微场景任务中评估 QPLEX，包含 14 个主流微场景和 3 个新提出的超级困难微场景，如图8代表九个任务的学习曲线，其中横坐标 Timesteps 代表时间步，范围从 $0.0M$ 到 $2.0M$，纵坐标 Median Test Win Rate 代表测试胜率中位数，每一条深色曲线代表中位数，其周围的浅色区域代表 1/4 分位数到 3/4 分位数之间的范围，因此对于每种场景，至少进行 4 次实验才能得到最终结果。

4.4 消融研究与分析

消融研究，即删除或替换算法及模型的一部分，以验证一些关键模块的有效性。本节对 QPLEX 进行了两项消融实验来研究多头注意力结构的影响和 QPLEX 参数数量的影响。

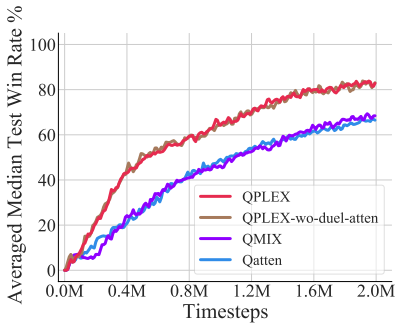

图 9 去掉 duel-attention 结构进行对比

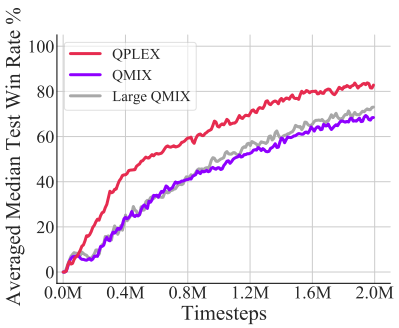

图 10 与相同参数数量的 QMIX 进行对比

如图9所示，为了证明 QPLEX 优于其他方法不是因为多头注意力技巧，论文使用一层前向网络来代替多头注意力结构，得到模型 QPLEX-wo-duel-atten，将这个模型和 QPLEX、QMIX、Qatten 进行对比测试，可以发现 QPLEX-wo-duel-atten 能够达到与 QPLEX 相似的性能，这表明 QPLEX 优于其他多智能体强化学习方法主要是由于它的值分解结构（Duplex Dueling），而不是多头注意力技巧。

此外，由于多头注意力结构，QPLEX 使用了更多的参数。为了证明 QPLEX 不是因为参数数量更多而比 QMIX 性能更优，论文引入了与 QPLEX 具有相同参数数量的 Large QMIX 模型进行对比测试。图10表明，增加 QMIX 的参数量不能从根本上提高它的性能，QPLEX 仍然远远优于 Large QMIX。这一结果也证明了 QPLEX 的主要优势来自其新颖的值分解结构（Duplex Dueling）而不是参数数量。

5 不足与改进

多智能体深度强化学习领域方兴未艾，还有许多问题亟待解决，前景无限光明，但道路也必定曲折。基于 Dueling DQN 的多智能体值分解方式存在一定的局限性，因此列出其缺点以及一些可以尝试改进的方向：

（1）本文提出的**值分解方式较为简单直接**，即直接将多智能体系统完全割裂为多个单智能体的组合，但大规模多智能体系统内部可能存在许多局部的协作团队，发挥作用的可能只是部分智能体，

目前的值分解式方法大多无法解决这个问题，因此可以尝试基于协作图的值分解模式，利用无向协作图表示联合动作值函数的高阶分解，每个节点表示一个智能体，每条边表示两个智能体之间的效用。基于协作图可以探索更通用更灵活的值分解模式，提高性能表现。

（2）本文提出的值分解方式**没有完备的理论框架**。尽管目前提出的一些值分解式方法在实践中有着比较好的效果，但它们有着不同的假设条件和限制条件。值分解式方法具体适用于什么类型的多智能体任务、能否适用于所有的多智能体任务还有待探究。可以尝试初步建立一个通用的可解释的值分解模型，并拓展到 AC 架构下用于解决连续动作空间的多智能体协作任务。

6 结论

近些年来深度强化学习飞速发展，成为了最热门的 AI 技术之一，在许多领域得到了广泛应用。深度强化学习与多智能体系统结合诞生了多智能体深度强化学习这一研究方向。本次课程大作业从零开始，理解并掌握强化学习、深度强化学习、多智能体强化学习以及基于值分解的多智能体强化学习的基础原理和算法脉络。

值分解是多智能体深度强化学习中的一个主流思想，其先分散式地学习每个智能体的个体动作值函数，然后集中式地利用个体动作值拟合联合动作值函数，可以在环境平稳性和智能体数量可拓展性之间取得平衡，但已有的值分解方法要么限制值分解的表达能力，要么放松 IGM 一致性的约束条件。Dueling DQN 是针对经典 DQN 算法的三大改进之一，以此为切入点，QPLEX 将 Dueling DQN 分解 Q 值的设计融合到基于值的多智能体强化学习算法中，完美实现了对现有多智能体强化学习算法的扩展和性能提升。

本次课程大作业理解并复现了论文 VDN、QMIX 和 QPLEX。原理部分，包含 QPLEX 的核心网络架构以及 Joint Dueling 和 Individual Dueling 原理；实验部分，在矩阵游戏、两态多智能体马尔可夫游戏和星际争霸 II 微场景环境下进行了实验复现，并对实验结果进行了相应评价，此外，还针对 QPLEX 的核心模块从多头注意力技巧和参数数量两方面进行了消融实验，证明 QPLEX 的优势在于其新颖的 Duplex Dueling 模块。

7 组员贡献

- 曹博文：QPLEX 论文整理 +QPLEX 部分实验复现
- 王春力：QMIX 论文整理 +QPLEX 部分实验复现
- 庄镇华：VDN 论文整理 +QPLEX 部分实验复现 + 补充整合

参考文献

1 Tan M. Multi-agent reinforcement learning: Independent vs. cooperative agents[C]Proceedings of the tenth international conference on machine learning. 1993: 330- 337.

2 Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S. A., Botvinick, M. Machine theory of mind[C]International conference on machine learning. PMLR, 2018: 4218-4227.

3 Sunehag P, Lever G, Gruslys A, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward[C]Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems. Richland, USA: IFAAMAS, 2018: 2085–2087.

4 Rashid T, Samvelyan M, Witt C S, et al. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning[C]Proceedings of the 35th International Conference on Machine Learning. New York, USA: ACM, 2018: 4292–4301.
5 Son K, Kim D, Kang W J, et al. QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning[C]Proceedings of the 36th International Conference on Machine Learning. New York, USA: ACM, 2019: 5887–5896.
6 Wang T, Wang J, Zheng C, et al. Learning nearly decomposable value functions via communication minimization[C]Proceedings of the 8th International Conference on Learning Representations. Amherst, USA: OpenReview.net, 2020.
7 Castellini J, Oliehoek F A, Savani R, et al. The representational capacity of action-value networks for multi-agent reinforcement learning[C]Proceedings of the 18th International Conference on Autonomous Agents and Multi Agent Systems. Richland, USA: IFAAMAS, 2019: 1862–1864.
8 Böhmer W, Kurin V, Whiteson S. Deep coordination graphs[C]Proceedings of the 37th International Conference on Machine Learning. New York, USA: ACM, 2020: 980–991.
9 Yang Y, Hao J, Liao B, et al. Qatten: a general framework for cooperative multiagent reinforcement learning[J/OL]. ArXiv:2002.03939, 2020[2020–03–08]. http://arxiv.org/abs /2002.03939.