多智能体强化学习中向多个独立顾问学习

Sriram Ganapathi Subramanian
VectorInstitute，多伦多，加拿大 滑铁卢大学，滑铁卢，加拿大
sriram.subramanian@vectorinstitute.ai

Matthew E. Taylor 阿尔伯塔大学，埃德蒙顿，加拿大 阿尔伯塔机器智能研究所，埃德蒙顿，加拿大
matthew.e.taylor@ualberta.ca

Kate Larson 滑铁卢大学
滑铁卢，加拿大
kate.larson
@uwaterloo.ca

Mark Crowley 滑铁卢大学 滑铁卢，加拿大
m
crowley@uwaterloo.ca

arXiv:2301.11153v2 [cs.LG] 2 Mar 2023

摘要

多智能体强化学习通常面临样本效率低下的问题，即学习合适的策略需要使用大量数据样本。向外部演示者学习是缓解这一问题的一种可行方案。然而，该领域的大多数既有方法都假设只有一位演示者。利用多个在环境不同方面具备专业知识的知识来源（即顾问），可以显著加快智能体在复杂环境中的学习速度。本文研究多智能体强化学习中同时向多个独立顾问学习的问题。该方法采用双层 *Q*-学习架构，并将这一框架从单智能体扩展到多智能体环境。我们提出了一系列有理论依据的算法，通过在每个状态下评估顾问，并随后利用顾问指导动作选择，将一组顾问的知识融入学习过程。我们还给出了理论收敛性和样本复杂度保证。在实验中，我们在三个不同的测试平台上验证了该方法，结果表明，我们的算法优于基线方法，能够有效整合不同顾问的综合专业知识，并学会忽略不良建议。

关键词

多智能体系统；多智能体强化学习；从动作建议中学习；强化学习；样本效率

ACM 引用格式：
Sriram Ganapathi Subramanian、Matthew E. Taylor、Kate Larson 和 Mark Crowley。2023年。从多智能体强化学习中的多个独立顾问处学习。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，IFAAMAS，共27页。

1 引言

强化学习（RL）已在诸多挑战中取得超越人类的表现，例如 Atari 游戏 [26], 围棋 [35], ，以及开门、学习视觉运动策略等简单的机器人任务 [20]。然而，要在复杂的现实世界问题中复制这些成功并非易事。原因之一是，这些问题通常具有多智能体结构，即有不止一个学习型智能体同时参与，从而导致复杂的动态变化。尽管多智能体强化学习（MARL）领域取得了研究进展 [11], ，现有算法的样本效率较低仍是一个问题，给将 MARL 应用于复杂问题带来了重大障碍 [33]。

利用外部知识来源来加速多智能体强化学习训练是一种解决方案 [2], ，在文献中得到了广泛支持 [33]。然而，大多数先前工作都包含两个限制性假设。首先，所有示范都必须来自单一演示者 [4]。在复杂的多智能体强化学习环境中，由于智能体需要学习能够同时满足应对不断变化的对手和环境这两个目标的策略 [22], ，学习者很可能会从多个知识来源中受益，这些来源分别擅长环境的不同部分或任务的不同方面。其次，所有示范都近乎最优（即来自“专家”） [29]。在实践中，这些知识来源通常并非最优，我们统称之为顾问（以区别于专家）。

本文提出了一种方法，可同时利用多个不同的（次优）顾问进行多智能体强化学习训练。由于顾问可能会在不同状态下给出相互冲突的建议，算法需要解决这些冲突，才能有效利用所有顾问。我们提出了一种双层学习架构，并构建了一种 *Q*-学习算法，以便在多智能体强化学习中同时纳入多个顾问，并在 Li 等人 [21] 针对单智能体强化学习开展的先前工作基础上加以改进。该架构的一层用于评估顾问，另一层则学习各动作的价值。此外，我们将该方法扩展为一种适用于集中训练、分散执行（CTDE）设置的演员-评论家变体 [24]。由于强化学习是一种不动点迭代方法[40], ，我们给出了收敛结果，证明我们的 *Q*-学习算法会收敛到纳什均衡 [27]（在常见假设下）。此外，我们针对两种不同类型的学习率，对该 *Q*-学习算法进行了详细的有限时间分析。最后，我们在三个不同的多智能体测试平台上，通过实验研究了该方法，并将其与标准基线方法进行了比较。

由于我们放宽了多智能体强化学习（MARL）中关于向示范者学习的两项限制性假设，我们希望这种方法能推动其在现实世界中的应用取得成功，例如自动驾驶 [10]和野火扑救 [15]。在这些应用中，多智能体强化学习方法可以利用现有的（次优）解决方案作为顾问，加快训练。

第*22*届自主智能体与多智能体系统国际会议论文集，*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年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。

2 相关工作

这项工作与基于专家演示的强化学习 （RLED） [29]的方法最为相关。一种知名的 RLED 技术是深度 Q-基于示范的学习 （DQfD） [12], , 它结合了时序差分（TD）损失、L2 正则化损失和分类损失，以鼓励动作接近演示者的动作。另一种方法归一化演员-评论家 （NAC） [16], 舍弃了分类损失，在面对不完美示范时更加稳健。然而，由于缺少分类损失，在示范质量较高时，NAC 的表现往往不如 DQfD。另一种方法人类智能体迁移 （HAT） [42], 利用分类器从有限的示范中提取信息，而基于置信度的人类智能体迁移（CHAT） [47] 则通过使用置信度度量来防范次优示范，从而改进 HAT。教师-学生框架 [44], 也是一种相关方法，其中预训练策略（教师）可为学习型智能体（学生）提供有限的建议。后续研究将这一框架扩展到交互式学习 [1], , 但该领域几乎所有工作都假设教师具备中等水平的专业知识。此外，这些方法都是彼此独立的，主要适用于单智能体环境，可能无法直接应用于多智能体强化学习场景。

此外，外部知识源也被用于多智能体强化学习 [33], , 其中先前工作通常假设专家近乎最优[30,49] ，或仅适用于受限场景，例如完全合作或零和竞争博弈 [28, 34, 35, 46, 52]。Leno 等人 [34] 提出了一种框架，使智能体除了从环境奖励中学习外，还能在共享学习环境中向同伴学习。这里的同伴可以是次优的，但这项工作仅适用于合作环境。其他研究则为分层学习提供了合作式教学框架 [18, 50]。对于多智能体常和环境，多智能体强化学习决策建议(ADMIRAL-DM) [38] 是一种 Q学习方法，能够融入来自单个在线次优顾问的实时信息。

许多先前工作的一个局限是，假设演示仅有单一来源。在多智能体强化学习中，有可能从不同知识来源获得顾问，而这些顾问会提供相互冲突的建议。对于单智能体环境，Li 等人 [21] 提出了双层Q学习算法（TLQL），可在强化学习中纳入多个顾问。TLQL 维护两个 Q网络，其中第一个 Q网络（高层）跟踪每位顾问的表现，第二个 Q网络（低层）则学习每个动作的质量。我们在 TLQL 的基础上进行改进，使其适用于多智能体强化学习环境。

3 背景

随机博弈： N人随机博弈表示为一个元组 S, A^1、……、A^N, r^1、……、r^N, P, γ, 其中 S是状态空间，A^j是智能体 $j \in 1 \quad N \quad S \times \{$、……、$A^1 \times \quad \times A^N \rightarrow \quad P \quad S \times$ $\}$的动作空间，而 r^j: $\cdots$ R 是 j的奖励函数。此外，: $A^1 \times \cdots A^N \rightarrow \Omega(S)$ 是转移概率，用于根据当前状态和所有智能体的联合动作确定下一个状态，其中 Ω是概率分布。最后，$\gamma \in [0, 1)$ 是折扣因子。在每个时间 t，所有智能体都会观察全局状态 s并采取局部动作 a^j [32]。联合动作 $\boldsymbol{a} = \{a^1$、……、$a^N\}$ 决定了 j 的即时奖励 r^j 以及系统的下一个状态 $s\prime$。每个智能体都会学习一种合适的策略，以对其对手作出最佳响应。策略记为 $\pi^j: S \rightarrow \Omega(A^j)$。令 $\boldsymbol{\pi} \triangleq (\pi^1$、……、$\pi^N)$ 表示所有智能体的联合策略。在状态 s下， j 在联合策略 $\boldsymbol{\pi}$ 下的价值函数为$v_{\boldsymbol{\pi}}^j(s) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\pi,\ P}[r_t^j|s_0 = s, \boldsymbol{\pi}]$。这表示所有智能体从状态 s开始遵循策略 $\boldsymbol{\pi}$时， j的期望折扣未来奖励。与价值函数相关的是动作价值函数，也称为 Q-函数。智能体 j在策略 $\boldsymbol{\pi}$下的 Q-函数定义为： $Q_{\boldsymbol{\pi}}^j(s,\ \boldsymbol{a}) = r^j(s,\ \boldsymbol{a}) + \gamma\,\mathbb{E}_{s'\sim P}[v_{\boldsymbol{\pi}}^j(s\prime)]$。

我们考虑的设定是一般和随机博弈，其中不同智能体的奖励函数可以任意关联。在这一设定中，纳什均衡 通常被视为解概念 [13], , 其中所有 $s \in S$ 和所有 j 的联合策略$\boldsymbol{\pi}_* = [\pi_*^1, \ldots, \pi_*^N]$ 满足 $v^j(s;\ \pi_*^j,\ \boldsymbol{\pi}_*^{-j}) \geq v^j(s;\ \pi^j,\ \boldsymbol{\pi}_*^{-j})$。这里，$\boldsymbol{\pi}_*^{-j} \triangleq [\pi_*^1, \ldots, \pi_*^{j-1}, \pi_*^{j+1}, \ldots, \pi_*^N]$ 表示除 j 之外所有智能体的联合策略。在纳什均衡中，每个智能体都会对其他智能体采取最佳响应，偏离这一响应必然会导致更差的结果。此外，Hu 和 Wellman [13] 证明，智能体 j 使用每个阶段的纳什收益进行 Q-更新，最终会收敛到其纳什 Q 值（Q_*^j）；该值是当所有智能体在无限期内遵循纳什均衡联合策略时，智能体 j 获得的动作值。

双层 Q学习：TLQL算法 [21] 使得单智能体能够在多个顾问同时存在、且提供相互冲突的示范的情况下进行学习。这里的挑战在于，确定在给定状态下应信任哪位顾问。为此，TLQL包含两个 Q表：高层 Q表（简称高层Q）和低层 Q表（简称低层Q）。高层Q存储 s, ad 对的值，其中 $ad \in AD$ 表示一位顾问（ AD 表示所有顾问的集合）。除每位顾问外，高层Q还存储遵循强化学习策略的值。低层Q 维护每个状态-动作对的值。

每个时间步，智能体都会观测状态，并使用 ϵ-贪心策略从高层Q 中选择一位顾问（或强化学习策略）。如果高层Q 返回一位顾问，就执行该顾问推荐的动作。如果返回的是强化学习策略，则根据 ϵ-贪心策略 [39]，从低层Q 中选择并执行一个动作。使用标准的 Q学习贝尔曼更新 [48]更新低层Q。随后，通过一个**同步**步骤更新高层Q。在此步骤中，执行顾问给出的动作时，只需将该顾问在高层Q中的值设为低层Q中该动作的值。最后，使用关系 $highQ(s, RL) = \max_a lowQ(s, a)$ 更新强化学习策略对应的高层Q。由于单智能体 Q学习中的策略改进保证 [39], 这种对高层Q 的同步更新保留了收敛保证。

TLQL 存在两个重要局限。首先，表示顾问值的高-Q还依赖于强化学习策略，这是通过同步步骤实现的。这个 Q 值表示在当前状态下采取顾问建议的动作，然后从下一个状态开始遵循强化学习策略的价值。这个定义存在问题，因为在训练初期，强化学习策略并非最优，而目标是依靠外部建议者来加速学习，并完全避免使用强化学习策略。由于每个状态下都是使用强化学习策略对顾问进行评估，因此在强化学习策略得到改进之前，顾问集合中最有效的顾问很可能一直不会被采用。

在这一阶段，直接遵循强化学习策略本身或许是可行的，但这就违背了从顾问处学习的目的。其次，学习开始时并未对顾问进行评估。因此，不可能从可用顾问中找出最适合遵循的顾问。虽然 TLQL 只是遵循一种 ϵ-贪心探索策略，但这种方法可能需要许多数据样本才能找出合适的顾问。我们针对这两个局限进行了处理。

4 多智能体强化学习中的双层架构

我们考虑一个一般和随机博弈，其中有一组智能体正在学习策略，其目标是在环境以及其他智能体面前实现最佳响应。每个智能体 j都可以访问一组有限的（可能并非最优的）独立顾问 AD^j。我们使用 ad^j 表示 j 的一个顾问，其中 $ad^j \in \{ad_1^j、\ ...\ ...、\ ad_{|AD^j|}^j\}$。每个顾问 ad^j都可以由 j查询，以便在随机博弈的每个状态下获得动作建议。这些在线顾问向智能体提供实时动作建议，帮助其学习动态适应对手。我们考虑集中式训练设置，并假设：1）顾问是固定的，不进行学习；2）智能体与其顾问之间的通信是免费的；3）学习智能体之间不存在直接通信；4）环境是完全可观测的（即，智能体可以观测全局状态、所有动作以及所有奖励）；5）状态和动作空间是离散的。尽管理论保证要求这些假设成立，但我们将展示，在实践中可以放宽其中的若干假设。

为了使 TLQL 适用于多智能体环境，我们按照实践中的常见做法，将两个 Q函数都用联合动作进行参数化 [22]。此外，我们不在高层Q 表中维护强化学习策略，也不执行同步步骤。在多智能体环境中，为了保留收敛结果，这些步骤已不再必要，因为我们没有策略改进保证（不同于单智能体环境）[41]。相反，我们选择采用概率策略重用（PPR）技术 [6]，其中一个超参数（$\epsilon_r \in [0,1]$）决定在训练期间的每个时间步，遵循任一顾问（即使用高层Q）或使用智能体自身策略（即使用低层Q）进行动作选择的概率。训练开始时，该超参数的值较高（最大程度依赖可用的顾问），随后随时间（线性）衰减。训练过程中经过有限个时间步后，该超参数的值降为 0（不再依赖任何顾问），智能体仅使用其低层Q（自身策略）进行动作选择。这有两方面的作用：1）在时间限制内（$t \to \infty$），学习型智能体有机会通过从环境中学习来摆脱不佳建议；2）最终，训练好的智能体可以独立部署（无需能够访问任何顾问）。

我们提出的多智能体双层Q-学习（MA-TLQL）算法的总体结构如图 1 所示。由于我们处于完全可观测环境中，如 [13], 所述，我们规定每个智能体都维护其他智能体的 Q表副本，以便获取其他智能体在当前状态下的联合动作。如果无法维护此类副本，智能体可以像先前工作 [38, 51]中那样，使用先前观测到的其他智能体动作来组成联合动作。我们采用双层架构，每个智能体都会维护一个高-Q 以及一个低-Q。该

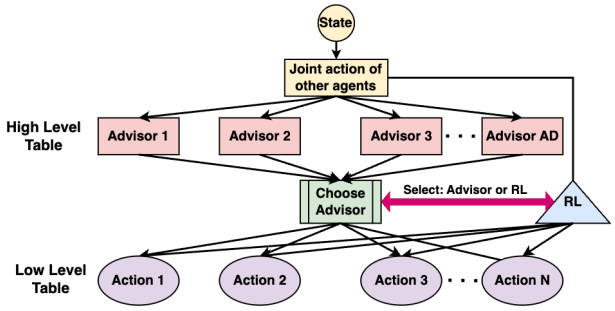


图 1：MA-TLQL 的结构，适用于可访问一组 AD 位顾问的代表性智能体

高-Q为s，$\boldsymbol{a}^{-j}$，ad^j元组提供一个值，其中 $\boldsymbol{a}^{-j} = \{a^1, \ .....,\ a^{j-1},\ a^{j+1},\ .....,\ a^N\}$ 是除智能体 j之外所有智能体的联合动作。该高-Q是智能体 j在状态 s和联合动作 $\boldsymbol{a}^{-j}$下，对顾问 ad^j所作的值估计。高-Q 估计通过以下评估更新进行更新：

$$highQ_{t+1}^j(s, \boldsymbol{a}^{-j}, ad^j) = highQ_t^j(s, \boldsymbol{a}^{-j}, ad^j) + \alpha\Big((r_t^j + \gamma highQ_t^j(s', \boldsymbol{a'}^{-j}, ad^j) - highQ_t^j(s, \boldsymbol{a}^{-j}, ad^j)\Big), \quad (1)$$

其中，s和 s' 分别是 t和 $t+1$处的状态，α是学习率。此外，$\boldsymbol{a}^{-j}$和 $\boldsymbol{a'}^{-j}$分别是 s处和 s'处的联合动作。

如前所述，使用一个超参数来决定是遵循顾问还是采用强化学习策略。如果智能体遵循顾问，则使用高-Q，通过集成选择技术选出一位顾问。我们将 $\mathrm{Q}^j = \{highQ^j(s, \boldsymbol{a}^{-j}, ad_1^j) \ldots, highQ^j(s, \boldsymbol{a}^{-j}, ad_m^j)\}$记为一组 M 位顾问（其中 $|M| = m$）对同一动作 a^j向智能体 j提供建议时的高-Q 估计值。此处，ad_i^j表示 j中的一位顾问$i \in \{1, \ldots, m\}$。然后，状态 s下、联合动作 $\boldsymbol{a}^{-j}$所对应的动作 a^j的投票值记为 $\mathrm{V}^j(s, \boldsymbol{a}^{-j}, a^j)$，其计算方式如下：

$$\mathcal{V}^j(s, \boldsymbol{a}^{-j}, a^j) = \max \boldsymbol{Q}^j + \sum_{i=1, i \neq \arg\max_i highQ^j(s, \boldsymbol{a}^{-j}, ad_i^j)}^m \frac{1}{\mu(s)} highQ^j(s, \boldsymbol{a}^{-j}, ad_i^j). \quad (2)$$

这里，$\mu(s)$表示智能体访问状态 s的次数。在公式 2 中，如果某个动作受到多个顾问的推荐，其投票值就是所有推荐该动作的顾问的高-Q 估计值的加权和。每个高-Q 估计值（最佳高-Q 估计值除外）都按相应状态被访问次数的倒数加权。这样，当某个状态被多次访问时，智能体很可能会遵循具有最佳高-Q 估计值的顾问的建议（个体智慧）。当某个状态仅被访问少数几次时，则很可能会选择多数顾问建议的动作（群体智慧）。根据公式 2，如果某个动作 a^j仅由一位顾问推荐，那么该动作 a^j 的投票值将等于该顾问的高-Q 估计值。计算出所有动作的投票值后，执行投票值最高的动作，并且智能体使用公式 1 更新推荐该动作的顾问的高-Q 估计值 j。

如果智能体决定使用其强化学习策略，它会使用自己的低层Q，其中包含 s，$\boldsymbol{a}^{-j}$，a^j 元组的值（每个动作对应一个值）。在每个步骤，低层Q 都会通过以下控制更新进行更新：

$$
\begin{aligned}
&lowQ_{t+1}^j(s, \boldsymbol{a}^{-j}, a^j) = lowQ_t^j(s, \boldsymbol{a}^{-j}, a^j) \\
&+\alpha\big(r_t^j + \gamma \max_{a'^j} lowQ_t^j(s', \boldsymbol{a}'^{-j}, a'^j) - lowQ_t^j(s, \boldsymbol{a}^{-j}, a^j)\big).
\end{aligned} \tag{3}
$$

现在我们将介绍 MA-TLQL 如何解决 TLQL 的两个局限。第一个局限是 TLQL 中高层Q依赖于强化学习策略。请注意，MA-TLQL 中的高层Q维护的是顾问自身的 Q 值，即从当前状态开始遵循顾问策略所得到的值（见式 1）。因此，顾问价值与强化学习策略之间的耦合被消除了（无需同步）。第二个局限是在 TLQL 中难以选出合适的顾问。MA-TLQL 在学习早期使用集成技术来选择顾问。在后期，它会根据高层Q的估计值，转而遵循最佳顾问的策略，从而解决 TLQL 的这一局限。在附录 J 中，我们给出了一个玩具示例，以说明 TLQL 的局限。

我们在附录 A（算法 1）中提供了 MA-TLQL算法表格实现的完整伪代码。此外，我们将此方法扩展到大型状态-动作环境，采用基于神经网络的实现（算法 2），其中使用了目标网络和经验回放缓冲区，与深度 Q-学习（DQN）算法 [26] 中一样。我们还提供了一种适用于集中训练、分散执行 [24] 的演员-评论家实现（算法 3）。我们将此算法称为多智能体双层演员-评论家 （MA-TLAC）。在 MA-TLAC 中，每个智能体都有两个演员和两个评论家（高层和低层），相应的 Q-函数充当评论家，相应的策略则充当演员。在这种集中训练、分散执行方法中，智能体在训练期间可以获取全局信息（包括其他智能体的动作和奖励），但在执行期间只需访问各自的局部观测。因此，我们的方法适用于 Lowe 等人 [24] 所述的部分可观测环境。MA-TLAC 也适用于连续状态空间环境（更多详情请参阅附录 A）。

5 理论结果

我们给出了表格型 MA-TLQL 的收敛保证，并刻画了其收敛速率。为获得这些结果，我们基于一些先前工作，这些工作提供了关于随机迭代函数性质的若干基本结果 [3, 5]。我们将这些结果应用于一般和随机博弈中的 MA-TLQL，并采用了 Hu 和 Wellman 提出的三个假设 [13], ，其中前两个是标准假设 [40]。

假设 1. 每个 $s \in \mathcal{S}$ 和 $a^j \in A^j$，对于每个智能体 j 都会被无限次访问，且奖励函数（$\forall j$）有界。

假设 2 s t $\boldsymbol{a}$ $0 \leq \alpha_t$ s $\boldsymbol{a}$ < 1 $\sum_{t=0}^{\infty} \alpha_t$ s $\boldsymbol{a}$ $=$.对于所有 ,
∞ $\sum_{t=0}^{\infty}[\alpha_t$ s $\boldsymbol{a}$ $]^2 < \infty$
, 和 ,(,),(,),(,)。

假设 3. 在随机博弈的每个阶段博弈中，纳什均衡都是全局最优解或鞍点。

第三个假设限制了随机博弈的性质。多项先前工作指出，该假设具有一定限制性，但要从理论上证明在包含两个或更多智能体的一般和随机博弈中， Q-学习方法的收敛性，就必须作出这一假设。然而在实践中，即使违反该假设，仍有可能观察到Q-学习方法收敛 [13, 38, 51]。

现在我们证明理论结果。所有定理陈述均在此给出，而证明见附录B–D。首先，我们给出低-Q的收敛保证。回顾一下，PPR 技术保证，MA-TLQL 对高-Q 的依赖仅持续到训练期间的某个有限时间步。在此步骤之后，智能体仅使用其低-Q 进行动作选择。由于定理 1 $t \to$ 中的收敛结果是在时间极限（ ）下给出的，因此高-Q 的影响可在该结果中忽略。

定理1。 在假设*1*、*2*、*3*成立的情况下，智能体 j 的低-Q 值在极限情况下会收敛到其纳什 Q 值 $(t \to \infty)$。

接下来，我们给出 MA-TLQL算法的样本复杂度界。我们不显式考虑高-Q 值，而是假设底层联合策略的覆盖时间为 L。覆盖时间给出了一个上界：从任意状态-联合动作对出发，访问所有状态-联合动作对至少一次所需的时间步数。此外，由于极限情况下 $(t \to \infty)$ 的动作选择仅基于低-Q 值，因此我们最关注低-Q的样本复杂度，其中对高-Q 的依赖实际上由 L表示。

关于样本复杂度，正如 [5], 中所做的那样，我们区分两种学习率。考虑以下低-Q 的方程（改写公式 3，并为简洁起见省略 low），

$$
\begin{aligned}
Q_{t+1}^j(s_t, \boldsymbol{a}_t) = &\big(1 - \alpha_t^\omega(s_t, \boldsymbol{a}_t)\big)\big(Q_t^j(s_t, \boldsymbol{a}_t)\big) \\
&+\alpha_t^\omega(s_t, \boldsymbol{a}_t)\big(r_t^j + \gamma \max_{a^j} Q_t^j(s_{t+1}, \boldsymbol{a}_{t+1})\big).
\end{aligned} \tag{4}
$$

$\alpha_t^\omega(s,$ $\boldsymbol{a}) =$ 1 [s $\boldsymbol{a}$#(, , $t)^\omega$] 的值，其中 $\#(s,$ $\boldsymbol{a},$ $t)$ 表示截至 t，联合动作 $\boldsymbol{a}$ 在 s执行的次数。此处，我们考虑 $\omega \in (1/2, 1]$。当 $\omega = 1$ 时，学习率为线性；当 $\omega \in (1/2,$ $1)$ 时，学习率为多项式。

下一个定理给出了多项式学习率情况下收敛所需时间步数的下界。根据假设 1，我们进一步指出，智能体 j 的所有奖励均有界于 $R^j{}_{\max}$。我们考虑变量 $Q^j{}_{\max}$，它表示智能体 j 可能达到的最大低-Q 值，其上界为 $Q^j{}_{\max} = R^j{}_{\max}/(1-\gamma)$。此外，我们还使用另一个变量 $\beta = (1-\gamma)/2$，以简洁地呈现接下来的结果。

定理 2. 设对于智能体 j，以至少 $1-\delta$ 的概率有 $||Q_T^j - Q_*^j||_\infty \leq \epsilon$。当多项式学习率的因子为 ω 时，低-Q 的收敛速率上界 Q_T^j 由下式给出（其中 Q_*^j 是智能体 j 的纳什 Q 值）

$$
\begin{aligned}
T = \Omega\Bigg(&\Bigg(\frac{L^{1+3\omega} Q_{\max}^{2,j} \ln(\frac{|S|\Pi_i|A_i|Q_{\max}^j}{\delta\beta\epsilon})}{\beta^2\epsilon^2}\Bigg)^{1-\omega}/L \\
&+\Bigg((\frac{L}{\beta} \ln \frac{Q_{\max}^j}{\epsilon} + 1)/2\Bigg)^{\frac{1}{1-\omega}}\Bigg).
\end{aligned} \tag{5}
$$

假设所有智能体的动作空间相同（即 $|A_1| = |A_2| = \cdots = |A_N| = |A|$），我们注意到其对智能体数量的依赖为 $\ln |A|^N = N\ln |A|$。总体而言，这使得其对智能体数量的依赖基于 ω 呈次线性，这远优于近期研究：这些研究报告称，在一般和随机博弈环境（智能体数量任意）中学习并收敛到纳什均衡 [23, 36] 时，对智能体数量呈指数级依赖。此外，定理 2 中对状态空间和动作空间的依赖为次线性（$\ln |S|$），对覆盖时间的依赖为 Ω（$L^{2\omega-3\omega^2} + L^1/^{1-\omega}$），即呈多项式依赖。

下一个定理讨论线性学习率的情形。

定理 3. 具体而言，以至少 $1-\delta$ 的概率，对于智能体 j，$||Q_T^j - Q_*^j||_\infty \leq \epsilon$。低-Q、 Q_T^j 在采用线性学习率时的收敛速率上界为

$$T = \Omega\Big((L+\psi L+1)^{\frac{1}{\beta}} \ln \frac{Q_{\max}^j}{\epsilon} \frac{Q_{\max}^{2,j} \ln(\frac{|S||\Pi_i||A_i|Q_{\max}^j}{\delta\beta\epsilon\psi})}{\beta^2\epsilon^2\psi^2}\Big), \quad (6)$$

其中， ψ是满足 $\psi \leq 0.712$的小型任意正常数。

定理 3 表明，该上界关于智能体数量呈线性关系，而关于状态空间和动作空间呈次线性关系。这种对智能体数量的线性依赖也优于以往的结果 [23, 36]。需要注意的是，根据 $Q^j_{\max}$ 和 ϵ 的取值，定理 3 中对覆盖时间的依赖可能远差于定理 2。由于 ϵ 的值较小，这种依赖性必然差于多项式学习率情况下所得到的结果。此外，与定理 2 中的多项式依赖相比，对 $Q^j{}_{\max}$ 的依赖呈指数关系。最后两个定理说明，在我们的算法中使用多项式学习率而非线性学习率所带来的性能优势。

6 实验与结果

我们考虑三个不同的实验领域，分别对应竞争性、合作和混合设置，其中每个智能体都可以访问由四名顾问组成的集合。我们使用 MA-TLQL 和 MA-TLAC 的神经网络实现，以及另外 5 种基线方法：DQN [26], DQfD [12], CHAT [47], ADMIRAL-DM [38], 和 TLQL [21]。在附录 F 中，我们汇总了这些基线方法的特征，并进一步说明了我们的选择。由于 CHAT 和 ADMIRAL-DM 假设存在单一顾问，因此，在多顾问设置中，我们采用加权随机策略方法来实现这两种算法，具体如 Li 等人 [21] 所述。如果不同顾问在同一状态下提供不同的动作，则根据建议各动作的顾问数量为每个动作赋予权重。对于 DQfD，在预训练期间 [12],，我们使用所有可用顾问的顾问示范填充经验回放缓冲区。对于所有实验，我们将在此处描述关键细节，完整说明见附录 K。所有实验均重复进行 30 次，并报告平均值和标准差。为检验统计显著性，我们采用非配对双侧 t 检验，并报告 p-值，其中 $p < 0.05$ 被视为具有显著性。检验将表现最佳的算法（通常为 MA-TLQL）与排名第二的基线方法以及最佳/平均顾问表现进行比较。我们共进行七项实验。所有实验的代码均已开源 [37]。附录 K 汇总了我们的全部实验设置。附录 L 提供了超参数细节，附录 M 包含实际运行时间。

实验 1–4 使用 Pom- merman [31] 的竞争性双智能体版本。该环境较为复杂，每个状态包含约 200 个与智能体位置和特殊元素（例如炸弹）相关的信息。奖励函数较为稀疏：智能体只有在终止状态才会收到 {−1, 0, +1} 奖励。实验分两个阶段进行。第一阶段为训练阶段，我们的算法和基线方法与一个标准 DQN 对手进行 50,000 回合的训练，并绘制累积奖励。在此阶段，算法可以使用顾问来加速训练。第二阶段为执行阶段，我们让训练好的策略与 DQN 对战 1000 回合，以测试其性能，并绘制各算法的胜率（获胜游戏所占的比例）。在此阶段，智能体无法访问顾问，不会采取探索性动作，也不会进行学习。这四项实验中的所有顾问都是基于规则的智能体。

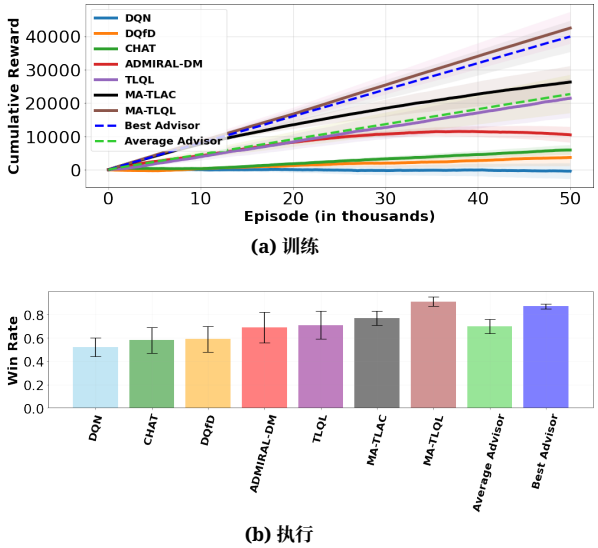
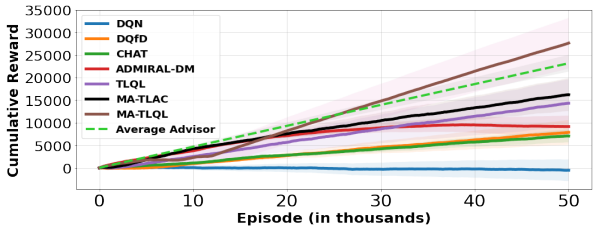


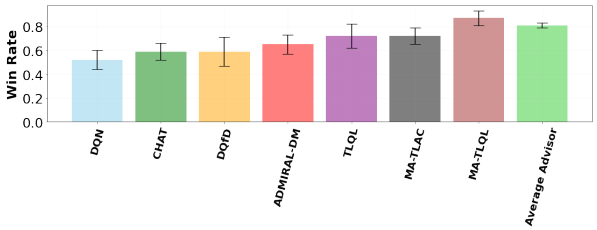
图 2：两智能体 Pommerman，配有四位不同质量的充足顾问（实验 1）

实验 1： 我们的第一个实验使用了四位顾问，他们的质量从顾问 1 到顾问 4 依次递减。其中，顾问 1 是最优秀的顾问，能够教会智能体赢得 Pommerman 游戏所需的所有技能；顾问 4 则只会建议随机动作。在 Pommerman 中，智能体需要掌握一组固定的六项技能才能获胜 [31]。由于这组顾问能够教授所有这些技能，我们称智能体拥有一组足够的顾问。我们分别在图 2(a) 和图 2(b) 中绘制了训练和执行表现，其中包括最优秀顾问和平均顾问（顾问 1–4 的平均表现）相对于 DQN 的表现。MA-TLQL 的表现最佳（$p < 0.01$），并且是训练和执行阶段中唯一表现优于最优秀顾问的算法（$p < 0.11$）。MA-TLAC 的表现优于平均顾问（$p < 0.04$）。其他算法的表现均未超过平均顾问。CHAT 和 ADMIRAL-DM 无法有效利用并区分不同顾问。DQfD 使用预训练，但这在非平稳多智能体环境中效果不佳。在多智能体强化学习中，通过在线建议进行学习更可取。此外，DQfD 和 CHAT 都是独立的技术，不会主动跟踪对手的表现。尽管 TLQL 能够从多位顾问那里学习，但其独立特性以及顾问价值与强化学习策略的耦合，降低了它在多智能体环境中的有效性。MA-TLQL 在训练和执行阶段的表现都优于 MA-TLAC（$p < 0.01$）。如前所述，Q学习算法系列在使用最大动作价值时往往会引入正偏差，从而导致

给出最佳响应 [45]。这解释了 MA-TLQL 为何表现出色。我们得出结论，MA-TLQL 能够有效利用一组优劣不一的顾问。此外，图 2(a) 中的训练结果表明，借助顾问，MA-TLQL 能够比基线方法更快地学得更优策略（$p < 0.01$）。图 2(b) 中的评估结果表明，在所有训练回合数相同的算法中，MA-TLQL 在不使用任何顾问的情况下部署时表现最佳（$p < 0.01$）。这两项结果都表明 MA-TLQL 具有更高的样本效率。附录 E 中的补充实验表明，与基线方法相比，MA-TLQL 更倾向于依赖优质顾问而非劣质顾问，进一步体现了其优越性。



(a) 训练

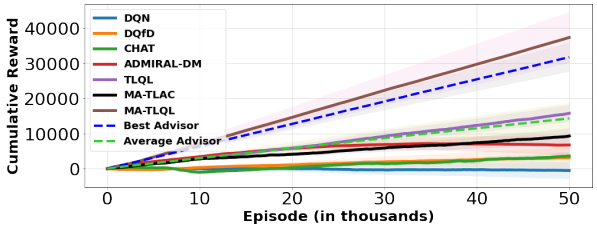


(b) 执行

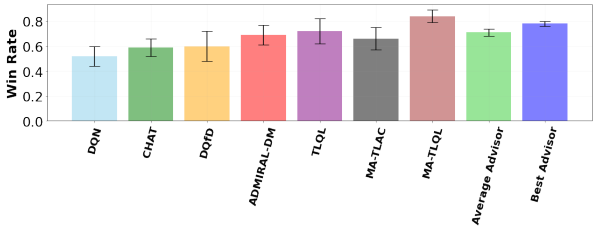
图 3：由四位能力相当且足以提供指导的顾问组成的双智能体 Pommerman（实验 2）

实验 2： 我们使用与实验 1 相同的领域，但采用另一组顾问。现在，四位顾问都能教授彼此截然不同的 Pommerman 技能。例如，顾问 1 能教会智能体如何躲避敌人（除此之外什么也不教），顾问 2 能教会智能体如何获取必要的强化道具（除此之外什么也不教——详情见附录 K）。在自身专业知识涉及范围之外的状态下，这些顾问会随机提供动作建议。这组顾问同样足以提供指导。现在，学习智能体必须决定在当前状态下听从哪位顾问的建议。从图 3(a) 和 (b) 中的训练与执行结果可以看出，MA-TLQL 的总体表现最佳（$p < 0.02$），超过了四位顾问的平均表现（$p < 0.05$）。由于四位顾问的能力相当，在本实验中，我们仅将四位顾问的平均表现作为比较基准。我们由此得出结论：在学习过程中，MA-TLQL 能够利用一组各自拥有不同专业知识的顾问所提供的综合知识。

实验 3： 我们使用与实验 1 相同的领域，但采用一组不同的四位顾问。这组顾问与首次实验中的顾问组相似，其中顾问 1 在整个领域中给出最佳建议，而顾问 4 则采取随机策略。



（a）训练



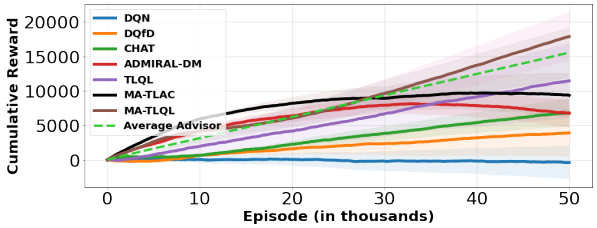
（b）执行

图 4：具有四位不同质量且能力不足的顾问的双智能体 Pommerman（实验 3）

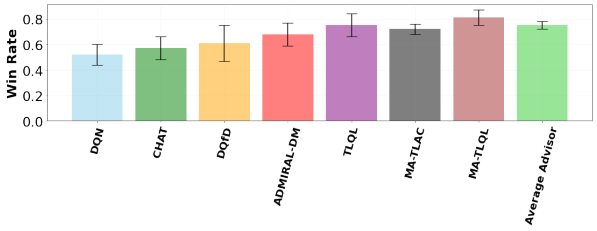
然而，这组顾问无法教授在 Pommerman 中获胜所需的所有策略（即 Pommerman技能），因此构成的顾问集合并不充分（详情见附录 K）。除了向顾问学习，智能体还必须从环境中学习。图 4 中的训练和执行结果表明，MA-TLQL 的表现优于其他方法；它是唯一一种胜过最佳顾问（$p < 0.05$）和所有基线方法（$p < 0.02$）的算法。令人惊讶的是，TLQL 的表现优于 MA-TLAC（$p < 0.02$），这可能是由于 Q-learning 的正偏差所致。该实验进一步印证了这一观察：MA-TLQL 能够向优质顾问学习，并避开劣质顾问（另见附录 E）。由于 MA-TLQL 的表现胜过最佳顾问，该实验表明，MA-TLQL 既能向顾问学习，也能通过与环境直接交互来学习，因此与其他仅从环境中学习的算法相比，其样本效率大幅提高。这一点在训练和执行过程中均有所体现。

实验 4： 这与实验 2 类似：四位顾问具有相似的质量，但每位顾问都掌握不同的 Pommerman 技能。然而，本实验中的顾问不足以教授 Pommerman 中的所有技能，智能体还必须从环境中学习。图5中的结果表明，与基线方法（$p < 0.04$）和顾问（$p < 0.05$）相比，MA-TLQL 能够利用顾问的综合专业知识并从环境中学习，从而获得最佳性能。这使 MA-TLQL 的样本效率高于先前算法。

实验5： 现在，我们转向 Pommerman 的四智能体版本，即 two vs. two。这是一个混合设置，因为智能体需要学习合作技能和竞争技能。总的来说，这是一个更复杂的领域，状态空间也更大。与实验 1 类似，我们考虑四位质量各异的充分顾问。我们分两个阶段进行：训练（50,000 回合）和执行（1000 回合）。图6中的训练和执行结果

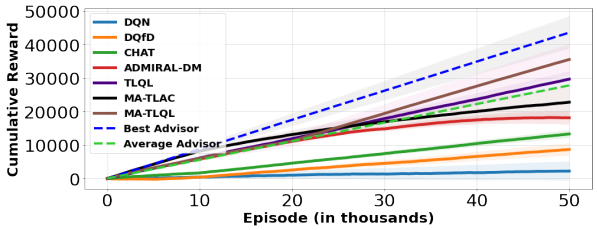


(a) 训练

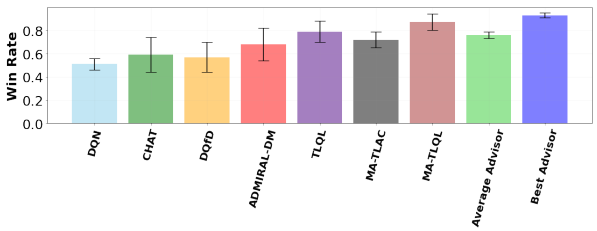


(b) 执行

图5：四名质量相似但能力不足的顾问参与的双智能体 Pommerman（实验 4）



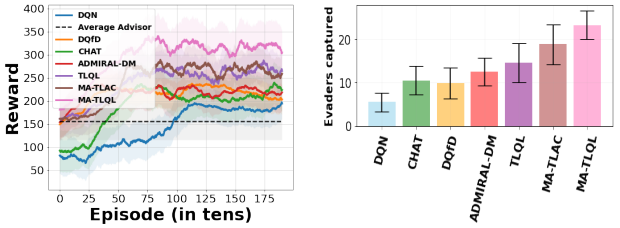
(a) 训练



(b) 执行

图6：团队（混合）Pommerman（实验5）

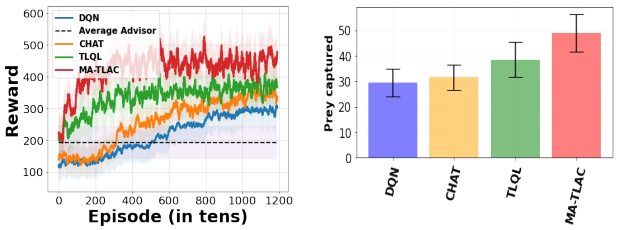
表明，与基线方法（$p < 0.04$）相比，MA-TLQL 的表现最佳，但不如表现最好的可用顾问。由于这是一个更复杂的领域，MA-TLQL 需要更长的训练时间才能学到良好的策略。不过，MA-TLQL 的表现仍优于四位顾问的平均表现（$p < 0.03$）。我们得出结论：尽管 MA-TLQL 在难度更高的混合设置中的表现有所下降，但仍优于所有其他基线方法，并且能够区分优质顾问和劣质顾问（另见附录 E）。从两者来看从图 6 的训练和执行结果可以看出，与其他基线方法相比，MA-TLQL 具有更高的样本效率。



(a) 训练　　(b) 执行

图 7：合作追捕设置（实验 6）

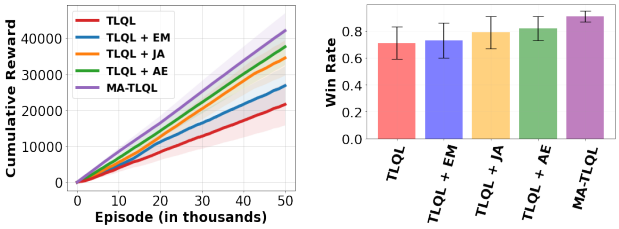
实验 6： 本实验转向合作追捕领域 [8]。八个追捕者学习智能体将学习捕获一组由 30 个随机移动的目标（逃避者）组成的目标集（详情见附录 K）。我们使用四个预训练的 DQN 网络作为顾问，分别在 500、1000、1500 和 2000 个回合上进行学习，地。我们再次分为两个阶段——训练和执行。在训练阶段，所有算法均训练 2000 回合。随后，在执行阶段使用训练好的网络运行 100 回合，不再进行进一步训练，也不受顾问影响。图 7(a) 绘制了训练期间获得的回合奖励，图 7(b) 则绘制了执行阶段捕获的目标数量，其中 MA TLQL 的表现最佳（$p < 0.03$）。因此，MA-TLQL 在合作环境中也能优于所有基线方法 .



(a) 训练　　(b) 执行

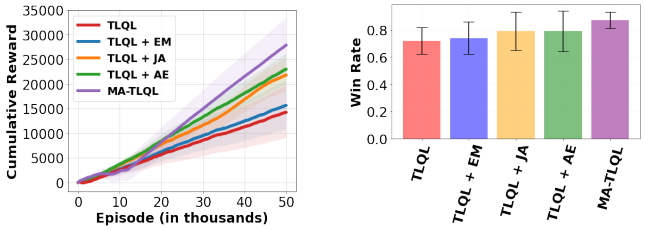
图 8：混合捕食者-猎物设置（实验7）

实验7： 这项最终实验考察了一个混合合作与竞争的捕食者-猎物环境，该环境属于多粒子环境（MPE）套件 [24]。我们的实现使用离散动作空间和连续状态空间（更多详情见附录 K）。共有八个捕食者试图捕获八只猎物（猎物不会被移除，而是在被捕获后重新生成）。在我们的实验中，每种算法都用于训练捕食者，而猎物则使用标准 DQN 对手进行训练。实验分为训练和执行两个阶段，其建模方式为集中训练、分散执行（CTDE）设置。在该设置中，每个智能体在训练期间都会获得所有其他智能体的动作和奖励信息，但在执行期间只能获得局部观测。由于该环境要求在执行期间实现去中心化，我们省略了完全集中式的 MA-TLQL 和 ADMIRAL-DM。我们也省略了 DQfD



(a) 训练　　(b) 执行

图 9：使用实验 1 的消融结果



(a) 训练　　(b) 执行

图10：使用实验 2 的消融结果

因为它此前表现不佳。与实验 6 一样，我们使用四个预训练的 DQN （捕食者）网络作为顾问（分别训练了 1000 、 2000 、 7000 和 12000 个回合）。训练进行了 12000 个回合，执行进行了 100 个回合。图 8(a) 中的训练结果（回合奖励图）显示，与其他算法相比，MA-TLAC 的样本效率最高，因为它比其他算法更善于利用现有顾问，因此表现优于它们（$p < 0.04$）。图 8(b) 的执行结果绘出了各算法捕获猎物的平均数量。MA-TLAC 在执行期间同样优于其他算法（$p < 0.03$）。

在七项实验的所有 p 值中，我们注意到大多数观测结果在统计上均显著。尽管 MA-TLQL 在许多实验中的表现优于最佳顾问，但其中一些比较在统计上并不显著（即， $p \geq 0.05$）。本文的主要实验考虑的是固定建议者，但我们的算法也可以与学习型／变化中的顾问一同实现（见附录 G）。在附录 I 中，我们研究了顾问数量不同时的表现。此外，我们的算法还可以与先前工作 [9] 中采用的对手建模技术结合使用（更多详情见附录 H）。

7 消融研究

本节将对 MA-TLQL 的三个组成部分进行消融研究。这三个部分使其有别于 Li 等人 [21] 先前提出的 TLQL 算法。回顾一下，这三个部分分别是：i）联合动作（JA）更新，ii）集成方法（EM），以及 iii）顾问评估（AE）。在这项消融研究中，我们将考虑 Pommerman 的双智能体版本，其中有四个充分顾问，它们的质量各不相同（实验 1）或相似（实验 2）。

与实验 1 对应的消融结果如图 9 所示。图中除了展示 TLQL 和 MA-TLQL 的表现外，还展示了分别加入三个组件之一的 TLQL 的表现。在图 9(a) 和 (b) 中，加入任一组件的 TLQL 表现都优于原版TLQL。采用集成方法的 TLQL（即 TLQL+EM）能够优于原版TLQL，因为在训练初期，顾问的 Q 值并不准确，而集成技术会选择给定顾问集合中多数顾问都认同的顾问动作（与我们在第 4 节中的讨论一致）。回顾一下，由四位不同顾问组成的集合中，各顾问的能力依次递减：前三位顾问能够教授一些有用的 Pommerman技能，而最后一位顾问只是随机顾问（见附录 K）。使用集成方法可以避免使用随机顾问，因为前三位顾问更有可能对某个动作达成一致，从而提高智能体选择该动作的可能性。此外，我们发现，与原版TLQL 中采用的独立更新相比，TLQL 从联合动作更新（即 TLQL+JA）中获益显著。联合动作更新会明确考虑其他智能体的策略，因此在多智能体环境中，相较于独立更新，能够提供更强的最佳响应。最后，与其他组件相比，在高-Q 表中采用顾问评估的 TLQL（即 TLQL+AE）带来的收益最大。正如第3节和第 4 节所讨论的，原版TLQL 中的高-Q 定义存在局限，因为通过高-Q 进行的顾问评估与不准确的强化学习策略耦合在一起（而顾问评估能够解决这一局限）。此外，从图 9 可以看出，与原版 TLQL 以及分别加入三个组件之一的 TLQL 实现相比，MA-TLQL（整合了全部三个组件）的表现最佳（$p < 0.05$）。因此，MA-TLQL 能够无缝整合 TLQL 各个组件的优势，展现出其优越性。

我们还使用实验2进行了类似的消融研究（见图10）。与图 9 一样，我们发现，包含这三个组件中任一组件的 TLQL，其表现都优于原版TLQL。由于我们有四位质量相似的顾问，且每位顾问都擅长不同的 Pommerman技能，因此它们在动作选择上的一致性预计较低。因此，集成技术（即 TLQL+EM）仅比原版TLQL略有提升。不过，另外两个组件（即 TLQL+JA 和 TLQL+AE）则显著提升了 TLQL 的表现。最后，整合了全部三个组件的 MA-TLQL 表现最佳（$p < 0.03$）。

8 结论

本文提出了一种在多智能体强化学习中向多个独立顾问学习的原则性方法。受 Li 等人启发， [21], 我们为多智能体环境提出了一种双层架构。我们讨论了 TLQL 的两个局限性，并在我们的方法中解决了这些局限性。此外，我们针对 MA-TLQL 的学习提供了不动点保证和样本复杂度上界。我们还提供了一种能够在集中训练、分散执行范式下运行的演员-评论家实现。进一步地，我们对合作、竞争性和混合设置下的 MA-TLQL 与 MA-TLAC 进行了广泛的实验分析，结果表明，这些算法能够恰当地利用一组顾问，并且表现优于基线方法。未来，我们希望考虑人类顾问，并进一步探索现实世界背景下的一些研究方向。

致谢

用于开展本研究的资源由安大略省政府和加拿大政府通过 CIFAR、NSERC 以及赞助 Vector Institute 的企业提供。本研究的部分工作在阿尔伯塔大学的智能机器人学习实验室（IRL）完成，该实验室部分得到了以下机构研究资助的支持：阿尔伯塔机器智能研究所（Amii）、加拿大CIFAR人工智能讲席（Amii）、Compute Canada、华为、Mitacs 和 NSERC。

参考文献

[1] Ofra Amir、Ece Kamar、Andrey Kolobov 和 Barbara J. Grosz。2016年。面向智能体训练的交互式教学策略。载于 *IJCAI*。IJCAI出版社，美国纽约州纽约市，2016年7月9日至15日，804–811。[2] Samuel Barrett、Avi Rosenfeld、Sarit Kraus 和 Peter Stone。2017年。即时结交朋友：与新队友合作。人工智能 242（2017年），132–171。[3] Dimitri P. Bertsekas 和 John N. Tsitsiklis。1996年。神经动态规划。优化与神经计算丛书，第3卷。Athena Scientific，美国Chestnut Street。[4] Tim Brys、Anna Harutyunyan、Halit Bener Suay、Sonia Chernova、Matthew E. Taylor 和 Ann Nowé。2015年。通过塑形从示范中进行强化学习。载于 *IJCAI，2015*年*7*月*25*日至*31*日。AAAI出版社，阿根廷布宜诺斯艾利斯，3352–3358。[5] Eyal Even-Dar 和 Yishay Mansour。2003年。Q学习的学习率。机器学习研究期刊 5（2003年），1–25。[6] Fernando Fernández 和 Manuela M. Veloso。2006年。在强化学习智能体中进行概率策略重用。载于第五届自主智能体与多智能体系统国际联合会议（*AAMAS 2006*），*2006*年*5*月*8*日至*12*日。ACM，日本函馆，720–727。[7] Yang Gao、Huazhe Xu、Ji Lin、Fisher Yu、Sergey Levine 和 Trevor Darrell。2018年。从不完美示范中进行强化学习。载于 *ICLR，2018*年*4*月*30*日至*5*月*3*日，研讨会论文集。OpenReview.net，加拿大不列颠哥伦比亚省温哥华。[8] Jayesh K Gupta、Maxim Egorov 和 Mykel Kochenderfer。2017年。使用深度强化学习进行协作式多智能体控制。载于 *AAMAS*。施普林格，IFAAMAS，巴西圣保罗，66–83。[9] He He、Jordan Boyd-Graber、Kevin Kwok 和 Hal Daumé III。2016年。深度强化学习中的对手建模。载于 国际机器学习会议。PMLR，美国纽约市，1804–1813。[10] Bassam Helou、Aditya Dusi、Anne Collin、Noushin Mehdipour、Zhiliang Chen、Cristhian Lizarazo、Calin Belta、Tichakorn Wongpiromsarn、Radboud Duintjer Tebbens 和 Oscar Beijbom。2021年。理性人群：面向基于证据且可解释的驾驶行为模型。载于 *IROS*。IEEE，捷克共和国布拉格，6708–6715。[11] Pablo Hernandez-Leal、Bilal Kartal 和 Matthew E. Taylor。2019年。多智能体深度强化学习的综述与批评。自主智能体与多智能体系统 33，6（2019年11月1日），750–797。https://doi.org/10.1007/s10458- 019-09421-1[12] Todd Hester、Matej Vecerík、Olivier Pietquin、Marc Lanctot、Tom Schaul、Bilal Piot、Dan Horgan、John Quan、Andrew Sendonaris、Ian Osband、Gabriel Dulac-Arnold、John P. Agapiou、Joel Z. Leibo 和 Audrunas Gruslys。2018年。从示范中进行深度Q学习。载于 *AAAI，2018*年*2*月*2*日至*7*日。AAAI出版社，美国路易斯安那州新奥尔良。[13] Junling Hu 和 Michael P Wellman。2003年。一般和随机博弈中的纳什Q学习。*JMLR* 4，11月（2003年），1039–1069。[14] Tommi Jaakkola、Michael I. Jordan 和 Satinder P. Singh。1994年。关于随机迭代动态规划算法的收敛性。《神经计算》 6，6（1994年），1185–1201。https://doi.org/10.1162/neco.1994.6.6.1185[15] Piyush Jain、Sean CP Coogan、Sriram Ganapathi Subramanian、Mark Crowley、Steve Taylor 和 Mike D Flannigan。2020年。野火科学与管理中机器学习应用的综述。环境评论 28，4（2020年），478–505。[16] Mingxuan Jing、Xiaojian Ma、Wenbing Huang、Fuchun Sun、Chao Yang、Bin Fang 和 Huaping Liu。2020年。软专家指导下从不完美示范中进行强化学习。载于 *AAAI，2020*年*2*月*7*日至*12*日。AAAI出版社，美国纽约州纽约市，5109–5116。[17] Beomjoon Kim、Amir-massoud Farahmand、Joelle Pineau 和 Doina Precup。2013年。从有限示范中学习。载于 *NeurIPS*。Morgan Kaufmann出版社，美国内华达州太浩湖，2859–2867。[18] Dong-Ki Kim、Miao Liu、Shayegan Omidshafiei、Sebastian Lopez-Cot、Matthew Riemer、Golnaz Habibi、Gerald Tesauro、Sami Mourad、Murray Campbell 和 Jonathan P. How。2020年。为协作智能体学习分层教学策略。载于 *AAMAS，2020*年*5*月*9*日至*13*日。IFAAMAS，新西兰奥克兰，620–628。[19] Vijay R. Konda 和 John N. Tsitsiklis。1999年。演员-评论家算法。载于 NeurIPS。麻省理工学院出版社，美国科罗拉多州丹佛。[20] Sergey Levine、Chelsea Finn、Trevor Darrell 和 Pieter Abbeel。2016年。深度视觉运动策略的端到端训练。《机器学习研究杂志》17，1（2016年），1334–1373。[21] Mao Li、Yi Wei 和 Daniel Kudenko。2019年。双层Q学习：从冲突示范中学习。《知识工程评论》34（2019年）。[22] Michael L. Littman。1994年。马尔可夫博弈：多智能体强化学习的框架。载于 ICML，1994年7月10日至13日。Morgan Kaufmann，美国新泽西州新不伦瑞克，157–163。[23] Qinghua Liu、Tiancheng Yu、Yu Bai 和 Chi Jin。2021年。自我博弈下基于模型的强化学习的精确分析。载于 ICML（机器学习研究论文集，第139卷）。PMLR，线上会议，7001–7010。[24] Ryan Lowe、Yi Wu、Aviv Tamar、Jean Harb、Pieter Abbeel 和 Igor Mordatch。2017年。混合合作竞争环境中的多智能体演员-评论家。载于 NeurIPS，2017年12月4日至9日。Morgan Kaufmann出版社，美国加利福尼亚州长滩，6379–6390。[25] Laetitia Matignon、Guillaume J Laurent 和 Nadine Le Fort-Piat。2012年。合作马尔可夫博弈中的独立强化学习者：关于协调问题的综述。《知识工程评论》27，1（2012年），1–31。[26] Volodymyr Mnih、Koray Kavukcuoglu、David Silver、Andrei A Rusu、Joel Veness、Marc G Bellemare、Alex Graves、Martin Riedmiller、Andreas K Fidjeland、Georg Ostrovski 等。2015年。通过深度强化学习实现人类水平的控制。《自然》518，7540（2015年），529–533。[27] John Nash。1951年。非合作博弈。《数学年刊》（1951年），286–295。[28] Shayegan Omidshafiei、Dong-Ki Kim、Miao Liu、Gerald Tesauro、Matthew Riemer、Christopher Amato、Murray Campbell 和 Jonathan P. How。2019年。在合作式多智能体强化学习中学习教学。载于 AAAI，2019年1月27日至2月1日。AAAI出版社，美国夏威夷州檀香山，6128–6136。[29] Bilal Piot、Matthieu Geist 和 Olivier Pietquin。2014年。处理专家示范的增强型贝尔曼残差最小化。载于 ECML-PKDD，2014年9月15日至19日，8725 549 564 - Vol. . Springer，法国南锡，–。[30] Tummalapalli Sudhamsh Reddy、Vamsikrishna Gopikrishna、Gergely V. Zaruba 和 Manfred Huber。2012年。去中心化非合作多智能体系统的逆强化学习。载于 IEEE系统、人类与控制论国际会议论文集（SMC 2012），2012年10月14日至17日。IEEE，韩国首尔，1930–1935。[31] Cinjon Resnick、Wes Eldridge、David Ha、Denny Britz、Jakob Foerster、Julian Togelius、Kyunghyun Cho 和 Joan Bruna。2018年。Pommerman：多智能体试验场。arXiv预印本 arXiv:1809.07124（2018年）。[32] Lloyd S Shapley。1953年。随机博弈。《美国国家科学院院刊》39，10（1953年），1095–1100。[33] Felipe Leno Da Silva 和 Anna Helena Reali Costa。2019年。多智能体强化学习系统迁移学习综述。JAIR 64（2019年），645–703。[34] Felipe Leno Da Silva、Ruben Glatt 和 Anna Helena Reali Costa。2017年。在多智能体强化学习中同时进行学习与建议。载于 AAMAS，2017年5月8日至12日。ACM，巴西圣保罗，1100–1108。[35] David Silver、Aja Huang、Chris J Maddison、Arthur Guez、Laurent Sifre、George Van Den Driessche、Julian Schrittwieser、Ioannis Antonoglou、Veda Panneershelvam、Marc Lanctot 等。2016年。利用深度神经网络和树搜索掌握围棋。《自然》529，7587（2016年），484。[36] Ziang Song、Song Mei 和 Yu Bai。2021年。何时能够以较大的样本效率学习拥有大量参与者的一般和马尔可夫博弈？arXiv预印本 arXiv:2110.04184（2021年）。[37] Sriram Ganapathi Subramanian。2023年。多智能体强化学习中向多个独立顾问学习。https://github.com/Sriram94/matlql [38] Sriram Ganapathi Subramanian、Kate Larson、Matthew Taylor 和 Mark Crowley。2022年。多智能体顾问Q学习。《人工智能研究杂志》74（2022年），1–74。[39] Richard S Sutton 和 Andrew G Barto。1998年。强化学习导论。第135卷。MIT出版社，剑桥。[40] Csaba Szepesvari 和 Michael L Littman。1999年。基于值函数的强化学习算法的统一分析。《神经计算》11，8（1999年），2017–2060。[41] Ming Tan。1993年。多智能体强化学习：独立智能体与协作智能体。载于 ICML。剑桥大学出版社，美国马萨诸塞州阿默斯特，330–337。[42] Matthew E. Taylor、Halit Bener Suay 和 Sonia Chernova。2011年。将强化学习与不同能力水平的人类示范相结合。载于 AAMAS，2011年5月2日至6日。IFAAMAS，台湾台北，617–624。[43] Justin K Terry、Benjamin Black、Mario Jayakumar、Ananth Hari、Luis Santos、Clemens Dieffendahl、Niall L Williams、Yashas Lokesh、Ryan Sullivan、Caroline Horsch 和 Praveen Ravi。2020年。PettingZoo：面向多智能体强化学习的 Gym。（2020年）。[44] Lisa Torrey 和 Matthew E. Taylor。2013年。低预算教学：智能体在强化学习中为智能体提供建议。载于 AAMAS，2013年5月6日至10日。IFAAMAS，美国明尼苏达州圣保罗，1053–1060。

[45] Hado van Hasselt。2010年。双重Q学习。载于 *NeurIPS*。Curran Associates, Inc.，加拿大不列颠哥伦比亚省温哥华，2613–2621。[46] Yixi Wang、Wenhuan Lu、Jianye Hao、Jianguo Wei 和 Ho-fung Leung。2018年。通过结合师生机制的强化社会学习实现高效的惯例涌现。载于 *AAMAS，2018*年*7*月*10*日至*15*日。IFAAMAS / ACM，瑞典斯德哥尔摩，795–803。[47] Zhaodong Wang 和 Matthew E. Taylor。2017年。利用基于置信度的示范改进强化学习。载于 *IJCAI，2017*年*8*月*19*日至*25*日。ICJAI，澳大利亚墨尔本，3027–3033。[48] Christopher JCH Watkins 和 Peter Dayan。1992年。Q学习。机器学习 8，3-4（1992年），279–292。[49] Kevin Waugh、Brian D. Ziebart 和 Drew Bagnell。2011年。计算理性化：逆均衡问题。载于 *ICML，2011*年*6*月*28*日至*7*月*2*日。Omnipress，美国华盛顿州贝尔维尤，1169–1176。[50] Tianpei Yang、Weixun Wang、Hongyao Tang、Jianye Hao、Zhaopeng Meng、Hangyu Mao、Dong Li、Wulong Liu、Yingfeng Chen、Yujing Hu 等。2021年。一种高效的多智能体强化学习迁移学习框架。*NeurIPS*，线上会议 34（2021年）。[51] Yaodong Yang、Rui Luo、Minne Li、Ming Zhou、Weinan Zhang 和 Jun Wang。2018年。平均场多智能体强化学习。载于 *ICML*，第80卷。PMLR，瑞典斯德哥尔摩，5571–5580。[52] Dayong Ye、Tianqing Zhu、Zishuo Cheng、Wanlei Zhou 和 S Yu Philip。2020年。多智能体强化学习中的差分建议。*IEEE* 《控制论汇刊》（2020年）。

A 算法伪代码

算法 1 给出了我们基于 Q- 学习的算法（MA-TLQL）的表格实现的完整伪代码。所有智能体在第 2 行初始化一个低-Q 表和一个高-Q 表。然后，在每个状态下，所有智能体在第 8–20 行选择执行一个动作。该动作可以来自顾问，也可以来自第 4 节所述的强化学习策略。接着执行该动作，并在第 21 行观测下一个状态和奖励。最后，根据第 4 节给出的公式，更新低-Q和高-Q 对应的 Q 值（第 22 行和第 23 行）。在训练过程中，ϵ，的值从较高值线性衰减至接近零的值（第 24 行）。

为了使算法 1 适用于高维状态和动作空间，我们在算法 2 中给出了 MA-TLQL 的一种基于函数逼近的实现。这里使用神经网络作为函数逼近器，并采用独立的目标网络和经验回放缓冲区进行训练，正如著名的 DQN 算法 [26] 所介绍的那样。智能体维护一个高-Q 网络和两个低-Q 网络（评估网络和目标网络），并使用时序差分（T.D.）误差，根据第 4 节给出的更新方程更新这些网络。如果无法获得随机博弈的完整状态，智能体可以直接使用其观测代替状态，这适用于大多数基于函数逼近的强化学习方法。

我们还将算法 2 扩展为算法 3 中描述的演员-评论家实现。该算法称为多智能体双层演员-评论家（MA-TLAC）。该算法将策略用作演员，将 Q-值用作评论家，这与先前工作 [19] 一致。我们保留两个演员和两个评论家，以体现该算法的双层（高层和低层）结构。高层演员决定顾问，高层评论家则利用时序差分误差帮助训练高层演员。类似地，低层演员决定适当的动作，低层评论家则提供用于训练的时序差分误差。在 MA-TLAC 中，由于我们使用单独的演员网络来选择顾问，因此不采用公式 2 中的集成技术。相反，高层演员直接从给定的顾问中为当前状态选择一个。该算法的优点是可以使用流行的集中训练、分散执行范式 [24], 实现，因为演员在选择动作或顾问时不需要其他智能体的动作。在集中训练、分散执行中，训练期间可以获得全局信息（即来自其他智能体的信息），但执行期间则无法获得。这种基于集中训练、分散执行的实现还使我们的方法能够用于部分可观测领域，因为演员可以利用局部观测来选择动作或顾问，而评论家在训练期间可以使用联合动作和状态，如 Lowe 等人 [24]所述。此外，由于 MA-TLAC 不使用集成技术（公式 2），它也适用于连续状态空间环境（不同于 MA-TLQL，后者仅适用于具有离散状态空间的环境）。

本节提供的所有算法伪代码均假设所有智能体在学习时使用相同的算法步骤，并且能够像先前工作 [13] 中那样保存其他智能体更新的副本。如果无法做到这一点，智能体就会直接使用观测到的其他智能体先前动作进行更新。

算法 1 MA-TLQL 表格方法

1：令 Ad^j 表示智能体 j 可用的顾问集合。2：对于所有 $j \in 1$、……、N、$s \in S$ 和 $a^j \in A^j$：$lowQ^j(s, a^j, \boldsymbol{a}^{-j}) \leftarrow 0$ 其中 $\boldsymbol{a}^{-j} = [a^1, \ldots, a^{j-1}, a^{j+1}, \ldots, a^N]$：对于所有 $s \in S$ 和所有 $j \in 1$、……、N：$highQ^j(s, \boldsymbol{a}^{-j}, ad^j) \leftarrow 0$ 3 4：为超参数 ϵ 和 ϵ' 及 η 初始化一个值。5：当训练尚未结束时执行：6：对于每个智能体 j，获取当前状态 s。7：对于每个智能体 j，使用各自的副本和所有智能体之前的动作，获取其他智能体在状态 s 下的联合动作 $\boldsymbol{a}^{-j}$。8：对于每个智能体 j，令 u 为一个介于 0 和 1 之间的均匀随机数。9：如果 $u < \epsilon'$，则：10：令 u' 为一个介于 0 和 1 之间的均匀随机数。11：如果 $u' < \eta$，则：12：根据智能体 j 在当前状态及其他智能体联合动作下的高-Q 值，利用式 2 选择一个顾问，并将其动作作为当前动作 a_t^j。13：否则：14：从 Ad^j 中随机选择一个顾问作为顾问 ad^j，并将其动作作为当前动作 a_t^j。15：结束 if。16：否则，如果 $u > \epsilon'$ 且 $u < \epsilon$，则：17：从动作空间 A^j 中随机选择一个动作作为动作 a_t^j。18：否则：19：利用 s 以及其他智能体的联合动作，从低-Q 值中选择一个贪心动作 a_t^j。20：结束 if。21：执行联合动作 $\boldsymbol{a}$，观察联合奖励 $\boldsymbol{r}$ 和下一个状态 $s' \quad \boldsymbol{a} = [a^1, \ldots, a^N] \quad \boldsymbol{r} = [r^1, \ldots, r^N]'$，22 其中 和 3：使用式 更新智能体 j 的低-Q 值。利用各自的副本和其他智能体之前的动作，获取其他智能体的下一个动作 $\boldsymbol{a}'^{-j}$。23：如果选择了顾问，则使用式 1 更新智能体 j 的该顾问的高-Q 值。24：每个回合结束时，线性衰减 ϵ'。25：结束 while

Alg算法 2MA-TLQL 神经网络方法

1 j：

2 $\ldots N$

令 Ad^j 表示智能体可用的顾问集合：对于所有 $j \in 1$、，初始化 Q_{ϕ^j}、$Q_{\phi^j_-}$（表示低-Q）。对于所有 $j \in 1$、……、N，初始化 Q_{θ^j}、$Q_{\theta^j_-}$（表示高-Q）。3：为超参数 ϵ 和 ϵ' 及 η 初始化一个值。4：当训练尚未结束时执行：5：对于每个智能体 j，获取当前状态 s。6：对于每个智能体 j，使用各自的副本和所有智能体之前的动作，获取其他智能体在状态 s 下的联合动作 $\boldsymbol{a}^{-j}$。7：对于每个智能体 j，令 u 为一个介于 0 和 1 之间的均匀随机数。8：如果 $u < \epsilon'$，则：9：令 u' 为一个介于 0 和 1 之间的均匀随机数。10：如果 $u' < \eta$，则：11：根据式 2 中智能体 j 在当前状态及其他智能体联合动作下的高-Q 值，利用高-Q、Q_{θ^j} 选择一个顾问，并将其动作作为当前动作 a_t^j。12：否则：13：从 Ad^j 中随机选择一个顾问作为顾问 ad^j，并将其动作作为当前动作 a_t^j。14：结束 if。15：否则，如果 $u > \epsilon'$ 且 $u < \epsilon$，则：16：从动作空间 A^j 中随机选择一个动作作为动作 a_t^j。17：否则：18：利用 s 以及其他智能体的联合动作，从低-Q、Q_{ϕ^j} 中选择一个贪心动作 a_t^j。19：结束 if。20：执行联合动作 $\boldsymbol{a}$，观察联合奖励 $\boldsymbol{r}$ 和下一个状态 s'，其中 $\boldsymbol{a} = [a^1, \ldots, a^N]$ 和 $\boldsymbol{r} = [r^1, \ldots, r^N]$。21：对于每个智能体 j，将 $s, \boldsymbol{a}, \boldsymbol{r}, s' \boldsymbol{a}'$ 存入经验回放缓冲区 $\mathbf{D}^j$，其中 $\boldsymbol{a} = [a^1, \ldots, a^N]$, $\boldsymbol{a}' = [a'^1$、……、$a'^N]$。利用各自的副本和其他智能体之前的动作，获取其他智能体的下一个动作 $\boldsymbol{a}'^{-j}$。22：如果使用了顾问，则对于每个智能体 j，将 $s, \boldsymbol{a}, \boldsymbol{r}, s' \boldsymbol{a}' ad^j$ 存入经验回放缓冲区 $\mathbf{D}'^j$，其中 ad^j 是该顾问。23：将下一个状态 s' 设为当前状态 s。24：每个回合结束时，线性衰减 ϵ'。25：当 $\mathbf{j} = 1$ 至 $\mathbf{N}$ 时执行：26：从 $\mathbf{D}^j$ 中采样一个包含 $\mathbf{K}$ 个经验 $s, \boldsymbol{a}, \boldsymbol{r}, s' \boldsymbol{a}'$ 的小批量。27：根据式 3 设置 $y^j = r^j + \gamma \mathbf{max}_{a'^j} Q_{\phi^j_-}(s' \boldsymbol{a}'^{-j}, a'^j)$。28：通过最小化损失 $\mathbf{L}(\phi^j) = \frac{1}{K} \sum (y^j - Q_{\phi^j}(s, \boldsymbol{a}^{-j}, a^j))^2$，更新 Q 网络 ϕ^j。29：从 $\mathbf{D}'$30 中采样一个包含 $\mathbf{K}$ 个经验 $s, \boldsymbol{a}, \boldsymbol{r}, s' \boldsymbol{a}' ad^j$ 的小批量。30：根据式 1 设置 $y^j = r^j + \gamma Q_{\theta^j_-}(s' \boldsymbol{a}'^{-j}, ad^j)$。31：通过最小化损失 $\mathbf{L}(\theta^j) = \frac{1}{K} \sum (y^j - Q_{\theta^j}(s, \boldsymbol{a}^{-j}, ad^j))^2$，更新 Q 网络 θ^j。32：结束 **while**。33：每隔 $\mathbf{T}$ 步，通过复制评估网络来更新每个智能体的目标网络参数：$\phi^j_- \leftarrow \phi^j$ 和 $\theta^j_- \leftarrow \theta^j$。34：结束 **while**

算法 3 MA-TLAC

1：令 Ad^j 表示智能体 j 可用的一组顾问 2：初始化 Q_{ϕ^j}、$\pi_{\phi^j_-}$，即所有 $j \in \{1, \ldots, n\}$ 的低层评论家网络和演员网络 3：初始化 Q_{θ^j}、$\pi_{\theta^j_-}$，即所有 $j \in \{1, \ldots, n\}$ 的高层评论家网络和演员网络 4：为超参数 ϵ 和 ϵ' 及 η 初始化一个值 5：当训练尚未完成时执行 6：对于每个智能体 j，获取当前状态 s 7：对于每个智能体 j，令 u 为 0 到 1 之间的均匀随机数 8：如果 $u < \epsilon'$ 9：令 u' 为 0 到 1 之间的均匀随机数 10：如果 $u' < \eta$，则 11：使用高层演员 π_{θ^j}，为智能体 j、当前状态 s 选择一个顾问 ad^j，并将其动作作为当前动作 a^j 12：否则 13：将顾问 ad^j 设为来自 Ad^j 的随机顾问，并将其动作作为当前动作 a^j 14：结束条件判断 15：否则，如果 $u > \epsilon'$ 且 $u < \epsilon$，则 16：将动作 a^j 设为动作空间 A^j 中的随机动作 17：否则 18：使用 s 从低层演员 π_{ϕ^j} 中选择一个贪心动作 a^j 19：结束条件判断 20：执行联合动作 $\boldsymbol{a}$，观测联合奖励 $\boldsymbol{r}$ 和下一个状态 s'，其中 $\boldsymbol{a} = [a^1, \ldots, a^N]$ 和 $\boldsymbol{r} = [r^1, \ldots, r^N]$ 21：对于每个智能体 j，获取状态 s

22 $s \ \boldsymbol{a}^{-j} \ a^j$ 下其他智能体的联合动作 $\boldsymbol{a}^{-j}$（观测到的其他智能体当前动作）：根据公式 3 设置 $y^j = r^j + \gamma \mathbf{max}_{a'^j} Q_{\phi^j}(\prime\prime, \prime)$ 23：对于每个 j，通过最小化损失 $\mathbf{L}(\phi^j) = (y^j - Q_{\phi^j}(s, \boldsymbol{a}^{-j}, a^j))^2$ 来更新低层评论家 24：对于每个 j，使用关系式 $A(s, \boldsymbol{a}^{-j}, a^j) = y^j - \sum_{a^j} \pi_{\phi^j_-}(a^j|s) Q_{\phi^j}(s, \boldsymbol{a}^{-j}, a^j)$ 计算优势估计 25：对于每个 j，使用对数损失 $\mathbf{J}(\phi^j_-) = \mathbf{log}^{\pi_{\phi^j_-}}(a^j|s) A(s, \boldsymbol{a}^{-j}, a^j)$ 更新低层演员 26：如果智能体 j 使用了顾问 ad^j，则更新该顾问的 Q 估计值。27：对于每个 j，根据公式 1 设置 $y^j = r^j + \gamma Q_{\theta^j}(s' \boldsymbol{a}'^{-j}, ad^j)$ 28：从相应的副本中获取其他智能体的下一个动作 $\boldsymbol{a}'^{-j}$ 29：对于每个 j，通过最小化损失 $\mathbf{L}(\theta^j) = (y^j - Q_{\theta^j}(s, \boldsymbol{a}^{-j}, ad^j))^2$ 来更新高层评论家，其中 ad^j 是智能体 j 选择的顾问 30：对于每个 j，使用关系式 $A(s, \boldsymbol{a}^{-j}, ad^j) = y^j - \sum_{ad^j} \pi_{\theta^j_-}(ad^j|s) Q_{\theta^j}(s, \boldsymbol{a}^{-j}, ad^j)$ 计算优势估计 31：对于每个 j，使用对数损失 $\mathbf{J}(\theta^j_-) = \mathbf{log}^{\pi_{\theta^j_-}}(a^j|s) A(s, \boldsymbol{a}^{-j}, ad^j)$ 更新高层演员 32：将下一个状态设为当前状态 $s = s'$ 33：在每个回合结束时，线性衰减 ϵ' 34：结束循环

B 定理1的证明

定理1。 给定假设*1*、*2*、*3*，智能体 j 的低层 Q 值在极限情况下收敛到其纳什 Q 值（$t \to \infty$）。

证明。我们的证明思路将参照 Subramanian 等人 [38] 中的定理 3。

我们来考虑一个先前工作中的引理。

引理1。 一个随机迭代过程

$$\Delta_{t+1}(x) = (1 - \alpha_t(x))\Delta_t(x) + \alpha_t(x)F_t(x) \tag{7}$$

其中 $x \in X, t = 0, 1, \ldots, \infty$, 在满足以下性质时以概率 *1* 收敛到零（*w. p. 1*）：

1. 可能的状态集合 X 是有限的。

2. $0 \le \alpha_t(x) \le 1, \sum_t \alpha_t(x) = \infty, \sum_t \alpha_t^2(x) < \infty$ 以概率*1*成立，其中概率是针对学习率 α_t而言的。

3. $||\mathbb{E}\{F_t(x)|\mathscr{P}_t\}||_W \le \mathscr{K}||\Delta_t||_W + c_t$，其中 $\mathscr{K} \in [0, 1)$ 且c_t 以概率*1*收敛至零。

4. 方差$\{F_t(x)|\mathscr{P}_t\} \le K(1 + ||\Delta_t||_W)2$，其中 K 是某个常数。

这里，$\mathscr{P}_t$是一个递增的 σ-域序列，其中包含该过程的过去信息。特别地，我们假设 $\alpha_t, \Delta_t, F_{t-1} \in \mathscr{P}_t$。记号 $||\cdot||_W$表示某个（固定的）加权最大范数，而记号方差 表示方差。

证明。参见Jaakola等人的定理1 [14]， 证明。 □

本节中，由于我们只关注低-Q值，为了略微简化记法，我们将用 Q 表示低-Q 值。现在，我们使用以下公式定义纳什算子 P_t:

$$P_t Q^k(s, \boldsymbol{a}) = \mathbb{E}_{s' \sim p}[r_t^k(s, \boldsymbol{a}) + \gamma \pi_*^1(s') \cdots \pi_*^n(s') Q^k(s')] \tag{8}$$

其中，s'是时间 $t+1$的状态，$(\pi_*^1(s') \ldots, \pi_*^n(s'))$ 是阶段博弈 $(Q^1(s') \ldots, Q^n(s'))$ 的纳什均衡解，而 p是转移函数。 Q^k表示代表性智能体 k的 Q值。

引理2。 在假设 *3* 下，如公式 *8* 中定义的纳什算子构成一个压缩映射，其不动点是该博弈的纳什 Q值。

证明。参见 Hu 和 Wellman [13]的定理17。 □

现在，由于 P_t算子构成一个压缩映射，因此对于某个 $\gamma \in [0, 1)$ 和所有 Q^j， $||P_t Q^j - P_t Q_*^j|| \le \gamma ||Q^j - Q_*^j||$成立。这里， Q_*^j是智能体 j的纳什 Q值。

目标是应用引理1，证明 MATLQL 中的低-Q 收敛到纳什 Q 值。

引理1的前两个条件由假设 1 和假设 2 得到满足。现在，比较式 7 和式 3，可得 x 可与状态—联合动作对（s， $\boldsymbol{a}$）相关联，而 Δ_t（s_t， a_t）可与 Q_t^j（s， $\boldsymbol{a}$）$-Q_*^j$（s， $\boldsymbol{a}$）相关联。这里，Q_*^j（s， $\boldsymbol{a}$）是智能体 j 的纳什 Q值。

现在我们得到

$$\Delta_{t+1}(x) = (1 - \alpha_t(x))\Delta_t(x) + \alpha_t(x)F_t(x), \tag{9}$$

其中

$$\begin{aligned} F_t(x) &= r_t^j + \gamma v^{Nash,j}(s_{t+1}) - Q_*^j(s_t, \boldsymbol{a}_t) \\ &\quad + \gamma[\max_{a^j} Q_t^j(s_{t+1}, \boldsymbol{a}_{t+1}) - v^{Nash,j}(s_{t+1})] \\ &\triangleq r_t^j + \gamma v^{Nash,j}(s_{t+1}) - Q_*^j(s_t, \boldsymbol{a}_t) + C_t^j(s_t, \boldsymbol{a}_t) \\ &\triangleq F_t^{Q,j}(s_t, \boldsymbol{a}_t) + C_t^j(s_t, \boldsymbol{a}_t) \end{aligned} \tag{10}$$

智能体 j 的纳什价值函数 $v^{Nash,\ j}(s)$ 定义为：在所有智能体都遵循纳什策略 $\boldsymbol{\pi}_*$ 的情况下，智能体 j 获得的期望累积折扣未来奖励。

这里，我们令 F_t（s_t， $\boldsymbol{a}_t$）$= F^Q$，$_t{}^j$（s_t， $\boldsymbol{a}_t$）$= C_t^j$（s_t， $\boldsymbol{a}_t$）$= 0$，当且仅当（s， $\boldsymbol{a}$）$\neq$（s_t， $\boldsymbol{a}_t$）。令由（s_t， α_t， a_t^1，. . ., a_t^n， r_{t-1}, . . ., s_1， α_1， a_1， Q_0）提供的所有随机变量生成的 σ-域表示为 $\mathscr{P}_t$。现在，所有 Q-值均为 $\mathscr{P}_t$可测的，因此 Δ_t和 F_t， $\mathscr{P}_t$均可测，这满足引理1的可测性条件。

Hu 和 Wellman [13] 证明了该结果， $v^{Nash,\ j}(s_{t+1}) \triangleq v^j(s', \pi_*^1, \ldots\ldots, \pi_*^n) = \pi_*^1(s') \cdots \pi_*^n(s') Q_*^j(s')$成立（参见 [13]的引理 10 中的证明）。因此，根据引理2，我们可以证明$\mathbb{E}[F_t^Q]$ 构成一个压缩映射。这可以利用以下事实得到： $\mathbb{E}(P_t Q_*) = Q_*$（参见 [13]中的引理 11）。这里，范数是联合动作上的最大范数。

现在，对于所有 t，有如下结论：

$$||\mathbb{E}[F_t^{Q,j}(s_t, \boldsymbol{a}_t)|\mathscr{P}_t]|| \le \gamma ||Q_t^j(s_t, \boldsymbol{a}_t) - Q_*^j(s_t, \boldsymbol{a}_t)|| = \gamma ||\Delta_t|| \tag{11}$$

现在根据公式 10,

$$\begin{aligned} &||\mathbb{E}[F_t(s_t, \boldsymbol{a}_t)|\mathscr{P}_t]|| \\ &\le ||\mathbb{E}[F_t^{Q,j}(s_t, \boldsymbol{a}_t)|\mathscr{P}_t]|| + ||\mathbb{E}[C_t^j(s_t, \boldsymbol{a}_t)|\mathscr{P}_t]|| \\ &\le \gamma ||\Delta_t|| + ||\mathbb{E}[C_t^j(s_t, \boldsymbol{a}_t)|\mathscr{P}_t]|| \end{aligned} \tag{12}$$

只要 $c_t \ni ||\mathbb{E}[C_t^j(s_t, \boldsymbol{a}_t)|\mathscr{P}_t]||$ 以概率 1 收敛于 0（w. p. 1.），这就满足引理1的第三个条件。根据 C_t^j的定义和假设 3，可以证明 C_t^j在时间趋于无穷时收敛于 0（参见 Subramanian 等人的定理 3 [38]）。

因此，由引理1可知，过程 Δ_t 收敛到 0，因此，智能体 j 的低-Q 值收敛到纳什 Q值 Q_*^j。

□

C 定理 2 的证明

本节给出定理 2 的证明。为了给出这个上界，我们使用覆盖时间的概念 L。覆盖时间是指从任意初始状态出发，在 L 步内，所有智能体至少执行一次所有状态-联合动作对。与 Dar 和 Mansour [5], 类似，我们澄清一点：无需假设状态-联合动作对是由任何特定策略生成的。与附录 B 中一样，在本节中，我们将用 Q 表示低-Q 值。对于代表性智能体 j，我们将关注 $r_t^j = ||Q_t^j - Q_*^j||$的值，目标是给出直到 $r_t^j \le \epsilon$的时间上界。这里的范数表示所有状态和联合动作下 Q 值的最大差异。本

该定理的证明遵循 Dar 和 Mansour 的定理4 [5]。Dar 和 Mansour 的研究局限于单智能体 MDP，而我们的结果将 Dar 和 Mansour 的分析扩展到了 一般和随机博弈 setting。

根据公式4，我们考虑如下形式的随机迭代过程：

$$X^j_{t+1}(i) = (1-\alpha_t(i))X^j_t(i) + \alpha_t(i)((H_tX^j_t)(i) + w^j_t(i)). \quad (13)$$

如第5节所述，我们指定 $\beta = \frac{1-\gamma}{2}$。此外，考虑常数 $Q^j_{\max}$，其中 $Q^j{\max}$ 表示智能体 j 在随机博弈中可能获得的最低 Q 值的最大值。因此，关系 $||X_0|| \le Q^j{\max}$ 成立。进一步地，考虑序列 D^j_k，其中 $D^j_1 = Q^j{\max}$ 且 $D^j_{k+1} = (1-\beta)D^j_k$ 对所有 $k \ge 1$ 均成立。根据这一构造的性质，序列 D^j_k 保证收敛于0，因为在每个步骤中，D^j_k 的值都会持续乘以一个分数值。现在我们可以证明以下结果。

引理3。对于每个 k，都存在一个时间 τ_k，使得对于任意 $t \ge \tau_k$，均有 $||X^j_t|| \le D^j_k$。

证明。参见 Dar 和 Mansour [5] 的定理 8。 □

引理3保证，在时间 $t \ge \tau_k$，对于任意 i，$||X^j_t(i)||$ 的值都在区间 $[-D^j_k, D^j_k]$ 内。

我们可以陈述以下引理，其中给出直到 $D^j_i \le \epsilon$ 为止所需迭代次数的上界。

引理 4。对于 $m \ge \frac{1}{\beta}\ln(Q^j{\max}/\epsilon)$，有 $D^j_m \le \epsilon$。

证明。由于我们有 $D^j_1 = Q^j{\max}$ 且 $D^j_i = (1-\beta)D^j_{i-1}$，因此需要一个满足 $D^j_m = Q^j{\max}(1-\beta)^m \le \epsilon$ 的 m。对两边取对数，得到

$$\begin{aligned}
&\ln\left((Q^j_{\max})(1-\beta)^m\right) \le \ln(\epsilon)\\
&\ln(Q^j_{\max}) + m\ln(1-\beta) \le \ln(\epsilon)\\
&\implies \ln(Q^j_{\max}) - \ln(\epsilon) \le -m\ln(1-\beta) \quad (14)\\
&\implies \ln(Q^j_{\max}/\epsilon) \le m\textstyle\sum_{k=0}^{\infty}\beta^k/k\\
&\implies 1/\beta\ln(Q^j_{\max}/\epsilon) \le m.
\end{aligned}$$

在最后一步中，我们省略了高次幂，因为 β 是一个较小的分数。这就证明了该结论。□

现在，我们定义一个时间序列 τ_k，它对应于采用多项式学习率的 MA-TLQL 底层 Q-更新。令 $\tau_{k+1} = \tau_k + L\tau^\omega_k$。其中，术语 ω 表示学习率的衰减因子，其满足 $\alpha^\omega_t = \frac{1}{(t+1)^\omega}$，其中 $\omega \in (1/2, 1)$。术语 $L\tau^\omega_k$ 表示至少更新每个状态 - 联合动作对 τ^ω_k 次所需的步数。τ_k 与 τ_{k+1} 之间的时间记为第 k 次迭代。现在，我们给出状态-联合动作对被访问次数的定义。

定义1。令 $n(s, \boldsymbol{a}, t_1, t_2)$ 表示状态-联合动作对 $(s, \boldsymbol{a})$ 在时间步区间 $[t_1, t_2]$ 内执行的次数。

在给出合适的上界之前，我们先给出一些方程，用以说明式 13 中的随机迭代技术与式 4 中的 Q-更新之间的关系。

首先，我们给出随机博弈的形式化定义，这将有助于后续分析。考虑如下定义的随机博弈：

定义 2. 随机博弈定义为 $\langle S\ N, A\ P, R\ \gamma\rangle$，其中 S 是有限状态集，N 是有限智能体集，$|N| = n$，$\boldsymbol{A} = A^1 \times \ldots \times A^n$ 是联合动作集，其中 A^j 是智能体 j 的有限动作集，而 $\boldsymbol{a} = (a^1, \ldots, a^n) \in \boldsymbol{A}$ 是联合动作，其中智能体 j 执行动作 $a^j \in A^j$。此外，$P_{i,k}(\boldsymbol{a}) : S \times \boldsymbol{A} \times S \mapsto [0,1]$ 是转移函数，它给出当所有智能体在状态 i 下执行联合动作 $\boldsymbol{a} \in \boldsymbol{A}$ 时，从状态 i 转移到状态 k 的概率；$\boldsymbol{R(s, a)} = \{R^1(s, \boldsymbol{a}) \ldots, R^n(s, \boldsymbol{a})\}$ 是奖励函数集，其中 $R^j(s, \boldsymbol{a}) : S \times \boldsymbol{A} \mapsto \mathbb{R}^n$ 是智能体 j 的奖励函数，γ 是满足 $0 \le \gamma < 1$ 的折扣因子。

为此，我们定义一个算子 H，它可以表示为

$$(HQ^j)(i, \boldsymbol{a}) = \textstyle\sum_{k=0}^{|S|} P_{ik}(\boldsymbol{a})(R^j(i, \boldsymbol{a}) + \gamma \max_{b^j \in A^j} Q^j(k, \boldsymbol{b})). \quad (15)$$

这里，$\boldsymbol{b} = \{b^j, b^{-j}\}$，其中 b^{-j} 表示除智能体 j 之外所有智能体的联合动作。

用 H 重写 Q-函数，得到：

$$Q^j_{t+1}(i, \boldsymbol{a}) = (1-\alpha_t(i, \boldsymbol{a}))Q^j_t(i, \boldsymbol{a}) + \alpha_t(i, \boldsymbol{a})((HQ^j_t)(i, \boldsymbol{a}) + w^j_t(i, \boldsymbol{a})). \quad (16)$$

设 i 为在状态 i 下于时间 t 执行联合动作 $\boldsymbol{a}$ 后到达的状态，且 $r^j(i, \boldsymbol{a})$ 为智能体 j 在状态 i 观察到的奖励；则

$$\begin{aligned}
w^j_t(i, \boldsymbol{a}) = {} & r^j(i, \boldsymbol{a}) + \gamma \max_{b^j \in A^j} Q^j_t(\bar{i}, \boldsymbol{b})\\
& - \textstyle\sum_{k=0}^{|S|} P^j_{ik}(\boldsymbol{a})\Big(R^j(i, \boldsymbol{a}) + \gamma \max_{b^j \in A^j} Q^j_t(k, \boldsymbol{b})\Big). \quad (17)
\end{aligned}$$

根据这一构造，对于所有 t，w^j_t 均以 $Q^j{\max}$ 为上界，且其期望为零。此外，我们还将定义另外两个序列 $W^j_{t;\tau}$ 和 $Y^j_{t;\tau}$，其中 τ 表示某个初始时间。它们由以下方程给出。

$$W^j_{t+1;\tau_k}(s, \boldsymbol{a}) = (1-\alpha^\omega_t(i))W^j_{t;\tau_k}(s, \boldsymbol{a}) + \alpha^\omega_t(i)w^j_t(s, \boldsymbol{a}) \quad (18)$$

其中 $W^j_{\tau_k;\tau_k}(s, \boldsymbol{a}) = 0$。$W^j_{t;\tau_k}$ 的值界定了从任意 τ_k 开始时，$w^j_t(s, \boldsymbol{a})$ 对 Q^j_t 的值所作的贡献。现在我们有

$$Y^j_{t+1;\tau_k}(s, \boldsymbol{a}) = (1-\alpha^\omega_t(i))Y^j_{t;\tau_k}(s, \boldsymbol{a}) + \alpha^\omega_t(s, \boldsymbol{a})\gamma D^j_k \quad (19)$$

其中 $Y^j_{\tau_k;\tau_k} = D^j_k$。

接下来，我们可以给出一个引理，用于界定相对于序列 W^j_t；τ_k 和 Y^j_t；τ_k 的 Q-函数。

引理 5。 对于每个状态 s、联合动作 $\boldsymbol{a}$和时间 τ_k，都有

$$\begin{aligned} &-Y^j_{t;\tau_k}(s,\boldsymbol{a}) + W^j_{t;\tau_k}(s,\boldsymbol{a}) \\ &\le Q^j_*(s,\boldsymbol{a}) - Q^j_t(s,\boldsymbol{a}) \\ &\le Y^j_{t;\tau_k}(s,\boldsymbol{a}) + W^j_{t;\tau_k}(s,\boldsymbol{a}) \end{aligned} \tag{20}$$

证明。参见 Bertsekas 和 Tsitsiklis [3]中的引理 4.4。 □

由引理 5 可知，差值 r^j_t 的上界取决于 $Y^j_{t;\tau_k}$和 $W^j_{t;\tau_k}$的上界。因此，我们可以分别对 $Y^j_{t;\tau_k}$和 $W^j_{t;\tau_k}$给出上界，而这两个上界结合起来即可得到 r^j_t的一个上界。

我们先给出一个关于序列 $Y^j_{\tau_k;\tau_k}$性质的结果。

引理 6. 序列 $Y^j_{t;\tau_k}$ 是单调递减序列。

证明。根据公式 19，我们可以写出（从两边减去 γD^j_k），

$$\begin{aligned} &(Y^j_{t+1;\tau_k}(s,\boldsymbol{a}) - \gamma D^j_k) \\ &= (1-\alpha^\omega_t(i))Y^j_{t;\tau_k}(s,\boldsymbol{a}) + (1-\alpha^\omega_t(s,\boldsymbol{a}))\gamma D^j_k \\ &= (1-\alpha^\omega_t(i))(Y^j_{t;\tau_k}(s,\boldsymbol{a}) + \gamma D^j_k). \end{aligned} \tag{21}$$

现在，$||Y^j_{t+1;\tau_k}(s,\boldsymbol{a}) - \gamma D^j_k||$的收敛性由 $\lim_{n\to\infty}\Pi^n_{t=k}(1-\alpha^\omega_t(s,\boldsymbol{a})) = 0$可知。由此可见，序列 $(Y^j_{t+1;\tau_k}(s,\boldsymbol{a}) - \gamma D^j_k)$ 单调递减至 0，因此序列 $Y^j_{t;\tau_k}$单调递减至 γD_k。这就证明了我们的结论。

□

接下来，我们给出 $Y^j_{t;\tau_k}$ 的值的上界。

引理 7。 考虑公式4中给出的低-Q 更新，其中学习率为多项式形式，并假设对于任意 $t \ge \tau_k$，都有 $Y^j_{t;\tau_k}(s,\boldsymbol{a}) \le D_k$。那么对于任意 $t \ge \tau_k + L\tau^\omega_k = \tau_{k+1}$，都有 $Y^j_{t;\tau_k}(s,\boldsymbol{a}) \le D^j(\gamma + \frac{2}{e}\beta)$。

证明。对于每个状态-联合动作对 $s,\boldsymbol{a}$，我们可以确定 $n(s,a,\tau_k,\tau_{k+1}) \ge \tau^\omega_k$，因为覆盖时间为 L，且底层策略已执行 $L\tau^\omega_k$ 步（因为我们有关系 $\tau_{k+1} = \tau_k + L\tau^\omega_k$）。

令 $Y^j_{\tau_k,\tau_k}(s,\boldsymbol{a}) = \gamma D^j_k + \rho^j_{\tau_k}$，其中 $\rho_{\tau_k} = (1-\gamma)D^j_k$。现在我们有以下表达式，

$$Y^j_{t+1,\tau_k}(s,\boldsymbol{a}) = (1-\alpha^\omega_t)Y^j_{t;\tau_k}(s,\boldsymbol{a}) + \alpha^\omega_t\gamma D^j_k = \gamma D^j_k + (1-\alpha^\omega_t)\rho^j_t \tag{22}$$

其中 $\rho^j_{t+1} = \rho^j_t(1-\alpha^\omega_t)$。我们旨在证明，在时间 $\tau_{k+1} = \tau_k + \tau^\omega_k$之后，对于任意 $t \ge \tau^\omega_k$和任意 $t \ge \tau_{k+1}$，都有 $\rho^j_t \le \frac{2}{e}\beta D^j_k$。根据定义，我们可以将 ρ^j_t改写为，

$$\begin{aligned} \rho^j_t &= (1-\gamma)D^j_k\Pi^{t-\tau_k}_{l=1}(1-\alpha^\omega_{l+\tau_k}) \\ &= 2\beta D^j_k\Pi^{t-\tau_k}_{l=1}(1-\alpha^\omega_{l+\tau_k}) \\ &= 2\beta D^j_k\Pi^{t-\tau_k}_{l=1}(1-\tfrac{1}{|n(s,\boldsymbol{a},\tau_{k+1},l)|^\omega}) \end{aligned} \tag{23}$$

最后一个等式由 α^ω_t的定义得出 .

由于 τ^ω_k单调递减，

$$\rho^j_t \le 2\beta D^j_k(1-\frac{1}{\tau^\omega_k})^{t-\tau_k}. \tag{24}$$

对于 $t \ge \tau_k + \tau^\omega_k$，我们有，

$$\rho^j_t \le 2\beta D^j_k\Big(1-\frac{1}{\tau^\omega_k}\Big)^{\tau^\omega_k} \le \frac{2}{e}\beta D^j_k. \tag{25}$$

最后一步是根据 $\lim_{x\to\infty}(1-\frac{1}{x})^x = \frac{1}{e}$ 的事实得到的。

Hence, $Y_{t;\tau_k}(s,\boldsymbol{a}) \le (\gamma + \frac{2}{e}\beta)D_k$.

□

根据引理 7，我们得到了项 $Y^j_{t;\tau_k}$ 关于 $t = \tau_{k+1}$ 的一个上界。由于项 $Y^j_{t;\tau_k}$ 单调递减（由引理 6 得出）且为确定性项（由公式 19 和引理 5 得出），该上界将自动对所有 $t \ge \tau_{k+1}$ 成立。

接下来，我们用 $(1-\frac{2}{e})\beta D^j_k$ 给出项$-W^j_{t;\tau_k}$ 的上界。 $W^j_{t;\tau_k}(s,\boldsymbol{a})$ 和 $Y^j_{t;\tau_k}(s,\boldsymbol{a})$ 的上界之和将为 $(\gamma+\beta)D^j_k = (1-\beta)D^j_k = D^j_{k+1}$，正如所需。

现在我们给出序列 $\eta^{k,t,j}_i(s,\boldsymbol{a})$ 和 $W^{l,t}_{;\tau_k}$ 的定义。

定义 3. 令

$$\begin{aligned} W^j_{t;\tau_k}(s,\boldsymbol{a}) &= (1-\alpha^\omega_t(s,\boldsymbol{a}))W^j_{t-1;\tau_k}(s,\boldsymbol{a}) + \alpha^\omega_t(s,\boldsymbol{a})w^j_t(s,\boldsymbol{a}) \\ &= \textstyle\sum_{i=1}^{t-\tau_k}\eta^{k,t,j}_{i+\tau_k}(s,\boldsymbol{a})w^j_{i+\tau_k}(s,\boldsymbol{a}), \end{aligned}$$

其中

$$\eta^{k,t,j}_{i+\tau_k}(s,\boldsymbol{a}) = \alpha^\omega_{i+\tau_k}(s,\boldsymbol{a})\Pi^t_{l=\tau_k+i+1}(1-\alpha^\omega_l(s,\boldsymbol{a})). \tag{26}$$

为了给序列 $W^j_{t;[\tau_{k+1},\tau_{k+2}]}$ τ_k $t \in$ 确定上界，我们考虑区间 。首先，我们给出一个引理，用于界定该区间内的系数，并界定 $w^j_t(s,\boldsymbol{a})$ 在该区间内的影响。

引理 8. 令 $\tilde{w}^{j,t}_{i+\tau_k}(s,\boldsymbol{a}) = \eta^{k,t,j}_{i+\tau_k}(s,\boldsymbol{a})w^j_{i+\tau_k}(s,\boldsymbol{a})$，则对任意$t \in [\tau_{k+1},\tau_{k+2}]$，随机变量 $\tilde{w}^j_{i+\tau_k}(s,\boldsymbol{a})$均值为零，且有界于 $(L/\tau_k)^\omega Q^j{}_{\max}$。

证明。由于 $\eta^{k,t,j}_{i+\tau_k}((s,\boldsymbol{a})) = \alpha^\omega_{i+\tau_k}(s,\boldsymbol{a})\Pi^t_{l=\tau_k+i+1}(1-\alpha^\omega_l(s,\boldsymbol{a}))$，我们可以将 $\eta^{k,t,j}_{i+\tau_k}$分为两部分，第一部分是 $\alpha^\omega_{i+\tau_k}$，第二部分是$\mu = \Pi^t_{l=\tau_k+i+1}(1-\alpha^\omega_l)$。由于 μ 的上界为 1，因此，

$$\begin{aligned} \eta^{k,t,j}_{i+\tau_k}(s,\boldsymbol{a}) &\le \alpha^\omega_{i+\tau_k}(s,\boldsymbol{a}) \\ &= \tfrac{1}{|\#(s,\boldsymbol{a},i+\tau_k)|} \overset{*}{\le} (\tfrac{L}{i+\tau_k})^\omega \le (\tfrac{L}{\tau_k})^\omega. \end{aligned} \tag{27}$$

这里，$*$ 是根据覆盖时间的定义得出的：在长度为 τ的时间区间内，每个状态—联合动作对至少执行 τ/L 次。

因此，我们得到关系 $\eta^{k,\ t\ j}_{i+\tau_k}(s,\boldsymbol{a}) \le (L/\tau_k)^{\omega}$。

现在，考虑 $\tilde{w}_{i+\tau_k}(s,\boldsymbol{a})$ 的期望。根据定义，我们有，$w^{j}_{\tau_k+i}(s,\boldsymbol{a})$ 的均值为零，且对于任意历史和状态-联合动作对，其上界为 $Q^{j}_{\max}$，因此，

$$\begin{aligned}&\mathbb{E}[\tilde{w}_{i+\tau_k}(s,\boldsymbol{a})]\\&=\mathbb{E}[\eta^{k,t,j}_{i+\tau_k}(s,\boldsymbol{a})w_{i+\tau_k}(s,\boldsymbol{a})]\\&=\eta^{k,t,j}_{i+\tau_k}(s,\boldsymbol{a})\,\mathbb{E}[w_{i+\tau_k}(s,\boldsymbol{a})]=0.\end{aligned}\tag{28}$$

接下来，我们还可以证明它有界，

$$\begin{aligned}&|\tilde{w}_{i+\tau_k}(s,\boldsymbol{a})|\\&=|\eta^{k,t,j}_{i}(s,\boldsymbol{a})w_{i+\tau_k}(s,\boldsymbol{a})|\\&\le|\eta^{k,t,j}_{i}(s,\boldsymbol{a})Q^{j}_{\max}|\\&\le(L/\tau_k)^{\omega}Q^{j}_{\max}\end{aligned}\tag{29}$$

□

现在，定义 $W^{l,\ t\ \ j}_{;\tau_k}(s,\boldsymbol{a}) = \sum_{i=1}^{l}\tilde{w}^{t}_{i+\tau_k}(s,\boldsymbol{a})$。目标是证明这是一个具有有界差分的鞅差序列。

引理 9. 对于任意 $t\in[\tau_{k+1},\tau_{k+2}]$ 和 $1\le l\le t$，我们有 $W^{l,t\ \ j}_{;\tau_k}(s,\boldsymbol{a})$ 是一个满足以下条件的鞅序列，

$$|W^{l,j}_{t;\tau_k}(s,\boldsymbol{a})-W^{l-1,j}_{t;\tau_k}(s,\boldsymbol{a})|\le(L/\tau_k)^{\omega}Q^{j}_{\max}\tag{30}$$

证明。首先注意到，项 $W^{l,\ \ j}(s,\ \boldsymbol{a})$ 是一个鞅序列，因为

$$\begin{aligned}&\mathbb{E}[W^{l,j}_{t;\tau_k}(s,\boldsymbol{a})-W^{l-1,j}_{t;\tau_k}(s,\boldsymbol{a})|F_{\tau_k+l-1}]\\&=\mathbb{E}[\tilde{w}^{j,t}_{l+\tau_k}(s,\boldsymbol{a})|F_{\tau_k+l-1}]=0\end{aligned}\tag{31}$$

其中，变量 F_{τ_k+l-1} 表示 W^{l} 先前的所有值。

此外，根据引理 8，我们可以证明 $\tilde{w}^{j}_{l+\tau_k}(s,\ \boldsymbol{a})$ 有上界 $(L/\tau_k)^{\omega}Q^{j}_{\max}$，因此

$$\begin{aligned}&|W^{l,j}_{t;\tau_k}(s,\boldsymbol{a})-W^{l-1,j}_{t;\tau_k}(s,\boldsymbol{a})|\\&=\tilde{w}^{j}_{l+\tau_k}(s,\boldsymbol{a})\le(L/\tau_k)^{\omega}Q^{j}_{\max}\end{aligned}\tag{32}$$

□

下一个引理给出了项 $W^{j}_{t;\ \tau_k}$ 的上界。

引理 10. 考虑公式4中给出的低-Q 更新，采用多项式学习率。以至少 $1-\frac{\delta}{m}$ 的概率，对于任意 $t\in[\tau_{k+1},\tau_{k+2}]$，我们有：对每个状态-联合动作对 $|W^{j}_{t;\tau_k}(s,\boldsymbol{a})|\le(1-\frac{2}{e}\gamma D_k)$，

$$Pr\Big[\forall s,\boldsymbol{a}\forall t\in[\tau_{k+1},\tau_{k+2}]:|W_{t;\tau_k}(s,\boldsymbol{a})|\le(1-\tfrac{2}{e})\beta D_k\Big]\ge1-\frac{\delta}{m}\tag{33}$$

已知

$$\tau_k=\Theta\Big(\big(\frac{L^{1+3\omega}Q^{2,j}_{\max}\ln(Q^{j}_{\max}|S|\Pi_i|A|_i m/(\delta\beta D_k))}{\beta^2D_k^2}\big)^{1/\omega}\Big)\tag{34}$$

证明。对于每个状态-联合动作对，比较 $W^{l,\ t\ \ j}_{;\tau_k}(s,\ \boldsymbol{a})$ 和 $W^{j}_{t;\ \tau_k}(s,\ \boldsymbol{a})$，我们注意到 $W^{j}_{t;\ \tau_k}(s,\ \boldsymbol{a}) = W^{t-\tau_k+1,\ t\ \ \ j}_{;\ \tau_k}(s,\ \boldsymbol{a})$。

令 $l=n(s,\ \boldsymbol{a},\ \tau_k,\ t)$，则对于任意 $t\in[\tau_{k+1},\tau_{k+2}]$，我们有

$$\begin{aligned}&l\le\tau_{k+2}-\tau_k=\tau_{k+1}+L\tau^{\omega}_{k+1}-\tau_k\\&=\tau_k+L\tau^{\omega}_{k}+L(\tau_k+L\tau^{\omega}_{k})-\tau_k\\&=L\tau^{\omega}_{k}+L\tau_k+L^2\tau^{\omega}_{k}\le\Theta(L^{1+\omega}\tau^{\omega}_{k}).\end{aligned}\tag{35}$$

根据引理 9，我们可以对 $W^{t-\tau_k+1,t\ \ \ \ j}_{;\tau_k}(s,\boldsymbol{a})$ 应用 Azuma 不等式，其 $c_i=(L/\tau_k)^{\omega}Q^{j}_{\max}$。因此，我们可以推导出

$$\begin{aligned}&Pr\Big[|W_{t;\tau_k}(s,\boldsymbol{a})|\ge\tilde{\epsilon}|t\in[\tau_{k+1},\tau_{k+2}]\Big]\\&\le2e^{\frac{-\tilde{\epsilon}^2}{2\sum_{i=\tau_k+1}^{t}c_i^2}}\le2e^{\frac{-\tilde{\epsilon}^2}{2lc_i^2}}\\&\le2e^{-c\frac{\tilde{\epsilon}^2\tau_k^{2\omega}}{lL^{2\omega}Q^{2,j}_{\max}}}\\&\le2e^{-c\frac{\tilde{\epsilon}^2\tau^{\omega}_{k}}{L^{1+3\omega}Q^{2,j}_{\max}}}\end{aligned}\tag{36}$$

对于某个常数 $c>0$。我们可以设 $\tilde{\delta}_k=2e^{\frac{-c\tau^{\omega}_{k}\tilde{\epsilon}^2}{L^{1+3\omega}Q^{2\ j}_{\max}}}$，这对 $\tau^{\omega}_{k}=\Theta(\ln(1/\tilde{\delta}_k)L^{1+3\omega}Q^{2,j}_{\max}/\tilde{\epsilon}^2)$ 成立。

使用并集界，我们得到：

$$\begin{aligned}&Pr[\forall s,\forall\boldsymbol{a},\forall t\in[\tau_{k+1},\tau_{k+2}]:W^{j}_{t,\tau_k}(s,\boldsymbol{a})\le\tilde{\epsilon}]\\&\ge1-\textstyle\sum_{t=\tau_{k+1}}^{\tau_{k+2}}Pr[\forall s,\forall\boldsymbol{a},W^{j}_{t;\tau_k}(s,\boldsymbol{a})\ge\tilde{\epsilon}]\\&\ge1-\textstyle\sum_{t=\tau_{k+1}}^{\tau_{k+2}}Pr[\forall s,\forall\boldsymbol{a},|W^{j}_{t;\tau_k}(s,\boldsymbol{a})|\ge\tilde{\epsilon}]\\&\ge1-\textstyle\sum_{t=\tau_{k+1}}^{\tau_{k+2}}\tilde{\delta}_k|S|\Pi_i|A|_i\\&\ge1-(\tau_{k+2}-\tau_{k+1})\tilde{\delta}_k|S|\Pi_i|A|_i\end{aligned}\tag{37}$$

我们希望为每个状态-联合动作对设定一个概率水平 $1-\frac{\delta}{m}$。根据上式，我们得到：

$$\begin{aligned}&1-(\tau_{k+2}-\tau_{k+1})\tilde{\delta}_k|S|\Pi_i|A|_i=1-\delta/m\\&\implies(\tau_{k+2}-\tau_{k+1})\tilde{\delta}_k|S|\Pi_i|A|_i=\delta/m.\\&\implies\tilde{\delta}_k=\delta/(\tau_{k+2}-\tau_{k+1})m|S|\Pi_i|A|_i.\end{aligned}\tag{38}$$

因此，取 $\tilde{\delta}_k=\frac{\delta}{m(\tau_{k+2}-\tau_{k+1})|S||A|}$ 可确保每个状态—联合动作对都满足 $1-\frac{\delta}{m}$。由此我们得到，

$$\begin{aligned}\tau^{\omega}_{k}&=\Theta\Big(\frac{L^{1+3\omega}Q^{2,j}_{\max}\ln(|S|\Pi_i|A|_i m\tau^{\omega}_{k}/\delta)}{\tilde{\epsilon}^2}\Big)\\&=\Theta\Big(\frac{L^{1+3\omega}Q^{2,j}_{\max}\ln(|S|\Pi_i|A|_i mQ^{j}_{\max}/(\delta\tilde{\epsilon}))}{\tilde{\epsilon}^2}\Big)\end{aligned}\tag{39}$$

令 $\tilde{\epsilon} = (1-2/e)\beta D_k$，即可得到所需的上界。

□

既然我们已经界定了每次迭代达到所需概率 $1-\frac{\delta}{m}$ 所需的时间，接下来的引理将给出所有迭代中误差的上界。

引理 11. 考虑公式4中给出的低-Q 更新，其学习率为多项式形式。以概率 $1-\delta$，对于每次迭代 $k \in [1,m]$ 和时间 $t \in [\tau_{k+1}, \tau_{k+2}]$，我们有 $|W_t^j{}_{;\tau_k}(s,\boldsymbol{a})| \leq (1-\frac{2}{e})\beta D_k^j$，即

$$Pr\Big[\forall k \in [1,m], \forall t \in [\tau_{k+1}, \tau_{k+2}], \forall s, \boldsymbol{a} : |W_{t;\tau_k}^j(s,\boldsymbol{a})| \leq (1-\frac{2}{e})\beta D_k^j\Big] \geq 1-\delta \tag{40}$$

已知

$$\tau_0 = \Theta\Big((\frac{L^{1+3\omega} Q_{\max}^{2,j} \ln(Q_{\max}^j |S|\Pi_i|A|_i m/(\delta\beta\epsilon))}{\beta^2\epsilon^2})^{1/\omega}\Big) \tag{41}$$

证明。从引理 10 可得

$$Pr\Big[\forall t \in [\tau_{k+1}, \tau_{k+2}] : |W_{t;\tau_k}^j| \geq (1-\frac{2}{e})\beta D_k^j\Big] \leq \frac{\delta}{m} \tag{42}$$

利用并集界可得

$$\begin{aligned} &Pr[\forall k \leq m, \forall t \in [\tau_{k+1}, \tau_{k+2}] | W_{t;\tau_k}^j| \geq \tilde{\epsilon}] \\ &\leq \textstyle\sum_{k=1}^m Pr[\forall t \in [\tau_{k+1}, \tau_{k+2}] | W_{t;\tau_k}^j| \geq \tilde{\epsilon}] \leq \delta \end{aligned} \tag{43}$$

其中 $\tilde{\epsilon} = (1-\frac{2}{e})\beta D_k$。

□

下一个引理求解该递推关系 $\sum_{i=0}^{m-1} L\tau_i^\omega + \tau_0$ 并推导时间复杂度。

引理 12. 令

$$a_{k+1} = a_k + La_k^\omega = a_0 + \sum_{i=0}^{k} La_i^\omega \tag{44}$$

那么，对于任意常数 $\omega \in (0,1)$，$a_k = \Omega(a_0^{1-\omega}/L + L^{\frac{1}{1-\omega}}((k+1)/2)^{\frac{1}{1-\omega}})$。

证明。定义如下级数

$$b_{k+1} = \textstyle\sum_{i=0}^k Lb_i^\omega + b_0 \tag{45}$$

其初始条件为 $b_0 = L^{\frac{1}{1-\omega}}$。

现在，我们将 b_k 的下界取为 $(L(k+1)/2)^{\frac{1}{1-\omega}}$。我们使用归纳法证明这一假设。对于 $k=0$，

$$b_0 = L^{\frac{1}{1-\omega}} \geq (\frac{L}{2})^{\frac{1}{1-\omega}} \tag{46}$$

假设归纳假设对 $k-1$ 成立，并证明其对 k 成立，

$$\begin{aligned} b_k &= b_{k-1} + Lb_{k-1}^\omega \\ &= (Lk/2)^{\frac{1}{1-\omega}} + L(Lk/2)^{\frac{\omega}{1-\omega}} \\ &= L^{\frac{1}{1-\omega}}((k/2)^{\frac{1}{1-\omega}} + (k/2)^{\frac{\omega}{1-\omega}}) \\ &\geq L^{\frac{1}{1-\omega}}((k+1)/2)^{\frac{1}{1-\omega}}. \end{aligned} \tag{47}$$

对于 $a_0 \geq L^{\frac{1}{1-\omega}}$，我们可以将级数视为从 $b_k = a_0$ 开始。由于第一个时间步为 1，可以看出起点移动了 $\Theta(a_0^{1-\omega}/L)$。因此，总复杂度为 $\Omega(a_0^{1-\omega}/L + L^{\frac{1}{1-\omega}}((k+1)/2)^{\frac{1}{1-\omega}})$。

□

定理 2。我们说明，对于智能体 j，以至少 $1-\delta$ 的概率，$||Q_T^j - Q_*^j||_\infty \leq \epsilon$。低-$Q$、$Q_T^j$ 的收敛速率上界，在学习率因子为 ω 的多项式学习率下，由下式给出（其中 Q_*^j 是 j 的纳什 Q-值）

$$\begin{aligned} T = \Omega\Big(\Big(\frac{L^{1+3\omega} Q_{\max}^{2,j} \ln(\frac{|S|\Pi_i|A_i|Q_{\max}^j}{\delta\beta\epsilon})}{\beta^2\epsilon^2}\Big)^{1-\omega}/L \\ + \Big((\frac{L}{\beta}\ln\frac{Q_{\max}^j}{\epsilon}+1)/2\Big)^{\frac{1}{1-\omega}}\Big). \end{aligned}$$

证明。定理 2 的证明由引理 7、11、4 和 12 推出。

具体而言，将引理 11 中的 a_0 值（τ_0 的值）以及引理 4 中的 k 值（m 的下界）代入引理 12 的关系式。由引理 7 和引理 11 可知，引理 5 所要求的条件得到满足，从而可以给出下界。

□

D 定理 3 的证明

在本节中，我们旨在证明，对于某个正常数 $\psi \leq 0.712$，第 k 次迭代的规模为 $L(1+\psi)\tau_k$。覆盖时间性质保证，在 $(1+\psi)L\tau_k$ 步内，每一对状态—联合动作至少执行 $(1+\psi)\tau_k$ 次。在这种情况下，时间序列为 $\tau_{k+1} = \tau_k + (1+\psi)L\tau_k$。与上一节一样，我们将先对 $Y_t^j{}_{;\tau_k}$ 给出上界，再对 $W_t^j{}_{;\tau_k}$ 给出上界。与附录 B 一样，本节中我们也用 Q 表示低-Q 值。本定理的证明遵循 Dar 和 Mansour [5] 中的定理 5。Dar 和 Mansour 的工作仅限于单智能体马尔可夫决策过程，而我们的结果将 Dar 和 Mansour 的分析拓展到了一般和随机博弈的情形。

引理 13. 考虑公式4中给出的低-Q 更新，其学习率为线性学习率。假设对于 $t \geq \tau_k$，我们有 $Y_t^j{}_{;\tau_k}(s,\boldsymbol{a}) \leq D_k^j$。那么，对于任意 $t \geq \tau_k + (1+\psi)L\tau_k = \tau_{k+1}$，我们有 $Y_t^j{}_{;\tau_k}(s,\boldsymbol{a}) \leq (\gamma + \frac{2}{2+\psi}\beta)D_k^j$。

证明。对于每个状态-联合动作对，我们可以确定 $n(s,\boldsymbol{a},\tau_k,\tau_{k+1}) \geq (1+\psi)\tau_k$，因为根据覆盖时间的定义，在一个包含 $(1+\psi)L\tau_k$ 步的区间内，每个状态-联合动作对至少被访问 $(1+\psi)\tau_k$ 次。

令 $Y_{\tau_k,\tau_k}^j(s,\boldsymbol{a}) = \gamma D_k^j + \rho_{\tau_k}^j$，其中 $\rho_{\tau_k}^j = (1-\gamma)D_k^j$。现在有，

$$\begin{aligned} \rho_t^j &= (1-\gamma)\Pi_{l=1}^{t-\tau_k}(1-\alpha_{l+\tau_k}) \\ &= 2\beta D_k^j \Pi_{l=1}^{t-\tau_k}(1-\alpha_{l+\tau_k}) \\ &= 2\beta D_k^j \Pi_{l=1}^{t-\tau_k}(1-\frac{1}{|n(s,\boldsymbol{a},\tau_{k+1},l)|}), \end{aligned} \tag{48}$$

其中，最后一个等式是由以下事实得出的：$\alpha_t = {}_{n\ s}{}^{1}{}_{(,\boldsymbol{a},0,t)}$。由于 τ_k 单调递减，利用 $t = \tau_k + (1+\psi)L\tau_k$，可得（使用 $\psi < 0.712$），

$$\begin{aligned}
\rho_t &\le 2D_k^j\beta(1-\tfrac{1}{(1+\psi)\tau_k})^{t-\tau_k} \\
&\le 2D_k^j\beta(1-\tfrac{1}{(1+\psi)\tau_k})^{1+\psi L\tau_k} \\
&\le 2D_k^j\beta(1-\tfrac{1}{(1+\psi)\tau_k})^{1+\psi\tau_k} \\
&\le \tfrac{2D_k^j\beta}{e} \\
&\le \tfrac{2D_k^j\beta}{2+\psi}
\end{aligned} \tag{49}$$

因此，$Y_t^{j};\ {}_{\tau_k}(s,\ \boldsymbol{a}) \le (\gamma + \frac{2}{2+\psi}\beta)D_k^j$。

□

以下引理使我们能够使用 Azuma 不等式。

引理14。对于任意 $t \ge \tau_k$ 和 $1 \le l \le t$，我们有 $W^{l,\ t}{}^{j};\ {}_{\tau_k}(s,\boldsymbol{a})$ 是一个鞅序列，它满足：

$$|W_{t;\tau_k}^{l,j}(s,\boldsymbol{a}) - W_{t;\tau_k}^{l-1,j}(s,\boldsymbol{a})| \le \frac{Q_{\max}^j}{n(s,\boldsymbol{a},0,t)} \tag{50}$$

证明。$W^{l,\ t}{}^{j};\ {}_{\tau_k}(s,\ \boldsymbol{a})$ 是一个鞅差序列，因为：

$$\begin{aligned}
&\mathbb{E}[W_{t;\tau_k}^{l,j}(s,\boldsymbol{a}) - W_{t;\tau_k}^{l-1,j}(s,\boldsymbol{a})|F_{\tau_k+l-1}] \\
&= \mathbb{E}[\eta_{\tau_k+l}^{k,t,j}(s,\boldsymbol{a})w_{\tau_k+l}^j(s,\boldsymbol{a})|F_{\tau_k+l-1}] \\
&= \eta_{\tau_k+l}^{k,t,j}(s,\boldsymbol{a})\,\mathbb{E}[w_{\tau_k+l}^j(s,\boldsymbol{a})|F_{\tau_k+l-1}] = 0
\end{aligned} \tag{51}$$

对于线性学习率，我们有 $\eta^{k,\ \ t\,j,}_{\tau_k+l}(s,\ \boldsymbol{a}) \le \alpha_{l+\tau_k} = 1/n(s,\ \boldsymbol{a},\ 0,\ t)$，因此

$$\begin{aligned}
&|W_{t;\tau_k}^{l,j}(s,\boldsymbol{a}) - W_{t;\tau_k}^{l-1,j}(s,\boldsymbol{a})| \\
&= \eta_{\tau_k+l}^{k,t,j}(s,\boldsymbol{a})|w_{\tau_k+l}^j(s,\boldsymbol{a})| \\
&\le \tfrac{Q_{\max}^j}{n(s,\boldsymbol{a},0,t)}.
\end{aligned} \tag{52}$$

□

以下引理给出了随机项 $W_t^{j};\ {}_{\tau_k}$ 的上界。

引理 15. 考虑公式4中给出的低-Q 更新，采用线性学习率。以至少 $1-\frac{\delta}{m}$ 的概率，对于每个状态-联合动作对 $|W_t^{j};{}_{\tau_k}(s,\boldsymbol{a})| \le \frac{\psi}{2+\psi}\beta D_k^j$，对任意 $t > \tau_{k+1}$ 和任意正常数 $\psi \le 0.712$，都有：

$$Pr\left[\forall t \in [\tau_{k+1},\tau_{k+2}] : W_{t;\tau_k}^j(s,\boldsymbol{a}) \le \tfrac{\psi}{2+\psi}\beta D_k^j\right] \ge 1-\tfrac{\delta}{m} \tag{53}$$

鉴于 $\tau_k \ge \Theta(\frac{Q^{2}{}_{,\max}{}^{j}\ln(Q_{\max}^j|S|\Pi_i|A|_i m)/(\psi\delta\beta D_k)}{\psi^2\beta^2 D_k^2})$。

证明。根据引理14，我们可以对该项应用 Azuma 不等式，$W^{t-\tau_k+1},{}_{t}{}^{j}{}_{;\tau_k}$（注意 $W^{t-\tau_k+1},{}_{t}{}^{j}{}_{;\tau_k} = W_{t;\tau_k}$），并对任意 $t \ge \tau_{k+1}$ 使用表达式 $c_i = \Theta\left(\frac{Q_{\max}^j}{n\ s\ a\ 0(,,,t)}\right)$。因此，我们推导得出，

$$Pr[|W_{t;\tau_k}^j \ge \tilde{\epsilon}|] \le 2e^{\frac{-2e\tilde{\epsilon}^2}{\sum_{i=\tau_k}^t c_i^2}} \le 2e^{-c\frac{\tilde{\epsilon}^2 n(s,\boldsymbol{a},\tau_k,t)}{Q_{\max}^{2,j}}} \tag{54}$$

其中 c 为某个正常数 .

定义：

$$\begin{aligned}
&\zeta_t(s,\boldsymbol{a}) = 1,\ \text{if } \alpha_t(s,\boldsymbol{a}) \ne 0 \\
&\zeta_t(s,\boldsymbol{a}) = 0,\ \text{otherwise}.
\end{aligned} \tag{55}$$

利用并集界以及在长度为 $(1+\psi)L\tau_k$ 的时间区间内，每个状态-联合动作对至少被访问 $(1+\psi)\tau_k$ 次这一性质，可得

$$\begin{aligned}
&Pr\left[\forall t \in [\tau_{k+1},\tau_{k+2}] : |W_{t;\tau_k}^j(s,\boldsymbol{a})| \ge \tilde{\epsilon}\right] \\
&\le Pr[\forall t \ge ((1+\psi)L+1)\tau_k : |W_{t;\tau_k}^j(s,\boldsymbol{a})| \ge \tilde{\epsilon}] \\
&\le \textstyle\sum_{t=((1+\psi)L+1)\tau_k}^{\infty} Pr\left[|W_{t;\tau_k}^j(s,\boldsymbol{a})| \ge \tilde{\epsilon}\right] \\
&\le \textstyle\sum_{t=((1+\psi)L+1)\tau_k}^{\infty} \zeta_t(s,\boldsymbol{a}) 2e^{-c\frac{\tilde{\epsilon}^2 n(s,\boldsymbol{a},0,t)}{Q_{\max}^{2,j}}} \\
&\le 2e^{-c\frac{\tilde{\epsilon}^2((1+\psi)\tau_k)}{Q_{\max}^{2,j}}} \textstyle\sum_{t=0}^{\infty} e^{-\frac{t\tilde{\epsilon}^2}{Q_{\max}^{2,j}}} \\
&= \frac{2e^{-c\frac{\tilde{\epsilon}^2((1+\psi)\tau_k)}{Q_{\max}^{2,j}}}}{1-e^{\frac{-\tilde{\epsilon}^2}{Q_{\max}^{2,j}}}} \\
&= \Theta(\tfrac{Q_{\max}^{2,j}e^{\frac{-c'\tau_k\tilde{\epsilon}^2}{Q_{\max}^{2,j}}}}{\tilde{\epsilon}^2})
\end{aligned} \tag{56}$$

其中 c，为某个正常数。令 $\underline{\delta_{m|S|\Pi_i|A|_i} = \Theta\left(\frac{e^{-c}}{{}'Q^{2,j}_{\max}}\right)}$，该条件在 $\tau_k = \Theta(\frac{Q^{2}{}_{,\max}{}^{j}\ln(Q_{\max}^j|S|\Pi_i|A|_i m/(\delta\tilde{\epsilon}))}{\tilde{\epsilon}^2})$ 时成立，并且该表达式 $\tilde{\epsilon} = \frac{\psi}{2+\psi}\beta D_k^j$ 保证了对于每个 $t \ge \tau_{k+1}$（从而对于任意 $t \in [\tau_{k+1},\tau_{k+2}]$），以至少 $1-\frac{\delta}{m}$ 的概率，该结论对每个状态-联合动作对都成立。

□

我们已对每次迭代达到所需的 $1-\frac{\delta}{m}$ 概率所需的时间给出了上界。以下引理给出了所有迭代中误差的上界。

引理 16. 考虑公式4中给出的低-Q 更新，采用线性学习率。以概率 $1-\delta$，对于每个迭代 $k \in [1,m]$、时间 $t \in [\tau_{k+1},\tau_{k+2}]$ 以及任意正常数 $\psi \le 0.712$，我们有 $|W_t^{j};{}_{\tau_k}| \le \frac{\psi\beta D_k^j}{2+\psi}$，即

$$Pr\left[\forall k \in [1,m], \forall t \in [\tau_{k+1},\tau_{k+2}] : |W_{t;\tau_k}^j| \le \tfrac{\psi\beta D_k^j}{2+\psi}\right] \ge 1-\delta \tag{57}$$

given that $\tau_0 = \Theta\left(\frac{Q_{\max}^{2,j}\ln(Q_{\max}^j|S||A|m/(\delta\beta\epsilon\psi))}{\psi^2\beta^2\epsilon^2}\right)$

证明. 根据引理 15，我们知道

$$Pr\left[\forall t \in [\tau_{k+1}, \tau_{k+2}] : |W^j_{t;\tau_k}| \geq \frac{\psi\beta D^j_k}{2+\psi}\right] \leq \frac{\delta}{m}. \tag{58}$$

根据并集界，我们有：

$$\begin{aligned} & Pr\left[\forall k \leq m, \forall t \in [\tau_{k+1}, \tau_{k+2}] : |W^j_{t;\tau_k}| \geq \frac{\psi\beta D^j_k}{2+\psi}\right] \\ & \leq \sum_{k=1}^{m} Pr\left[\forall t \in [\tau_{k+1}, \tau_{k+2}] | W^j_{t;\tau_k}| \geq \frac{\psi\beta D^j_k}{2+\psi}\right] \\ & \leq \delta. \end{aligned} \tag{59}$$

□

定理 3. 设以至少 $1-\delta$ 的概率，对于一个智能体 j, $||Q^j_T - Q^j_*||_\infty \leq \epsilon$。采用线性学习率时，低-$Q$, Q^j_T 的收敛速率上界为

$$T = \Omega\left((L+\psi L+1)^{\frac{1}{\beta}} \ln \frac{Q^j_{\max}}{\epsilon} \frac{Q^{2,j}_{\max} \ln(\frac{|S||\Pi_i||A|_i Q^j_{\max}}{\delta\beta\epsilon\psi})}{\beta^2\epsilon^2\psi^2}\right),$$

其中 ψ 是满足 $\psi \leq 0.712$ 的任意小正常数

证明。定理 3 可由引理 16、引理 13、引理 4，以及以下事实推出：$a_{k+1} = a_k + (1+\psi)La_k = a_0((1+\psi)L+1)^k$。

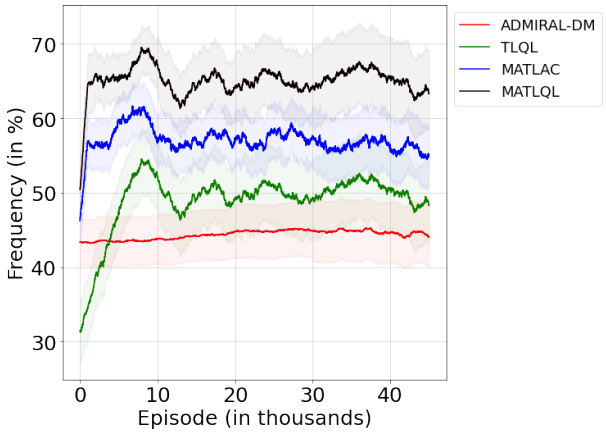
具体而言，将引理 4 中的 k （ m 的下界）以及引理 16 中的 a_0 （ τ_0 的值）代入，可知引理 13 和引理 16 满足引理 5 中下界的条件。

□

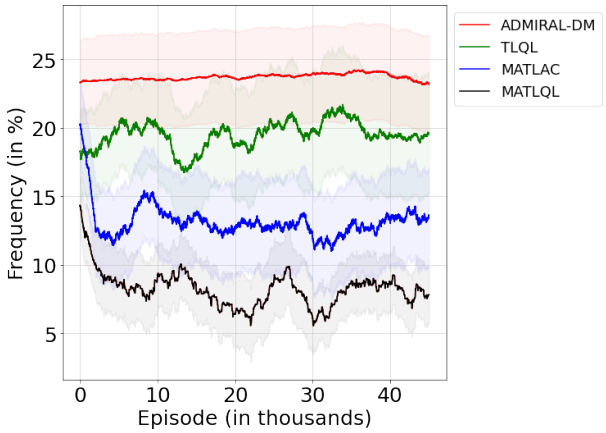
倾听顾问的频率

本节绘制了我们部分实验中听取各位顾问的频率。我们希望表明，与其他相关基线方法相比，MA-TLQL 会更多地听取优质顾问的意见，并更倾向于避开劣质顾问。在这些实验中，我们将 TLQL [21] 和 ADMIRAL-DM [47]算法作为比较对象。由于 CHAT 的实现与 ADMIRAL-DM 使用相同的顾问选择方法（加权随机策略方法），因此我们将在这些结果中省略 CHAT（其性能与 ADMIRAL-DM 相近）。DQfD 使用预训练，并非以在线方式选择顾问，因此我们也将在这些实验中省略 DQfD。

我们考虑第6节中的第一个实验（实验 1），其中有一组质量各异的四位充分顾问。第一位顾问（顾问 1）的质量优于其他顾问，因此智能体必须更多地听取这位顾问的建议。另一方面，顾问 4 只会提出随机动作，因此智能体应该避免听取这位顾问的建议。在图 11(a) 中，我们绘制了一条曲线，表示在算法有机会听取某位可用顾问建议的所有时间步中，算法听取顾问 1 建议的时间步所占的百分比。从图中可以看出，从训练开始到结束，与其他基线方法相比，MA-TLQL 都会更多地听取这位顾问（顾问 1）的建议。由于 MA-TLQL 使用集成技术来选择顾问，因此在训练早期阶段具有明显优势。此外，由于 MA-TLQL 会独立于强化学习策略对顾问进行显式评估，因此它能够



（a）优质顾问（顾问 1）

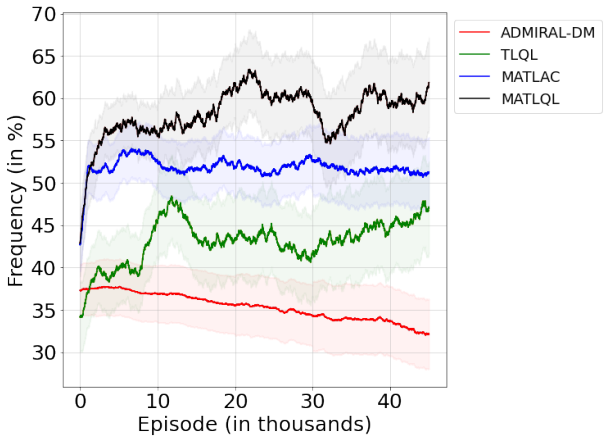


（b）劣质顾问（顾问 4）

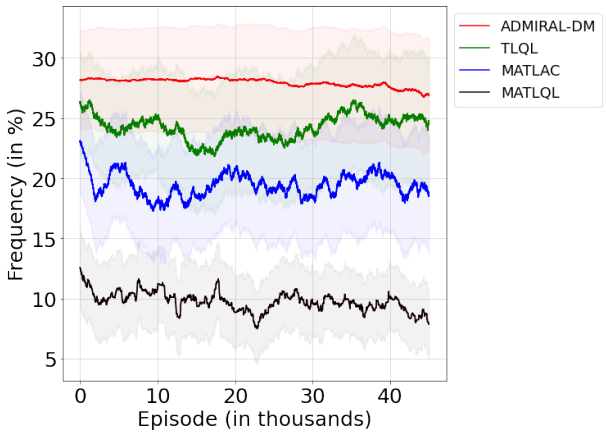
图 11：在包含四位不同质量的充分顾问的双智能体 Pommerman 实验中听取顾问的频率（实验 1）

与其他基线方法相比，更多地听取正确顾问的建议。与其他基线方法（TLQL、ADMIRAL-DM）相比，MA-TLAC 也会更多地听取优质顾问的建议。由于 TLQL 将顾问评估与强化学习策略耦合在一起，因此与 MA-TLQL 相比，它听取优质顾问建议的次数要少得多。此外，TLQL 将强化学习策略视为高层表格的一部分，因此对顾问的依赖程度较低。当有优质顾问可用时，这可能会成为问题。在图 11(b) 中，我们绘制了各算法听取劣质顾问（顾问 4）建议的百分比。可以看到，与所有其他算法相比，MA-TLQL 对这位顾问的依赖程度最低。这进一步印证了我们的观察：在这种多智能体环境中，MA-TLQL 最有可能选择听取正确顾问的建议。

我们绘制了第6节实验 3 中对顾问 1 和顾问 4 的倾听比例。该实验的顾问数量不足四位，且他们的质量依次递减。图 12(a) 和 (b) 的结果表明，MA-TLQL 更倾听优质顾问，较少倾听劣质顾问，这与我们在实验 1 中的观察一致。

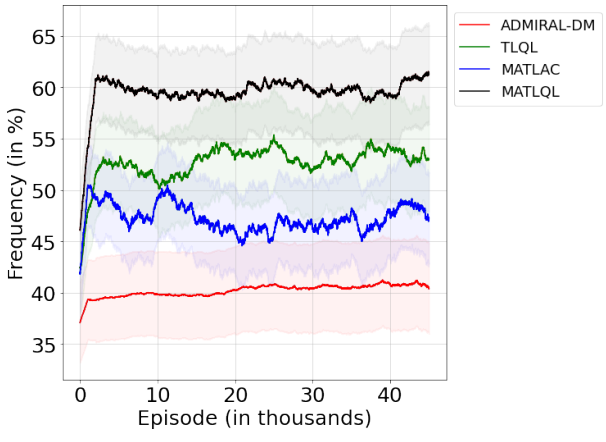


(a) 优质顾问（顾问 1）

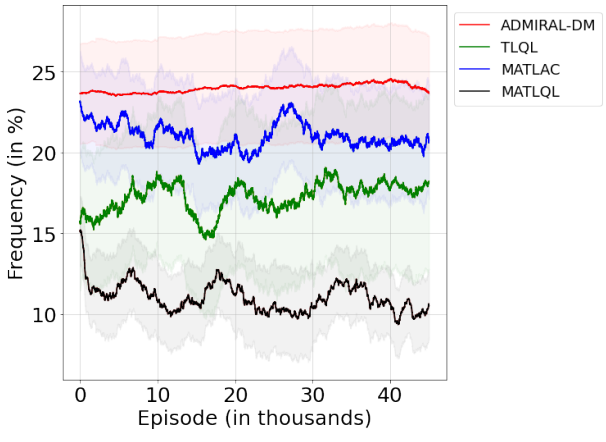


(b) 劣质顾问（顾问 4）

图 12：在由四位质量各异且信息不足的顾问参与的双智能体 Pommerman 实验中，听取顾问的频率（实验 3）



(a) 优质顾问（顾问 1）



(b) 劣质顾问（顾问 4）

图 13：Pommerman 团队实验中听取顾问的频率，实验使用了四位质量各异的充分顾问（实验5）

同样，我们在图 13(a) 和 (b) 中绘制了实验5所用 Pommerman 团队环境（混合设置）中倾听优质顾问和劣质顾问的比例。这里展示的是同一队伍中两名 Pommerman 智能体之一的结果（另一名智能体的结果相似）。我们再次注意到，与其他算法相比，MA-TLQL 更倾听正确的顾问，也更善于避开劣质顾问。

F 所考虑算法的性质

在表1中，我们汇总了本文所考察的所有算法。算法之间的差异源于算法的性质（独立或多智能体）、其自然支持从冲突示范（即来自多个顾问的示范）中学习的能力，以及其自然支持的示范类型（离线或在线）。离线演示是指在“训练阶段”开始之前从顾问处收集并存入记忆缓冲区的示范；在该阶段，算法通过与环境交互进行训练。通常，这些示范会像若干先前工作中所做的那样，用于在常规训练之前的“预训练”阶段训练算法 [7, 12, 17]。另一种方式是在线示范，即在训练阶段实时获取（而非预先收集）。在这种情况下，智能体可以针对当前游戏情境，直接主动向顾问获取动作建议。

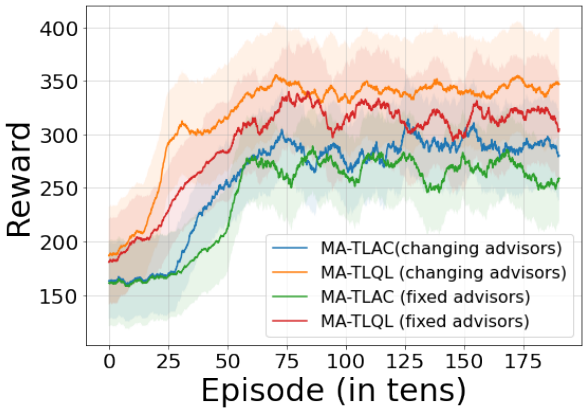
总体而言，在本文考察的所有实验中，我们发现，考虑其他智能体的动作以形成最佳响应的算法（非独立算法），表现优于将所有其他智能体视为环境一部分的独立算法。这种现象的一个原因可能是，独立算法破坏了强化学习方法中的马尔可夫假设 [41]。此外，我们发现，从在线建议中学习的算法，表现优于从离线建议中学习的算法。如果算法

算法	《自然》的性质	顾问数量 顾问们	Type of 示范
DQN [26]	独立	不通过建议进行学习 来自建议	不适用
DQfD [12]	独立	One	离线
CHAT [47]	独立	One	在线
ADMIRAL-DM [38]	多智能体	One	在线
TLQL [21]	独立	多个	在线
MA-TLQL（我们的方法）	多智能体	多个	在线
MA-TLAC（我们的方法）	多智能体	多于一个	在线

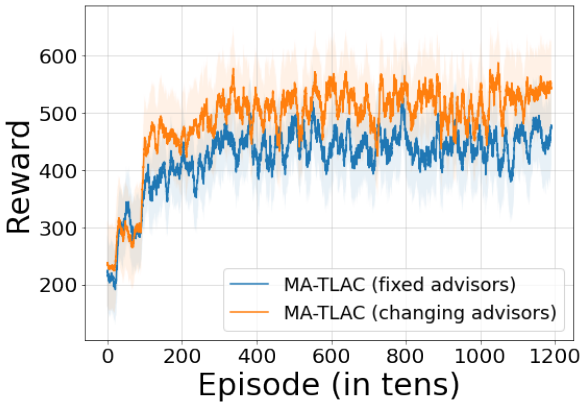
表1：本文所考虑的所有算法的描述

从在线建议中学习，就能够实时利用顾问的知识，应对其他智能体动态变化的情况。由于在多智能体环境中，训练前通常无法获得其他智能体/对手，因此，与在单智能体训练中相当成功的离线演示不同，离线演示在多智能体训练中并不十分有效[12]。本文考察了建议来自多个独立知识源的环境。在许多状态下，不同顾问会给出相互冲突的建议。因此，与无法自然支持在多个相互冲突的顾问指导下学习的算法相比，能够有效化解不同顾问所提供信息之间冲突的算法更为成功。如表1所示，具备全部三项理想属性（即支持多智能体更新、能够从相互冲突的顾问那里学习、支持离线演示）的仅有两种算法：MA-TLQL 和 MA-TLAC。在第6节考察的七项实验中，它们在大多数实验中都取得了最佳表现。

在本文中，我们没有将 HMAT [18] 和 LeCTR [28]视为基线方法，尽管它们也可以归类为动作建议方法。由于两个重要原因，我们认为这些算法不适合作为基准。首先，它们都仅适用于双智能体合作设置，而我们关注的是更一般的领域，包括智能体数量超过两个的领域，以及竞争性和混合动机环境。其次，我们关注的是从一组外部建议者那里进行独立学习，而 HMAT 和 LeCTR 都侧重点对点学习（部分原因是它们关注教学团队，而我们的工作则针对更一般的多智能体学习问题）。



（a）追逐环境



（b）捕食者-猎物环境

图14：在学习中的顾问发生变化以及顾问固定的情况下，MA-TLQL 和 MA-TLAC 的性能。这里，（a）对应实验 6 中描述的设置，（b）对应实验7 中描述的设置，详见第6节。

G MA-TLQLWITH 学习型建议者

本文所有实验中，我们都采用了固定且不变化的顾问，具体说明见第 4 节。如果顾问是固定的，那么可以通过 MA-TLQL 中的高-Q 更新确定其 Q值。顾问的固定特性使我们能够使用与定理1类似的论证，为高-Q 值提供收敛理论保证（假设在极限情况下进行无限次更新）。不过，在实验中，我们仍可以考虑其他类型的顾问：它们在训练阶段主动学习，因此会主动更新策略（但使用这类顾问并不具有任何收敛理论保证）。本节将重新考察第6节中的实验 6 和实验7，并研究 MA-TLQL 和 MA-TLAC 在学习型建议者下的表现。为了检验统计显著性，我们采用非配对双侧 t 检验并报告 p值，与此前的实验一致。

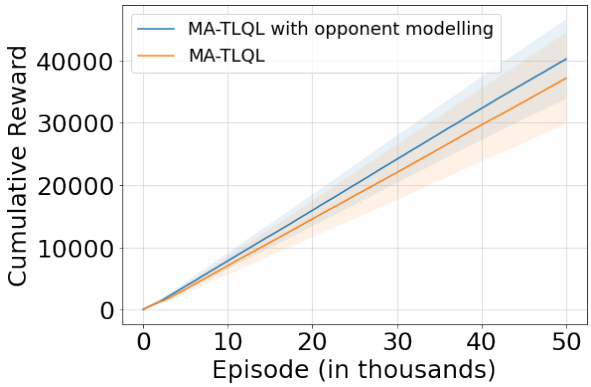
首先，我们考察实验 6，其中使用了 Pursuit 合作环境，并将四个 DQN 预训练网络作为顾问。现在，我们将考虑同一组四个顾问，并称其为“固定建议者”。此外，我们会让同样的四个 DQN 预训练网络在被主动用于动作建议的同时继续训练。我们将这组顾问称为“变化中的顾问”。图14(a)绘出了 MA-TLQL 和 MA-TLAC 与固定及变化中的顾问组配合时的表现（回合奖励图）。使用变化中的顾问组时，我们发现 MA-TLQL 和 MA-TLAC 的表现优于使用固定建议者时的对应方法（$p < 0.04$）。由于顾问在训练阶段主动学习并调整策略，因此它们能够在环境的不同阶段向 MA-TLQL 和 MA-TLAC 提供更好的动作建议。这也体现为它们更优异的表现。

同样，我们重新考察了采用捕食者-猎物环境的实验7，并考虑两组不同的顾问。在实验7中，我们使用了四个预训练的 DQN 网络作为顾问。我们将重复使用同一组四个顾问，并将其称为“固定建议者”。持续训练的对应版本则称为“变化中的顾问”。我们在图14(b)中绘出了 MA-TLAC 在捕食者-猎物实验中的表现，再次发现，使用变化中的顾问组的 MA-TLAC 优于使用固定建议者组的 MA-TLAC（$p < 0.03$）。

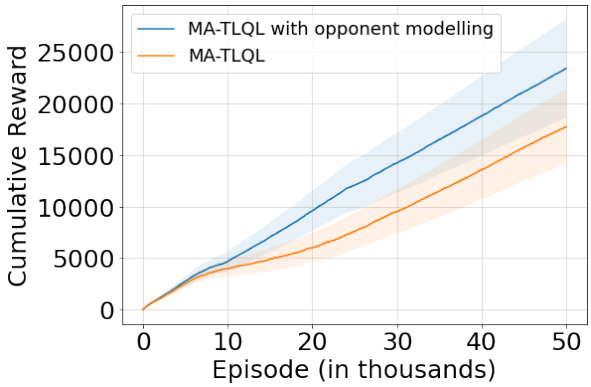
因此，从实验结果来看，MA-TLQL 和 MA-TLAC 仍可与能够取得良好实证表现的变化中的顾问配合使用。然而，这些顾问的非平稳性使我们无法提供收敛的理论保证。这类似于在多智能体环境中使用独立算法：这些算法既没有收敛到局部最优解的理论保证，也没有收敛到全局最优解的理论保证；但从实证来看，先前的多项工作指出，这些算法在各种多智能体环境中表现良好 [11, 25]。

H 使用对手建模的 MA-TLQL

本文所有实验中的 MA-TLQL 均采用多智能体更新方式，仅跟踪对手动作（通过考虑上一个动作），并未进行任何主动对手建模。本文的核心重点是向顾问学习，因此我们没有采用任何特定的对手建模技术，以保持算法简单。不过，在本节中，我们将结合一种对手建模技术实现 MA-TLQL：该技术使用一个独立的神经网络（包含 2 个各有 50 个神经元的 ReLU 层和一个输出层）来预测对手的动作。该网络以状态和对手上一个动作为输入，并预测对手的动作。智能体利用预测得到的对手动作计算最佳响应。最后，我们使用实际观察到的对手动作定义交叉熵损失函数，以训练该网络。我们将在第6节重新讨论实验 3 和实验 4；在这两项实验中，我们使用了四个质量各不相同且相似的不足的顾问。现在，我们将采用相同的实验流程（使用相同的顾问集），考察采用和不采用主动对手建模时 MA-TLQL 的表现。为检验统计显著性，我们使用非配对双侧 t 检验，并报告 p-值。



(a) 一对一 Pommerman：质量不同的顾问不足



(b) 一对一 Pommerman：质量相似的顾问不足

图 15：使用四个不足的顾问时，采用和未采用对手建模的 MA-TLQL 的性能。此处，(a) 对应实验 3 中的设置，(b) 对应实验 4 中的设置，详见第6节。

我们在图 15 中绘制了性能曲线。从图 15(a) 和 (b) 可以看出，采用对手建模的 MA-TLQL 优于不进行任何主动对手建模的 MA-TLQL算法。在图 15(a) 中，我们看到对手建模仅带来了小幅性能提升（$p < 0.2$）。原因之一是，在该实验中，不采用对手建模的 MA-TLQL 已经能够通过向顾问高效学习，展现出很高的性能。另一方面，图 15(b) 显示，MA-TLQL 在采用对手建模时，性能有了显著提升（$p < 0.04$）。这里，不采用对手建模的 MA-TLQL 性能不如图 15(a) 中的表现，而对手建模则显著改善了性能。

I MA-TLQL 与不同数量的顾问

本节中，我们在第 6 节实验 6 所采用的追捕环境中，使用不同数量的顾问训练 MA-TLQL。

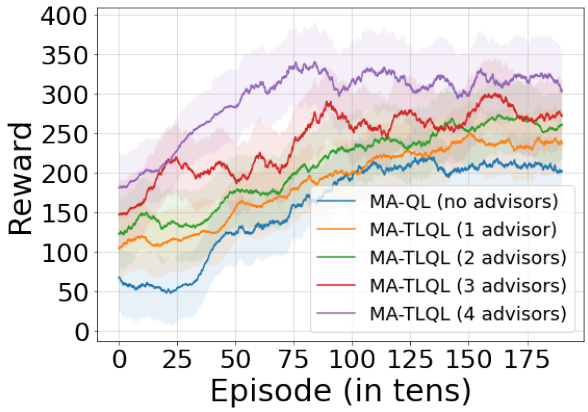


图 16：追捕环境中采用不同数量顾问的 MA-TLQL（实验 6 使用的设置，详见第6节）

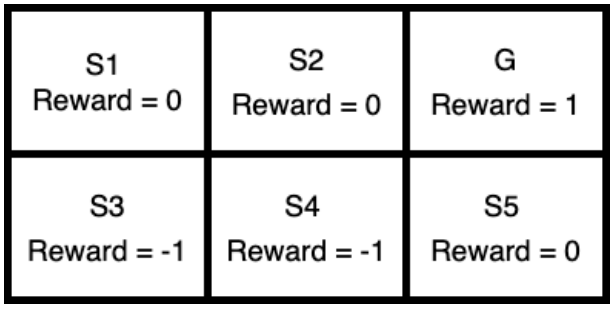


图 17：用于比较 TLQL 和 MA-TLQL 更新的玩具环境

回顾一下，在实验 6 中，我们将四个经过预训练的 DQN 网络作为顾问（分别经过 500、1000、1500 和 2000 回合的预训练）。为这些顾问编号后，我们将经过 500 回合预训练的顾问称为顾问 1，将经过 1000 回合预训练的顾问称为顾问 2，将经过 1500 回合预训练的顾问称为顾问 3，将经过 2000 回合预训练的顾问称为顾问 4。在本实验中，我们首先在没有顾问的情况下训练 MA-TLQL（记为 MA-QL），随后从顾问集合中选取一位顾问加入，继续训练 MA-TLQL。MA-QL 始终从低层Q 中选择动作（由于没有顾问，因此不维护高层Q ）。我们的目标是研究存在不同数量顾问时 MA-TLQL 的性能。为检验统计显著性，我们采用非配对双侧 t 检验，并报告 p-值。

结果见图 16。可以看出，不使用任何顾问训练的 MA-QL 表现最差。使用一位顾问（顾问 1）训练的 MA-TLQL 表现优于 MA-QL（$p < 0.06$）。接下来，我们使用两位顾问（顾问 1 和顾问 2）训练 MA-TLQL，其表现优于 MA-QL 和使用一位顾问训练的 MA-TLQL（$p < 0.1$）。同样，使用三位顾问（顾问 1、顾问 2 和顾问 3）的 MA-TLQL 表现优于使用两位顾问的情况（$p < 0.09$），而使用全部四位顾问的 MA-TLQL 表现最佳（$p < 0.04$）。这一结果表明，随着加入更优秀的顾问（优于当前集合中的顾问），MA-TLQL 的表现能够持续提升；这说明 MA-TLQL 能够从可用顾问集合中识别合适的顾问，并在多智能体环境中利用不同可用顾问的专业知识。根据 p值，我们注意到，使用全部四位顾问的 MA-TLQL 表现最佳这一观察结果具有统计显著性。尽管我们观察到随着每位顾问的加入，表现持续提升，但根据 p值，其中一些比较并未达到统计显著性。

J 说明性示例

在本节中，我们将展示一个玩具示例：与本文提出的 MA-TLQL 更新相比，Li 等人 [21] 给出的 TLQL 更新需要更长时间才能找出一组顾问中的最佳顾问。我们将使用图 17 所示的单智能体网格世界环境，而不是多智能体环境。按照 Li 等人 [21]的提议，TLQL 更新将使用贝尔曼更新来更新低层Q，随后通过同步步骤更新其高层Q。在 MA-TLQL 中，低层Q 将采用式 3 中给出的控制更新，但以单智能体方式进行（无需考虑联合动作），因为此环境中只有一个智能体。高层Q 将以单智能体方式采用式 1 中给出的评估更新。此外，为简单起见，在此示例中，我们不会在 MA-TLQL 的动作选择中使用式 2 所示的集成选择策略，而只使用高层Q 和低层Q 更新，以便在给定状态下评估顾问和动作。这里有一组包含 6 个状态的状态集合 $\{S1, S2, S3, S4, S5, G\}$，智能体从状态 $S1$ 出发，尝试到达目标状态 G。状态 $S3$、 $S4$ 和 G 是终止状态，智能体在状态 $S3$ 和 $S4$中获得 -1 的奖励，在状态 G中获得 +1 的奖励。状态 $S5$ 仅为对称性而存在（实际中无法到达）。智能体在每个状态都可以采取两个动作 $\{R, D\}$ 中的一个（分别表示向右和向下）。在此环境中可以看出，智能体需要在状态 $S1$ 和 $S2$ 采取动作 R，才能获得最大奖励。智能体可以使用两个顾问 $A1$ 和 $A2$：其中， $A1$是最优顾问，会在每个状态提供正确的动作（向右）； $A2$ 是次优顾问，以 0.5 的概率提供动作 R，并以 0.5 的概率提供动作 D。此领域中的所有状态转移都是确定性的。此外，我们设定学习率 (α) 为 0.1，折扣因子 (γ) 为 0.9。

现在，我们来考察 MA-TLQL 和 TLQL 中的 Q-更新（高层和低层）。初始时，我们任意将所有 Q-值设为 0。

在初始时间步（$t = 0$），假设智能体从初始状态 $S1$开始。两位顾问都建议采取动作 R。随后，智能体执行该动作，并首先使用以下方程更新其低-Q：

(ST./AC.)	右	Down
S1	0	0
S2	0	0

(a) 时间 $t = 1$的低-Q 价值

(ST./AC.)	Adv. 1	Adv. 2
S1	0	0
S2	0	0

(b) TLQL 在时间 $t = 1$的高-Q

(ST./AC.)	Adv. 1	Adv. 2
S1	0	0
S2	0	0

(c) MA-TLQL 在时间 $t = 1$的高-Q

表 2：时间 $t = 1$的 TLQL 和 MA-TLQL 更新。各列表示动作，各行表示状态。

$$
\begin{aligned}
&lowQ_1(S1, R)\\
&\quad = lowQ_0(S1, R) + \alpha\Big(r + \gamma \max_a lowQ_0(S2, a) - lowQ_0(S1, R)\Big)\\
&lowQ_1(S1, R) = 0 + 0.1\Big(0 + 0.9 \times 0 - 0\Big)\\
&lowQ_1(S1,\ R) = 0.
\end{aligned}
\tag{60}
$$

现在，TLQL 会将两位顾问的高-Q 值设为 0（同步）。MA-TLQL 的更新也会使两位顾问的值均为 0。在表 2 中，我们列出了非终止状态（$S1$ 和 $S2$）的 Q值。

在下一个时间步 $t = 2$，智能体处于状态 $S2$。再次假设两位顾问都建议采取正确动作。现在，low-Q 更新如下:

$$
\begin{aligned}
&lowQ_2(S2,\ R)\\
&= lowQ_1(S2, R) + \alpha\Big(r + \gamma \max_a lowQ_1(G, a) - lowQ_1(S2, R)\Big)\\
&lowQ_2(S2, R) = 0 + 0.1\Big(1 + 0.9 \times 0 - 0\Big)\\
&lowQ_2(S2,\ R) = 0.1
\end{aligned}
\tag{61}
$$

我们规定，当下一个状态为终止状态时，时序差分（T.D.）目标就是奖励本身。在这种情况下，TLQL 和 MA-TLQL 两种算法也会具有相同的 high-Q 值。时间 $t = 2$ 的 Q-值列于表 3 中。

现在，我们进入时间 $t = 3$。由于在上一个时间步已达到目标，智能体会重置回初始状态 $S1$。现在，假设两位顾问在状态 $S1$再次指定正确动作。低-Q 值更新如下，

(ST./AC.)	右	Down
S1	0	0
S2	0.1	0

（a）时间 $t = 2$的低-Q 价值

(ST./AC.)	Adv. 1	Adv. 2
S1	0	0
S2	0.1	0.1

（b）时间 $t = 2$的 TLQL 高-Q

(ST./AC.)	Adv. 1	Adv. 2
S1	0	0
S2	0.1	0.1

(c) MA-TLQL 在时间Q处的高-Q at time $t = 2$

表 3：TLQL 和 MA-TLQL 在时间 $t = 2$的更新

(ST./AC.)	右	Down
S1	0.009	0
S2	0.1	0

(a) 时间 $t = 3$的低-Q 价值

(ST./AC.)	Adv. 1	Adv. 2
S1	0.009	0.009
S2	0.1	0.1

(b) 时间 $t = 3$的 TLQL 高-Q

(ST./AC.)	Adv. 1	Adv. 2
S1	0.009	0.009
S2	0.1	0.1

(c) MA-TLQL 在时间 $t = 3$的高-Q

表 4：TLQL 和 MA-TLQL 在时间 $t = 3$的更新

$$
\begin{aligned}
&lowQ_3(S1,\ R)\\
&= lowQ_2(S1, R) + \alpha\Big(r + \gamma \max_a lowQ_2(S2, a) - lowQ_2(S2, R)\Big)\\
&lowQ_3(S1, R) = 0 + 0.1\Big(0 + 0.9 \times 0.1 - 0\Big)\\
&lowQ_3(S1, R) = 0.009
\end{aligned}
\tag{62}
$$

同样，MA-TLQL 和 TLQL 的高-Q 值将与低-Q 值相同，并列于表 4 中。

接下来，智能体移动到状态 $S2$。此时为时间 $t = 4$。设顾问 $A2$ 在此状态指定动作 D（顾问 $A1$ 始终指定 R）。再假设智能体选择在此状态听取 $A2$的意见（两位顾问的Q-值相同，因此智能体对选择哪位顾问没有偏好），于是它从 $A2$执行动作 D（下）。现在，low-Q 的 Q值可以更新为:

$$
\begin{aligned}
lowQ_4(S2, D) &= lowQ_4(S2, D)\\
&\quad + \alpha\Big(r + \gamma \max_a lowQ_4(S4, a) - lowQ_4(S2, D)\Big)\\
lowQ_4(S2,\ D) &= -0.1
\end{aligned}
\tag{63}
$$

MA-TLQL 更新的高-Q 价值如下，

(ST./AC.)	右	Down
S1	0.009	0
S2	0.1	-0.1

(a) 时间 $t = 4$的 Low-Q 价值

(ST./AC.)	Adv. 1	Adv. 2
S1	0.009	0.009
S2	0.1	-0.1

(b) TLQL 高-Q 在时间 $t = 4$

(ST./AC.)	Adv. 1	Adv. 2
S1	0.009	0.009
S2	0.1	-0.01

(c) MA-TLQL 高-Q 在时间 $t = 4$

表 5：TLQL 和 MA-TLQL 在时间 $t = 4$的更新

(ST./AC.)	右	Down
S1	0.0171	0
S2	0.1	-0.1

(a) 时间 $t = 5$的低-Q 价值

(ST./AC.)	Adv. 1	Adv. 2
S1	0.0171	0.0171
S2	0.1	-0.1

(b) 时间 $t = 5$的 TLQL 高-Q

(ST./AC.)	Adv. 1	Adv. 2
S1	0.0171	0.0072
S2	0.1	-0.01

(c) MA-TLQL 在时间 $t = 5$的高-Q

表 6：时间 $t = 5$的 TLQL 和 MA-TLQL 更新

$$\begin{aligned} highQ_4(S2,A2) &= highQ_3(S2,A2) \\ &\quad +\alpha\Big(r+\gamma highQ_3(S4,A2) - highQ_3(S2,A2)\Big) \\ highQ_4(S2,A2) &= 0.1+0.1(-1-0.1) \\ highQ_4(S2,A2) &= -0.01 \end{aligned} \tag{64}$$

时间 $t = 4$ 的所有 Q-值均列于表 5。

由于状态 $S4$ 是终止状态，智能体回到了状态 $S1$。现在时间为 $t = 5$。在此状态下，假设两位顾问都指定动作 R。此时，智能体选择执行该动作，并更新其 low-Q，

$$\begin{aligned} lowQ_5(S1,R) &= lowQ_4(S1,R) \\ &\quad +\alpha\Big(r+\gamma \max_a lowQ_4(S2,a) - lowQ_4(S2,R)\Big) \\ lowQ_5(S1,R) &= 0.009+0.1\Big(0+0.9\times 0.1-0.009\Big) \\ lowQ_5(S1,R) &= 0.0171. \end{aligned} \tag{65}$$

由于两位顾问都指定了动作 R，在 TLQL 更新中，两位顾问在 high-Q 中被赋予相同的 Q-值。现在，顾问 $A1$ 的 high-Q 估计值在 MA-TLQL 更新中也将与 low-Q 值相同。然而，MA-TLQL 更新中顾问 $A2$ 的 high-Q 估计值如下：

$$\begin{aligned} highQ_5(S1,A2) &= highQ_4(S1,A2) \\ &\quad +\alpha\Big(r+\gamma highQ_4(S2,A2) - highQ_4(S1,A2)\Big) \\ highQ_5(S1,A2) &= 0.009+0.1\Big(0+0.9\times -0.01-0.009\Big) \\ highQ_5(S1,A2) &= 0.009-0.0018=0.0072 \end{aligned} \tag{66}$$

在此阶段（时间 $t = 5$），所有 Q值均列于表 6 中。比较 . TLQL 和 MA-TLQL 的高-Q 估计值可以看出，在 TLQL 更新中（表 6b），智能体在状态 $S1$ 下无法区分遵循顾问 $A1$ 还是顾问 $A2$ （ Q值相同），而在状态 $S2$ 下则会决定选择顾问 $A1$。即使经过 5 个更新步骤，TLQL 仍无法确定正确的顾问（因为对于这两个状态，顾问 $A1$ 都优于 $A2$）。相比之下，表 6c 中的 MA-TLQL 更新清楚地显示，对于两个状态，顾问 $A1$ 的 Q值都高于顾问 $A2$。该示例展现了这样一种情况：与 Liet al. [21] 提出的原版TLQL 更新相比，MA-TLQL 能更快地区分优质顾问和劣质顾问。

K 实验详情

本节将完整介绍我们所有实验领域的细节，包括奖励函数和所用顾问的相关信息。对于 Pommerman 和追捕环境，我们假设其他智能体的所有动作都可以直接观察到（完全可观测）、在智能体之间共享，或由游戏引擎提供以执行集中式更新。对于 MPE 环境，其他智能体的动作仅在训练期间可观测，在执行期间不可观测（集中训练、分散执行）。

表 7 汇总了所有实验设置及其对应的顾问配置。

K.1 Pommerman

在 Pommerman 中，学习游戏取胜所需的全部技能包括：1）躲避敌人；2）获取强化道具（炸弹/生命）；3）消灭敌人；4）炸开墙壁以开辟路线；5）与队友协同（团队版本中） [31]。

在我们的实验中，我们考虑了两个 Pommerman 领域。前四个实验使用 Pommerman 的双智能体版本，第五个实验使用四智能体团队版本的 Pommerman。在训练和执行实验中，每个回合都对应一场完整的 Pommerman 游戏，棋盘随机生成，最多进行 800 步后结束。游戏会在所有步数执行完毕，或两名 Pommerman 智能体中的一名死亡时结束（双智能体版本）。在团队版本中，游戏会在所有步数执行完毕（最多 800 步），或两支队伍中的一支全员阵亡时结束。

第一个实验（实验 1）使用了四位能力各异的充分顾问。第一位顾问（顾问 1）能够教授赢得 Pommerman 游戏所需的全部策略（即第6节中提到的各种 Pommerman技能）。顾问 2 能够教授在对手非常靠近智能体时击杀对手的动作，以及有助于躲避敌人的防御策略。第三位顾问（顾问 3）只能教授

实验	领域	Type	顾问	# of 训练 ing 智能体
1	Two- 智能体 Pommer- man	竞争性 4 个	足够的顾问——问，质量各不——同	2
2	Two- 智能体 Pommer—— man	竞争性 4 个	足够的顾问——具有相似质量	2
3	Two- 智能体 Pommer- man	竞争性 4 个	不足的顾问- 具有不同- 不同质量	2
4	Two- 智能体 Pommer- man	竞争性 4 个	不足的顾问 具有相似质量	2
5	四 智能体 Pommer- man	混合	4 个足够的顾问具有不同智能体质量	4
6	追逐 SISL	合作 4 个不	足的顾问- 问，其质量各不相同- 同的质量	8
7	捕食者- Prey MPE	混合 （集中训练、分散执行）	4 个质量不佳的顾问质量各不相同的顾问	8

表 7：实验设置说明

有助于逃离敌人的防御策略，无法教授击杀敌人所需的进攻策略。最后一位顾问（顾问 4）只会提供随机动作。因此，所有智能体都应尽可能听从第一位顾问的建议，并且必须完全避开第四位顾问。

在第二个实验（实验 2）中，我们使用一组新的四位顾问。在这里，第一位顾问只教授防御技能（躲避敌人），第二位顾问只能教授进攻技能（消灭敌人），第三位顾问只能教授能够获取强化道具的策略，第四位顾问则教授寻找并炸开木墙的方法，从而为游戏开辟各种路径。由此可见，没有任何一位顾问能够教授在 Pommerman 中获胜所需的全部策略。然而，四位顾问共同起来能够教授所需的策略，因此需要恰当地利用他们。

在实验 3 中，我们使用了一组质量逐渐下降的四位顾问，与第一个实验中的设置相同；然而，这些顾问都无法教授寻找并消灭敌人的策略（技能集合不充分）。此外，没有任何顾问能够教授寻找木墙并将其炸开的必要技能。因此，这组顾问不足以帮助智能体赢得 Pommerman 游戏。第一位顾问（顾问 1）可以教授躲避敌人的策略、在敌人紧邻智能体时将其消灭，以及获取强化道具。第二位顾问（顾问 2）只能教授有助于躲避敌人或消灭附近敌人的策略。顾问 3 只能教授与躲避敌人有关的策略，而顾问 4 只会提供随机建议。

在实验 4 中，我们使用了一组与第二个实验相似的顾问；然而，这些顾问无法教授赢得 Pommerman 所需的足够技能。第一位顾问（顾问 1）只教授躲避敌人的防御技能。第二位顾问（顾问 2）帮助学习一种能够消灭紧邻智能体的敌人的策略。第三位顾问（顾问 3）可以教授有助于获取强化道具的策略，而第四位顾问（顾问 4）可以教授炸开木墙所需的技能，但前提是智能体非常靠近墙壁。

团队领域中的第五个实验（实验 5）使用与实验 1 相同的一组顾问。

Pommerman 环境由 Resnick 等人 [31] 以 Apache 2 许可证发布。

K.2 追捕领域

追捕领域最初由 Gupta 等人介绍 [8], ，我们使用 Petting Zoo 环境 [43] 提供的实现（以 MIT 许可证发布）。游戏中有 8 个追捕智能体相互合作，在环境中捕捉 30 个逃避者。我们采用与 Terry 等人 [43] 相同的奖励函数和环境参数，并做了一些小幅修改。在我们的设置中，智能体击中（标记）逃避者可获得 +1奖励，捕获逃避者可获得 +30 奖励。当至少两个追捕智能体包围一个逃避者时，该逃避者即被判定为捕获。被捕获的逃避者会从环境中移除。在游戏的每个时间步，所有智能体都会受到 -0.1 的紧迫性惩罚。每个回合最多包含 500 步。所有逃避者均被捕获或所有步数均已完成时，回合结束。在所有时间步中，所有追捕者获得相同的奖励（全局奖励结构），奖励由所有智能体共享。每个追捕者都能观察自身周围的 7×7 网格（整个网格为 16×16），这意味着追捕者可以完整获取该可观测网格内其他追捕者和逃避者的信息，但无法获取网格外的信息。

K.3 多粒子环境（捕食者-猎物）

多粒子环境（MPE）最初由 Lowe 等人发布， [24] 作为一组测试平台，用于测试适用于具有通信和

障碍物交互特征的合作及混合合作-竞争设置的算法。我们从这组测试平台中选用了 Simple Tag 环境，该环境属于捕食者-猎物设置，其中三个捕食者试图捕获一只猎物。我们使用 Petting Zoo 库中的环境定义 [43]（依据 MIT 许可证发布）。这里所有智能体的动作空间均为离散空间，包含 5 个动作（四个基本方向以及一个表示不移动的动作）。捕食者和猎物都具有连续观测空间，对应于所有智能体（包括智能体自身）的位置和速度。捕食者在击中（碰撞/标记）猎物时获得 +10奖励；猎物被任一捕食者击中时受到 -10 的惩罚（猎物不会从环境中移除）。猎物离开比赛场地时还会受到少量额外惩罚（更多详情请参阅 Terry 等人 [43] ）。在所有时间步中，所有捕食者获得相同的奖励（全局奖励结构），奖励由所有捕食者共享。我们的环境包含八个捕食者和八只猎物，此外还有五个会阻挡捕食者和猎物路径的障碍物。每场游戏包含 500 个训练或执行步骤。我们将该领域建模为集中训练、分散执行设置：训练期间，所有智能体采取的动作和获得的奖励对每个智能体均可见，但执行期间不可见。

L 超参数与实现细节

我们实现基线方法时使用的所有超参数，要么与介绍这些算法的相应论文所推荐的值相同，要么与之非常接近。我们的算法也采用与基线算法相似的超参数，但出于性能和计算效率方面的考虑，有少数例外。

DQN [26], CHAT [47], TLQL [21], ADMIRAL-DM [38], 和 MA-TLQL 的实现采用几乎相同的超参数。这些算法的学习率为 0.01，折扣因子为 0.9，经验回放池大小为 2×10^6，固定探索率为 0.9。目标网络使用硬替换策略，每经过 10 次学习迭代便替换一次。评估网络和目标网络均包含 3 个全连接层（2 个含 50 个神经元的 ReLU 层，以及一个输出层）。批量大小设为 32。

对于 CHAT 的实现，我们采用 Wang 和 Taylor [47] 提出的基于神经网络的置信度变体（NNHAT）。我们将置信度阈值设为 0.6，并使用 3 个全连接层（其中 2 层各含 50 个神经元，并以 ReLU 作为激活函数，另加一个输出层）。出于性能方面的考虑，同时由于我们实验中使用的顾问是基于规则的智能体或预训练网络，而非 CHAT 设计中所用的人类顾问，我们在 CHAT 中直接使用顾问，而不像 [47] 中那样根据分类器模型准备基于决策的规则。由于在 CHAT 中，这些顾问被视为提取出的规则（而非实际的演示者/顾问），因此我们也允许 CHAT 在执行阶段依赖顾问。

我们的实验在 TLQL 中使用了 PPR 技术，而 Li 等人 [21] 的研究中并未使用该技术。这是出于两个原因。首先，与单智能体环境不同，独立 Q-学习方法在多智能体环境中没有策略改进保证（如第 4 节所述） [41], ，因此，原版TLQL 并不能保证像 Li 等人 [21] 所提出的那样停止依赖顾问。其次，如果不使用 PPR，实验中 TLQL 的训练性能会非常好（因为它可以无限依赖顾问），但执行性能则会很差（因为执行期间无法使用顾问）。使用 PPR 时，TLQL 的执行性能最佳。

关于 PPR 技术的实现，TLQL、ADMIRAL-DM、MA-TLAC 和 MA-TLQL 的实现均从 $\epsilon_\prime = 1$ 开始，并在训练结束时线性衰减至 0。执行期间顾问不会产生影响，因此，执行期间的 $\epsilon_\prime = 0$ 。

为与 Li 等人 [21], 的描述保持一致，TLQL 的实现使用总共 3 个网络（低-Q 的评估网络和目标网络，以及高-Q）。高-Q 和低-Q 网络使用全连接层，其架构与 DQN 中描述的相同。MA-TLQL 的实现使用四个网络（分别用于低-Q 和高-Q 的两组评估网络和目标网络），配置也与 DQN 中描述的相同。此外，MA-TLQL 为高-Q 使用第二个经验回放缓冲区，同时保留低-Q 的经验回放缓冲区。两个缓冲区的内存大小均为 2×10^6。

关于 DQfD [12], ，我们将 1×10^6 设为演示缓冲区大小，并在预训练期间执行 50,000 次小批量更新。经验回放缓冲区的大小是演示缓冲区的两倍。N 步回报权重为 1.0，监督损失权重为 1.0，L2 正则化权重为 10^{-5}。ε 贪心探索率为 0.9。折扣因子为 0.99，学习率为 0.002。网络架构包含 3 个全连接层（2 个各含 24 个神经元的 ReLU 层和一个输出层）。DQfD 的预训练数据来自一个数据缓冲区，该缓冲区与一系列顾问相互竞争的游戏有关。

MA-TLAC 使用两个演员网络和两个评论家网络。网络架构与 MA-TLQL 中所述的相同。演员网络的学习率为 10^{-6}，评论家网络的学习率为 10^{-3}。

对于所有训练实验，我们使用一组 30 个随机种子（1 – 30）。对于执行实验，我们使用一组新的 30 个随机种子（31 – 60）。

M 墙钟时间

所有实验的训练均在一台虚拟机上进行，该虚拟机配备 2 个 Nvidia A100 GPU，GPU 显存为 40 GB。CPU 使用 AMD EPYC 处理器，内存为 125 GB。Pommerman 实验平均需要 12 小时墙钟时间才能完成，而追逐实验平均需要 15 小时墙钟时间才能完成。