arXiv:2302.01188v1 [cs.LG] 2 Feb 2023

最佳可能 Q 学习

Jiechuan Jiang [1] **Zongqing Lu** [1]

摘要

全局信息（即其他智能体的动作）不可获取的完全去中心化学习，是合作式多智能体强化学习中的一项根本挑战。然而，由于所有智能体同时更新策略，转移概率呈现非平稳性，因此大多数去中心化算法在理论上无法保证收敛和最优性。为应对这一挑战，我们提出最佳可能算子，这是一种新颖的去中心化算子，并证明如果每个智能体都独立地通过该算子更新其个体状态-动作价值，那么智能体的策略将收敛至最优联合策略。此外，为了使更新更加高效和实用，我们简化了该算子，并证明简化后的算子仍能保证收敛和最优性。通过实例化这一简化算子，得到的完全去中心化算法最佳可能 *Q* 学习 (BQL)不受非平稳性的影响。实验结果表明，在多种合作式多智能体任务中，BQL 相较于基线方法取得了显著提升。

1. 引言

协作式多智能体强化学习（MARL）旨在训练一组智能体，使其最大化累积共享奖励，这对于现实世界中的应用具有重要意义，包括物流（Li 等人，2019年）、交通信号控制（Xu 等人，2021年）、电力调度（Wang 等人，2021b）和游戏（Vinyals 等人，2019年）。尽管现有的大多数方法遵循集中训练与分散执行（CTDE）范式，但在许多训练期间无法获取所有智能体信息的场景中，每个智能体都必须在没有集中信息的情况下独立学习。因此，完全去中心化学习中，智能体只能利用局部经验，而无法获取其他智能体的动作，这种学习方式非常理想（Jiang & Lu，2022年）。

然而，在完全去中心化学习中，由于其他智能体被视为环境的一部分，且它们同时更新各自的策略，从单个智能体的视角来看，转移概率将呈现非平稳性。因此，大多数去中心化算法的收敛性例如，独立 Q 学习（IQL）（Tan，1993）在理论上无法得到保证。多智能体交替 Q 学习（MA2QL）（Su 等人，2022年）保证收敛到纳什均衡，但当存在多个均衡时，收敛到的均衡未必是最优均衡（Zhang等人，2021a）。分布式 IQL （ Lauer&Riedmiller，2000 年）和 I2Q （ Jiang& Lu，2022 年）能够学习最优联合策略，但仅适用于确定性环境。如何保证在随机环境中收敛到最优联合策略，仍是一个尚未解决的问题。

为应对这一挑战，我们提出了最佳可能算子，这是一种用于更新每个智能体的个体状态-动作价值的新型去中心化算子，并证明在该算子下，各智能体的策略会收敛于最优联合策略。然而，执行最佳可能算子的效率较低，因而不具备实际可行性，因为每次更新都需要计算所有可能转移概率的期望值，并将状态-动作价值更新为其中的最大值。因此，我们进一步提出了简化最佳可能算子。在每次更新时，简化算子仅计算其中一个可能转移概率的期望值，并单调地更新状态-动作价值。我们证明，在简化算子下，各智能体的策略同样会收敛于最优联合策略。对于表格情形，我们分别使用Q表实例化简化算子；对于复杂环境，则使用神经网络进行实例化。在Q表实例化中，非平稳性会被自然地避免；而在神经网络实例化中，回放缓冲区中的非平稳性不再是缺点，而是收敛的必要条件。

我们提出的算法 最佳可能 ***Q*** 学习 **(BQL)** 是完全去中心化的，无需使用其他智能体的信息。我们在多种多智能体协作任务上评估了 BQL，即随机博弈、基于 MPE 的微分博弈（Lowe等人，2017年）、多智能体 MuJoCo（de Witt等人，2020b）、SMAC（ Samvelyan 等人，2019年）和 GRF（Kurach等人，2020年），涵盖完全可观测与部分可观测、确定性与随机、离散与连续环境。

[1]北京大学计算机科学学院。通讯作者：Jiechuan Jiang < jiechuan.jiang@pku.edu.cn>、Zongqing Lu < zongqing.lu@pku.edu.cn>。

环境。实证结果表明，BQL 显著优于基线方法。据我们所知，BQL 是首个能够保证在随机环境中收敛至全局最优解的去中心化算法。最佳可能算子的更多简化形式与具体实例仍有待进一步探索。我们相信，BQL 可以成为完全去中心化学习的一种新范式。

2. 方法

2.1. 预备知识

考虑具有状态空间 $\mathcal{S}$ 和联合动作空间 $\mathcal{A}$ 的 N 智能体 MDP $M_{\text{env}} =< \mathcal{S}, \mathcal{O}, \mathcal{A}, R, P_{\text{env}}, \gamma >$。每个智能体 i 选择一个单独的动作 a_i，环境根据联合动作 $\boldsymbol{a}$ 和转移概率 $P_{\text{env}}(s'|s, \boldsymbol{a})$ 转移到下一个状态 s'。为简化理论分析，我们假设所有智能体都能获得状态 s，尽管在实践中，每个智能体 i 都可以使用局部观测 $o_i \in \mathcal{O}$ 或轨迹来做出决策。所有智能体都获得共享奖励 $r = R(s, s') \in [r_{\min}, r_{\max}]$，并学习最大化期望折扣回报 $\mathbb{E}\sum_{t=0}^{\infty}\gamma^t r_t$。在完全去中心化设置中，$M_{\text{env}}$ 是部分可观测的，因为每个智能体 i 只能观测自己的动作 a_i，而无法观测联合动作 $\boldsymbol{a}$。从每个智能体i的角度来看，存在一个 MDP $M_i =< \mathcal{S}, \mathcal{A}_i, R, P_i, \gamma >$，其具有单独的动作空间 $\mathcal{A}_i$ 和转移概率

$$P_i(s'|s, a_i) = \sum_{\boldsymbol{a}_{-i}} P_{\text{env}}(s'|s, a_i, \boldsymbol{a}_{-i})\,\boldsymbol{\pi}_{-i}(\boldsymbol{a}_{-i}|s) \quad (1)$$

其中 $\boldsymbol{\pi}_{-i}$ 表示除智能体 i之外所有智能体的联合策略，$\boldsymbol{a}_{-i}$也同理。根据（1），转移概率 P_i 取决于其他智能体的策略 $\boldsymbol{\pi}_{-i}$。由于其他智能体持续更新其策略，P_i会变得 非平稳。对于非平稳转移概率，独立Q学习的收敛[1]

$$Q_i(s, a_i) = \mathbb{E}_{P_i(s'|s,a_i)}\left[r + \gamma \max_{a_i'} Q_i(s', a_i')\right] \quad (2)$$

无法得到保证，而在完全分散式设置中如何学习最优联合策略也颇具挑战性。在下一节中，我们提出最佳可能算子，这是一种新颖的完全去中心化算子，理论上能够保证在随机环境中收敛到最优联合策略。

2.2. 最佳可能算子

首先，我们来考虑最优联合Q值

$$Q(s, \boldsymbol{a}) = \mathbb{E}_{P_{\text{env}}(s'|s,\boldsymbol{a})}\left[r + \gamma \max_{\boldsymbol{a}'} Q(s', \boldsymbol{a}')\right], \quad (3)$$

它是最优联合策略$\boldsymbol{\pi}^*(s) = \arg\max_{\boldsymbol{a}} Q(s, \boldsymbol{a})$的期望回报。基于最优联合Q值，对于每个智能体 i，我们定义 $\max_{\boldsymbol{a}_{-i}} Q(s, a_i, \boldsymbol{a}_{-i})$，其满足以下不动点方程：

$$\begin{aligned}
&\max_{\boldsymbol{a}_{-i}} Q(s, a_i, \boldsymbol{a}_{-i}) \\
=&\max_{\boldsymbol{a}_{-i}} \mathbb{E}_{P_{\text{env}}(s'|s,\boldsymbol{a})}\left[r + \gamma \max_{a_i'} \max_{\boldsymbol{a}'_{-i}} Q(s, a_i', \boldsymbol{a}'_{-i})\right] \quad (4)\\
=&\mathbb{E}_{P_{\text{env}}(s'|s,a_i,\boldsymbol{\pi}^*_{-i}(s,a_i))}\left[r + \gamma \max_{a_i'} \max_{\boldsymbol{a}'_{-i}} Q(s, a_i', \boldsymbol{a}'_{-i})\right] \quad (5)
\end{aligned}$$

其中 $\boldsymbol{\pi}^*_{-i}(s, a_i) = 是_{\boldsymbol{a}_{-i}} Q(s, a_i, \boldsymbol{a}_{-i})$ 在给定 a_i 的情况下其他智能体的最优条件联合策略。式（4）由对式（3$\boldsymbol{a}_{-i}$）两边取 max 得到，而式（5）则是将 $\boldsymbol{\pi}^*_{-i}(s, a_i)$ 并入 P_{env}所得。由此得到以下引理。

引理 2.1。如果每个智能体 i 学习独立价值函数 $Q_i(s, a_i) = \max_{\boldsymbol{a}_{-i}} Q(s, a_i, \boldsymbol{a}_{-i})$，并将动作选为 $\arg\max_{a_i} Q_i(s, a_i)$，那么当且仅有一个最优联合策略时，智能体将获得该最优联合策略[2]。

证明。 由于 $\max_{a_i}\max_{\boldsymbol{a}_{-i}} Q(s, a_i, \boldsymbol{a}_{-i}) = \max_{\boldsymbol{a}} Q(s, \boldsymbol{a})$，且最优联合策略只有一个，因此，$\arg\max_{a_i} Q_i(s, a_i)$ 是智能体 i 在最优联合动作中的动作 $\boldsymbol{a}$。 □

根据引理 2.1，要获得最优联合策略，就要让每个智能体 i 学习价值函数 $Q_i(s, a_i) = \max_{\boldsymbol{a}_{-i}} Q(s, a_i, \boldsymbol{a}_{-i})$。为此，我们提出了一个新的算子，用于以完全去中心化的方式更新 Q_i：

$$Q_i(s, a_i) = \max_{P_i(\cdot|s,a_i)} \mathbb{E}_{P_i(s'|s,a_i)}\left[r + \gamma \max_{a_i'} Q_i(s', a_i')\right]. \quad (6)$$

给定 s 和 a_i，由于其他智能体的策略 $\boldsymbol{\pi}_{-i}$不同，会有许多 $P_i(s'|s, a_i)$。为降低复杂度，我们仅考虑确定性策略，因为当最优联合策略只有一个时，该策略必定是确定性的（Puterman，1994年）。因此，算子（6）仅对确定性 $\boldsymbol{\pi}_{-i}$下的转移概率 $P_i(s'|s, a_i)$取最大值。直观来说，该算子不断追求“最佳可能期望回报”，直到 Q_i 达到最优期望回报 $\max_{\boldsymbol{a}_{-i}} Q(s, a_i, \boldsymbol{a}_{-i})$，因此我们将算子（6）命名为最佳可能算子。接下来，我们将从理论上证明，在最佳可能算子作用下，$Q_i(s, a_i)$ 收敛至 $\max_{\boldsymbol{a}_{-i}} Q(s, a_i, \boldsymbol{a}_{-i})$，从而使智能体学得最优联合策略。令 $Q_i^k(s, a_i)$ 表示更新 k 中的价值函数，且 $Q_i(s, a_i) := Q_i^{\infty}(s, a_i)$。于是，我们得到以下引理。

引理 2.2. 若 Q_i^0 初始化为最小回报$\frac{r_{\min}}{1-\gamma}$，则在最佳可能算子下，$\max_{\boldsymbol{a}_{-i}} Q(s, a_i, \boldsymbol{a}_{-i}) \geq Q_i^k(s, a_i)\ \forall s, a_i, \forall k$。

[1]为简单起见，除非另有说明，本文将最优值 Q^* 称为 Q。

[2]我们可以采用 I2Q 中提出的简单方案来处理多个最优联合策略，相关内容见附录 D。

证明。我们通过归纳法证明该引理。首先，由于Q_i^0初始化为最小回报，因此$\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i}) \geq Q_i^0(s,a_i)$。然后，假设$\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i}) \geq Q_i^{k-1}(s,a_i)\ \forall s,a_i$。令$\arg\max_{P_i(s'|s,a_i)} \mathbb{E}_{P_i(s'|s,a_i)}\left[r+\gamma \max_{a_i'} Q_i^{k-1}(s',a_i')\right]= P_i^*(s'|s,a_i)$，则有

$$
\begin{aligned}
&\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i}) - Q_i^k(s,a_i) \\
=&\max_{\boldsymbol{a}_{-i}} \sum_{s'} P_{\text{env}}\left(s'|s,a_i,\boldsymbol{a}_{-i}\right)\left[r+\gamma \max_{a_i'} \max_{\boldsymbol{a}'_{-i}} Q(s',a_i',\boldsymbol{a}'_{-i})\right] \\
&\quad - \sum_{s'} P_i^*(s'|s,a_i)\left[r+\gamma \max_{a_i'} Q_i^{k-1}(s',a_i')\right] \\
\geq& \sum_{s'} P_i^*(s'|s,a_i)\left[r+\gamma \max_{a_i'} \max_{\boldsymbol{a}'_{-i}} Q(s',a_i',\boldsymbol{a}'_{-i})\right] \\
&\quad - \sum_{s'} P_i^*(s'|s,a_i)\left[r+\gamma \max_{a_i'} Q_i^{k-1}(s',a_i')\right] \\
=&\gamma \sum_{s'} P_i^*(s'|s,a_i) \max_{a_i'}\left(\max_{\boldsymbol{a}'_{-i}} Q(s',a_i',\boldsymbol{a}'_{-i}) - Q_i^{k-1}(s',a_i')\right) \\
\geq&\gamma \sum_{s'} P_i^*(s'|s,a_i)\left(\max_{\boldsymbol{a}'_{-i}} Q(s',a_i'^*,\boldsymbol{a}'_{-i}) - Q_i^{k-1}(s',a_i'^*)\right) \\
\geq&0,
\end{aligned}
$$

其中$a_i'^* = \arg\max_{a_i'} Q_i^{k-1}(s',a_i')$。因此，在更新$k$中该结论成立。根据归纳法原理，该引理对所有更新均成立。□

直观地说，$\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i})$是执行动作$a_i$后的最优期望回报，因此它是$Q_i(s,a_i)$的上界。此外，根据引理 2.2，我们有以下引理。

引理2.3。$Q_i(s,a_i)$在最佳可能算子下收敛至$\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i})$。

证明。为清晰起见，我们用$P_{\text{env}}\left(s'|s,a_i,\boldsymbol{\pi}_{-i}^*\right)$表示$P_{\text{env}}\left(s'|s,a_i,\boldsymbol{\pi}_{-i}^*(s,a_i)\right)$。由（5）和（6），可得

$$
\begin{aligned}
&\left\|\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i}) - Q_i^k(s,a_i)\right\|_\infty \\
=&\max_{s,a_i}\left(\sum_{s'} P_{\text{env}}\left(s'|s,a_i,\boldsymbol{\pi}_{-i}^*\right)\left[r+\gamma \max_{a_i'} \max_{\boldsymbol{a}'_{-i}} Q(s',a_i',\boldsymbol{a}'_{-i})\right]\right. \\
&\left. - \sum_{s'} P_i^*(s'|s,a_i)\left[r+\gamma \max_{a_i'} Q_i^{k-1}(s',a_i')\right]\right) \leftarrow \text{(Lemma 2.2)} \\
\leq&\max_{s,a_i}\left(\sum_{s'} P_{\text{env}}\left(s'|s,a_i,\boldsymbol{\pi}_{-i}^*\right)\left[r+\gamma \max_{a_i'} \max_{\boldsymbol{a}'_{-i}} Q(s',a_i',\boldsymbol{a}'_{-i})\right]\right. \\
&\left. - \sum_{s'} P_{\text{env}}\left(s'|s,a_i,\boldsymbol{\pi}_{-i}^*\right)\left[r+\gamma \max_{a_i'} Q_i^{k-1}(s',a_i')\right]\right) \\
\leq&\gamma \max_{s',a_i'}\left(\max_{\boldsymbol{a'}_{-i}} Q(s',a_i',\boldsymbol{a'}_{-i}) - Q_i^{k-1}(s',a_i')\right) \\
=&\gamma\left\|\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i}) - Q_i^{k-1}(s,a_i)\right\|_\infty.
\end{aligned}
$$

我们有$\left\|\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i}) - Q_i^k(s,a_i)\right\|_\infty \leq \gamma^k \left\|\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i}) - Q_i^0(s,a_i)\right\|_\infty$。令$k\to\infty$，则$Q_i(s,a_i)\to\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i})$，因此引理成立。□

根据引理2.1 和 2.3，我们立即得到：

定理2.4。智能体在最佳可能算子下学习到最优联合策略。

2.3. 简化最佳可能算子

最佳可能算子保证收敛到最优联合策略。然而，要执行（6），每次更新时，每个智能体i都必须计算所有可能转移概率的期望值，并将Q_i更新为最大的期望值，这会带来过高的计算成本。因此，我们引入一个辅助价值函数$Q_i^{\text{e}}(s,a_i)$，并将（6）简化为两个算子。首先，在每次更新时，我们随机选择一个~可能的转移概率P_i，针对每个(s,a_i)，并通过以下方式更新$Q_i^{\text{e}}(s,a_i)$：

$$
Q_i^{\text{e}}(s,a_i) = \mathbb{E}_{\tilde{P}_i(s'|s,a_i)}\left[r+\gamma \max_{a_i'} Q_i(s',a_i')\right]. \tag{7}
$$

$Q_i^{\text{e}}(s,a_i)$表示所选转移概率的期望值。然后，我们通过以下方式单调更新$Q_i(s,a_i)$：

$$
Q_i(s,a_i) = \max\left(Q_i(s,a_i), Q_i^{\text{e}}(s,a_i)\right). \tag{8}
$$

我们将（7）和（8）统称为简化最佳可能算子。通过执行简化最佳可能算子，$Q_i(s,a_i)$能够高效地向最大期望值更新。由此得到以下引理。

引理 2.5. $Q_i(s,a_i)$在简化最佳可能算子作用下收敛到$\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i})$。

证明。根据 (8)，由于$Q_i(s,a_i)$单调递增，在更新k中，$Q_i^k(s,a_i) \geq Q_i^{k-1}(s,a_i)$。与引理 2.2 的证明类似，我们可以很容易地证明，在 (7) 和 (8) 下，$\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i}) \geq Q_i^k(s,a_i)$。因此，$\{Q_i^k(s,a_i)\}$是一个递增且有上界的序列。根据单调收敛定理，$\{Q_i^k(s,a_i)\}$在$k\to\infty$时收敛，并令$Q_i(s,a_i) := Q_i^\infty(s,a_i)$。

然后我们证明，收敛后的值$Q_i(s,a_i)$等于$\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i})$。由于单调性和收敛性，当$k > K$，$Q_i^k(s,a_i) - Q_i^{k-1}(s,a_i) \leq \epsilon$时，无论在更新$k$中选择哪个~$P_i$，$\forall \epsilon, s, a_i, \exists K$都成立。由于每个都有可能被选中，因此在选择$(, a_i) =$时

参数最大值$_{P_i(s'|s,a_i)}$ $\mathbb{E}_{P_i(s'|s,a_i)}$ $\left[r+\gamma$最大值$_{a_i'}$ $Q_i^{k-1}(s', a_i')\right]$ $=P_i^*(s'|s,a_i)$，通过执行（7）和（8），我们得到

$$Q_i^{k-1}(s,a_i)+\epsilon \geq Q_i^k(s,a_i) \geq Q_i^{\mathrm{e}}(s,a_i)$$
$$=\sum_{s'} P_i^*(s'|s,a_i)\left[r(s,s')+\gamma \max_{a_i'} Q_i^{k-1}(s',a_i')\right].$$

根据引理的证明2.3，我们有

$$\max_{s,a_i}\left(\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i})-Q_i^{\mathrm{e}}(s,a_i)\right)$$
$$\leq \gamma \max_{s,a_i}\left(\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i})-Q_i^{k-1}(s,a_i)\right).$$

用 s^*，a_i^* 表示

$$\arg\max_{s,a_i}\left(\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i})-Q_i^{k-1}(s,a_i)\right).$$

由于 $Q_i^{k-1}(s,\ a_i)+\epsilon \geq Q_i^{\mathrm{e}}(s,\ a_i)$,

$$\max_{\boldsymbol{a}_{-i}} Q(s^*,a_i^*,\boldsymbol{a}_{-i})-Q_i^{k-1}(s^*,a_i^*)-\epsilon$$
$$\leq \gamma \max_{\boldsymbol{a}_{-i}} Q(s^*,a_i^*,\boldsymbol{a}_{-i})-\gamma Q_i^{k-1}(s^*,a_i^*).$$

那么，我们有

$$\left\|\max_{\boldsymbol{a}_{-i}} Q(s,a_i,\boldsymbol{a}_{-i})-Q_i^{k-1}(s,a_i)\right\|_\infty \leq \frac{\epsilon}{1-\gamma}.$$

因此，$Q_i(s,\ a_i)$ 收敛于 $\max_{\boldsymbol{a}_{-i}} Q(s,\ a_i,\ \boldsymbol{a}_{-i})$. □

根据引理2.1和2.5，我们还可以得到：

定理2.6。智能体通过简化的最佳可能算子学习最优联合策略。

2.4. 最佳可能 Q 学习

最佳可能 ***Q*** 学习（BQL）基于简化的最佳可能算子实例化。我们首先考虑表格情形下的 Q 表学习。关键挑战在于，如何在确定性 $\boldsymbol{\pi}_{-i}$学习过程中获得所有可能的转移概率。为解决这一问题，整个训练过程被划分为 M 个训练周期。在第 m个训练周期，每个智能体i 随机且独立地初始化一个确定性策略$\hat{}\pi_i^m$，并选取状态的一个子集 S_i^m。随后，每个智能体 i使用该确定性策略与环境交互

$$\begin{cases}\arg\max_{a_i} Q_i(s,a_i) & \text{if } s \notin S_i^m,\\ \hat{\pi}_i^m(s) & \text{else.}\end{cases}$$

每个智能体 i 在回放缓冲区 $\mathcal{D}_i^m$中存储独立的经验（$s,\ a_i,\ s',\ r$）。由于 P_i 取决于 $\boldsymbol{\pi}_{-i}$，且智能体采用确定性策略行动，因此 $\mathcal{D}_i^m$在确定性 $\boldsymbol{\pi}_{-i}$下只包含一个 P_i。由于其他智能体修改其策略 $\boldsymbol{\pi}_{-i}$时，P_i 会发生变化，因此执行随机初始化策略 $\hat{}\pi_i^m$

算法1每个智能体的表 i**1：初始化表** Q_i **和** Q_i^{e}**。2：对于** $m=1,\ldots,M$**，执行 3：初始化回放缓冲区** $\mathcal{D}_i^m$**和探索策略**$\hat{}\pi_i^m$**。4：所有智能体与环境交互，并将经验（**s, a_i, s', r**）存入** $\mathcal{D}_i^m$**。5：对于** $t=1,\ldots,n\ update$**，执行 6：从** $\mathcal{D}_i^1,\cdots,\mathcal{D}_i^m$**中随机选择一个缓冲区** $\mathcal{D}_i^j$**。7：通过从** $\mathcal{D}_i^j$ **中采样，根据（7）更新** Q_i^{e}**。8：通过从** $\mathcal{D}_i^j$ **中采样，根据（8）更新** Q_i**。9：结束循环 10：结束循环**

算法2 每个智能体使用神经网络的 BQL i

1：初始化神经网络 Q_i **和** Q_i^{e}**，以及目标网络** $\bar{Q}_i^{\mathrm{e}}$**。2：初始化回放缓冲区** $\mathcal{D}_i$**。3：对于** $t=1,\ldots,$ $n\ iteration$**，执行 4：所有智能体与环境交互，并将经验 (**s, a_i, s', r**) 存入** $\mathcal{D}_i$**。5：从** $\mathcal{D}_i$**中采样一个小批量。6：通过最小化 (9) 更新** Q_i^{e}**。7：通过最小化 (10) 更新** $\bar{Q}_i$**。8：更新目标网络** $\bar{Q}_i^{\mathrm{e}}$**。9：结束循环**

在第 S_i^m轮次的 m 上，不仅有助于每个智能体 i 探索状态-动作对，也有助于其他智能体探索可能的转移概率。当 M 足够大时，对于任意 $(s,\ a_i)$ 对，都可以在回放缓冲区中找到任意 $P_i(s,\ a_i)$。

第 m轮次交互结束后，每个智能体 i 都有一个缓冲区序列 $\{\mathcal{D}_i^1,\cdots,\mathcal{D}_i^m\}$，其中每个缓冲区的转移概率各不相同。在第 m轮次的训练阶段，每个智能体 i从 $\{\mathcal{D}_i^1,\cdots,\mathcal{D}_i^m\}$ 中随机选择一个回放缓冲区 $\mathcal{D}_i^j$，并从 $\mathcal{D}_i^j$ 中采样小批量 $\{s,a_i,s',r\}$，根据 (7) 更新 Q表 $Q_i^{\mathrm{e}}(s,a_i)$，然后从$\mathcal{D}_i^j$ 中采样小批量，根据 (8) 更新 $Q_i(s,a_i)$。Q表的实现总结于算法1中。

接下来，我们分析收集缓冲区序列的样本效率。简化最佳可能算子要求，任何可能的 $P_i(s,\ a_i)$ $(s,\ a_i)$ 对都能在单个缓冲区中找到，但不考虑同一缓冲区中不同状态-动作对的转移概率之间的关系。因此，理想情况下，BQL 只需 $|\mathcal{A}_i|\times|\mathcal{A}_{-i}|=|\mathcal{A}|$个较小的缓冲区，就能覆盖任意 P_i 对应的所有 $(s,\ a_i)$ 对，这使经验收集非常高效。我们对此作直观说明，并分析得出 BQL 的样本复杂度与联合Q学习（3）相近，见

附录 B。

在状态-动作空间较大或连续的复杂环境中，遵循表格情形下的经验收集方式效率低下且成本高昂。在表格情形下，智能体无法在每轮交互过程中更新策略，并且每轮都需要足够的样本才能准确估计期望（7）。因此，在复杂环境中，与 IQL 一样，每个智能体 i 仅维护一个回放缓冲区$\mathcal{D}_i$，其中包含所有历史经验，并采用与 IQL 相同的 ϵ-贪心策略（不使用随机初始化的确定性策略 $\hat{}\pi_i$）。接着，我们使用神经网络 Q_i 和 Q_i^{e}来实现简化最佳可能算子。 Q_i^{e}通过最小化以下目标进行更新：

$$\mathbb{E}_{s,a_i,s',r\sim\mathcal{D}_i}\left[\left(Q_i^{\mathrm{e}}\left(s,a_i\right)-r-\gamma Q_i(s',a_i'^{*})\right)^2\right], \quad (9)$$
$$a_i'^{*}=\arg\max_{a_i'} Q_i(s',a_i').$$

并且 Q_i 通过最小化以下内容进行更新：

$$\mathbb{E}_{s,a_i\sim\mathcal{D}_i}\left[w(s,a_i)\left(Q_i\left(s,a_i\right)-\bar{Q}_i^{\mathrm{e}}(s,a_i)\right)^2\right], \quad (10)$$
$$w(s,a_i)=\begin{cases}1 & \text{if } \bar{Q}_i^{\mathrm{e}}(s,a_i)>Q_i\left(s,a_i\right)\\ \lambda & \text{else.}\end{cases}$$

$\bar{}Q_i^{\mathrm{e}}$是 Q_i^{e}的软更新目标网络。当 $\lambda=0$时，(10)等价于(8)。然而，当 $\lambda=0$时，更新中 Q_i 的正向随机噪声可能会不断累积，从而导致价值高估。因此，我们在(10)中采用加权最大值，并令 $0<\lambda<1$ 以抵消正向随机噪声。在连续动作空间中，按照 DDPG（Lillicrap 等人，2016年）的做法，我们通过最大化 $Q_i(s,\pi_i(s))$来训练策略网络 $\pi_i(s)$，以替代 $\arg\max_{a_i} Q_i(s,a_i)$。神经网络的实现总结于算法2中。

简化最佳可能算子对于神经网络实现很有意义。由于只有一个缓冲区 $\mathcal{D}_i$，我们无法执行(6)，但仍可在 $\mathcal{D}_i$ 上执行(7)和(8 $\mathcal{D}_i$。当其他智能体更新其策略时， $\mathcal{D}_i$ 中的转移概率会持续变化。如果 $\mathcal{D}_i$充分经历所有可能的转移概率，$Q_i(s,\ a_i)$ 将收敛到最大值$_{\boldsymbol{a}_{-i}} Q(s,\ a_i,\ \boldsymbol{a}_{-i})$，智能体便会学得最优联合策略。也就是说，回放缓冲区中的非平稳性不再是缺点，而是 BQL 的必要条件。

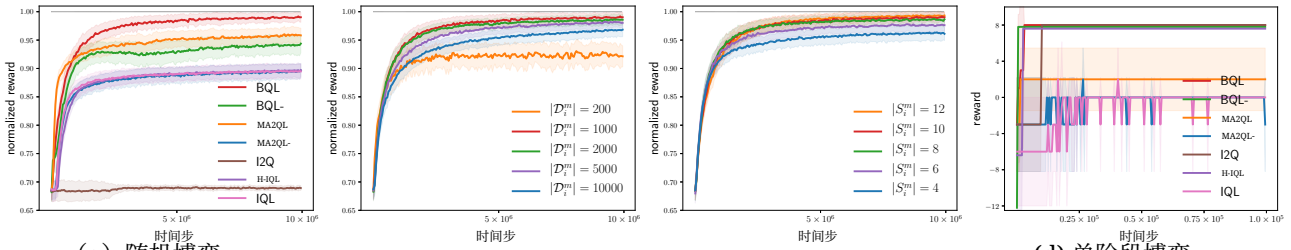
3. 相关工作

大多数现有的多智能体强化学习方法（Lowe 等，2017；Iqbal & Sha，2019；Wang 等，2020；Zhang等，2021b；Su & Lu，2022；Peng 等人，2021；Li 等，2022；Sunehag 等人，2018；Rashid 等人，2018；Son 等人，2019；Wang 等，2021a；Rashid 等人，2020）都遵循集中训练与分散执行（CTDE）的范式，即在训练过程中以集中方式获取所有智能体的信息。与这些方法不同，我们专注于完全去中心化学习，在这种学习中无法获取全局信息。最直接的去中心化方法，即，独立Q学习（Tan，1993）和独立近端策略优化（IPPO）（de Witt 等人，2020a），无法保证所学策略收敛，因为从每个智能体的角度来看，转移概率并非平稳，这是由于所有智能体同时学习策略所致。多智能体交替 Q 学习（MA2QL）（Su 等人，2022）保证收敛到纳什均衡，但当存在多个纳什均衡时，收敛后的均衡未必是最优均衡。此外，为了获得理论保证，该方法必须以同策略方式训练，且不能使用回放缓冲区，因此样本效率较低。遵循乐观估计原则，滞后 IQL（Matignon 等人，2007年）对价值惩罚设置较低的学习率。分布式 IQL（Lauer&Riedmiller，2000年）是滞后 IQL 的一种特殊情形，其较低学习率为零；该方法仅在确定性环境中保证收敛到最优解。I2Q（Jiang &Lu，2022年）让每个智能体基于理想转移概率执行独立Q学习，并且只有在确定性环境中才能学得最优策略。我们的 BQL 是首个能够在随机环境中收敛到最优联合策略的完全去中心化算法。

在下一节中，我们将 BQL 与这些 Q 学习变体进行比较（分布式 IQL 已包含在滞后 IQL 中）。与在策略算法进行比较，例如 IPPO，尤其是在完全去中心化设置中样本效率不高，因此不在本文关注范围内，相关内容留待附录讨论。带通信的去中心化方法（Zhang等人，2018；Konan等人，2021；Li和He，2020年）允许智能体根据时变通信信道与邻近智能体共享信息。然而，这些方法不遵循完全去中心化设置，因此不在本文讨论范围内。

4. 实验

在实验中，我们首先在随机生成的合作随机博弈上使用Q表测试BQL，以验证其收敛性和最优性。随后，为展示其在复杂任务上的性能，我们将使用神经网络的BQL与Q-learning变体在MPE版本微分博弈（Jiang & Lu，2022年）、多智能体 MuJoCo（Peng 等人，2021年）、SMAC（Samvelyan 等人，2019年）和GRF（Kurach 等人，2020年）上进行比较。实验涵盖完全可观测和部分可观测、确定性和随机性、离散和连续环境。由于我们考虑的是完全去中心化设置，BQL和基线方法不使用参数共享。结果以不同随机种子下的均值和标准差呈现。实验在英特尔i7-8700 CPU和NVIDIA



（a）随机博弈 (b) $|\mathcal{D}_i^m|$ (c) $|S_i^m|$ (d) 单阶段博弈

图 *1.* 合作随机博弈上的学习曲线。

GTX 1080Ti GPU上进行。每个MPE、MuJoCo和GRF任务的训练可在5小时内完成，而每个SMAC任务的训练可在20小时内完成。有关超参数的更多详细信息请参见附录E。

4.1. 随机博弈

为支持对 BQL 的理论分析，我们在随机博弈中测试 Q 表的实例化，这些博弈包含 4 个智能体、30 个状态，且具有无限时域。每个智能体的动作空间为 4，因此联合动作空间 $|\mathcal{A}| = 256$。初始状态的分布是均匀的。给定一个联合动作，每个状态都会根据转移概率转移到任意状态。每个游戏中的转移概率和奖励函数都是随机生成并固定的。我们随机生成了 20 个游戏，并在每个游戏中使用四个不同的随机种子训练智能体。

图中展示了 20 个游戏的平均归一化回报（以最优回报进行归一化）及标准差，见图 1a。由于非平稳性，IQL 无法学习最优策略。尽管采用乐观更新来缓解非平稳性，滞后 IQL（H-IQL）在随机环境中仍无法解决这一问题，其表现与 IQL 相近。在附录 A 中，我们详细分析了 H-IQL 与 BQL 之间的差异，并表明 H-IQL 是确定性环境下 BQL 的一种特殊情况。I2Q 在理想转移函数上执行 Q学习，其中下一状态确定性地取价值最高的状态，但这在随机任务中是不可能的。因此，I2Q 无法保证在随机环境中获得最优联合策略。MA2QL 保证收敛至纳什均衡，但收敛到的均衡可能并非最优均衡，因此 MA2QL 与最优策略之间存在性能差距。BQL 可以收敛至最优解，而其中的微小差距是由 Q表更新的拟合误差造成的。这验证了我们的理论分析。需要注意的是，在 Q表实例化中，MA2QL 和 BQL 使用了与 IQL 不同的经验收集方式，即探索策略和回放缓冲区。MA2QL 仅使用在策略经验，而 BQL 收集一系列小型缓冲区。然而，为了提高样本效率，在使用神经网络的复杂任务中，这两种方法必须采用与 *IQL* 相同的经验收集方式。MA2QL- 和 BQL- 分别表示采用相同经验收集方式的两种方法与 IQL 采用相同的经验收集方式。MA2QL- 在离策略经验上进行训练，受到非平稳性的影响，其表现与 IQL 相近。正如我们在第 2.4 节中所分析的，即使只使用一个缓冲区，只要这个非平稳缓冲区充分经历所有可能的转移概率，BQL 智能体也能收敛至最优解。尽管通过一个缓冲区经历所有可能的转移概率效率较低，BQL- 仍显著优于 IQL，这表明 BQL 在复杂任务中使用一个缓冲区具有潜力。

图 1b 展示了缓冲区大小 $\mathcal{D}_i^m$在轮次 m的影响。如果 $|\mathcal{D}_i^m|$太小，也就是说，200，则 $|\mathcal{D}_i^m|$ 中的经验不足以准确估算期望值（7）。如果 $|\mathcal{D}_i^m|$ 太大，也就是说，10000，则$|\mathcal{D}_i^m|$ 中的经验是冗余的，并且在总训练时间步数固定的情况下，缓冲区序列难以涵盖所有可能的转移概率。图 1c 展示了智能体执行随机初始化的确定性策略时所涉及状态数量的影响 $\hat{}\pi_i^m$以进行探索。较大的 $|S_i^m|$意味着对状态-动作对和可能转移概率的探索都更充分，从而带来更好的表现。

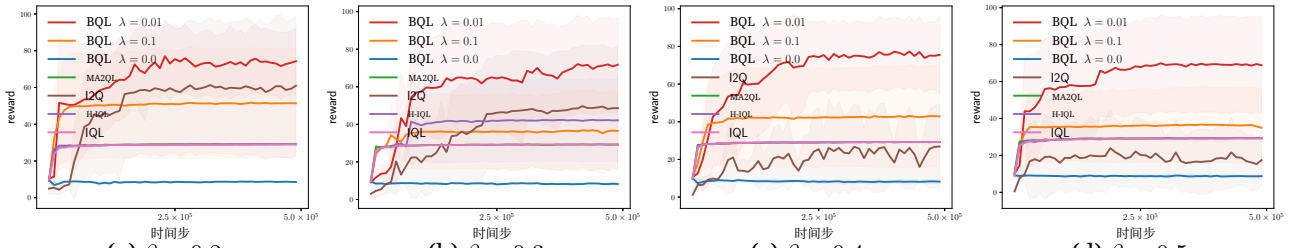
接下来，我们考虑一种在多智能体强化学习中广泛采用的单阶段博弈（Son 等人，2019年）。其中有 2 个智能体，每个智能体的动作空间为3。奖励矩阵为

$$\begin{vmatrix} a_1/a_2 & \mathcal{A}^{(1)} & \mathcal{A}^{(2)} & \mathcal{A}^{(3)} \\ \mathcal{A}^{(1)} & \mathbf{8} & -12 & -12 \\ \mathcal{A}^{(2)} & -12 & 0 & 0 \\ \mathcal{A}^{(3)} & -12 & 0 & 0 \end{vmatrix}$$

其中，奖励 8 是全局最优解，而奖励 0是次优纳什均衡。如图 1d所示，当第二个智能体的初始策略选择 $\mathcal{A}^{(2)}$ 或$\mathcal{A}^{(3)}$时，MA2QL 会收敛到次优纳什均衡。但 BQL 很容易收敛到全局最优解。

4.2. MPE

为了评估采用神经网络实现的 BQL 的有效性，我们采用3智能体的、基于 MPE 的微分博弈，该博弈用于 I2Q（Jiang & Lu，2022年），其中 3 个智能体可以在范围 $[-1, 1]$内移动。与原始的确定性版本不同，我们为其加入了随机性。在每个



(a) $\beta = 0.2$ (b) $\beta = 0.3$ (c) $\beta = 0.4$ (d) $\beta = 0.5$

图 2. 基于 MPE 的微分博弈上的学习曲线，采用不同 β。

时间步，智能体 i 执行动作 $a_i \in [-1, 1]$，智能体 i 的位置将以概率 $1-\beta$更新为 $x_i = \text{clip}(x_i + 0.1 \times a_i, -1, 1)$（即，更新后的位置限制在 $[-1, 1]$），或以概率 β更新为 $-x_i$。β 控制随机性。状态是位置向量 $\{x_1, x_2, x_3\}$。每个时间步的奖励函数为

$$r = \begin{cases} 0.5\cos(4l\pi) + 0.5 & \text{if } l \leq 0.25 \\ 0 & \text{if } 0.25 < l \leq 0.6 \\ 0.15\cos(5\pi(l - 0.8)) + 0.15 & \text{if } 0.6 < l \leq 1.0 \\ 0 & \text{if } l > 1.0 \end{cases}$$

其中 $l = \sqrt{\frac{2}{3}(x_1^2 + x_2^2 + x_3^2)}$。我们在附录的图 r 和 l 中展示了12 两者之间的关系。只有一个全局最优解（$l = 0$ 和 $r = 1$），但存在无限多个次优解（$l = 0.8$ 和 $r = 0.3$），而含有$r > 0.3$ 的狭窄区域被含有 $r = 0$的区域包围。因此，以完全去中心化的方式学习最优策略颇具挑战。每个回合包含100 个时间步，初始位置服从均匀分布。我们针对不同的随机性 β进行实验，并针对每种 β使用八个随机种子训练智能体。在连续环境中，BQL 和基线方法均基于深度确定性策略梯度构建。

如图2所示，由于转移概率非平稳，IQL 总是陷入局部最优（总奖励 ≈ 30）。在 $\beta = 0.3$的设置下，H-IQL 仅在一个随机种子下摆脱了局部最优。根据 I2Q 论文中的理论分析，随着随机性增加，I2Q 的价值估计误差会变大，这正是 I2Q 在 $\beta = 0.4$和0.5下表现不佳的原因。在神经网络实现中，MA2QL 和 BQL 使用与 IQL 相同的经验收集，因此不存在 MA2QL- 和 BQL-。MA2QL 会收敛到局部最优，因为它无法保证收敛后的均衡是全局最优解，尤其是在使用离策略数据训练时。BQL（$\lambda = 0.01$）在所有设置下都能在超过 4 个随机种子中摆脱局部最优，这证明了我们的优化目标（9）和（10）的有效性。全局最优解（总奖励 ≈ 100）与局部最优之间差距很大，导致 BQL 的方差也很大。在目标函数（10）中，λ 用于权衡执行最佳可能算子与抵消算子导致的过高估计。如图2所示，较大的 λ，即，0.1，会削弱 BQL 的效果；而过小的 λ，即，0，则会导致严重的过高估计并损害性能。

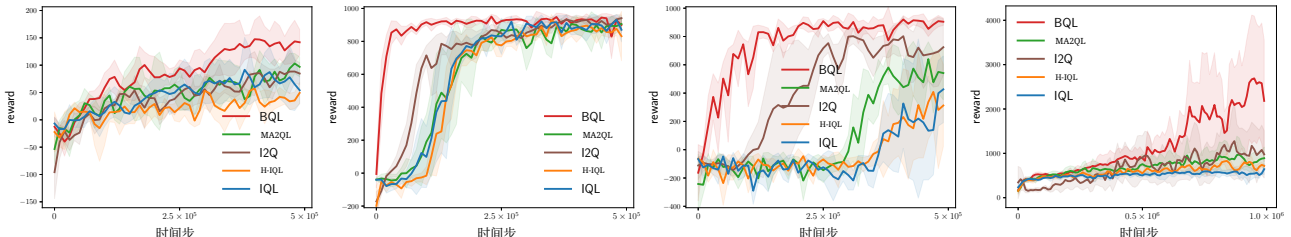
4.3. 多智能体 MuJoCo

为了评估 BQL 在部分可观测环境中的表现，我们采用多智能体 MuJoCo（Peng 等人，2021年），其中每个智能体独立控制机器人的一个或多个关节。在每项任务中，我们测试四个随机种子，并在图3中绘制学习曲线。这里，我们设定 $\lambda = 0.5$。在前三项任务中，每个智能体只能观测自身关节和躯干的状态（参数 agent obsk = 0）。BQL 获得了更高的奖励或学习速度快于基线方法，这验证了 BQL 可应用于部分可观测环境。在部分可观测环境中，BQL 基于观测$P_i(o_i'|o_i,\ a_i)$的转移概率执行，这也取决于 $\boldsymbol{\pi}_{-i}$。只有当一个观测 o_i 与一个状态 s唯一对应时，才能保证 BQL 的收敛性和最优性。已有研究证明，部分可观测马尔可夫决策过程（POMDP）中的最优性不可判定（Madani 等人，1999年），因此这并非 BQL 的局限。

在前三个任务中，我们只考虑部分可观测环境下的双智能体情况，因为当智能体数量更多时，过于有限的观测范围无法支持强大的策略。我们还在图 17 中测试了完全观测下的 3d 智能体 Humanoid。BQL 在这一多智能体任务中取得了显著的性能提升，这可以证明 BQL 具有**良好的可扩展性**。

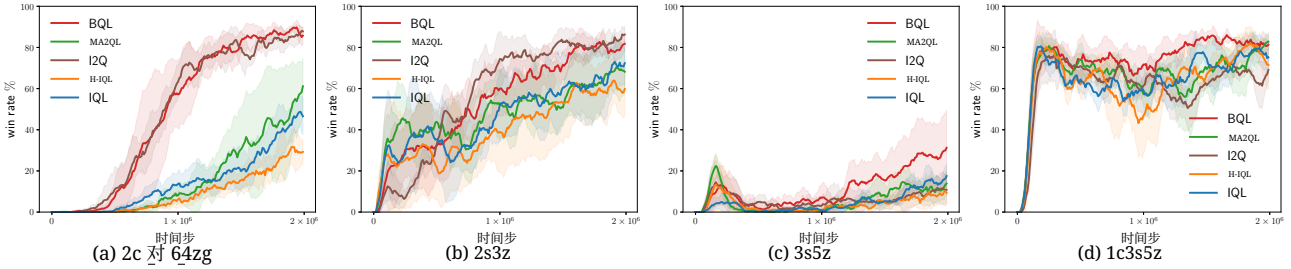
4.4. SMAC

我们还在部分可观测且随机的SMAC 任务（Samvelyan 等人，2019年），使用 SC2.4.10 版本，涵盖简单和困难地图（Yu 等人，2021年）。智能体数量在 2 到9之间。我们基于 PyMARL 的实现构建 BQL（Samvelyan 等人，2019年），并使用四个随机种子训练智能体。学习曲线如图4所示。总体而言，BQL 优于基线方法，这验证了 BQL 在高维复杂环境中也能取得性能提升



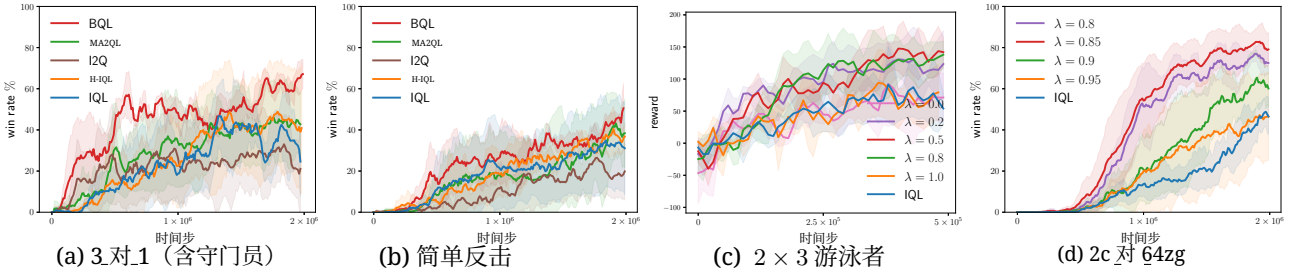
(a) 2×3 游泳者 (b) 2×4d 蚂蚁 (c) 6|2蚂蚁 (d) 17×1 人形机器人

图 *3.* 多智能体 MoJoCo 上的学习曲线。



(a) 2c 对 64zg (b) 2s3z (c) 3s5z (d) 1c3s5z

图*4.* SMAC 上的学习曲线。



(a) 3 对 1（含守门员） (b) 简单反击 (c) 2×3 游泳者 (d) 2c 对 64zg

图*5.* (a) 和 (b)：GRF 上的学习曲线。(c) 和 (d)：使用不同 λ的学习曲线。

任务。在 2c_对_64zg 中，考虑到非平稳转移概率，BQL 和 I2Q 相较于其他方法取得了显著提升。我们推测，智能体之间的相互作用在此任务中很强。

4.5. Google Research Football

Google Research Football (GRF)（Kurach等人，2020年）是一款基于物理的 3D 模拟器，智能体旨在掌握足球比赛。我们选取了两个奖励稀疏的学院任务：3 对 1（含守门员）（3 个智能体）-和简单反击（4 个智能体）。我们基于 PyMARL2（Hu等人，2021年）的实现构建 BQL，并使用四个随机种子训练智能体。尽管 I2Q 在某些 SMAC 任务中表现出与 BQL 相似的结果，但如图 5a 和 5b所示，BQL 在 GRF 中的表现优于 I2Q，因为 GRF 比 SMAC 更具随机性，而且随着随机性的增加，I2Q 的价值差距也会扩大。

4.6. 超参数 λ

我们进一步研究了 λ 在多智能体 MuJoCo 和 SMAC 中的有效性。在目标函数（10）中，λ 控制着在执行最佳可能算子与抵消该算子造成的过高估计之间的平衡。如图 5c 和 5d所示，λ 过大会削弱 BQL 的作用。当 $\lambda = 1.0$时，BQL 会退化为 IQL。过小，即，0，会导致过高估计。如果环境更复杂，例如SMAC，那么更容易出现过高估计，因此应将 λ设得较大。在 2×3 游泳者中，当 λ 落在区间 $[0.2, 0.8]$内时，BQL 能够获得性能提升，表明其对 λ具有鲁棒性。

5. 结论

我们提出最佳可能算子，并从理论上证明，如果每个智能体独立地通过该算子更新其个体状态-动作价值，那么智能体的策略将收敛至最优联合策略。随后，我们对该算子进行简化，推导出 BQL——首个能够保证在随机环境中收敛至全局最优解的去中心化多智能体强化学习算法。实验结果表明，在多种多智能体任务中，BQL 的表现优于基线方法。我们相信，BQL 可以成为完全去中心化学习的新范式。

参考文献

Achiam，J.《深度强化学习入门》。2018。

de Witt, C. S., Gupta, T., Makoviichuk, D., Makoviychuk, V., Torr, P. H., Sun, M., and Whiteson, S. 在《星际争霸多智能体挑战赛》中，独立学习是否就是你所需要的一切？*arXiv* 预印本 *arXiv:2011.09533*，2020a。

de Witt, C. S., Peng, B., Kamienny, P.-A., Torr, P., B¨ohmer, W., and Whiteson, S. 面向去中心化连续协作控制的深度多智能体强化学习。*arXiv* 预印本 *arXiv:2003.06709*，2020b。Hu, J., Jiang, S., Harding, S. A., Wu, H., and Liao, S.-w. 重新审视协作式多智能体强化学习中的实现技巧与单调性约束。*arXiv* 电子预印本，第 arXiv–2102 页，2021。Iqbal, S. and Sha, F. 面向多智能体强化学习的演员-注意力-评论家。载于国际机器学习会议（*ICML*），2019。Jiang, J. and Lu, Z. I2q：一种完全去中心化的 Q 学习算法。载于神经信息处理系统进展（*NeurIPS*），2022。Konan, S. G., Seraj, E., and Gombolay, M. 在协作式与拜占庭式去中心化团队协作中，利用互信息进行迭代理性推理。载于国际学习表征会议（*ICLR*），2021。Kurach, K., Raichuk, A., Stanczyk, P., Zajkac, M., Bachem, O., Espeholt, L., Riquelme, C., Vincent, D., Michalski, M., Bousquet, O., et al. Google Research Football：一种新型强化学习环境。载于 *AAAI* 人工智能会议论文集（*AAAI*），2020。Lauer, M. and Riedmiller, M. 一种用于协作式多智能体系统的分布式强化学习算法。载于国际机器学习会议（*ICML*），2000。Li, H. and He, H. 多智能体信赖域策略优化。*arXiv* 预印本 *arXiv:2010.07916*，2020。Li, X., Zhang, J., Bian, J., Tong, Y., and Liu, T. 面向复杂物流网络资源均衡的协作式多智能体强化学习框架。载于自主智能体与多智能体系统国际会议（*AAMAS*），2019。Li, Y., Xie, G., and Lu, Z. 多智能体策略梯度的差分优势估计。载于国际机器学习会议（*ICML*），2022。

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. 深度强化学习中的连续控制。载于国际学习表征会议 *(ICLR)*，2016年。Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mor- datch, I. 混合协作竞争环境中的多智能体演员-评论家。神经信息处理系统大会 *(NeurIPS)*，2017年。Madani, O., Hanks, S., and Condon, A. 概率规划和无限时域部分可观测马尔可夫决策问题的不可判定性。载于*AAAI/IAAI*，1999年。Matignon, L., Laurent, G. J., and Le Fort-Piat, N. 滞后Q学习：一种用于合作多智能体团队中分散式强化学习的算法。载于*IEEE/RSJ*智能机器人与系统国际会议 *(IROS)*，2007年。Peng, B., Rashid, T., Schroeder de Witt, C., Kamienny, P.-A., Torr, P., B¨ohmer, W., and Whiteson, S. FACMAC：因子化多智能体集中式策略梯度。神经信息处理系统进展 *(NeurIPS)*，2021年。Puterman, M. L. 马尔可夫决策过程：离散随机动态规划，1994年。Rashid, T., Samvelyan, M., De Witt, C. S., Farquhar, G., Foerster, J., and Whiteson, S. QMIX：用于深度多智能体强化学习的单调价值函数分解。载于国际机器学习会议 *(ICML)*，2018。Rashid, T., Farquhar, G., Peng, B., and Whiteson, S. 加权QMIX：扩展用于深度多智能体强化学习的单调价值函数分解。神经信息处理系统进展 *(NeurIPS)*，2020年。Samvelyan, M., Rashid, T., de Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G. J., Hung, C.-M., Torr, P. H. S., Foerster, J., and Whiteson, S. 星际争霸多智能体挑战赛。*arXiv* 预印本 *arXiv:1902.04043*，2019年。Son, K., Kim, D., Kang, W. J., Hostallero, D. E., and Yi, Y. QTRAN：通过变换学习因子分解以实现协作式多智能体强化学习。载于国际机器学习会议 *(ICML)*，2019年。Su, K. and Lu, Z. 散度正则化多智能体演员-评论家。载于国际机器学习会议 *(ICML)*，2022年。Su, K., Zhou, S., Gan, C., Wang, X., and Lu, Z. MA2QL：一种完全分散式多智能体的极简方法

强化学习。*arXiv*预印本 *arXiv:2209.08244*，2022年。

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zam- baldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., 等。基于团队奖励的合作式多智能体学习价值分解网络。载于国际自主智能体与多智能体系统会议 *(AAMAS)*，2018 年。

Tan, M. 多智能体强化学习：独立智能体与合作智能体。载于国际机器学习会议（*ICML*），1993。Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P. 等。《利用多智能体强化学习达到《星际争霸 II》中的宗师级水平》。《自然》，575 (7782):350–354，2019年。Wang, J., Ren, Z., Liu, T., Yu, Y. 和 Zhang, C. Qplex：双重决斗多智能体 Q 学习。载于国际学习表征会议（*ICLR*），2021a。Wang, J., Xu, W., Gu, Y., Song, W. 和 Green, T. C. 面向配电网络有功电压控制的多智能体强化学习。神经信息处理系统进展（*NeurIPS*），2021b。Wang, Y., Han, B., Wang, T., Dong, H. 和 Zhang, C. Dop：离策略多智能体分解策略梯度。载于国际学习表征会议（*ICLR*），2020年。Xu, B., Wang, Y., Wang, Z., Jia, H. 和 Lu, Z. 分层协同学习交通信号控制。载于*AAAI* 人工智能会议（*AAAI*），2021年。Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A. 和 Wu, Y. 近端策略优化在合作式多智能体游戏中的惊人效果。*arXiv* 预印本 *arXiv:2103.01955*，2021年。Zhang, K., Yang, Z., Liu, H., Zhang, T. 和 Basar, T. 联网智能体的完全分散式多智能体强化学习。载于国际机器学习会议（*ICML*），2018。Zhang, K., Yang, Z. 和 Bas¸ar, T. 多智能体强化学习：理论与算法精选综述。《强化学习与控制手册》，第 321–384 页，2021a。Zhang, T., Li, Y., Wang, C., Xie, G. 和 Lu, Z. FOP：最大熵多智能体强化学习的最优联合策略分解。载于国际机器学习会议（*ICML*），2021b。

A. 与滞后 IQL 的比较

当环境为确定性时，滞后 IQL 是 BQL 的一种特例。为了对此进行充分说明，我们重写 BQL 的损失函数

$$w(s,a_i)\left(Q_i\left(s,a_i\right)-\mathbb{E}_{\tilde{P}_i(s'|s,a_i)}\left[r+\gamma\max_{a_i'}Q_i(s',a_i')\right]\right)^2,$$

$$w(s,a_i)=\begin{cases}1 & \text{if } \mathbb{E}_{\tilde{P}_i(s'|s,a_i)}\left[r+\gamma\max_{a_i'}Q_i(s',a_i')\right]>Q_i\left(s,a_i\right)\\ \lambda & \text{else.}\end{cases}$$

如果 $\lambda=0$，BQL 的更新为

$$Q_i(s,a_i)=\max\left(Q_i(s,a_i),\mathbb{E}_{\tilde{P}_i(s'|s,a_i)}\left[r+\gamma\max_{a_i'}Q_i(s',a_i')\right]\right).$$

滞后 IQL 遵循损失函数

$$w(s,a_i)\left(Q_i\left(s,a_i\right)-r-\gamma\max_{a_i'}Q_i(s',a_i')\right)^2,$$

$$w(s,a_i)=\begin{cases}1 & \text{if } r+\gamma\max_{a_i'}Q_i(s',a_i')>Q_i\left(s,a_i\right)\\ \lambda & \text{else.}\end{cases}$$

如果 $\lambda=0$，滞后 IQL 会退化为分布式 IQL（Lauer& Riedmiller，2000年）

$$Q_i(s,a_i)=\max\left(Q_i(s,a_i),r+\gamma\max_{a_i'}Q_i(s',a_i')\right).$$

~BQL 对转移概率 $P_i(s'|s,\ a_i)$ 上的期望目标取最大值，而滞后 IQL 则对下一状态 s' 上的目标取最大值。当环境是确定性时，两者等价。然而，在随机环境中，滞后 IQL 无法保证收敛到全局最优解，因为环境并非总是转移到相同的 s'。BQL 在确定性和随机环境中都能保证达到全局最优解。

B. BQL 的效率

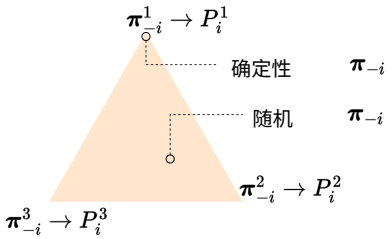


图 6. 给定一个（$s,\ a_i$）时其他智能体策略的空间 $\boldsymbol{\pi}_{-i}$。

我们将讨论为 BQL 收集回放缓冲区的效率。给定 (s,a_i) 对时其他智能体的策略空间 $\boldsymbol{\pi}_{-i}$ 是一个凸多胞体。为清晰起见，图 6 展示了一个三角形空间。每个 $\boldsymbol{\pi}_{-i}$ 对应一个 $P_i(s'|s,a_i)$。确定性策略 $\boldsymbol{\pi}_{-i}$ 位于顶点处，而多胞体的边和内部则是随机 $\boldsymbol{\pi}_{-i}$，即确定性策略的混合。由于 BQL 只考虑确定性策略，缓冲区序列只需在收集每个缓冲区 $\mathcal{D}_i^m$时采用确定性策略，覆盖所有顶点即可，因而效率很高。BQL 只需要 $|\mathcal{A}_i|\times|\mathcal{A}_{-i}|=|\mathcal{A}|$ 个小型缓冲区，这与状态空间 $|\mathcal{S}|$无关，便可满足简化最佳可能算子的要求：给定 (s,a_i) 对时，任何一个可能的 $P_i(s'|s,a_i)$ 都能在一个（理想情况下仅一个）缓冲区中找到。具体而言，需要 $|\mathcal{A}_i|$ 个缓冲区来覆盖动作空间；对于每个动作，则需要 $|\mathcal{A}_{-i}|$ 个缓冲区来覆盖转移空间。我们在图 7 中直观地展示了这一点。在 $\mathcal{D}_i^m$中的每个状态都需要 # 个样本来估计式 (7) 中的期望，因此样本复杂度

理想情况下，4 个缓冲区即可覆盖所有可能的 $P_i(s, a_i)$

$$< s^1, a_i^1, P_i^1(s^1, a_i^1) >, < s^2, a_i^1, P_i^1(s^2, a_i^1) >, < s^3, a_i^1, P_i^1(s^3, a_i^1) >$$

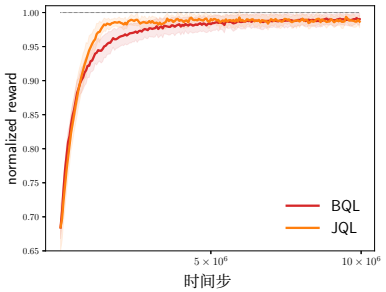
$$< s^1, a_i^1, P_i^2(s^1, a_i^1) >, < s^2, a_i^1, P_i^2(s^2, a_i^1) >, < s^3, a_i^1, P_i^2(s^3, a_i^1) >$$

$$< s^1, a_i^2, P_i^1(s^1, a_i^2) >, < s^2, a_i^2, P_i^1(s^2, a_i^2) >, < s^3, a_i^2, P_i^1(s^3, a_i^2) >$$

$$< s^1, a_i^2, P_i^2(s^1, a_i^2) >, < s^2, a_i^2, P_i^2(s^2, a_i^2) >, < s^3, a_i^2, P_i^2(s^3, a_i^2) >$$

图 7. 用于说明理想缓冲区数量的简单示例。 $|\mathcal{S}| = 3$、 $|\mathcal{A}_i| = 2$和 $|\mathcal{A}_{-i}| = 2$ 分别对应 P_i^1和 P_i^2。我们可以看到，任何 $P_i(s, a_i)$ 都能在这 4 个缓冲区中找到。

是 $O(|\mathcal{A}||S|\#)$。对于联合Q学习（3），它是在随机环境中保证收敛和最优性的已知最高效方法，每个状态-联合动作对 $(s, \boldsymbol{a})$ 需要 $\#$ 个样本来估计期望，因此样本复杂度也是 $O(|\mathcal{A}||S|\#)$。因此，就样本复杂度而言，BQL 与联合Q学习相近，这一点已在图8中通过实验证实。



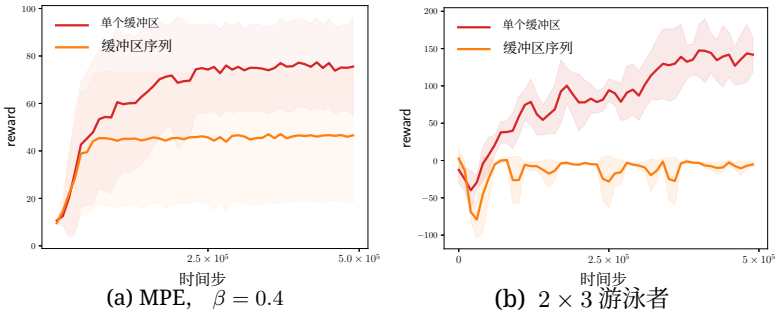
*Fi*图 *8*。 BQL 和联合Q学习（JQL）的学习曲线。BQL 的样本效率与 JQ 相近 L.

有人可能会问：“既然你获得了所有可能的转移概率，为什么不针对每种转移概率执行 IQL，并选择最高的值？”实际上，这种朴素算法也能学到最优策略，但其缓冲区收集成本远高于 BQL。朴素算法要求在一个缓冲区中找到整个状态-动作空间的任一种可能转移概率函数，这需要 $|\mathcal{A}_{-i}|^{|\mathcal{S}|}$ 个缓冲区。而训练 IQL $|\mathcal{A}_{-i}|^{|\mathcal{S}|}$ 次也非常耗费资源。BQL 只要求在一个缓冲区中找到任一状态-动作对的任一种可能转移概率，效率高得多。

然而，考虑到样本效率，采用神经网络的 BQL 仅维护一个回放缓冲区 $\mathcal{D}_i$，其中包含所有历史经验，这与 IQL 相同。P_i 在 $\mathcal{D}_i$ 中对应于其他智能体历史策略的平均值，而这是随机的。因此，为了保证最优性，理论上，采用单个缓冲区的 BQL 必须遍历几乎整个 $\boldsymbol{\pi}_{-i}$ 空间，这代价很高。如图1d所示，BQL-（采用单个缓冲区）的表现优于 IQL，但无法达到与 BQL（采用缓冲区序列）相近的结果，这表明维护单个缓冲区代价高昂，但仍然有效。在神经网络实例化中，我们在图9中展示了采用缓冲区序列的 BQL 的结果。由于样本效率的限制，缓冲区序列无法取得出色的表现；在复杂环境中，像 IQL 一样维护单个缓冲区是更好的选择。

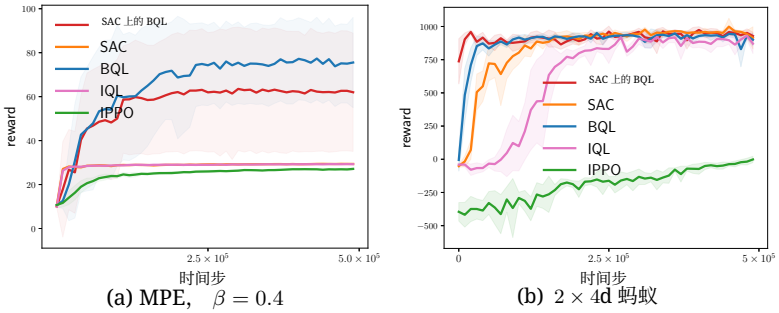
C. 其他基准算法

除了深度确定性策略梯度，BQL 也可以基于其他 Q 学习变体构建，例如，SAC。图 10 显示，BQL 在独立软演员-评论家上也能取得性能提升。独立近端策略优化（IPPO）（de Witt 等人，2020a）是一种在策略分散式多智能体强化学习基线。IPPO 不是 Q 学习方法，因此不能作为 BQL 的基础算法。在策略算法不会使用旧经验，因此样本效率较低（Achiam，2018），尤其是在完全分散式设置中



(a) MPE，$\beta = 0.4$　　(b) 2×3 游泳者

图 *9.* BQL 使用单个缓冲区和缓冲区序列。



(a) MPE，$\beta = 0.4$　　(b) 2×4d 蚂蚁

图*10*。 其他基准算法的学习曲线。

如图 10所示。因此，将离策略算法与在策略算法进行比较是不公平的。

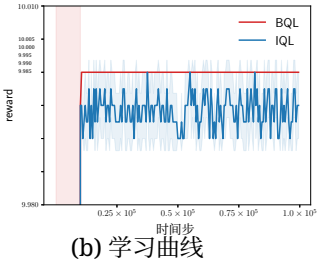
D. 多个最优联合策略

我们假设只有一个最优联合策略。当存在多个最优动作（具有最大 $Q_i(s, a_i)$）时，如果每个智能体任意选择一个最优独立动作，联合动作可能并非最优。为了解决这一问题，我们采用 I2Q（Jiang 和 Lu，2022）提出的简单技术。具体而言，我们设置一个性能容差 ε，并引入一个固定的、随机初始化的奖励函数 $\hat{r}(s, s') \in (0, (1-\gamma)\varepsilon]$。随后，所有智能体都执行 BQL，以学习由塑形后的 $\hat{} \hat{}$ 奖励 $r + \hat{r}$ 得到的 $Q_i(s, a_i)$。由于 $\hat{r} > 0, Q_i(s, a_i) > Q_i(s, a_i)$。在 $Q_i(s, a_i)$ 中，$\hat{r}$ 的最大贡献为 $(1-\gamma)\varepsilon/(1-\gamma) = \varepsilon$，$\hat{}$ 因此 r 的最小贡献为 $Q_i(s, a_i) - \varepsilon > Q_i(s, a_i) - \varepsilon$，这意味着根据 Q_i 选择动作时，最大性能下降为 $\varepsilon\hat{}$。在奖励函数 $r + \hat{r}$ 上找到多个最优联合策略的概率很小，因为 $\hat{r}(s, s')$ 是随机初始化的。因此，如果将 ε 设得足够小，BQL 就能解决具有多个最优联合策略的任务。不过，这项技术只是为了弥补理论结果中的假设而引入的。在经验上，这并非必要，因为复杂环境中通常只有一个最优联合策略。在所有实验中，我们都不对 BQL 和其他基线方法使用随机初始化的奖励函数，因此比较是公平的。

我们在一个具有两个最优联合策略的单阶段矩阵博弈中测试随机初始化的奖励函数 $(1, 2)$ 和$(2, 1)$，如图 11所示。如果智能体独立选择动作，它们可能会选择未协调的联合策略$(1, 1)$ 和 $(2, 2)$。IQL 无法收敛，但 BQL 智能体始终选择协调一致的动作，尽管最优策略与次优策略之间的价值差距非常小，这验证了随机初始化奖励的有效性。

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	9.99	**10**	0
$\mathcal{A}^{(2)}$	**10**	0	0
$\mathcal{A}^{(3)}$	0	0	0

(a) 矩阵博弈



(b) 学习曲线

图*11.* 在具有多个最优联合策略的单阶段矩阵博弈上的学习曲线。

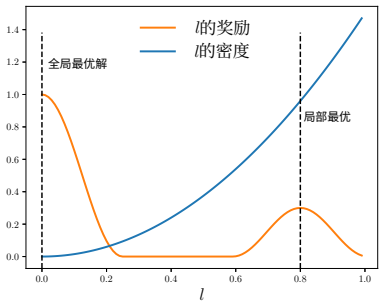


图 *12.* $l = \sqrt{\frac{2}{3}\sum_{i=0}^{3} x_i^2}$在 MPE 中的奖励曲线和密度。我们绘制了均匀状态分布的密度。全局最优解只有一个，但局部最优的密度很高。因此，去中心化智能体很容易学到局部最优策略。

E. 超参数

在基于 MPE 的微分博弈中， r 与 l 之间的关系如图12所示。

在 2×3 游泳者中，有两个智能体，每个智能体控制多智能体游泳者的 3 个关节。在 $6|2$ 蚂蚁中，有两个智能体。其中一个控制 6 个关节，另一个控制 2 个关节，依此类推。

在基于 MPE 的微分博弈和多智能体 MuJoCo 中，我们采用 SpinningUp（Achiam，2018年）实现，即 DDPG 的 SOTA 实现，并遵循 SpinningUp 中的所有超参数。折扣因子 $\gamma = 0.99$，学习率为$0。001$，使用 Adam 优化器，批量大小为100，回放缓冲区包含 5×10^5 个状态转移，隐藏单元数为256。

在 SMAC 中，我们采用 PyMARL（Samvelyan 等人，2019年）实现，并遵循 PyMARL 中的所有超参数。折扣因子 $\gamma = 0.99$，学习率为$0。0005$，使用 RMSprop 优化器，批量大小为32 个回合，回放缓冲区包含5000 个回合，隐藏单元数为64。我们采用 SMAC 的 SC2.4.10 版本。

在 GRF 中，我们采用 PyMARL2（Hu 等人，2021年）的实现，并遵循 PyMARL2 中的所有超参数。折扣因子为 $\gamma = 0.999$，学习率为 $0。0005$，使用 Adam 优化器，批量大小为 128 个回合，回放缓冲区包含2000 个回合，隐藏单元数为 256。我们使用 simple115 特征（一个概括游戏诸多方面的 115 维向量）作为观测，而不是 RGB 图像。

在基于 MPE 的微分博弈中，我们设置 $\lambda = 0.01$。在多智能体 MuJoCo 中，我们设置 $\lambda = 0.5$；在 SMAC 中，对于 2c 对 64zg，我们设置 $\lambda = 0.85$，对于其他任务设置 $\lambda = 0.8$。在 GRF 中，对于 3 对 1（含守门员），我们设置 $\lambda = 0.1$，对于简单反击设置 $\lambda = 0.4$。