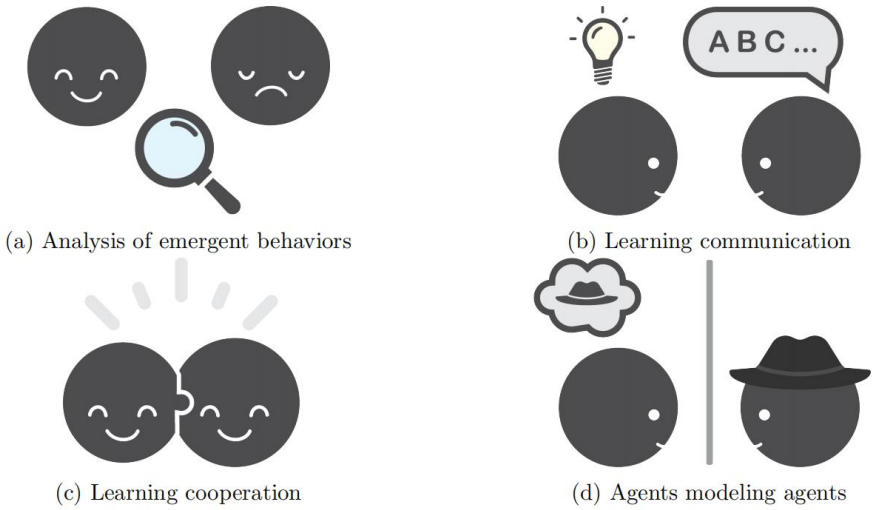
# Is multi-agent deep reinforcement learning the answer or the question? A brief survey

**Introduction：**

关于 MAL 现已经有三种方法总结：在不稳定环境中学习，智能体指导智能体，多智能体强化学习的迁移学习。




(a) Analysis of emergent behaviors (b) Learning communication

(c) Learning cooperation (d) Agents modeling agents

Figure 1: Categories of different MDRL works. (a) Analysis of emergent behaviors: evaluate single-agent DRL algorithms in multiagent scenarios. (b) Learning communication: agents learn with actions and through messages. (c) Learning cooperation: agents learn to cooperate using only actions and (local) observations. (d) Agents modeling agents: agents reason about others to fulfill a task (e.g., cooperative or competitive). For a more detailed description

多智能体难点包括多智能体信用分配，全局探索(lazy agent 问题)，相对过度泛化(学不够充分)。

**AAAI, ICML, ICLR, IJCAI and NeurIPS, and specialized conferences such as AAMAS, MARL 顶会.**

文章讲述了在多智能体环境中学习，协作学习，智能体指导智能体，知识重用下；单智能体深度强化学习下使用 MDRL 的实用性总结。

Our goal is to outline a recent and active area (i.e., MDRL), as well as to motivate future research to take advantage of the ample and existing literature in multi-agent learning.

**单智能体学习：**

**1. Q - learning & Reinforce (Monte Carlo Policy Gradient)**

MDPs 是在单智能体完全观测环境下能进行最优决策的充分模型。有一些解决完整描述 MDP 的技巧。其中一个是价值迭代，需要完整的马尔科夫状态模型描述。

**2. DRL & challenges**

PG 的主要局限性是高方差。Bootstrapping 技巧减小了方差，但增大了偏差。

深度学习有两个优点。第一点深度学习有助于状态之间泛化（学习更多状态之间的信息），提高了大规模状态空间 RL 问题的样本效率；第二点深度学习可以用来减少(或消除)手工设计特性来表示状态信息的需求，但训练数据由高度相关的序列组成，违背了独立条件。

总之，PG 方法在收敛性上比 valued-base 方法表现的更好。

减少相关性可以使用经验回放方法去保存 interactions<s, a, r, s'>.

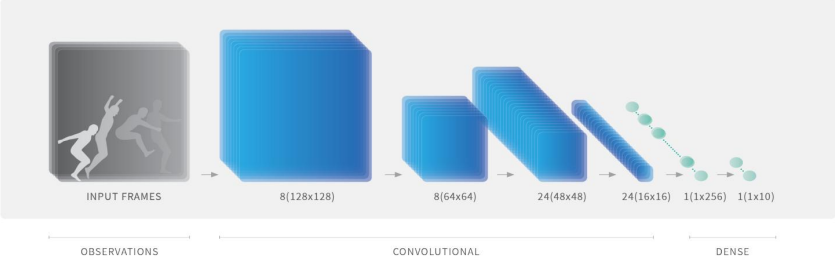



Figure 2: Deep Q-Network (DQN) [13]: Inputs are four stacked frames; the network is composed of several layers: *Convolutional* layers employ filters to learn features from high-dimensional data with a much smaller number of neurons and *Dense* layers are fully-connected layers. The last layer represents the actions the agent can take (in this case, 10 possible actions). Deep Recurrent Q-Network (DRQN) [85], which extends DQN to partially observable domains [42], is identical to this setup except the penultimate layer ($1 \times 256$ Dense layer) is replaced with a recurrent LSTM layer [86].

$$L_i(\theta_i) = \mathbb{E}_{s,a,r,s'}[(r + \gamma max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i))^2]$$

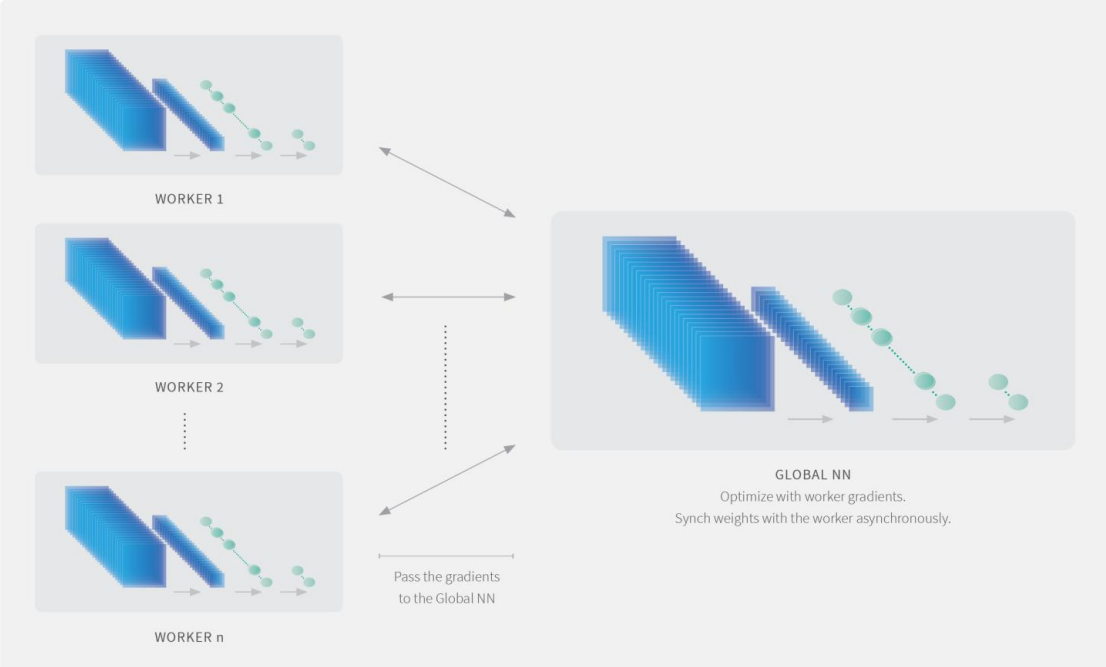

Figure 3: Representation of a DQN agent that uses an experience replay buffer [89, 90] to keep $\langle s, a, r, s' \rangle$ tuples for minibatch updates. The Q-values are parameterized with a NN and a policy is obtained by selecting (greedily) over those at every timestep.

灾难性遗忘，该现象将发生在先前由于不稳定训练导致网络效果差的学习任务中。

DDPG 是一个为解决该领域的 model-free 且 off-policy 的 AC 算法。和 DQN 使用的硬重设(直接权重复制)不同，DDPG 通过将一些采样噪声添加到其 actor 中来生成探测行为。

Asynchronous Advantage Actor-Critic (A3C)是一种使用并行异步训练方法提高效率的算法(使用 CPU 多线程)。所有智能体传递局部梯度给进行优化的全局网络同时异步的同步到每一个智能体。A3C 参数使用优势函数 A(st, at; θ v) = Q(s, a) - V (s)进行更新。

与 A3C 相比，UNREAL 使用优先级经验回放提高某些高回报的状态采样优先级。




有趣的是，PPO 和 TRPO 具有惊人的效果。但是这些方法常常偏离理论框架的预测，梯度估计与真实梯度的相关性很差，而值网络往往会对真实值函数产生不准确的预测。

Soft Actor-Critic (SAC)是最近学习随机策略的算法，两个 Q 函数(从 DDQN 得到灵感)和一个状态价值函数。

## 多智能体深度强化学习

### 1. MARL Challenges & Introduction

$\mathcal{T}$, and reward function, $\mathcal{R}$, depend on the actions $\mathcal{A} = A_1 \times ... \times A_\mathcal{N}$ of all, $\mathcal{N}$, agents, this means, $\mathcal{R} = R_1 \times ... \times R_\mathcal{N}$ and $\mathcal{T} = \mathcal{S} \times A_1 \times ... \times A_\mathcal{N}$.

Given a learning agent $i$ and using the common shorthand notation $-\boldsymbol{i} = \mathcal{N} \setminus \{i\}$ for the set of opponents, the value function now depends on the joint action $\boldsymbol{a} = (a_i, \boldsymbol{a_{-i}})$, and the joint policy $\boldsymbol{\pi}(s, \boldsymbol{a}) = \prod_j \pi_j(s, a_j)$.[6]

$$V_i^{\boldsymbol{\pi}}(s) = \sum_{\boldsymbol{a} \in \mathcal{A}} \boldsymbol{\pi}(s, \boldsymbol{a}) \sum_{s' \in \mathcal{S}} \mathcal{T}(s, a_i, \boldsymbol{a_{-i}}, s')[R_i(s, a_i, \boldsymbol{a_{-i}}, s') + \gamma V_i(s')]. \tag{4}$$

Consequently, the optimal policy is dependent on the other agents' policies,

$$\begin{aligned} &\pi_i^*(s, a_i, \boldsymbol{\pi_{-i}}) = \arg\max_{\pi_i} | V_i^{(\pi_i, \boldsymbol{\pi_{-i}})}(s) = \\ &\arg\max_{\pi_i} \sum_{\boldsymbol{a} \in \mathcal{A}} \pi_i(s, a_i) \boldsymbol{\pi_{-i}}(s, \boldsymbol{a_{-i}}) \sum_{s' \in \mathcal{S}} \mathcal{T}(s, a_i, \boldsymbol{a_{-i}}, s')[R_i(s, a_i, \boldsymbol{a_{-i}}, s') + \gamma V_i^{(\pi_i, \boldsymbol{\pi_{-i}})}(s')]. \end{aligned} \tag{5}$$

在对抗环境中(如零和博弈)，最优决策可以使用 min-max Q-learning 对抗模糊对手。

**有些算法能在强假设下将其他的智能体收敛到最优行为？？** e.g., Nash Q-learning and Friend-or-Foe Q-learning.

最近 MDRLR 的工作已经能处理一些高维的任务，并更注重于收敛性的保证。

有些优势函数使用反事实 baseline 进行衡量。这是独立 RL 算法在非完全信息的零和博弈中提供了新的收敛性质。

注意到，有些多智能体算法在并不注重收敛性下，学习对抗多级对手也有好的效果

## 2. MDRL 分类

我们从先前总结中提出了 4 种分类。紧急行为分析主要关注于分析和评估深度强化学习算法。在这个分类下我们主要分析三种行为，协作，竞争和混合型协作竞争行为。

## 3. 紧急行为

Table 1: These papers analyze *emergent behaviors* in MDRL. Learning type is either value-based (VB) or policy gradient (PG). Setting where experiments were performed: cooperative (CO), competitive (CMP) or mixed. A detailed description is given in Section 3.3.

| Work | Summary | Learning | Setting |
|---|---|---|---|
| Tampuu et al. [155] | Train DQN agents to play Pong. | VB | CO&CMP |
| Leibo et al. [156] | Train DQN agents to play sequential social dilemmas. | VB | Mixed |
| Lerer and Peysakhovich [176] | Propose DRL agents able to cooperate in social dilemmas. | VB | Mixed |
| Leibo et al. [159] | Propose Malthusian reinforcement learning which extends self-play to population dynamics. | VB | Mixed |
| Bansal et al. [158] | Train PPO agents in competitive MuJoCo scenarios. | PG | CMP |
| Raghu et al. [157] | Train PPO, A3C, and DQN agents in attacker-defender games. | VB, PG | CMP |
| Lazaridou et al. [161] | Train agents represented with NN to learn a communication language. | PG | CO |
| Mordatch and Abbeel [160] | Learn communication with an end-to-end differentiable model to train with backpropagation. | PG | CO |

同时在序列社交困境（满足一定不等式的马尔科夫博弈）的大背景下学习一个独立的 DQN。

Tit-for-Tat (TFT) strategy 维护了协作。为构建智能体算法使用自我对局和两种奖赏方法：私人奖赏和协作奖赏。

通常使用自我对局容易遗忘先验经验，因此使用马尔萨斯强化学习作为群体动态下的自我对局的拓展。**Malthusian reinforcement learning???**

他们使用 PPO 训练独立学习的智能体，同时进行两种主要的修改以处理多智能体环境的问题。

第一，他们使用探索奖赏，通过给环境的竞争奖赏分配更多的权重而将探索奖赏退火。

第二种是通过对旧版本的对手策略进行采样，而不是用最近的版本。

Agent 学习一种应急语言去通信，去完成任务。分析语义属性以自主创建通讯协议。

## 4. Communication

Table 2: These papers propose algorithms for *learning communication*. Learning type is either value-based ( or policy gradient (PG). Setting were experiments were performed: cooperative (CO) or mixed. A more det description is given in Section 3.4.

| Algorithm | Summary | Learning | Setting |
|---|---|---|---|
| Lazaridou et al. [161] | Train agents represented with NN to learn a communication language. | PG | CO |
| Mordatch and Abbeel [160] | Learn communication with an end-to-end differentiable model to train with backpropagation. | PG | CO |
| RIAL [162] | Use a single network (parameter sharing) to train agents that take environmental and communication actions. | VB | CO |
| DIAL [162] | Use gradient sharing during learning and communication actions during execution. | VB | CO |
| CommNet [163] | Use a continuous vector channel for communication on a single network. | PG | CO |
| BiCNet [164] | Use the actor-critic paradigm where communication occurs in the latent space. | PG | Mixed |
| MD-MADDPG [165] | Use of a shared memory as a means to multiagent communication. | PG | CO |
| MADDPG-MD [177] | Extend dropout technique to robustify communication when applied in multiagent scenarios with direct communication. | PG | CO |

通信是一组局部可观测环境下的协作智能体的功能之一。Reinforced Inter-Agent Learning (RIAL) and Differentiable Inter-Agent Learning (DIAL)是两种使用深度学习网络学习

通信的方法。用于在下一个时间戳和其他智能体通信。

RIAL 基于 DRQN，同时使用参数分享的概念，比如使用一个单一网络用于给所有的智能体参数分享。相比之下，DIAL 在学习时直接通过信道将梯度通信，同时在执行时将信息分解并映射为一组通信动作。

在 MD(memory-driven)-MADDPG 中，智能体使用共享的 memory 作为信道。

直接使用信息通信是被允许的。在这种情况下，其他智能体的信息有时会在训练中被忽略，因此推荐 the Message-Dropout MADDPG algorithm.

多智能体双向协调网络，在隐藏层中将通信代替了天赋空间。Multi-agent Bidirectionally Coordinated Network (BiCNet)

## 5. Cooperation

Table 3: These papers aim to *learn cooperation*. Learning type is either value-based (VB) or policy gradient (PG). Setting where experiments were performed: cooperative (CO), competitive (CMP) or mixed. A more detailed description is given in Section 3.5.

| Algorithm | Summary | Learning | Setting |
|---|---|---|---|
| Lerer and Peysakhovich [176] | Propose DRL agents able to cooperate in social dilemmas. | VB | Mixed |
| MD-MADDPG [165] | Use of a shared memory as a means to multiagent communication. | PG | CO |
| MADDPG-MD [177] | Extend dropout technique to robustify communication when applied in multiagent scenarios with direct communication. | PG | CO |
| RIAL [162] | Use a single network (parameter sharing) to train agents that take environmental and communication actions. | VB | CO |
| DIAL [162] | Use gradient sharing during learning and communication actions during execution. | VB | CO |
| DCH/PSRO [172] | Policies can overfit to opponents: better compute approximate best responses to a mixture of policies. | VB | CO & CMP |
| Fingerprints [168] | Deal with ER problems in MDRL by conditioning the value function on a fingerprint that disambiguates the age of the sampled data. | VB | CO |
| Lenient-DQN [35] | Achieve cooperation by leniency, optimism in the value function by forgiving suboptimal (low-rewards) actions. | VB | CO |
| Hysteretic-DRQN [166] | Achieve cooperation by using two learning rates, depending on the updated values together with multitask learn- | VB | CO |
| WDDQN [178] | Achieve cooperation by leniency, weighted double estimators, and a modified prioritized experience replay buffer. | VB | CO |
| FTW [179] | Agents act in a mixed environment (composed of teammates and opponents), it proposes a two-level architecture and population-based learning. | PG | Mixed |
| VDN [180] | Decompose the team action-value function into pieces across agents, where the pieces can be easily added. | VB | Mixed |
| QMIX [181] | Decompose the team action-value function together with a mixing network that can recombine them. | VB | Mixed |
| COMA [167] | Use a centralized critic and a counter-factual advantage function based on solving the multiagent credit assignment. | PG | Mixed |
| PS-DQN, PS-TRPO, PS-A3C [182] | Propose parameter sharing for learning cooperative tasks. | VB, PG | CO |
| MADDPG [63] | Use an actor-critic approach where the critic is augmented with information from other agents, the actions of all agents. | PG | Mixed |

他们的解决方案是在经验回放元组中添加信息，这有助于消除经验回放元组中采样数据的年龄歧义。减少经验回放的遗忘性。

多智能体 Importance Sampling 增加了采用共有动作的概率，所以当元组稍后被采样以进行训练时，Importance sampling 的相关性可以被计算。

**多智能体的指纹增加了其他智能体策略的估计。Multi-agent Fingerprints which adds the estimate (i.e., fingerprint) of other agents' policies.？？**

LDQN：宽容处理技巧的提出是为了克服一种叫做相对过度泛化(就是陷入局部最优)的

问题。保持宽容的心态以减少不协调噪声带来的影响。将最优化曲线的噪声减缓，让曲线更光滑以减少陷入局部最优的可能。

随着时间的推移，考虑到观察-动作对出现的频率，学习者对会降低其效用值的更新，因此对于更新就不会那么宽容了。

Weighted Double Deep Q-Network (WDDQN)基于两个近似估计网络。

在不同时间尺度下的 2 层分级式循环神经网络表达。

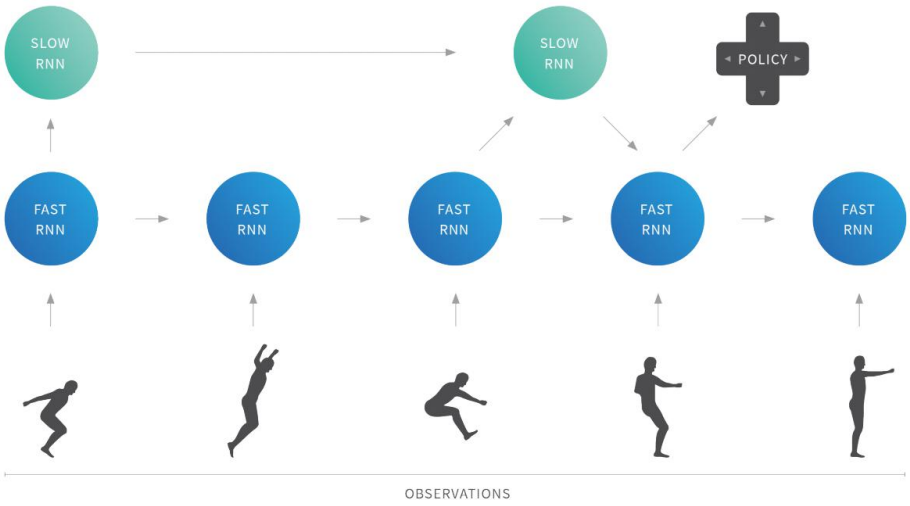



Figure 5: A schematic view of the architecture used in FTW (For the Win) [179]: two unrolled recurrent neural networks (RNNs) operate at different time-scales, the idea is that the *Slow RNN* helps with long term temporal correlations. Observations are latent space output of some convolutional neural network to learn non-linear features. Feudal Networks [229] is another work in single-agent DRL that also maintains a multi-time scale hierarchy where the slower network sets the goal, and the faster network tries to achieve them. Fedual Networks were in turn, inspired by early work in RL which proposed a hierarchy of Q-learners [230, 231].

QMIX 依赖于因式分解的思想，而不是求和，QMIX 构建了一个混合的网络结构将不同的局部值用非线性方法组合，以表达单调价值函数。使用 MDRL 方法进行价值函数的因式分解。

## 6. 智能体指导智能体

Table 4: These papers consider *agents modeling agents*. Learning type is either value-based (VB) or policy gradient (PG). Setting where experiments were performed: cooperative (CO), competitive (CMP) or mixed. A more detailed description is given in Section 3.6.

| Algorithm | Summary | Learning | Setting |
|---|---|---|---|
| MADDPG [63] | Use an actor-critic approach where the critic is augmented with information from other agents, the actions of all agents. | PG | Mixed |
| DRON [169] | Have a network to infer the opponent behavior together with the standard DQN architecture. | VB | Mixed |
| DPIQN, DPIRQN [171] | Learn policy features from raw observations that represent high-level opponent behaviors via auxiliary tasks. | VB | Mixed |
| SOM [170] | Assume the reward function depends on a hidden goal of both agents and then use an agent's own policy to infer the goal of the other agent. | PG | Mixed |
| NFSP [173] | Compute approximate Nash equilibria via self-play and two neural networks. | VB | CMP |
| PSRO/DCH [172] | Policies can overfit to opponents: better compute approximate best responses to a mixture of policies. | PG | CO & CMP |
| M3DDPG [183] | Extend MADDPG with minimax objective to robustify the learned policy. | PG | Mixed |
| LOLA [64] | Use a learning rule where the agent accounts for the parameter update of other agents to maximize its own reward. | PG | Mixed |
| ToMnet [174] | Use an architecture for end-to-end learning and inference of diverse opponent types. | PG | Mixed |
| Deep Bayes-ToMoP [175] | Best respond to opponents using Bayesian policy reuse, theory of mind, and deep networks. | VB | CMP |
| Deep BPR+ [184] | Bayesian policy reuse and policy distillation to quickly best respond to opponents. | VB | CO & CMP |

智能体的一个重要的能力就是对其他智能体行为进行推理以构建知道其他智能体的模型。

早些使用深度学习网络进行指导智能体的方法有 DRON。一个用于估计 Q 价值，另一个用于表达对手的策略。作者推荐使用不同的专家网络结合进行 Q 价值估计，一般是每一个专家网络负责一个对手的策略。

基于类型的博弈论推理：

Deep Policy Inference Q-Network (DPIQN)和他的循环网络版本，DPIRQN 直接从其他智能体的未处理观测学习策略特征。设置辅助任务以学习对手策略，辅助任务通过计算辅助 loss 对 loss 函数进行修改。辅助 loss 为在估计对手策略和对手实际策略(one-hot action vector)之间的交叉熵 loss 。

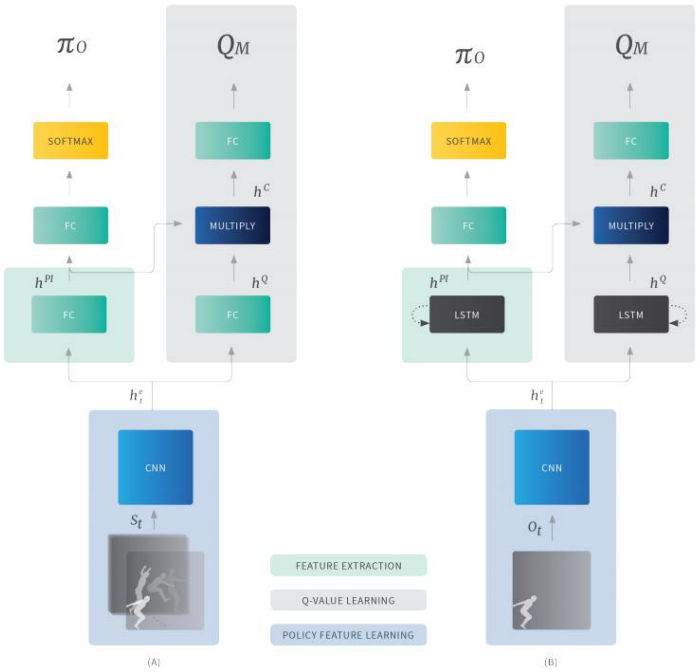



Figure 6: (a) Deep Policy Inference Q-Network: receives four stacked frames as input (similar to DQN, see Figure 2). (b) Deep Policy Inference Recurrent Q-Network: receives one frame as input and has an LSTM layer instead of a fully connected layer (FC). Both approaches [171] condition the $Q_M$ value outputs on the policy features, $h^{PI}$, which are also used to learn the opponent policy $\pi_o$.

作者使用适应性训练进程调整注意力(loss 函数的一个权重)以强调学习对手的策略特征或者是各个智能体的 Q 价值。

Self Other Modeling (SOM)提出了一种不同的方法，那就是使用智能体自己的策略作为预测对手动作的方法。SOM 使用两个网络，一个用作计算智能体的个体策略，另一个用于推断对手的目标。

可以从博弈论中的东西得到一些启发方法。The (N)FSP 模型是 PSRO 的泛化拓展版。Policy-Space Response Oracles (PSRO)

They extended the MADDPG algorithm to Mini-max Multi-agent Deep Deterministic Policy Gradients (M3DDPG)基于最糟糕的模式，所有的智能体互相竞争来更新策略的。

然而，他们并没有明确地考虑到其他 agent 的预期学习，而这正是另一种学习目标，即学习对手意识。Learning with Opponent-Learning Awareness (LOLA)学习对手意识。

因此，具有学习对手意识直接认定其他智能体会在策略学习时最大化奖赏，其中一种具有学习对手意识的方法就是接收对手的参数。

ToMnet 是一个由三个网络组成的结构：(i)学习历史信息的特征网络 (ii) a 一个精神状态

网络，它获取特征的输出和最近的 trajectory (iii)一个采用当前状态和其他网络的输出作为输入的预测网络。

Deep Bayesian Theory of Mind Policy (Bayes-ToMoP)另一个从心智理论中获得灵感的算法。

BPR+的局限性由自我对局中效果较差，因此 Deep Bayes-ToMoP 使用心智理论提供一个可以产生最优行为的高级推理策略。

**Implement:**

This section aims to provide directions to promote fruitful cooperation between sub-communities.

**1. 如何成功**

避免深度学习健忘症：在 MDRL 中采样

1. 处理独立学习者的不稳定问题

Hyper-Q 计算其他智能体混合策略的值，同时将该信息包含在状态表示中，有效地将学习问题转化为固定问题。

2. 解决多智能体的信用分配问题

信用分配目标在于准确的抓住每个智能体对于系统全局行为的贡献，推荐的方法是使用使用夹紧操作 a clamping operation，相当于将该智能体从队伍中移除，然后查看全局行为的变化。

3. 多任务学习

精炼技巧，大致就是将一个大型模型转换为小的模型，原先是一个监督学习和模型压缩的一个概念。策略精炼技巧被用于训练许多小网络，然后将其合并为任务特定的单一网络。

4. 辅助任务

一种可以考虑的辅助任务就是指导其他智能体的行为，在 MDRL 设定中有 DPIQN and DRPIQN。

5. 经验回放

经验回放方法加快信用分配传播的进程。

6. 两个近似估计网络

DDQN 使用两个近似估计网络，将动作选择和估计分开来，增强效果解决价值函数过度估计问题。

**2. 如何学习新的东西**

1. MDRL 的经验回放方法

在经验回放元组中添加有助于消除样本数据年龄歧义的信息是许多工作中采用的解决方案，无论是 value-base 方法还是 PG 方法。

2. 集中学习和分散执行

学习中添加额外信息以达到集中学习的目的，包括全局状态，动作或者奖赏。执行时拆解全局信息。

3. 参数共享

4. 循环网络

原来的网络面临学习过程中梯度消失的问题，因此使得它们对于长期记忆的遗忘问题，而 RNN 变种 LSTM 或者门控循环单元能处理这种问题。

Feudal Networks 提出一种分级方法，不同时间尺度的多个 LSTM 网络。比如对于每个 LSTM 网络，输入不同的观测以创建一个时间的分级结构，以便更好地解决 RL 问题的长期

信用分配问题。

5. MAL 的过拟合

解决方法是学习一组较优策略，从中学习一个策略，或者是，学习它们的混合策略。另一种方法是 robustify algorithms。

**3. MARL Benchmark**

**4. 未来挑战**

1. 复现模型，令人不安的趋势和消极的结果
2. 实施的挑战和超参数调优，额外的非平凡的优化-这些有时是必要的算法，以实现良好的性能。
3. 计算资源
4. 奥卡姆剃刀和烧烛分析

**5. 开放问题**

1. 对稀疏和延迟奖赏的挑战

为处理这个问题，最近的 MDRL 方法在每一步都应用密集奖赏，以允许智能体可以学习基本运动技能，然后随着时间的推移，减少这些密集的奖励，以支持环境奖励。人工增加立即奖赏积累学习。FTW 通过 2 层分级式优化以学习智能体自己内部的奖赏。

2. 对自我对局的角色位置
3. MDRL 组合性算法的挑战

搜索结合的 RL 方法依然有许多挑战，包括多智能体下搜索指数增长的动作空间，可以考虑使用搜索并行。

**Conclusion:**

首先，我们将最近的工作分为 4 类不同的话题：紧急行为，学习通信，学习协作，智能体指导智能体。

其次，我们举例了许多关键部分(包括经验回放和不同的奖赏)以适应 MDR 领域的 RL 和 MAL 方法。