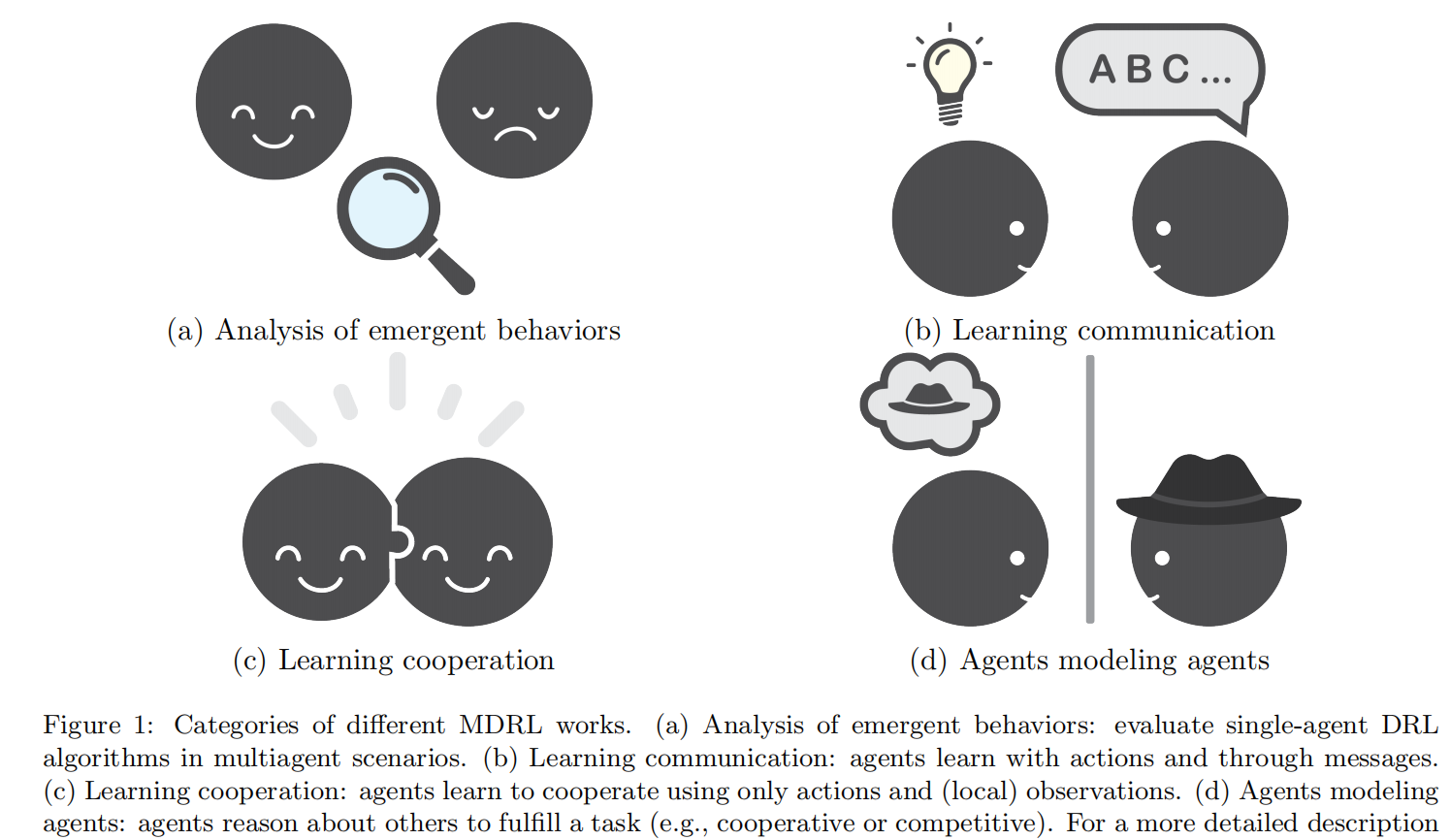
Is multi-agent deep reinforcement learning

the answer or the question? A brief survey

**Introduction：**

Three surveys related to MAL have been published: learning in non-stationary environments, agents modeling agents, and transfer learning in multi-agent RL.



The difficulties of multi-agent problem include multi-agent credit assignment, global exploration, and relative over-generalization.

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

This article contributes to the state of the art with a brief survey of the current works in MDRL in an effort to complement existing surveys on multi-agent learning, cooperative learning, agents modeling agents, knowledge reuse in multi-agent RL, and (single-agent) deep reinforcement learning.

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.

**Single agent learning：**

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

MDPs are adequate models to obtain optimal decisions in single agent fully observable environments. There are different techniques for solving MDPs assuming a complete description of all its elements. One of the most common techniques is **the value iteration algorithm,** which requires a complete and accurate representation of states, actions, rewards, and transitions.

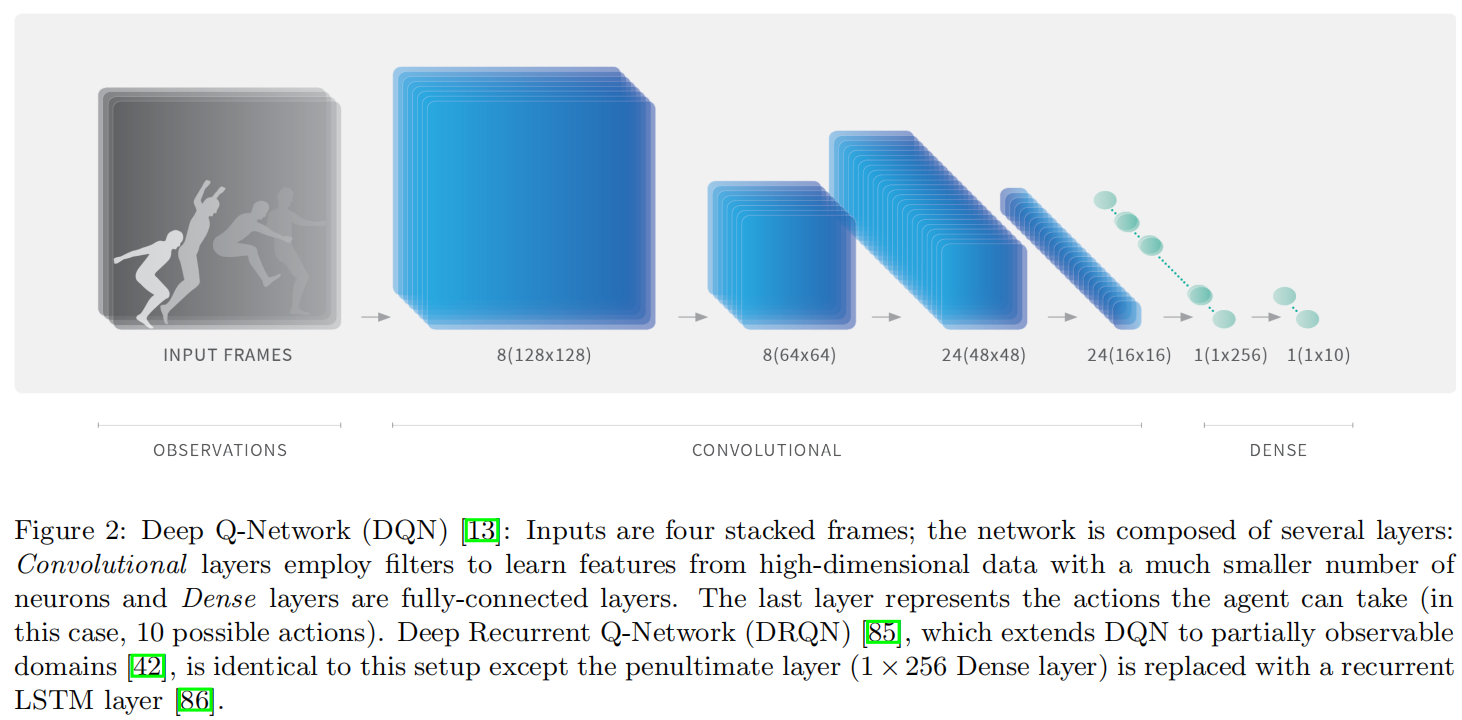
1. **DRL & challenges**

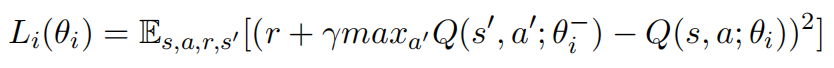
A main limitation is that policy gradient methods can have high variance. Bootstrapping reduces the variance but introduces bias.

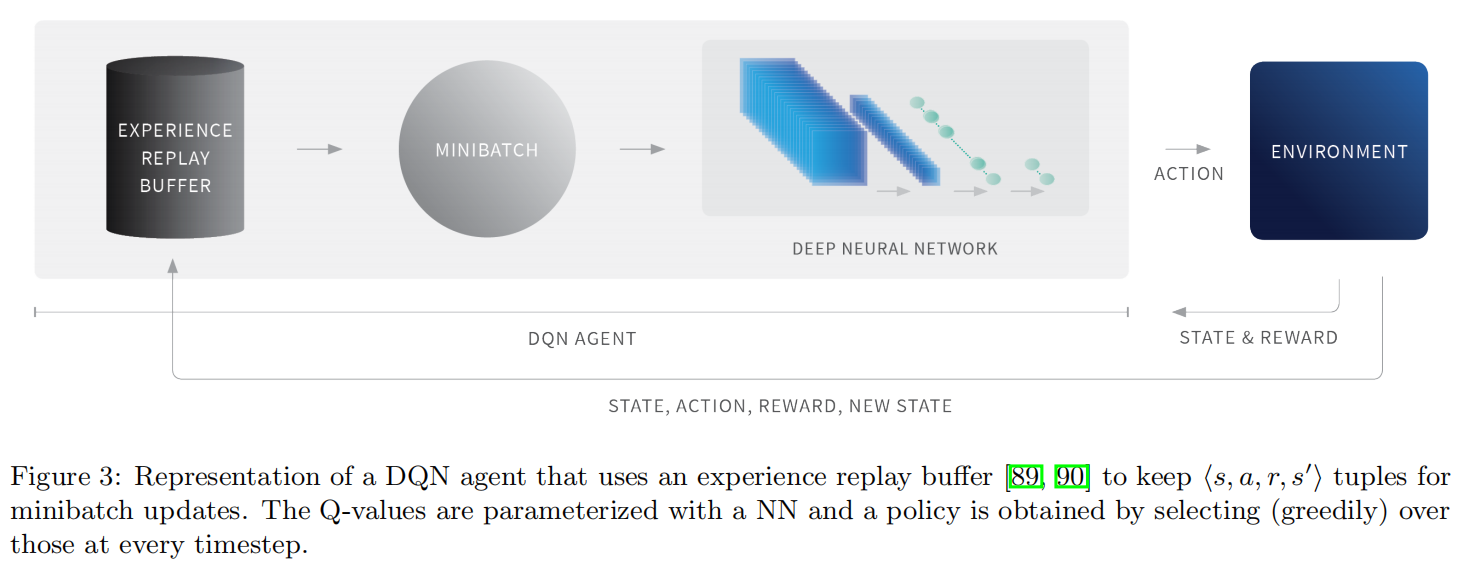
This has two advantages, first, deep learning helps to generalize across states improving the sample efficiency for large state-space RL problems. Second, deep learning can be used to reduce (or eliminate) the need for manually designing features to represent state information. Training data consists of highly correlated sequential agent-environment interactions, which violates the independence condition.

In general, stronger convergence guarantees are available for policy-gradient methods than for value-based methods.

Maintaining an experience replay (ER) buffer to store interactions<s, a, r, s’>.





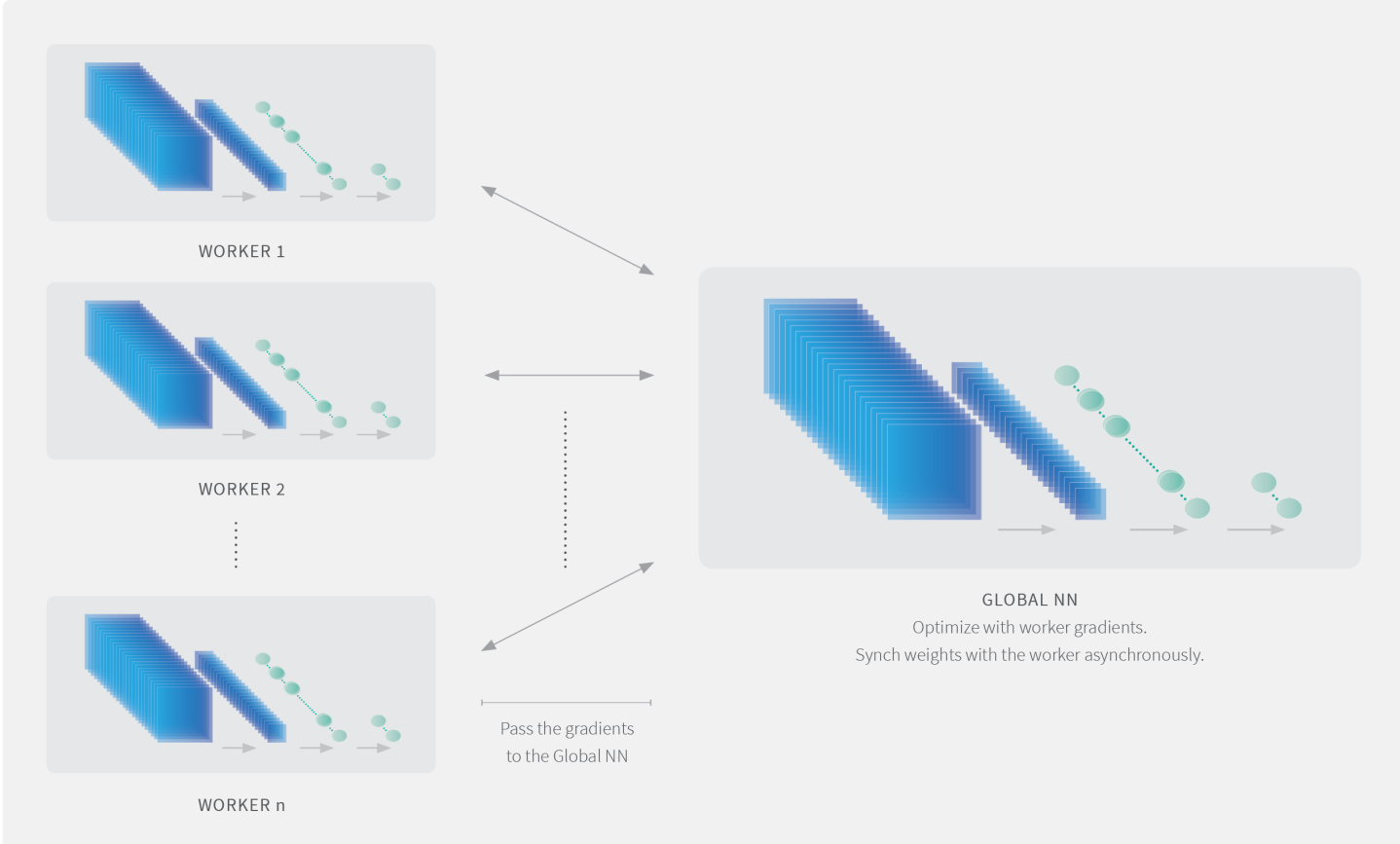


Catastrophic forgetting, this occurs when the trained neural network performs poorly on previously learned tasks due to a non-stationary training distribution.

DDPG is a model-free off-policy actor-critic algorithm for such domains. In contrast to the hard reset (direct weight copy) used in DQN. Given the off-policy nature, DDPG generates exploratory behavior by adding sampled noise from some noise processes to its actor policy.

Asynchronous Advantage Actor-Critic (A3C) [93] is an algorithm that employs a parallelize asynchronous training scheme (using multiple CPU threads) for efficiency. All the workers pass their computed local gradients to a global NN which performs the optimization and synchronizes with the workers asynchronously. A3C parameters are updated using the advantage function A(st, at; θv) = Q(s, a) - V (s).

In contrast to A3C, UNREAL uses a prioritized ER buffer, in which transitions with positive reward are given higher probability of being sampled.

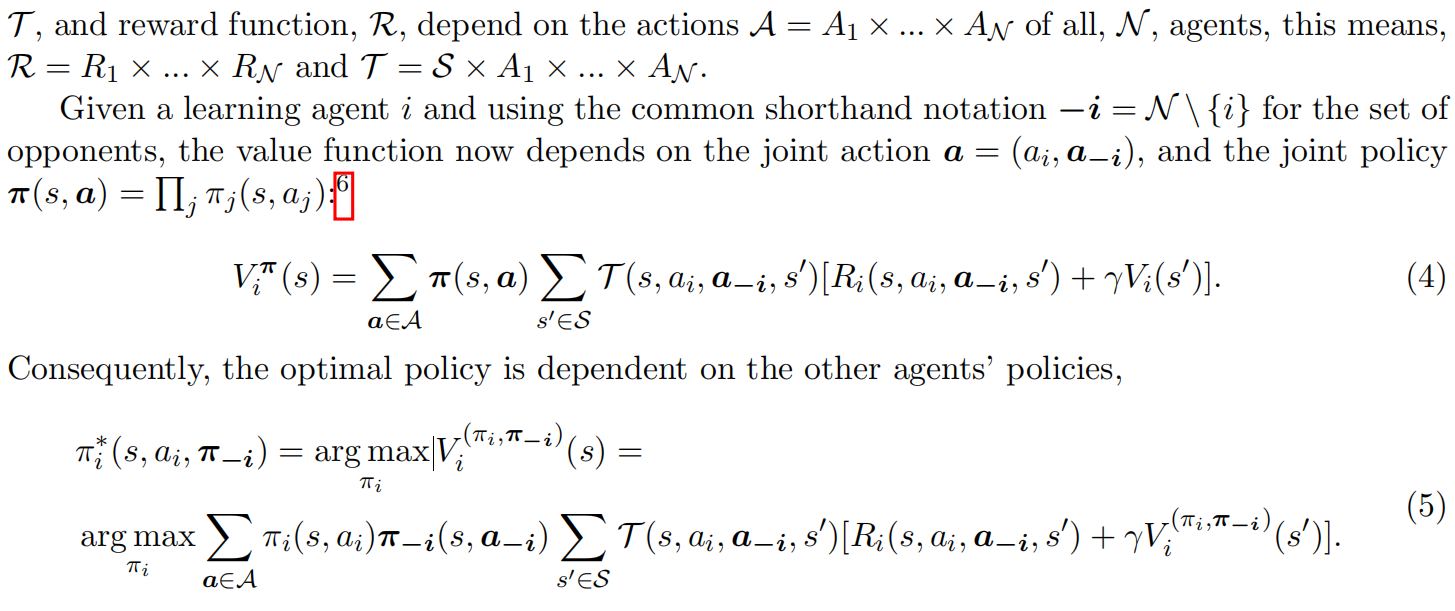


Interestingly, a recent work studying PPO and TRPO arrived at the surprising conclusion that these methods often deviate from what the theoretical framework would predict: gradient estimates are poorly correlated with the true gradient and value networks tend to produce inaccurate predictions for the true value function.

Soft Actor-Critic (SAC) is a recent algorithm that concurrently learns a stochastic policy, two Q-functions (taking inspiration from Double Q-learning) and a value function.

**Deep RL：For Multi-Agent: MADRL**

1. **MARL Challenges & Introduction**



In adversarial environments (zero-sum games) an optimal play can be guaranteed against an arbitrary opponent, i.e., Mini-max Q-learning.

Strong assumptions need be made about other agents to guarantee convergence to optimal behavior, e.g., Nash Q-learning and Friend-or-Foe Q-learning.

Recent work on MDRL have addressed scalability and have focused significantly less on convergence guarantees.

The advantage values are scaled counter-factual regrets. This lead to new convergence properties of independent RL algorithms in zero-sum games with imperfect information.

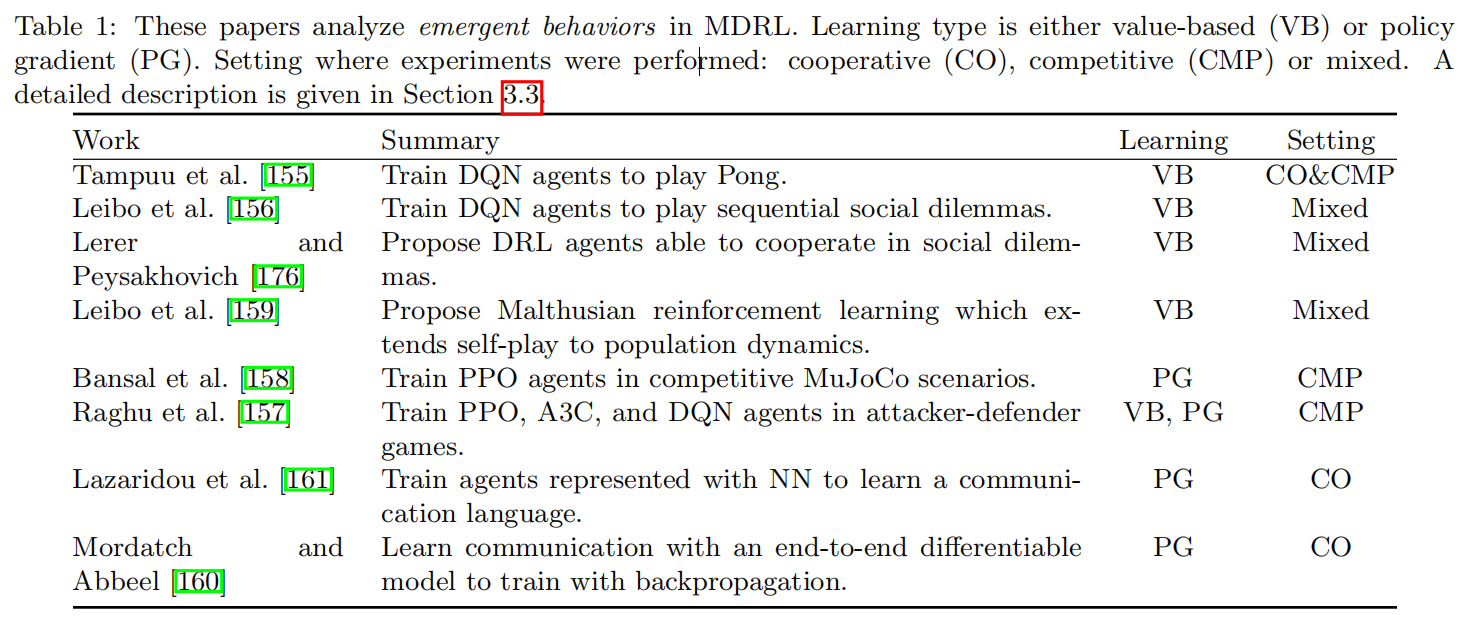
Note that instead of convergence, some MAL algorithms have proved learning a best response against classes of opponents.

1. **MDRL Categorization**

We propose 4 categories which take inspiration from previous surveys. Analysis of emergent behaviors their main focus is to analyze and evaluate DRL algorithms. In this category we found works which analyze behaviors in the three major settings: cooperative, competitive and mixed scenarios.

Cooperative, in this category we found works which analyze behaviors in the three major settings: cooperative, competitive and mixed scenarios.

1. **Emergent Behavior**



Meanwhile studied independent DQNs in the context of **sequential social dilemmas???**: a Markov game that satisfies certain inequalities.

Tit-for-Tat (TFT) strategy maintain cooperation. To construct the agents they used self-play and two reward schemes: selfish and cooperative.

Despite its common usage self-play can be brittle to forgetting past knowledge. **Malthusian reinforcement learning???** as an extension of self-play to population dynamics.

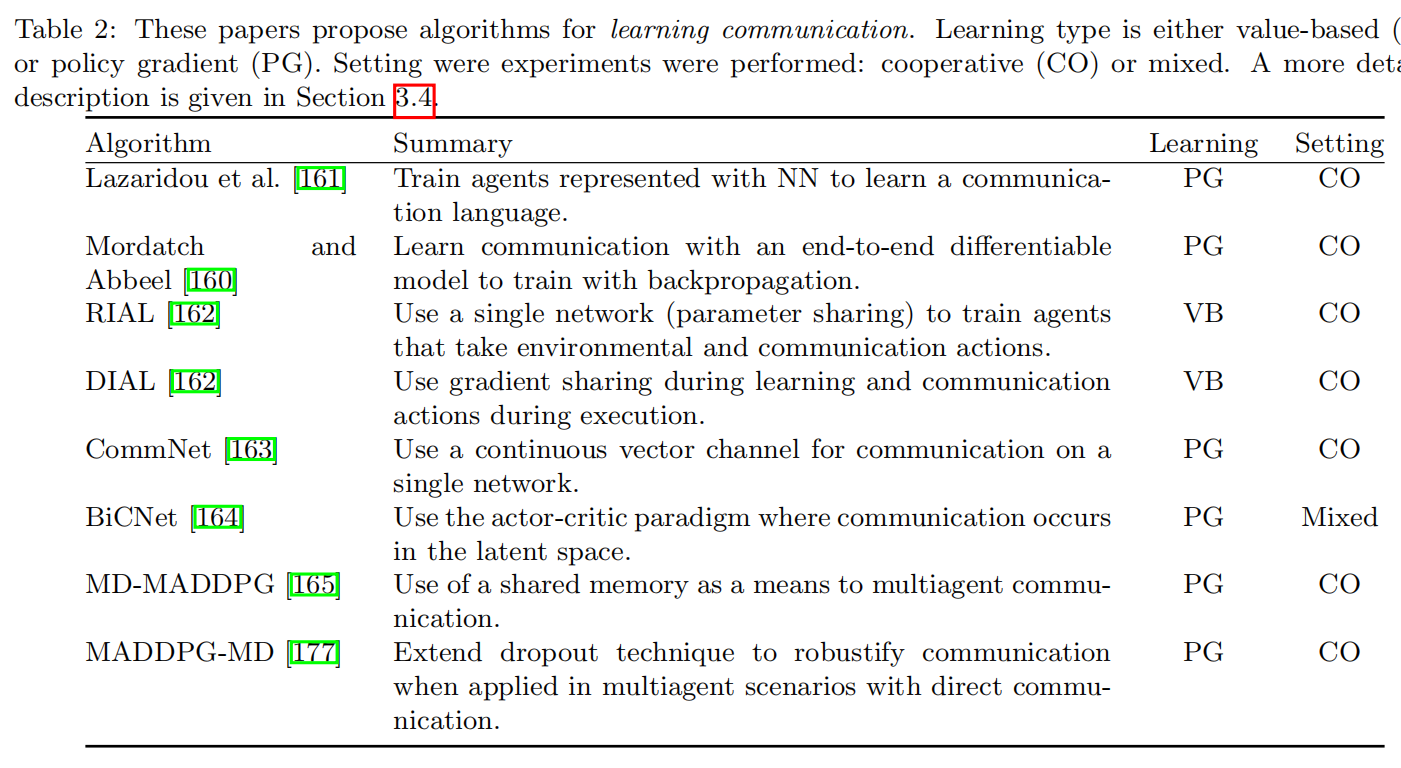
They trained independent learning agents with PPO and incorporated two main modifications to deal with the MAL nature of the problem.

First, they used exploration reward, this type of reward is annealed through time giving more weight to the environmental (competitive) reward. 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。

The second was opponent sampling which maintains a pool of older versions of the opponent to sample from, in contrast to using the most recent version.

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。

1. **Communication**



Communication is a set of cooperative agents in a partially observable environment. Reinforced Inter-Agent Learning (RIAL) and Differentiable Inter-Agent Learning (DIAL) are two methods using deep networks to learn to communicate. A message to communicate to other agents in the next timestep.

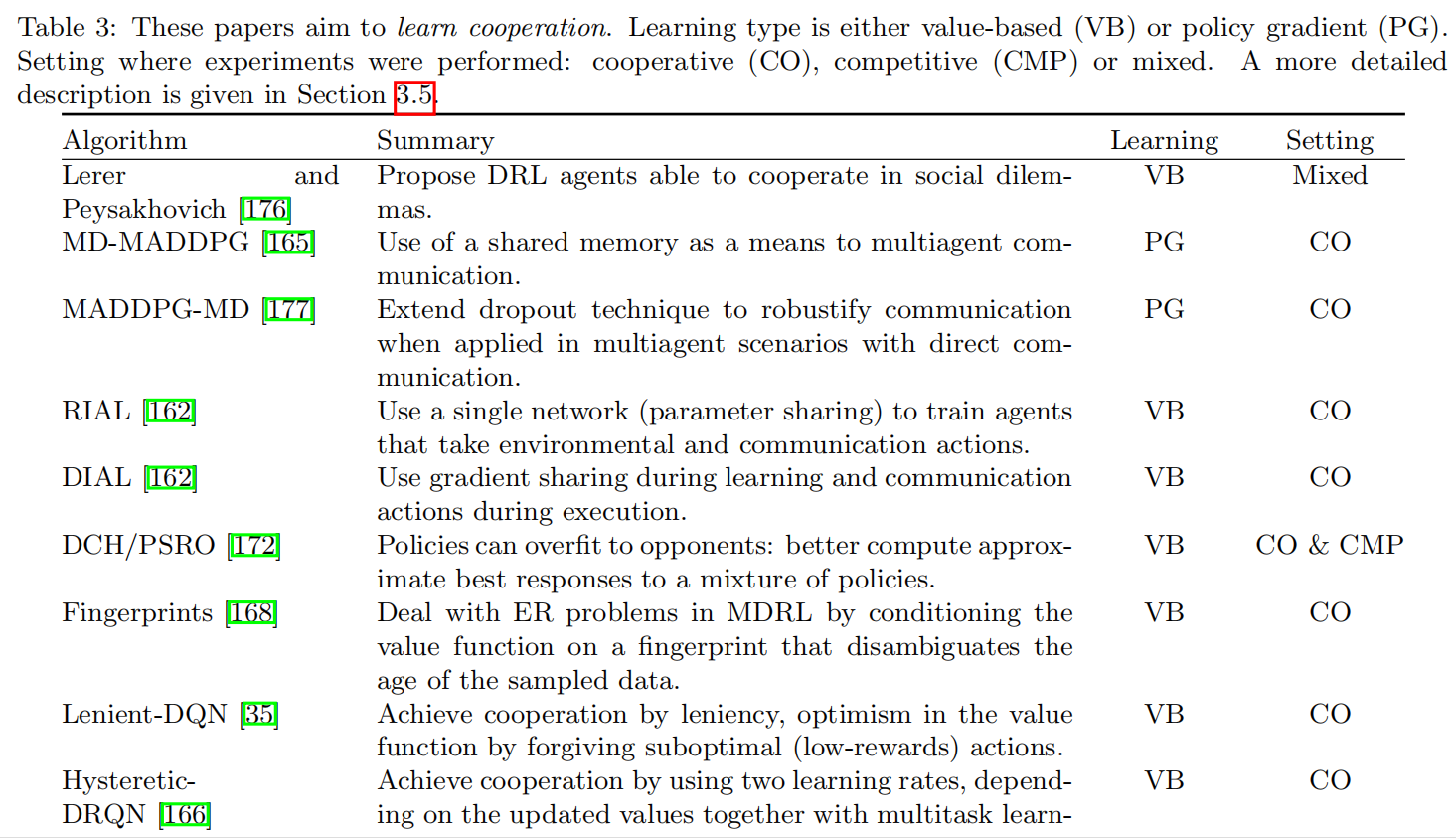
RIAL is based on DRQN and also uses the concept of parameter sharing, i.e., using a single network whose parameters are shared among all agents. In contrast, DIAL directly passes gradients via the communication channel during learning, and messages are discretized and mapped to the set of communication actions during execution.

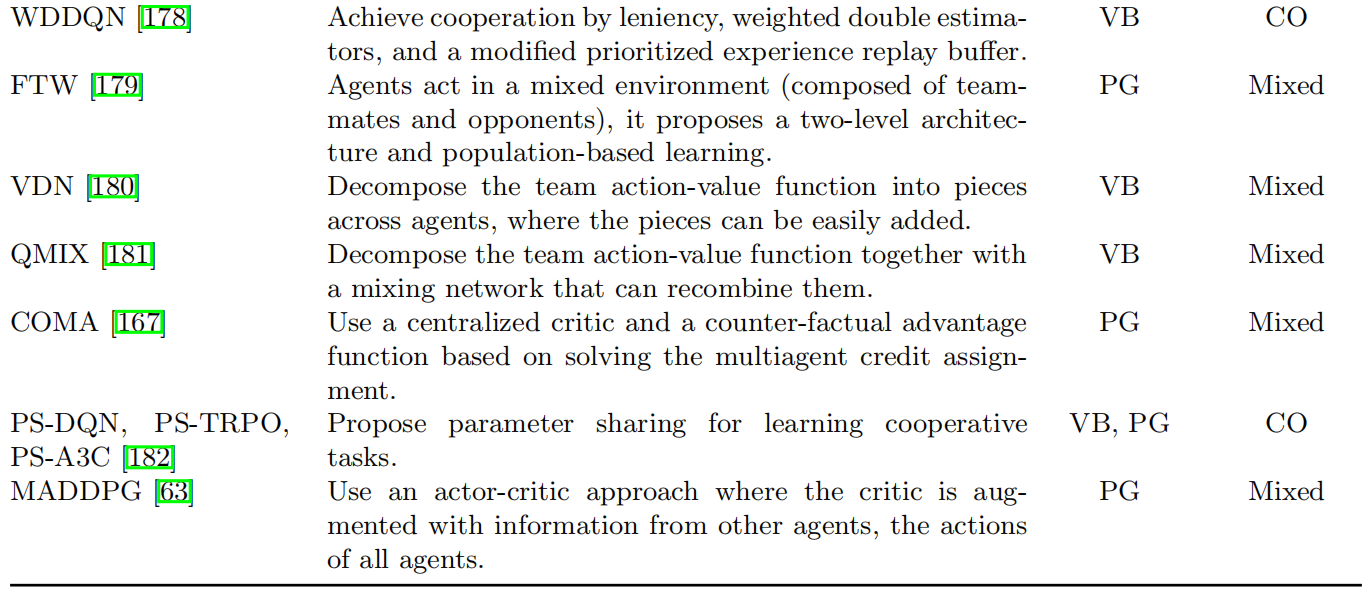
In MD(memory-driven)-MADDPG, the agents use a shared memory as a communication channel.

Direct communication through messages is allowed. In this case, the messages of other agents are dropped out at training time, thus the authors proposed the Message-Dropout MADDPG algorithm.

Multi-agent Bidirectionally Coordinated Network (BiCNet), communication takes place in the latent space (i.e., in the hidden layers).

1. **Cooperation**





Their solution is to add information to the experience tuple that can help to disambiguate the age of the sampled data from the replay memory. 经验回放的遗忘性。

Multi-agent Importance Sampling which adds the probability of the joint action so an importance sampling correction [70, 218] can computed when the tuple is later sampled for training.

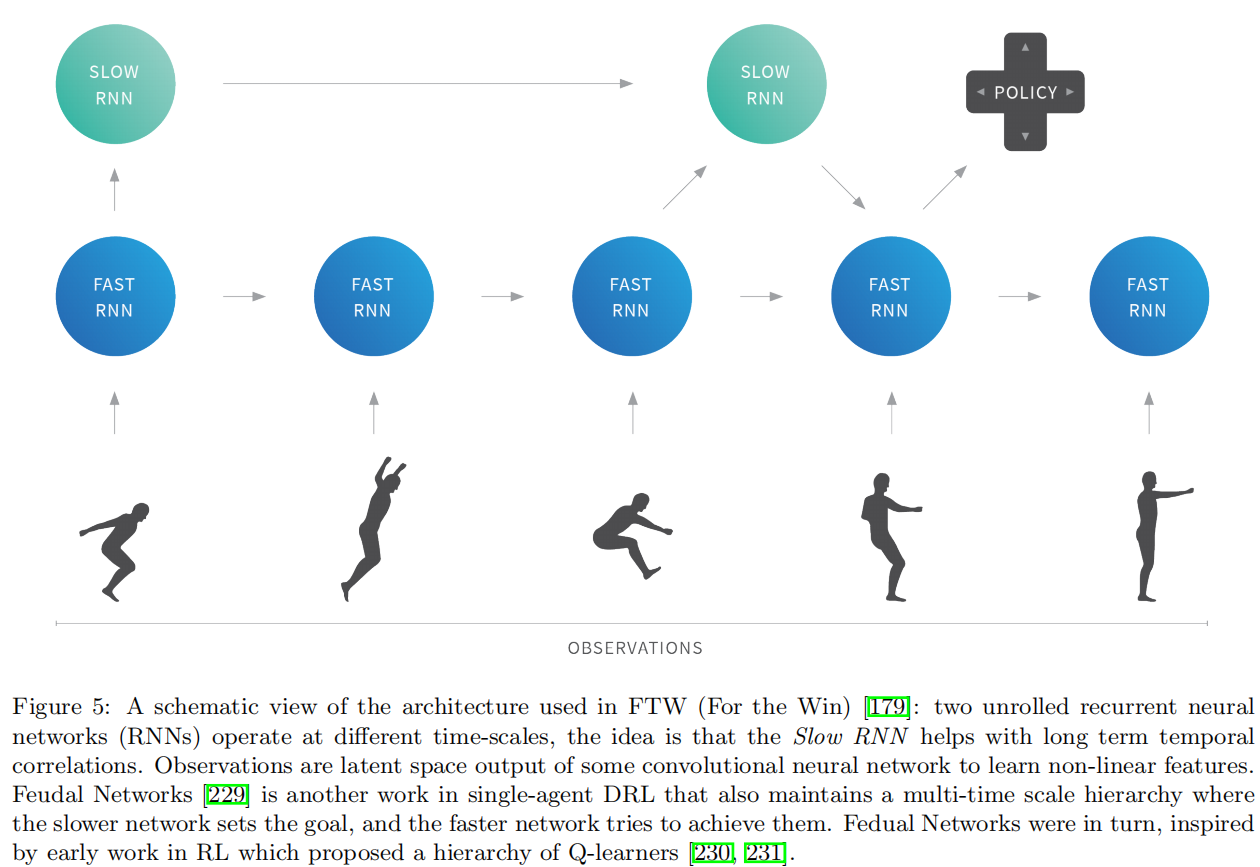
Multi-agent Fingerprints which adds the estimate (i.e., fingerprint) of other agents’ policies.

LDQN: The purpose of leniency is to overcome a pathology called relative over-generalization. Maintaining an optimistic disposition to mitigate the noise from transitions resulting in miscoordination, preventing agents from being drawn towards sub-optimal but wide peaks in the reward search space. 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。

Lenient learners over time apply less leniency towards updates that would lower utility values, taking into account how frequently observation-action pairs have been encountered.

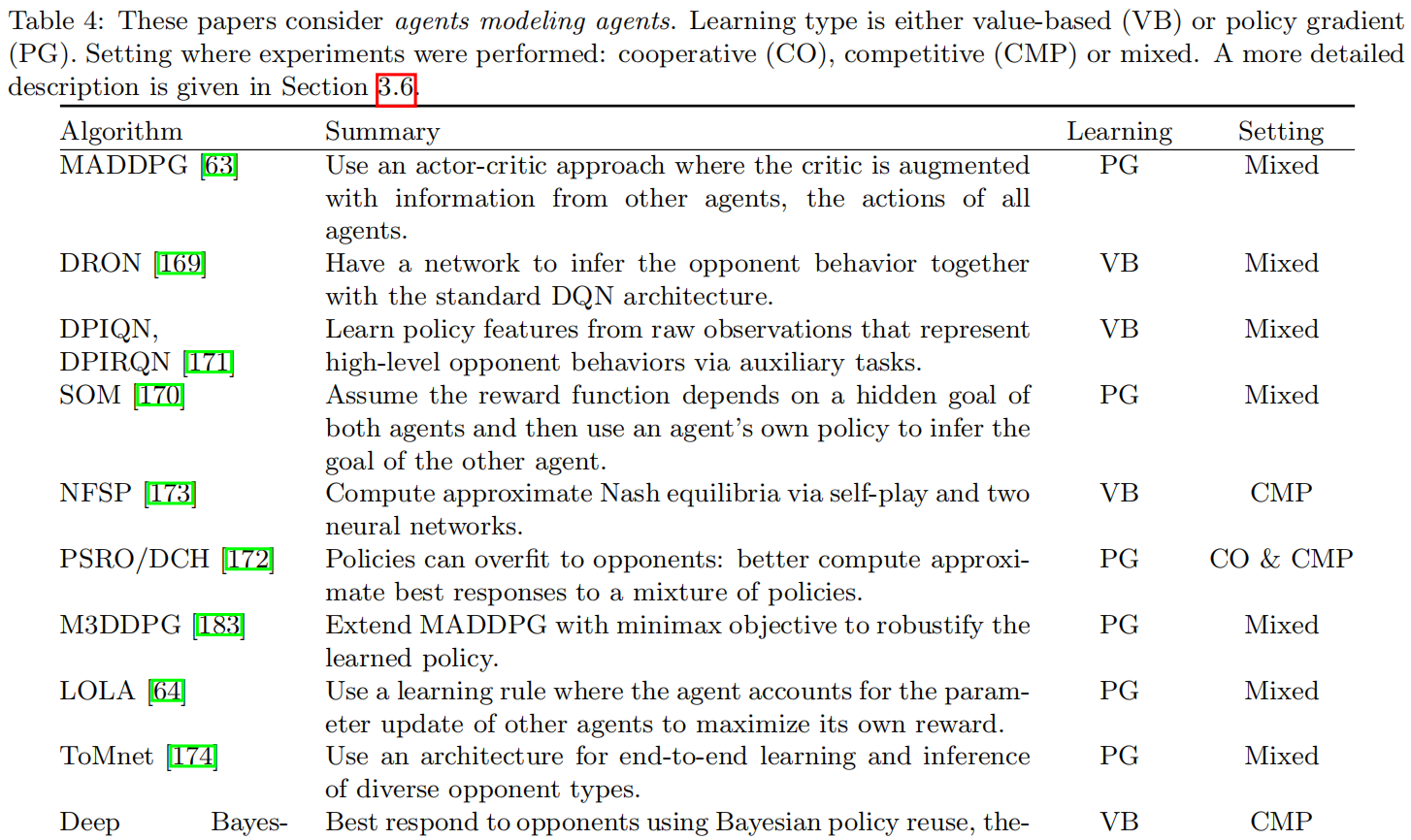
Weighted Double Deep Q-Network (WDDQN) is based on having double estimators.

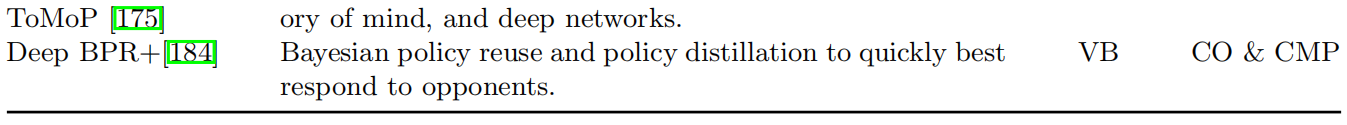
A hierarchical two-level representation with recurrent neural networks operating at different timescales.



QMIX relies on the idea of factorizing, however, instead of sum, QMIX assumes a mixing network that combines the local values in a non-linear way, which can represent monotonic action-value functions. **任务分解，value function 分解???** the factorization of value-functions in multi-agent scenarios using function approximator (MDRL) is an ongoing research topic.

1. **Agent models other agents**



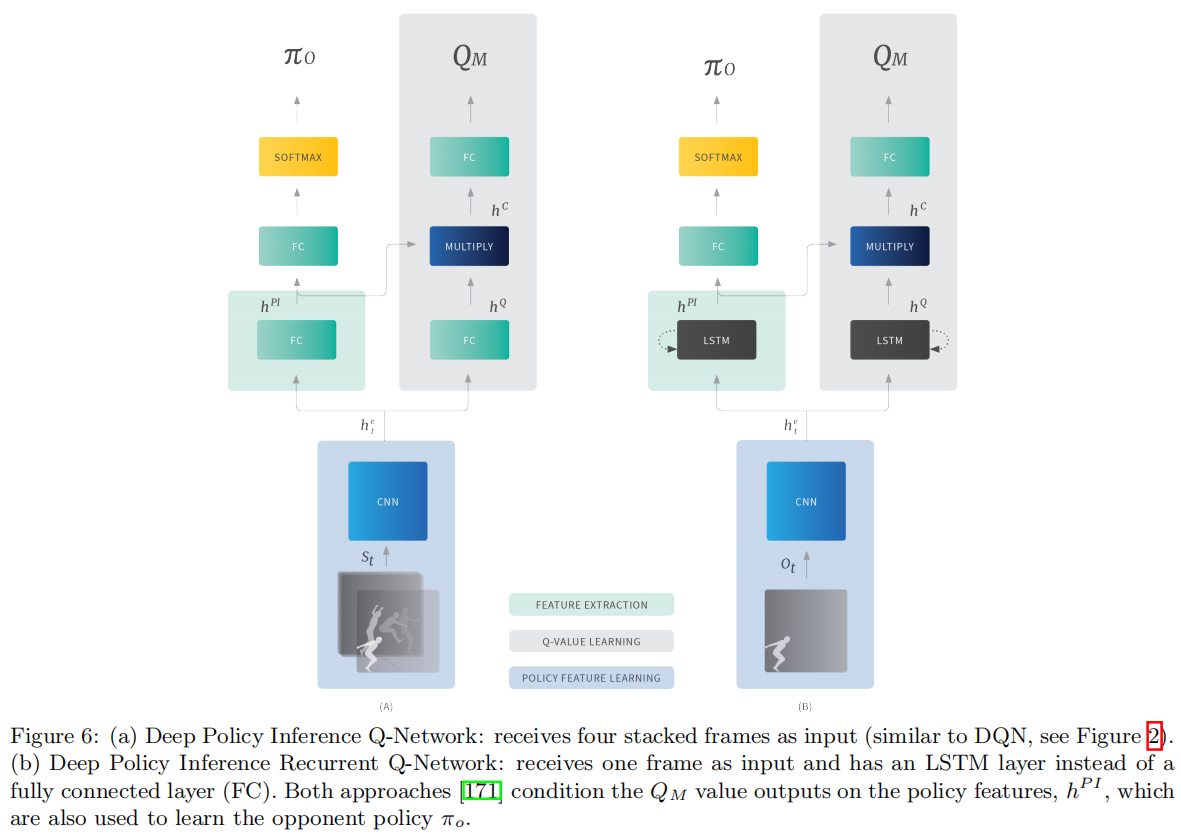


An important ability for agents to have is to reason about the behaviors of other agents by constructing models that make predictions about the modeled agents.

An early work for modeling agents while using deep neural networks was the Deep Reinforcement Opponent Network (DRON). one which evaluates Q-values and a second one that learns a representation of the opponent’s policy. The authors proposed to have several expert networks to combine their predictions to get the estimated Q value, the idea being that each expert network captures one type of opponent strategy.

type-based reasoning from game theory基于类型的博弈论推理

Deep Policy Inference Q-Network (DPIQN) and its recurrent version, DPIRQN learn policy features directly from raw observations of the other agents. Set the auxiliary task which is to learn the opponents’ policies. This auxiliary task modifies the loss function by computing an auxiliary loss: the cross entropy loss between the inferred opponent policy and the ground truth (one-hot action vector) of the opponent.



The authors used an adaptive training procedure to adjust the attention (a weight on the loss function) to either emphasize learning the policy features (of the opponent) or the respective Q values of the agent.

Self Other Modeling (SOM) proposed a different approach, this is, using the agent’s own policy as a means to predict the opponent’s actions. SOM uses two networks, one used for computing the agents’ own policy, and a second one used to infer the opponent’s goal.

可以从博弈论中的东西得到一些启发方法。The (N)FSP concept was further generalized in Policy-Space Response Oracles (PSRO)

They extended the MADDPG algorithm to Mini-max Multi-agent Deep Deterministic Policy Gradients (M3DDPG), which updates policies considering a worst-case scenario: assuming that all other agents act adversarailly.

However, they do not explicitly account for anticipated learning of the other agents, which is the objective of Learning with Opponent-Learning Awareness (LOLA)学习对手意识。

Therefore, a LOLA agent directly shapes the policy updates of other agents to maximize its own reward. One of LOLA’ s assumptions is having access to opponents’ policy parameters.

ToMnet has an architecture composed of three networks: (i) a character network that learns from historical information, (ii) a mental state network that takes the character output and the recent trajectory, and (iii) the prediction network that takes the current state as well as the outputs of the other networks as its input.

Deep Bayesian Theory of Mind Policy (Bayes-ToMoP) is another algorithm that takes inspiration from theory of mind.

A limitation of BPR+ is that it behaves poorly against itself (self-play), thus, Deep Bayes-ToMoP uses theory of mind to provide a higher-level reasoning strategy which provides an optimal behavior against BPR+ agents.

**Implement:**

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

1. **How to succeed**

Avoiding deep learning amnesia: examples in MDRL. 避免健忘症

1. Dealing with non-station in independent learners

Hyper-Q which accounts for the (values of mixed) strategies of other agents and includes that information in the state representation, which effectively turns the learning problem into a stationary one.

1. Multi-agent credit assignment

Credit assignment aims to capture an agent’s contribution to the system’s global performance. the wonderful life utility, which proposes to use a clamping operation that would be equivalent to removing that player from the team.

1. Multitask learning

Distillation, roughly defined as transferring the knowledge from a large model to a small model, was a concept originally introduced for supervised learning and model compression. Policy distillation was used to train a much smaller network and to merge several task-specific policies into a single policy.

1. Auxiliary tasks

One could think of extending these auxiliary tasks to modeling other agents’ behaviors, which is one of the key ideas that DPIQN and DRPIQN proposed in MDRL settings.

1. Experience replay

The concept of experience replay to speed up the credit assignment propagation process in single agent RL.

1. Double estimators

Double Q-learning works by keeping two Q functions and was proven to convergence to the optimal policy.

1. **How to learn for new**
2. Experience replay buffer in MDRL

Adding information in the experience tuple that can help disambiguate the sample is the solution adopted in many works, whether a value based method or a policy gradient method.

1. Centralized learning with decentralized execution

During learning additional information can be used (e.g., global state, action, or rewards) and during execution this information is removed.

1. Parameter sharing
2. Recurrent network

They suffer from the vanishing gradient problem, which renders them inefficient for long-term dependencies. However, RNN variants such as LSTM and GRU(Gated Recurrent Unit) addressed this challenge.

Feudal Networks proposed a hierarchical approach, multiple LSTM networks with different time-scales, i.e., the observation input schedule is different for each LSTM network, to create a temporal hierarchy so that it can better address the long-term credit assignment challenge for RL problems.

1. Overfitting in MAL

A solution is to have a set of policies (an ensemble)

and learn from them or best respond to the mixture of them [172, 63, 169]. Another solution

has been to robustify algorithms.

1. **MARL Benchmark**
2. **Future Challenges**
3. Reproducibility, troubling trends and negative results
4. Implementation challenges and hyperparameter tuning

Additional non-trivial optimization — these are sometimes necessary for the algorithms to achieve good performance.

1. Computational resource
2. Occam’s razor and ablative analysis奥卡姆剃刀和烧烛分析
3. **Open problems**
4. On the challenge of sparse and delayed rewards

To address this issue, recent MDRL approaches applied dense rewards(a concept originated in RL) at each step to allow the agents to learn basic motor skills and then decrease these dense rewards over time in favor of the environmental reward. Hand-crafted intermediate rewards to accelerate the learning and FTW lets the agents learn their internal rewards by a hierarchical two-tier optimization.

1. On the role of self-play
2. On the challenge of the combination nature of MDRL

How search and RL can be better combined for potentially new methods. One way to tackle this challenge指数增长的动作空间 within multi-agent scenarios is the use of search parallelization.

**Conclusion:**

First, we categorized recent works into four different topics: emergent behaviors, learning communication, learning cooperation, and agents modeling agents.

Then, we exemplified how key components (e.g., experience replay and difference rewards) originated in RL and MAL need to be adapted to work in MDRL.