



多智能体系统

多智能体系统

- 现实世界中复杂的任务所涉及的系统规模较为庞大。
- 将复杂任务可以分解为多个子任务。
- 这些任务需要智能体互相配合并且决策的结果会互相影响。

这样的系统可以被称为多智能体系统 (multi-agent system, MAS)

多智能体系统

多智能体系统中：

- 1) 至少有两个Agent
- 2) Agent之间存在着一定的关系, 如合作关系、竞争关系, 或者同时存在竞争与合作的混合关系
- 3) 每个Agent最终所获得的奖赏会受到其余Agent的影响。



...

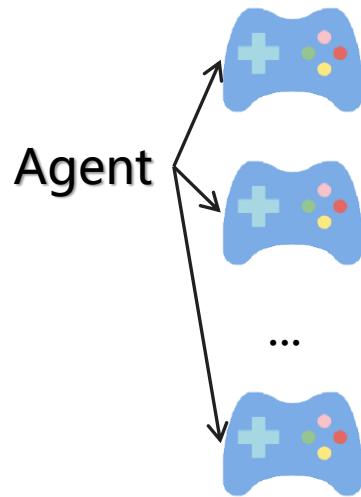


- 1) 控制守门员, 后卫, 前锋共11个球员取得比赛胜利
- 2) 可以每个球员移动, 冲刺, 传球, 射门等
- 3) 同队间合作获得比分、两队互相竞争, (如果给球员加入个人进球数的指标同队中既有竞争又有合作)

多智能体系统

为什么需要多agents ?

- 想象一下一个人学会同时控制两个球员，难度太大了！
- 对于计算机而言学会控制一个agent，即学会状态到动作的映射 $\pi: S \rightarrow A$ ，有现成的强化学习算法可用。
- 控制11个agent即学会 $\pi: S \times \dots \times S \rightarrow A \times \dots \times A$ ，输入输出空间太大了 $|S|^{11}, |A|^{11}$



多智能体系统

- 多智能体强化学习
 - 单智能体深度强化学习（去中心化训练）
 - 通信学习（中心化训练）
 - 协同学习（中心化训练）
 - 基于值函数
 - 基于Actor-Critic
- 多智能体-大语言模型

单智能体强化学习

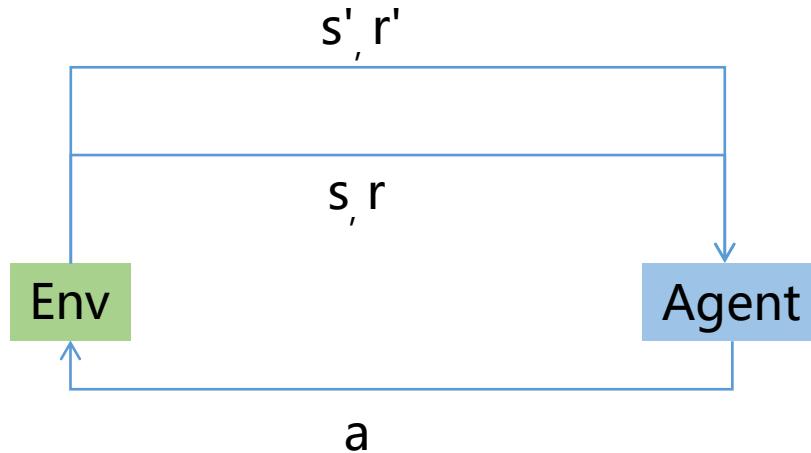
多智能体强化学习中学习策略的算法，是基于单智能体强化学习算法的

单智能体强化学习

强学习中智能体与环境的交互过程可以被建模为 MDP

- S : Agent 所处的有限状态空间
- A : Agent 的有限动作集合
- $R : S \times A \times S \rightarrow \mathbb{R}$: 奖励函数, 用以描述发生状态转移时 Agent 获得的环境反馈给它的奖赏
- $T(s' | s, a)$: 状态转移函数, 代表了从当前状态 $s \in S$ 执行了动作 $a \in A$ 之后到达 $s' \in S$ 的概率, 此概率是由环境本身的不确定性决定的.
- $\gamma \in [0, 1]$: 折扣因子. 如果 Agent 在环境中作了长期决策, γ 就是为了平衡立即奖赏和长期累计奖赏的折扣因子.

MDP 其实是希望找到一个最优策略 $\pi : S \rightarrow A$, 使折扣回报 $G = \sum_{t=1}^T \gamma^t r_t$ 最大化



单智能体强化学习

基于值函数

- Q值 $Q(s, a)$ 表示的是 agent 在当前状态 s 下执行 a ，之后根据策略决策，未来总收益的预测。
- V值 $V(s)$ 表示的是 agent 在当前状态 s 下，之后根据策略决策，未来总收益的预测。
- Q值的更新方式: $\hat{Q}(s, a) \leftarrow \hat{Q}(s, a) + \alpha \left[\left(r + \gamma \max_{a'} \hat{Q}(s', a') \right) - \hat{Q}(s, a) \right]$
- V值的更新方式: $\hat{V}(s) \leftarrow \hat{V}(S) + \alpha [(r + \gamma \hat{V}(s')) - \hat{V}(s)]$
- 预测动作的价值，执行价值更高的动作。使用时序差分（动态规划）的方法进行值的更新。
$$\sum_{t=0}^T \gamma^t r_t = r_0 + \gamma \sum_{t=1}^T \gamma^{t-1} r_t$$

基于 Actor-Critic

- Actor 负责执行动作，Critic 负责评价动作指导 Actor 学习。（相当球员与教练）
- Actor 不对动作打分，直接输出动作（或者动作的概率），可输出连续动作，但是无法单步更新。（需要完成一回合）
- Critic 更新：

$$\hat{V}(s) \leftarrow \hat{V}(s) + \alpha \delta$$

- Actor 更新： $\delta = (r + \gamma \hat{V}(s')) - \hat{V}(s)$

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi(s, a; \theta) \delta$$

单智能体强化学习

基于值函数

- Q值 $Q(s, a)$ 表示的是 agent 在当前状态 s 下执行 a ，之后根据策略决策，未来总收益的预测。
- V值 $V(s)$ 表示的是 agent 在当前状态 s 下，之后根据策略决策，未来总收益的预测。

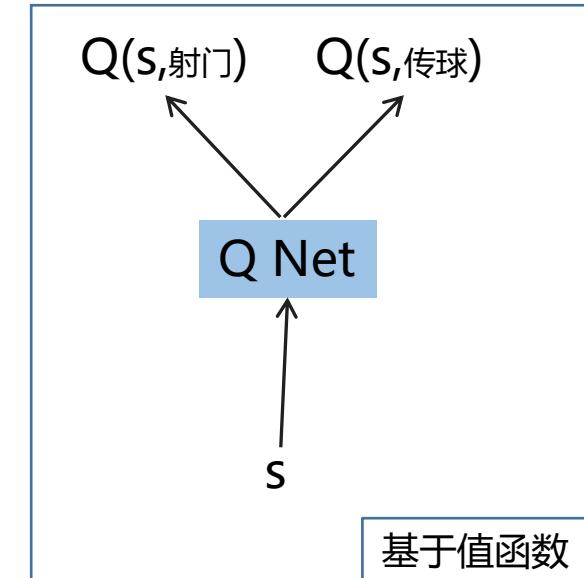
• Q值的更新方式: $\hat{Q}(s, a) \leftarrow \hat{Q}(s, a) + \alpha \left[\left(r + \gamma \max_{a'} \hat{Q}(s', a') \right) - \hat{Q}(s, a) \right]$

• V值的更新方式: $\hat{V}(s) \leftarrow \hat{V}(S) + \alpha [(r + \gamma \hat{V}(s')) - \hat{V}(s)]$

- 预测动作的价值，执行价值更高的动作。使用时序差分（动态规划）的方法进行值的更新。

	射门	传球	冲刺
1	0.3	$0.3 + 0.01 * (0 + 0.9 * 0.6 - 0.3)$	0.3
2	0.2	0.2	0.1
3	0.6	0.4	0.5

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi(s, a; \theta) \delta$$



单智能体强化学习

基于值函数

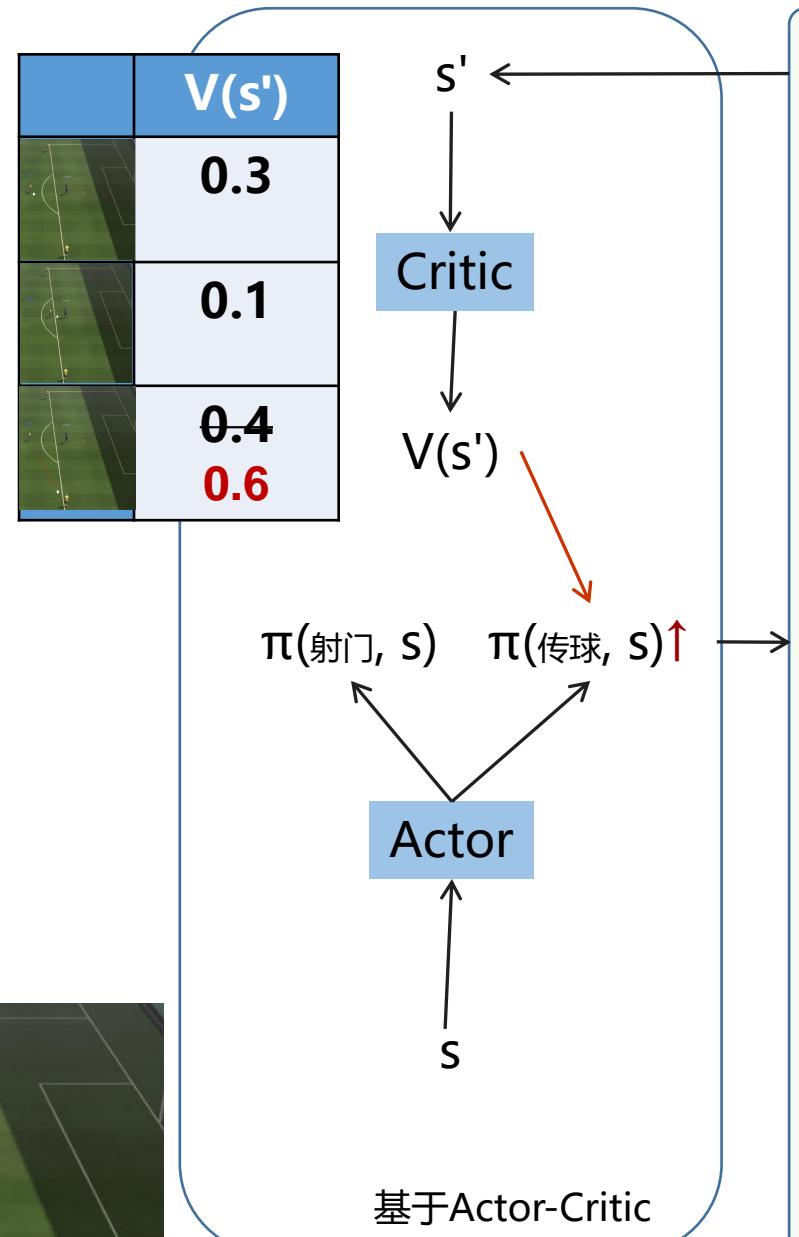
- Q值 $Q(s, a)$ 表示的是 agent 在当前状态 s 下执行 a ，之后根据策略决策，未来总收益的预测。
- V值 $V(s)$ 表示的是 agent 在当前状态 s 下，之后根据策略决策，未来总收益的预测。
- Q值的更新方式: $\hat{Q}(s, a) \leftarrow \hat{Q}(s, a) + \alpha \left[\left(r + \gamma \max_{a'} \hat{Q}(s', a') \right) - \hat{Q}(s, a) \right]$
- V值的更新方式: $\hat{V}(s) \leftarrow \hat{V}(s) + \alpha [(r + \gamma \hat{V}(s')) - \hat{V}(s)]$
- 预测动作的价值，执行价值更高的动作。使用时序差分（动态规划）的方法进行值的更新。

基于 Actor-Critic

- Actor 负责执行动作，Critic 负责评价动作指导 Actor 学习。（相当球员与教练）
- Actor 不对动作打分，直接输出动作（或者动作的概率），可输出连续动作，但是无法单步更新。（需要完成一回合）
- Critic 更新：

$$\hat{V}(s) \leftarrow \hat{V}(s) + \alpha \delta$$

- Actor 更新： $\delta = (r + \gamma \hat{V}(s')) - \hat{V}(s)$
- $$\theta \leftarrow \theta + \alpha \nabla_{\theta} \log \pi(s, a; \theta) \delta$$

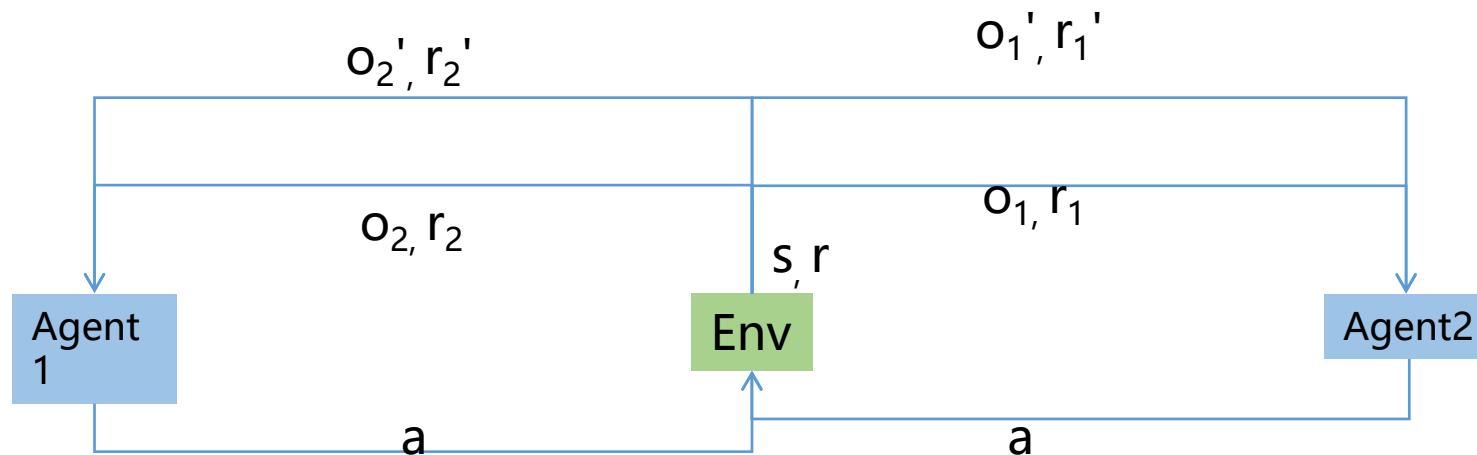


多智能体强化学习

非中心部分可观马尔可夫决策过程(Dec-POMDP)

- N : Agent 的数目, 通常 $N \geq 2$
- \mathbf{S} : 状态空间, 包含 Agent 和环境的信息 $\mathbf{S} = S \times \dots \times S$
- $\mathbf{A} = [A_1, A_2, \dots, A_N]$: 联合动作 \mathbf{a} 的空间, 其中 A_i 代表 $Agent_i$ 所能采取的局部动作 a_i 的集合
- $\mathbf{R} : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow \mathbb{R}^N$: 奖励函数, 用以描述发生状态转移时 Agent 获得的奖励. 在完全合作环境下, 所有 Agent 共享同一个奖励函数
- $\mathbf{T}(s' | s, a) : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow [0, 1]$: 状态转移函数, 用以描述在状态 s 下, Agent 采取联合动作 a 之后, 转移到状态 s' 的概率
- $\mathbf{O} = [O_1, \dots, O_N]$: 联合观测 \mathbf{o} 的集合
- $\mathbf{Z} : \mathbf{S} \times \mathbf{A} \rightarrow \mathbf{O}$: 观测函数
- $\gamma \in [0, 1]$: 折扣因子

每个Agent i 希望找到一个最优策略 $\pi : S \rightarrow A$, 使折扣回报 $G = \sum_{t=1}^T \gamma^t r_t$ 最大化

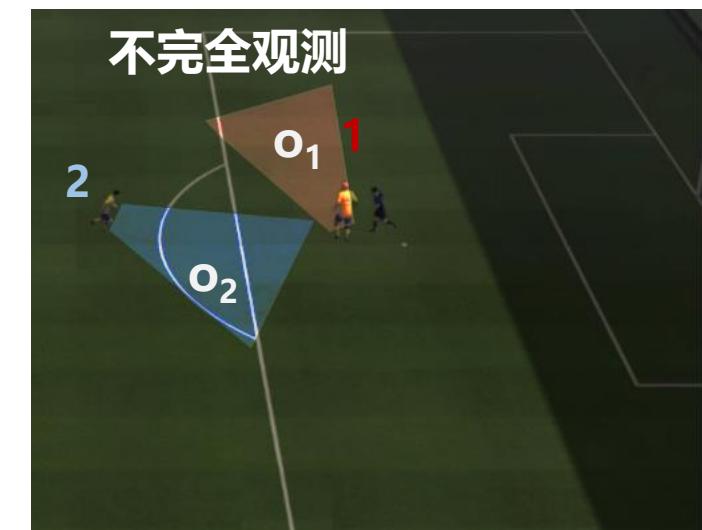
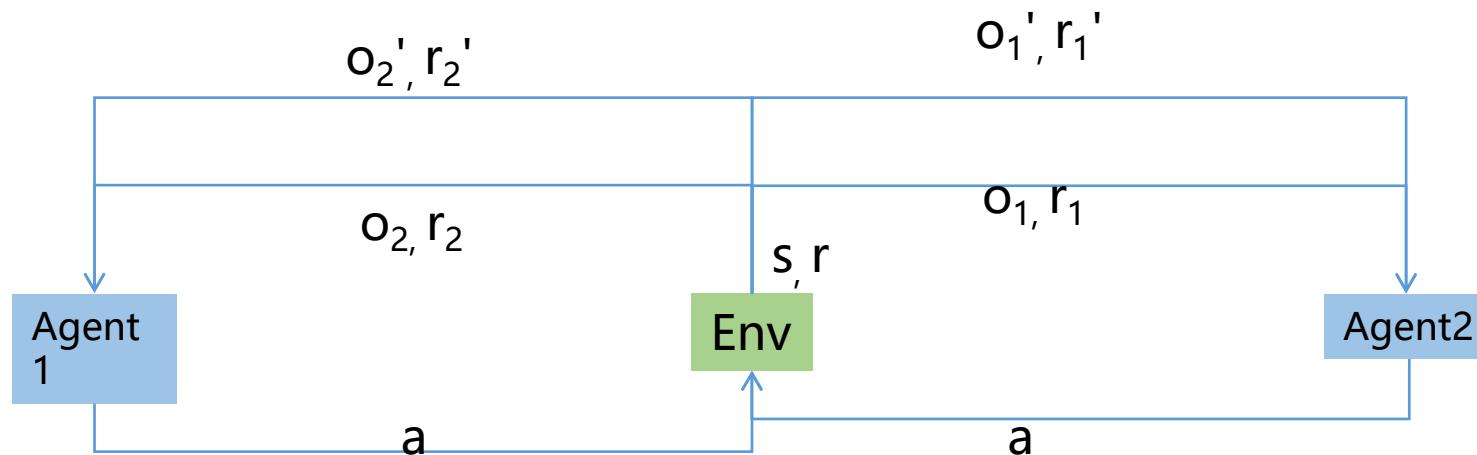


多智能体强化学习

非中心部分可观马尔可夫决策过程(Dec-POMDP)

- N : Agent 的数目, 通常 $N \geq 2$
 - \mathbf{S} : 状态空间, 包含 Agent 和环境的信息 $\mathbf{S} = S \times \dots \times S$
 - $\mathbf{A} = [A_1, A_2, \dots, A_N]$: 联合动作 \mathbf{a} 的空间, 其中 A_i 代表 $Agent_i$ 所能采取的局部动作 a_i 的集合
 - $\mathbf{R} : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow \mathbb{R}^N$: 奖励函数, 用以描述发生状态转移时 Agent 获得的奖励. 在完全合作环境下, 所有 Agent 共享同一个奖励函数
 - $\mathbf{T}(s' | s, a) : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow [0, 1]$: 状态转移函数, 用以描述在状态 s 下, Agent 采取联合动作 a 之后, 转移到状态 s' 的概率
 - $\mathbf{O} = [O_1, \dots, O_N]$: 联合观测 \mathbf{o} 的集合
 - $\mathbf{Z} : \mathbf{S} \times \mathbf{A} \rightarrow \mathbf{O}$: 观测函数
- $\gamma \in [0, 1]$: 折扣因子

每个Agent i 希望找到一个最优策略 $\pi : S \rightarrow A$, 使折扣回报 $G = \sum_{t=1}^T \gamma^t r_t$ 最大化

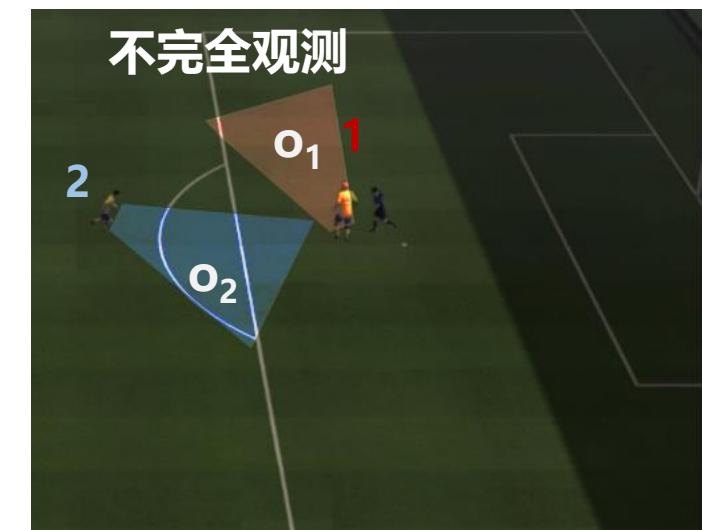
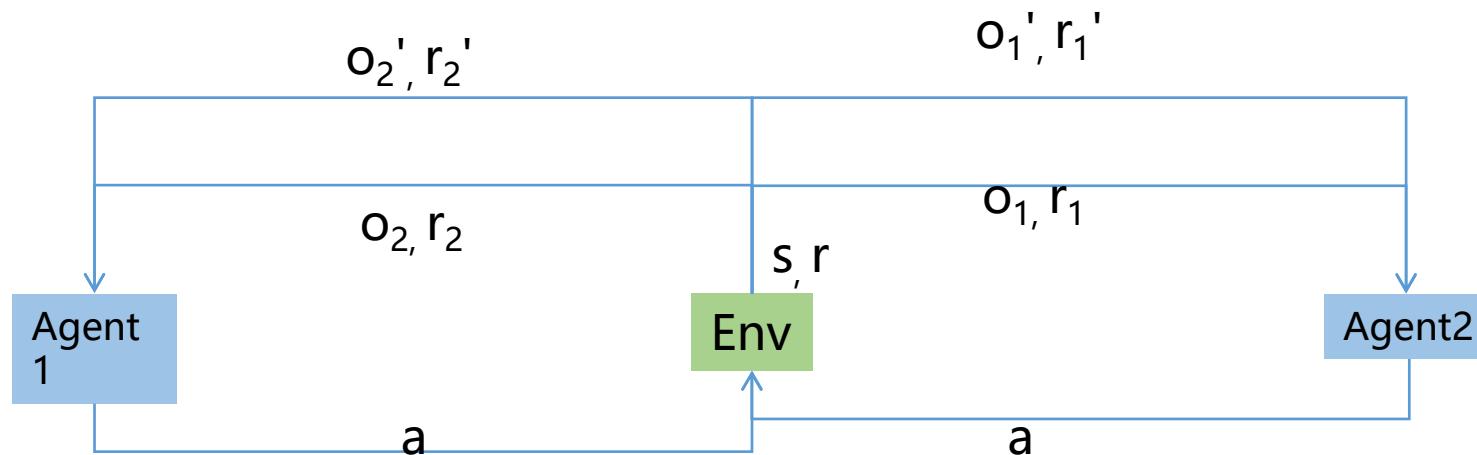


多智能体强化学习

非中心部分可观马尔可夫决策过程(Dec-POMDP)

- N : Agent 的数目, 通常 $N \geq 2$
 - \mathbf{S} : 状态空间, 包含 Agent 和环境的信息 $\mathbf{S} = S \times \dots \times S$
 - $\mathbf{A} = [A_1, A_2, \dots, A_N]$: 联合动作 \mathbf{a} 的空间, 其中 A_i 代表 $Agent_i$ 所能采取的局部动作 a_i 的集合
 - $\mathbf{R} : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow \mathbb{R}^N$: 奖励函数, 用以描述发生状态转移时 Agent 获得的奖励. 在完全合作环境下, 所有 Agent 共享同一个奖励函数
 - $\mathbf{T}(s' | s, a) : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow [0, 1]$: 状态转移函数, 用以描述在状态 s 下, Agent 采取联合动作 a 之后, 转移到状态 s' 的概率
 - $\mathbf{O} = [O_1, \dots, O_N]$: 联合观测 \mathbf{o} 的集合
 - $\mathbf{Z} : \mathbf{S} \times \mathbf{A} \rightarrow \mathbf{O}$: 观测函数
- $\gamma \in [0, 1]$: 折扣因子

每个Agent i 希望找到一个最优策略 $\pi : S \rightarrow A$, 使折扣回报 $G = \sum_{t=1}^T \gamma^t r_t$ 最大化



多智能体强化学习

非中心部分可观马尔可夫决策过程(Dec-POMDP)

- N : Agent 的数目, 通常 $N \geq 2$
- \mathbf{S} : 状态空间, 包含 Agent 和环境的信息 $\mathbf{S} = S \times \dots \times S$
- $\mathbf{A} = [A_1, A_2, \dots, A_N]$: 联合动作 \mathbf{a} 的空间, 其中 A_i 代表 $Agent_i$ 所能采取的局部动作 a_i 的集合
- $\mathbf{R} : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow \mathbb{R}^N$: 奖励函数, 用以描述发生状态转移时 Agent 获得的奖励. 在完全合作环境下, 所有 Agent 共享同一个奖励函数
- $\mathbf{T}(s' | s, a) : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow [0, 1]$: 状态转移函数, 用以描述在状态 s 下, Agent 采取联合动作 a 之后, 转移到状态 s' 的概率
- $\mathbf{O} = [O_1, \dots, O_N]$: 联合观测 \mathbf{o} 的集合
- $\mathbf{Z} : \mathbf{S} \times \mathbf{A} \rightarrow \mathbf{O}$: 观测函数
- $\gamma \in [0, 1]$: 折扣因子

相比于单智能体情形, 每个智能体在和环境交互的同时也在和其他智能体进行直接或者间接的交互。多智能体比单智能体环境更困难体现在以下几点:

- 每个智能体在不断更新自身策略, 因此在每个智能体的视角下, 环境是**非稳态的 (non-stationary)**。(即对于一个智能体而言, 即使在相同的状态下采取相同动作, 得到的**状态转移和奖励信号的分布可能在不断改变**)
- 训练评估的复杂度会增加, 可能需要大规模非中心化训练来提高效率。



多智能体强化学习

非中心部分可观马尔可夫决策过程(Dec-POMDP)

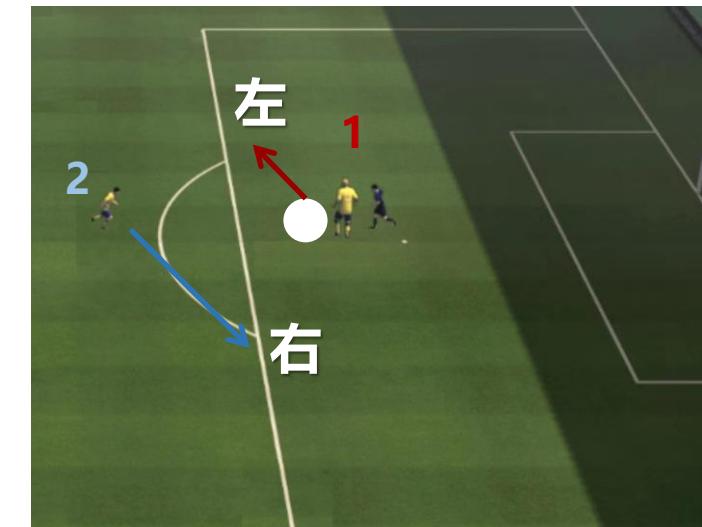
- N : Agent 的数目, 通常 $N \geq 2$
- \mathbf{S} : 状态空间, 包含 Agent 和环境的信息 $\mathbf{S} = S \times \dots \times S$
- $\mathbf{A} = [A_1, A_2, \dots, A_N]$: 联合动作 \mathbf{a} 的空间, 其中 A_i 代表 $Agent_i$ 所能采取的局部动作 a_i 的集合
- $\mathbf{R} : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow \mathbb{R}^N$: 奖励函数, 用以描述发生状态转移时 Agent 获得的奖励. 在完全合作环境下, 所有 Agent 共享同一个奖励函数
- $\mathbf{T}(s' | s, a) : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow [0, 1]$: 状态转移函数, 用以描述在状态 s 下, Agent 采取联合动作 a 之后, 转移到状态 s' 的概率
- $\mathbf{O} = [O_1, \dots, O_N]$: 联合观测 \mathbf{o} 的集合
- $\mathbf{Z} : \mathbf{S} \times \mathbf{A} \rightarrow \mathbf{O}$: 观测函数
- $\gamma \in [0, 1]$: 折扣因子

非稳态环境 :

$s: \{1: \text{在前}, 2: \text{在后}, \text{球: 在前}\}$
 $a: \{1: \text{传左}, 2: \text{跑右}\}$
 $s': \{1: \text{在前}, 2: \text{在右}, \text{球: 在左}\}, r: -1$

策略更新 :

$a: \{1: \text{传左}, 2: \text{跑右}\}$
 $s': \{1: \text{在前}, 2: \text{在左}, \text{球: 在右}\}, r: -1$



多智能体强化学习

非中心部分可观马尔可夫决策过程(Dec-POMDP)

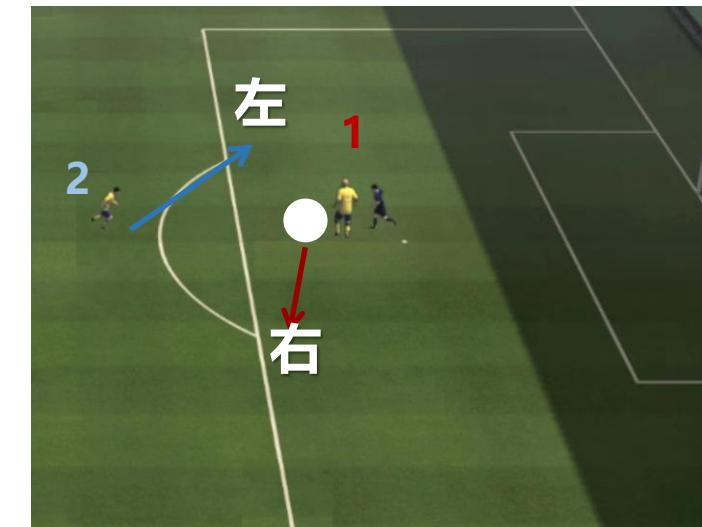
- N : Agent 的数目, 通常 $N \geq 2$
- \mathbf{S} : 状态空间, 包含 Agent 和环境的信息 $\mathbf{S} = S \times \dots \times S$
- $\mathbf{A} = [A_1, A_2, \dots, A_N]$: 联合动作 \mathbf{a} 的空间, 其中 A_i 代表 $Agent_i$ 所能采取的局部动作 a_i 的集合
- $\mathbf{R} : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow \mathbb{R}^N$: 奖励函数, 用以描述发生状态转移时 Agent 获得的奖励. 在完全合作环境下, 所有 Agent 共享同一个奖励函数
- $\mathbf{T}(s' | s, a) : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow [0, 1]$: 状态转移函数, 用以描述在状态 s 下, Agent 采取联合动作 a 之后, 转移到状态 s' 的概率
- $\mathbf{O} = [O_1, \dots, O_N]$: 联合观测 \mathbf{o} 的集合
- $\mathbf{Z} : \mathbf{S} \times \mathbf{A} \rightarrow \mathbf{O}$: 观测函数
- $\gamma \in [0, 1]$: 折扣因子

非稳态环境 :

s : {1: 在前, 2: 在后, 球: 在前}
 a : {1: 传左, 2: 跑右}
 s' : {1: 在前, 2: 在右, 球: 在左}, r : -1

策略更新 :

a : {1: 传左, 2: 跑右}
 s' : {1: 在前, 2: 在左, 球: 在右}, r : -1



多智能体强化学习

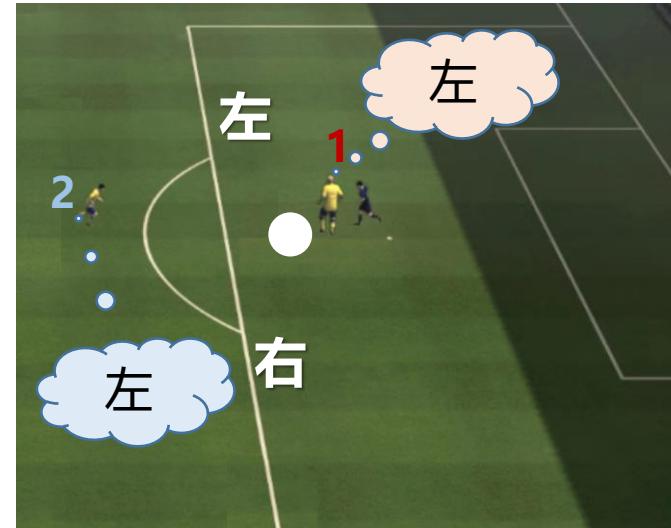
- **去中心化的训练:** 将单Agent强化学习算法直接应用到多Agents环境之中, 每个Agent之间相互独立执行各自的DRL算法。中心化的训练框架会面临环境**不平稳**的问题.
- **中心化的训练:** 可以解决环境非平稳性的问题, 通过把所有Agent联合建模来学习一个联合的策略, 该策略的输入是所有Agent的联合观测, 输出所有Agent的联合动作。去中心化训练去中心化执行(CTDE)是目前主流的框架。

中心化的训练

- 不能直接学习 $\pi: O \times \dots \times O \rightarrow A \times \dots \times A$, 输入输出空间太大了 $|O|^{11}, |A|^{11}$
 - 通信学习(Learning communication) : Agent之间存在显式的通信交互，共享环境信息以学会合作。
 - 协作学习(Learning cooperation): Agents没有显式通信, 只通过动作和各自的观测作出判断，但是训练中加入知晓全局的“教练”以学会合作。



通信 , $\pi: (O, M) \rightarrow A$

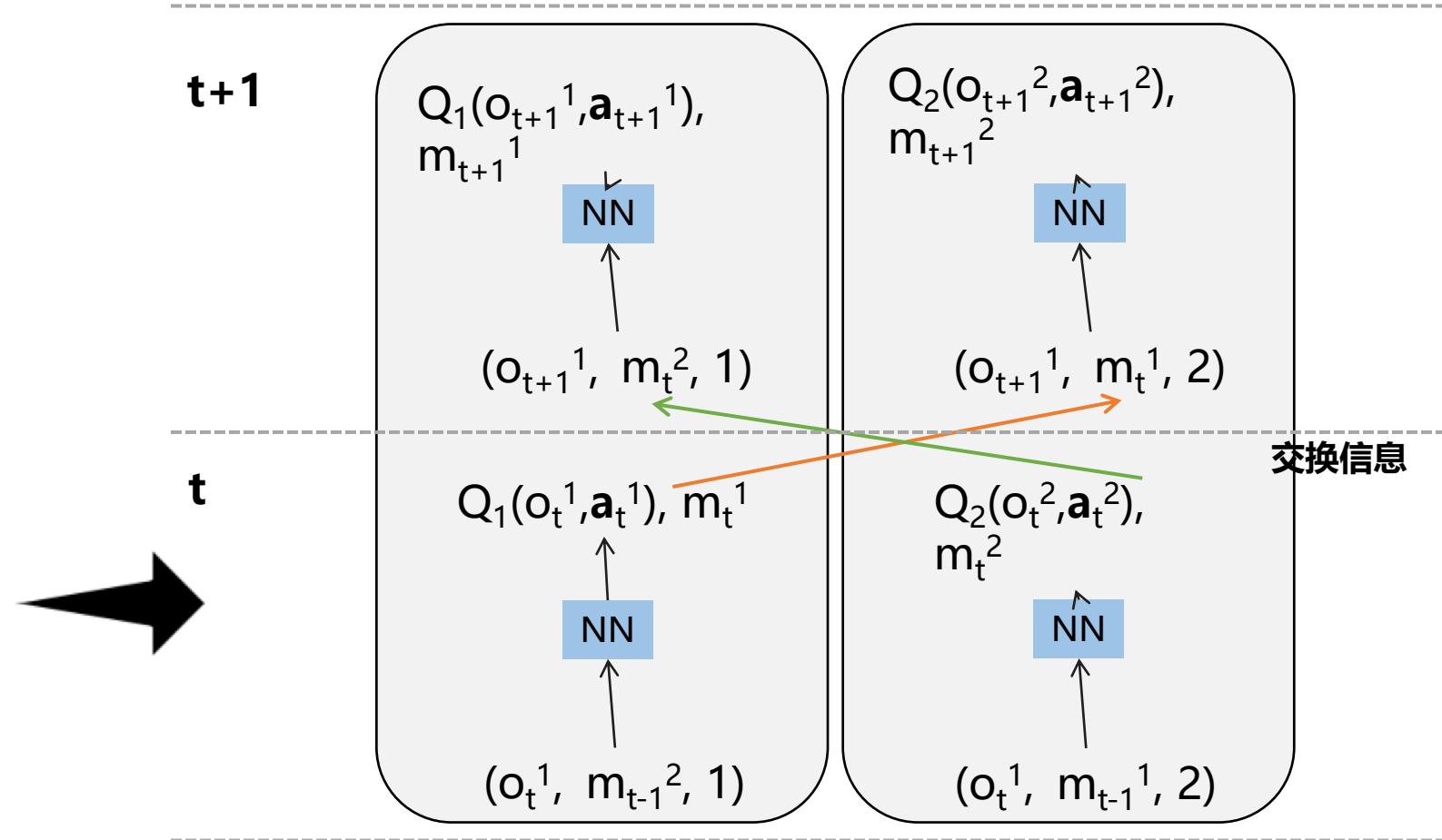
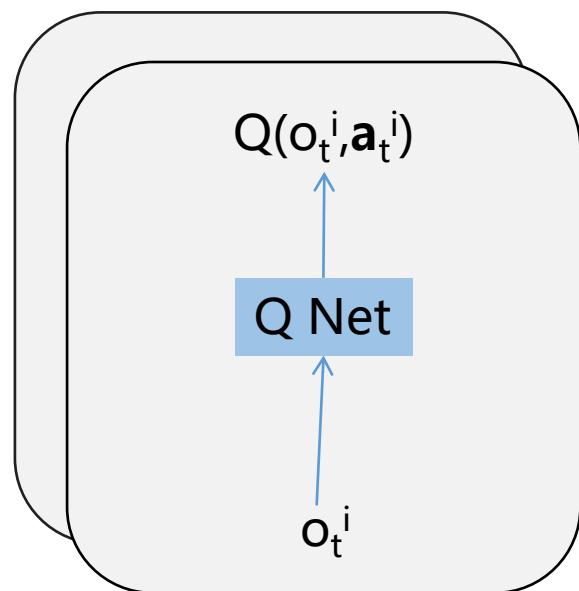


培养默契 $\pi: O \rightarrow A$

多智能体系统

- 多智能体强化学习
 - 单智能体深度强化学习（去中心化训练）
 - 通信学习（中心化训练）
 - 协同学习（中心化训练）
 - 基于值函数
 - 基于Actor-Critic
- 多智能体-大语言模型

RIAL, DIAL



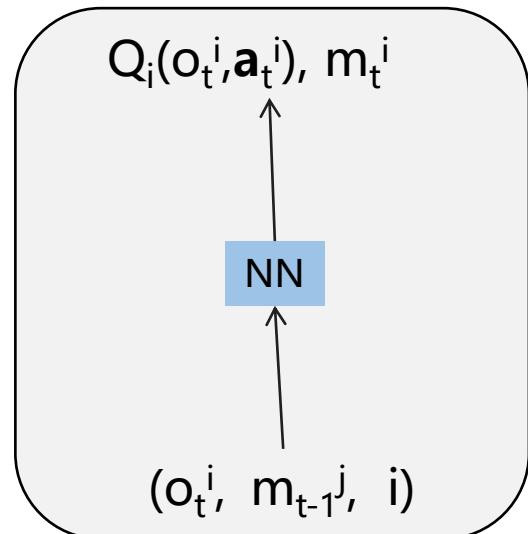
每个智能体i
只输入各自t时刻观测 o_t^i ,
输出动作 $a_t^i = \text{argmax}_a Q(o_t^i, a)$

智能体i
输入t时刻观测 o_t^i , 另一智能体t-1时刻信息 m_{t-1}^j ,
智能体编号i
输出动作 a_t^i , 和给另一智能体信息 m_t^i

RIAL, DIAL

- DRQN: DQN+RNN (LSTM)

决策需要获得完整的观测信息,加入RNN更好地拟合环境的状态

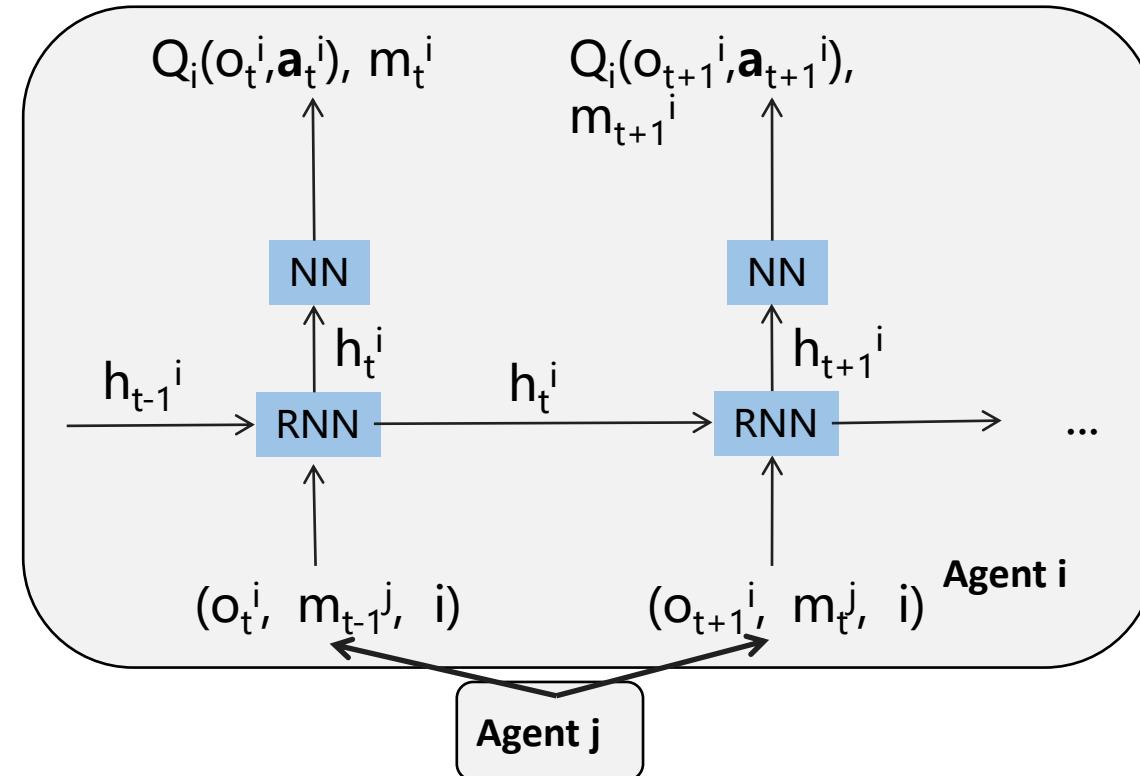


仅利用上一时刻观测和通信信息

(如图玩家转头可以收集一系列的观测视野存在RNN隐变量中)
结合历史观测，球员可获得更大视野和速度等额外观测信息

RIAL, DIAL

- DRQN: DQN+RNN (LSTM)

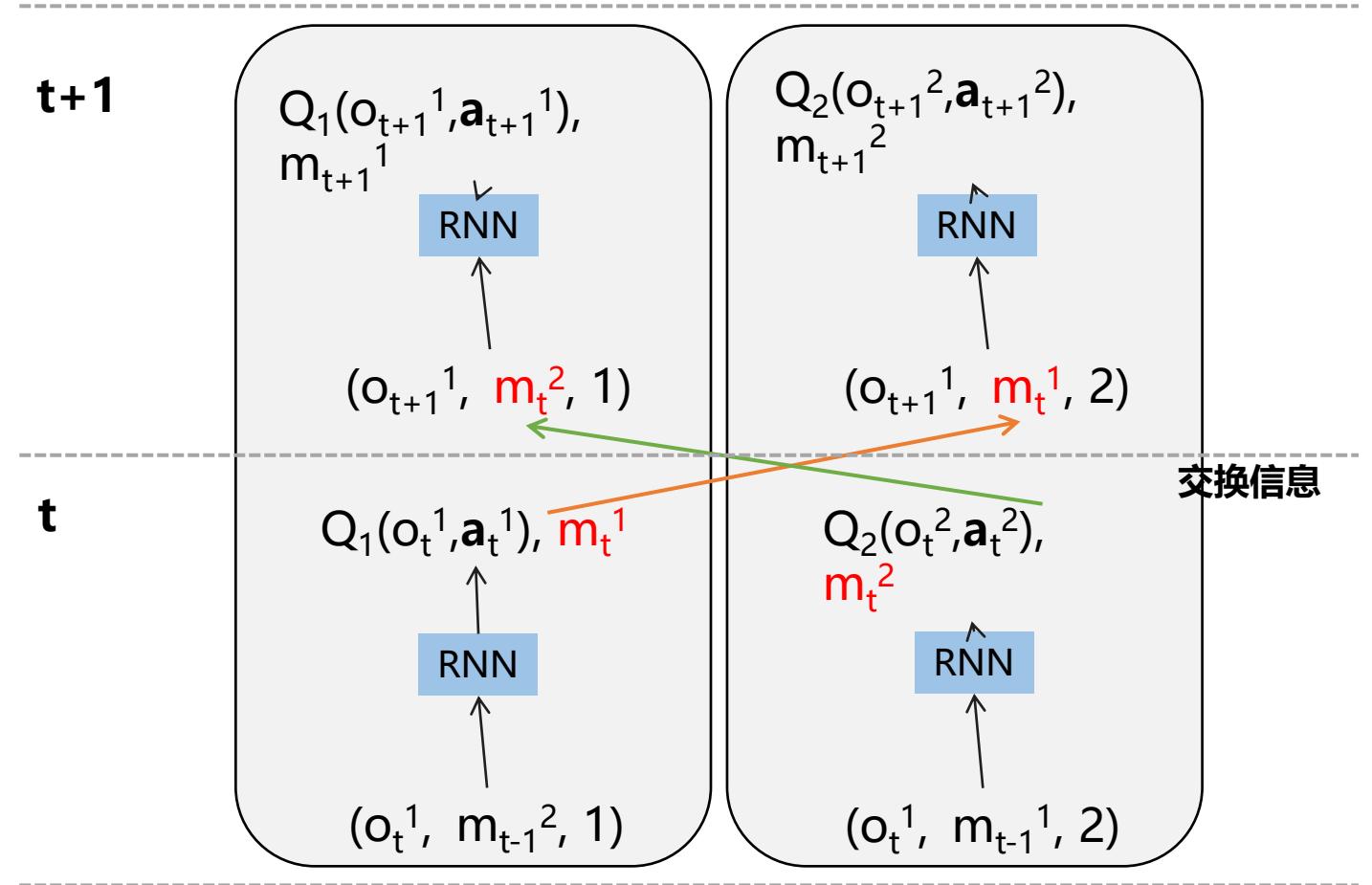


智能体*i*

- 输入当前观测 o_t^i , 另一智能体的通信信息 m_{t-1}^j , 智能体编号*i*以及上一时刻状态隐变量 h_{t-1}^i
- 获得当前隐变量 h_t^i (可理解为对环境状态 s_t 的估计)
- 输出动作 a_t^i , 和给另一智能体信息 m_t^i

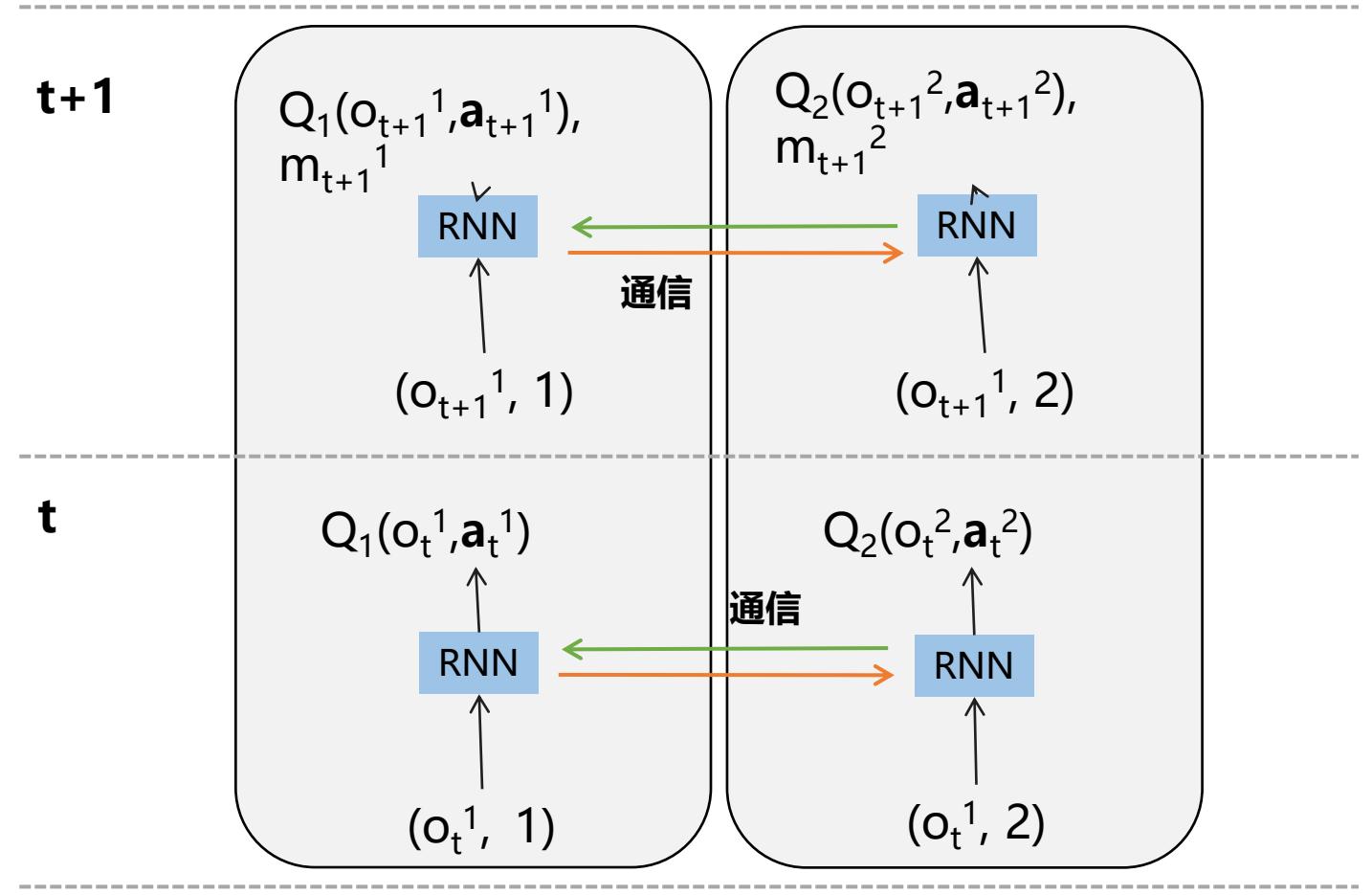
RIAL, DIAL

1. 通信信息 m ，是one-hot编码，不能直接通过反向传播更新
2. 交流的信息落后一步



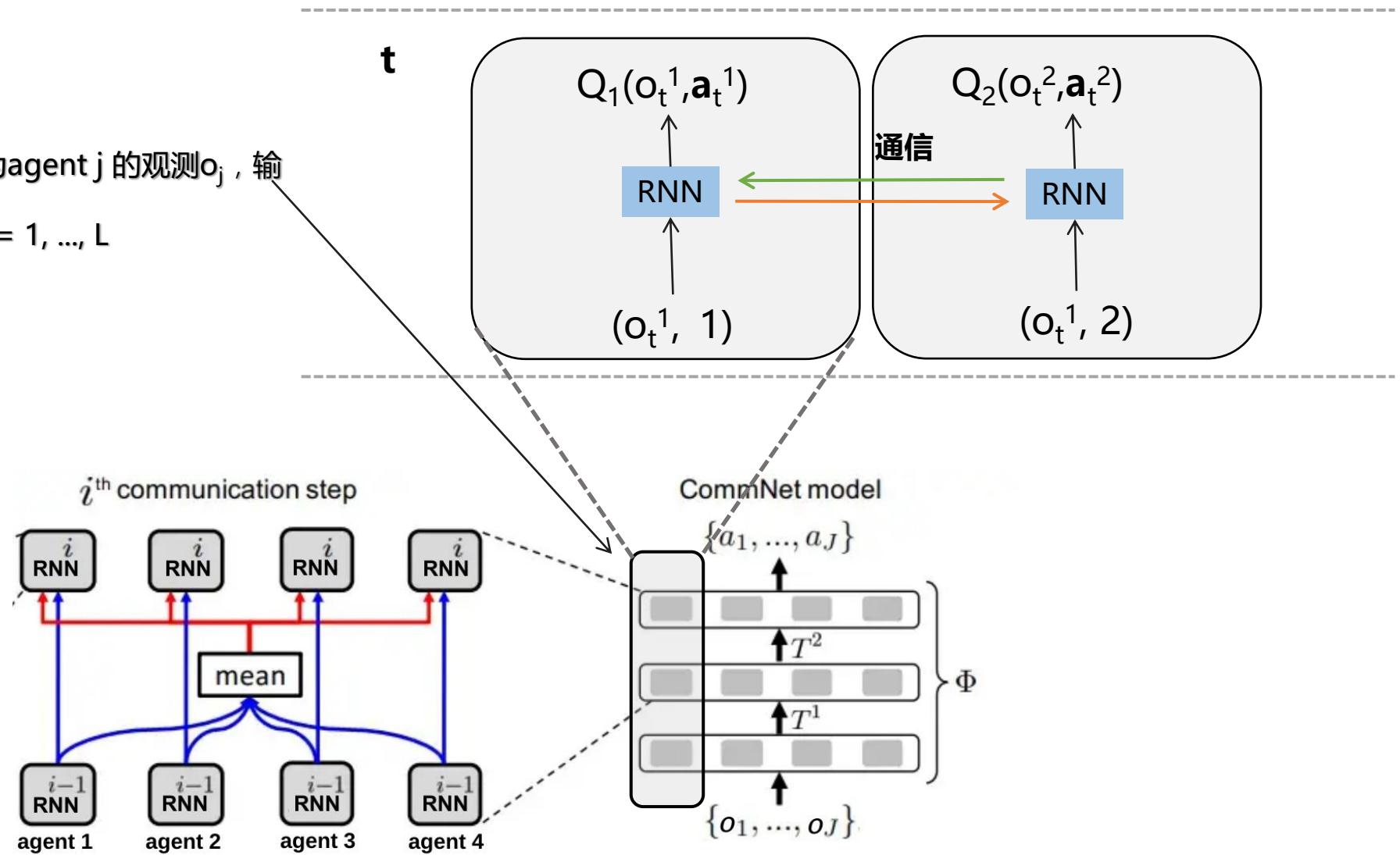
RIAL, DIAL

1. 通信信息 m ，是one-hot编码，不能直接通过反向传播更新
2. 交流的信息落后一步
3. 利用隐变量 h 直接通信？



CommNet

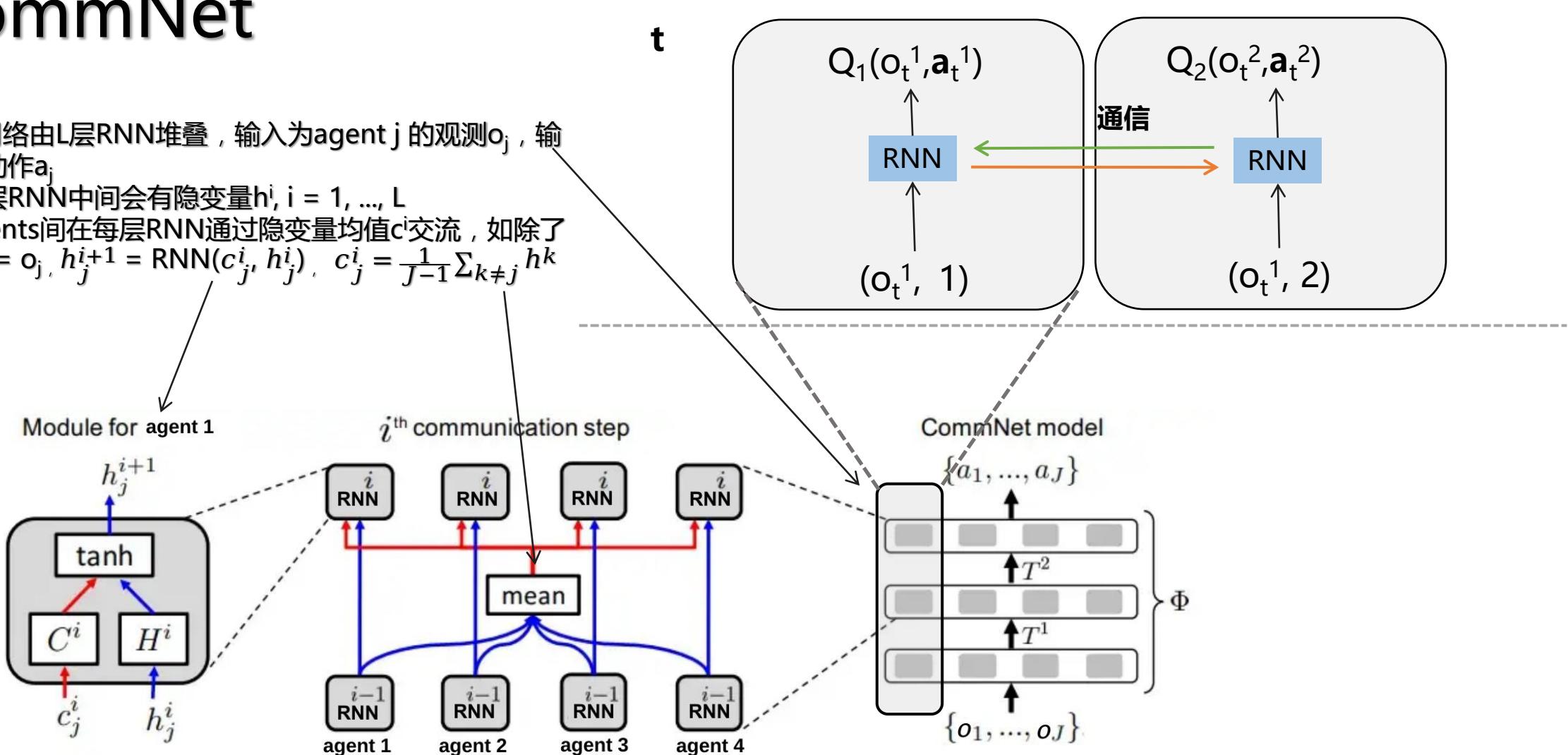
1. Q网络由L层RNN堆叠，输入为agent j 的观测 o_j ，输出动作 a_j
2. 每层RNN中间会有隐变量 h^i , $i = 1, \dots, L$



CommNet

1. Q网络由L层RNN堆叠，输入为agent j 的观测 o_j ，输出动作 a_j
2. 每层RNN中间会有隐变量 h^i , $i = 1, \dots, L$
3. agents间在每层RNN通过隐变量均值 c^i 交流，如除了

$$h_j^0 = o_j, h_j^{i+1} = \text{RNN}(c_j^i, h_j^i), c_j^i = \frac{1}{J-1} \sum_{k \neq j} h_k^i$$



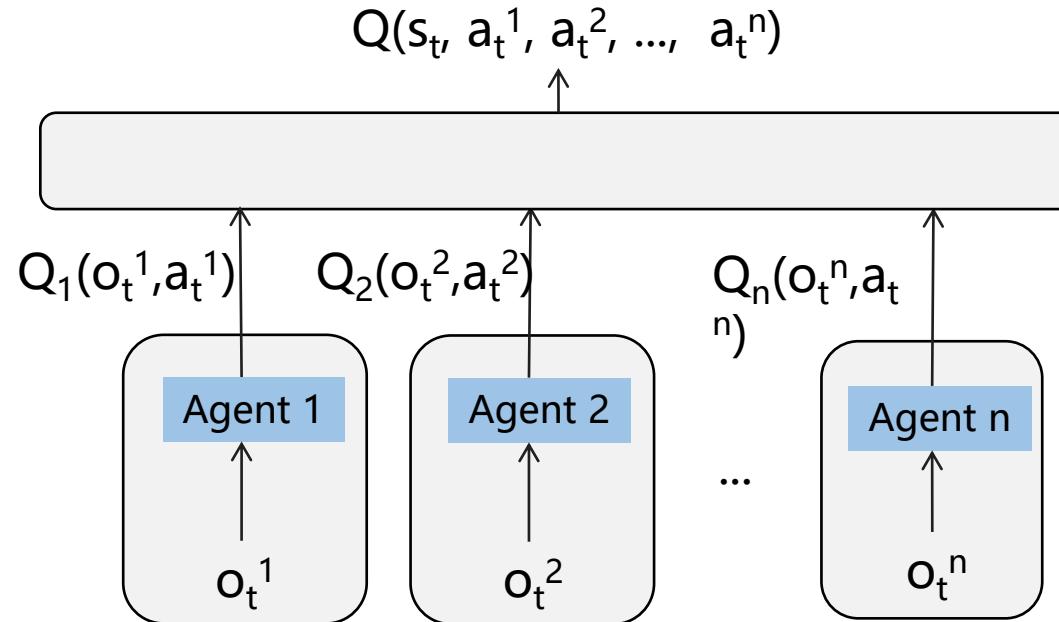
通信学习

- 现存工作: 静态通信用于合作的任务中, 不普遍使用。
- 如何学习一种动态变化的通信拓扑结构 (ATOC)。
- 非完全合作的情况下通信还需要考虑隐私信息的保护。

多智能体系统

- 多智能体强化学习
 - 单智能体深度强化学习（去中心化训练）
 - 通信学习（中心化训练）
 - 协同学习（中心化训练）
 - 基于值函数
 - 基于Actor-Critic
- 多智能体-大语言模型

基于值函数分解的方法



多智能体强化学习中一个核心问题是各智能体的**个体收益**和**团队收益**的关系该如何定义？

- 团队中存竞争、合作等多种复杂关系时，怎么将每个人的个人收益转述为任务需求的团队收益。
- 思想：全局Q值(**团队收益**)分解成局部Q值(**个体收益**)，每个智能体使用局部Q值指导个体学习。

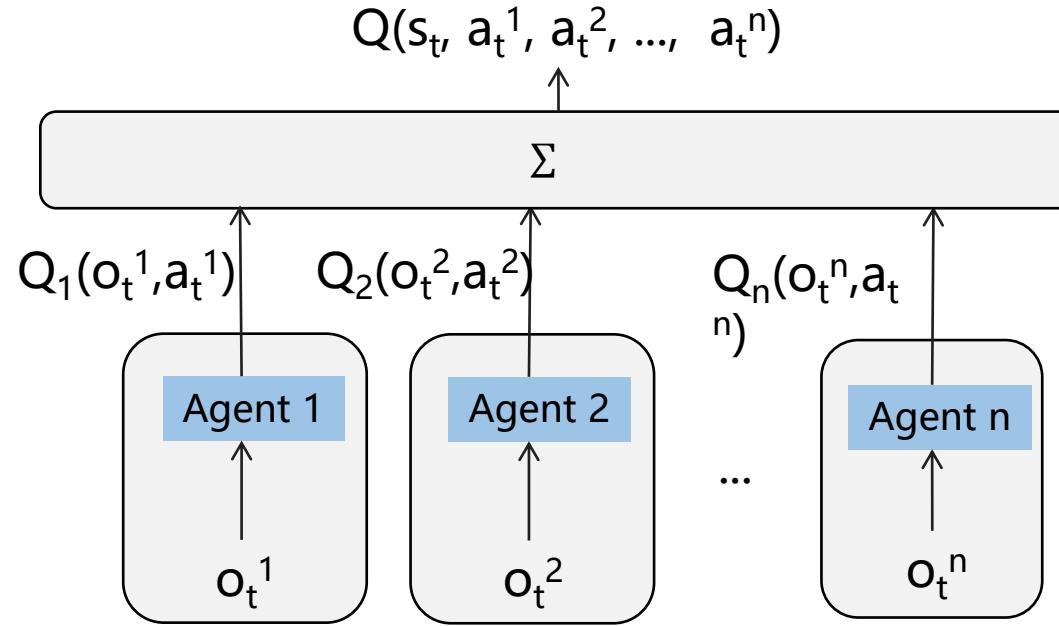
基于值函数分解的方法



多智能体强化学习中一个核心问题是各智能体的**个体收益**和**团队收益**的关系该如何定义？

- 团队中存竞争、合作等多种复杂关系时，怎么将每个人的个人收益转述为任务需求的团队收益。
- 思想：全局Q值(**团队收益**)分解成局部Q值(**个体收益**)，每个智能体使用局部Q值指导个体学习。

VDN

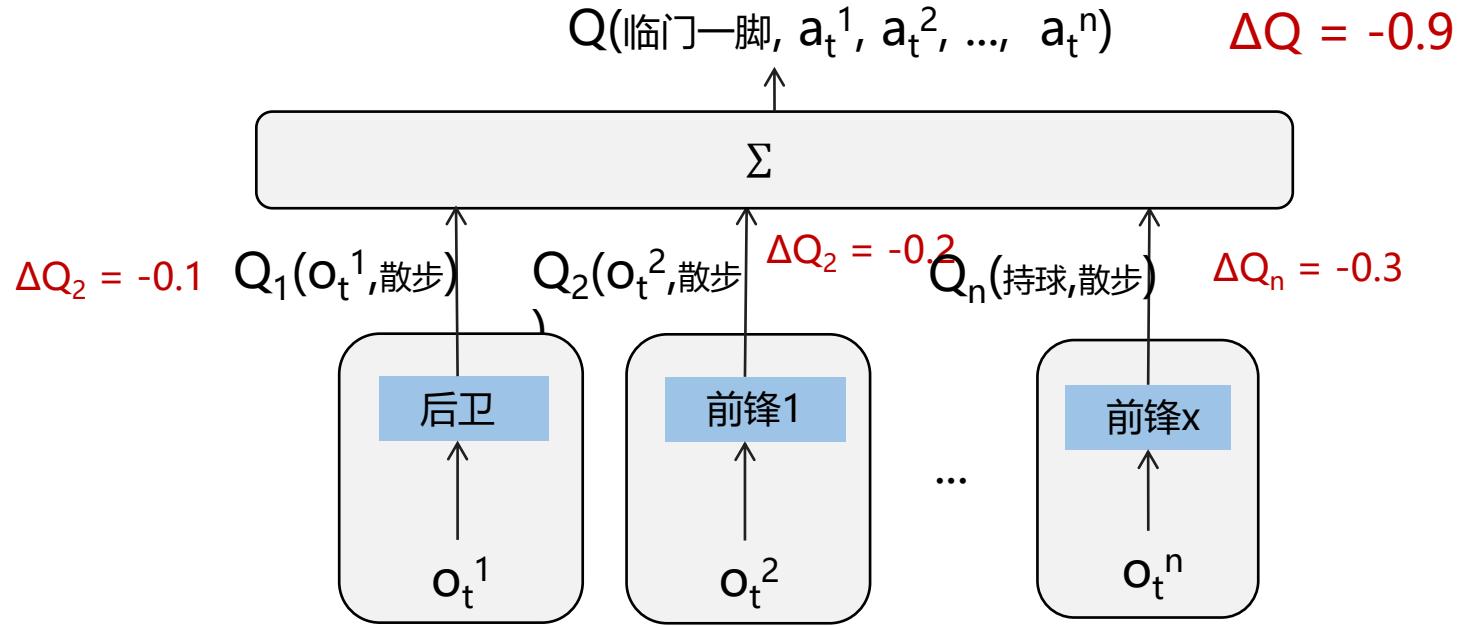


$$Q(s_t, a_t^1, a_t^2, \dots, a_t^n) = \sum_{i=1}^n Q_i(\tau^i, a^i), \quad \tau_i = [o_1^i, o_2^i, \dots, o_t^i]$$

VDN的假设：全局Q值能够被加性分解成局部Q_i值。

- 中心化训练时各智能体在算法的前向传播中将Q_i求和为全局Q值

VDN

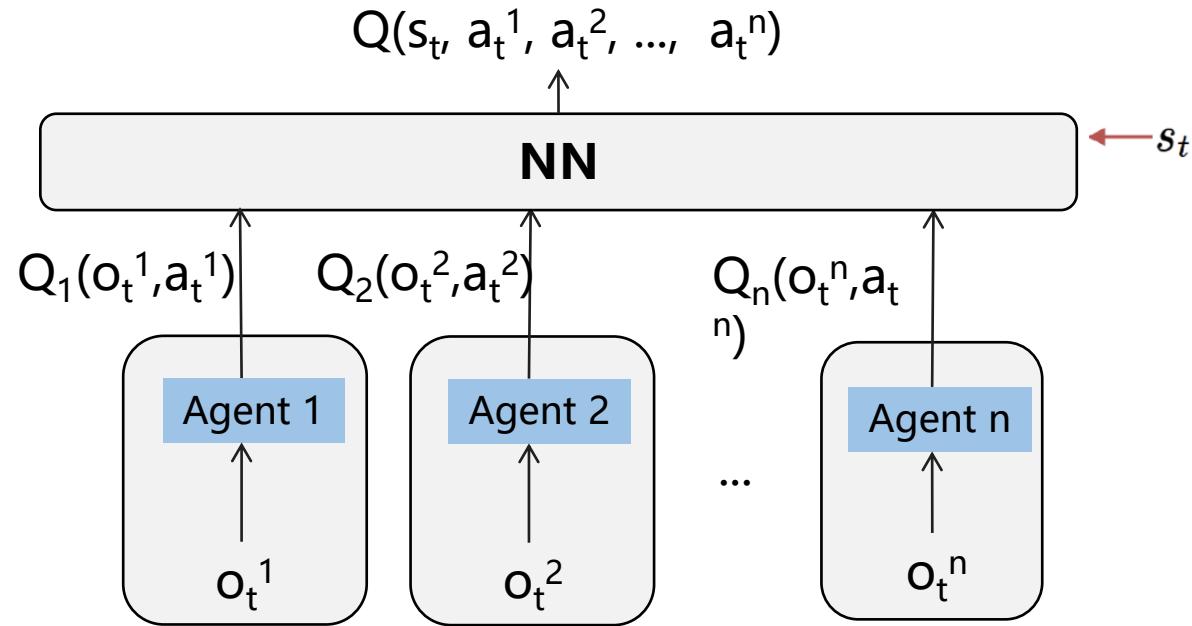


$$Q(s_t, a_t^1, a_t^2, \dots, a_t^n) = \sum_{i=1}^n Q_i(\tau^i, a^i), \tau_i = [o_1^i, o_2^i, \dots, o_t^i]$$

VDN的假设：全局Q值能够被加性分解成局部Q_i值。每时每刻每个智能体的职责分配一样？

- **加性分解值函数：**
 - 能解释完全合作的关系（如两个智能体迷宫中寻找对方）
 - 若是足球或是MOBA等存在有竞争关系的任务，线性关系不能进行合适的拟合。
- 如何进行更复杂的值分解？

QMIX 2020

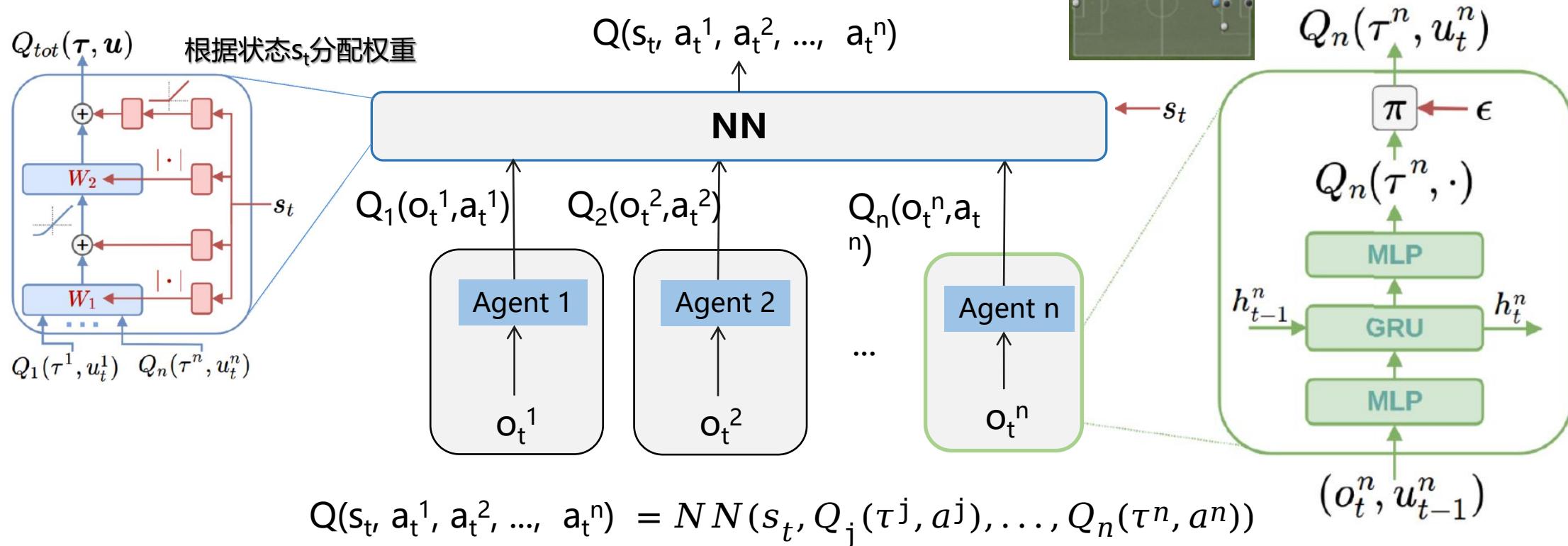


$$Q(s_t, a_t^1, a_t^2, \dots, a_t^n) = NN(s_t, Q_j(\tau^j, a^j), \dots, Q_n(\tau^n, a^n))$$

使用神经网络去拟合个体收益和团队收益的关系：

网络输入各个智能体的**个体收益**和**当前状态** s_t ，输出是**团队总收益**，根据状态 s_t 分配权重。

QMIX 2020



使用神经网络去拟合个体收益和团队收益的关系：

网络输入各个智能体的**个体收益**和**当前状态** s_t ，输出是**团队总收益**，根据**状态** s_t 分配权重。

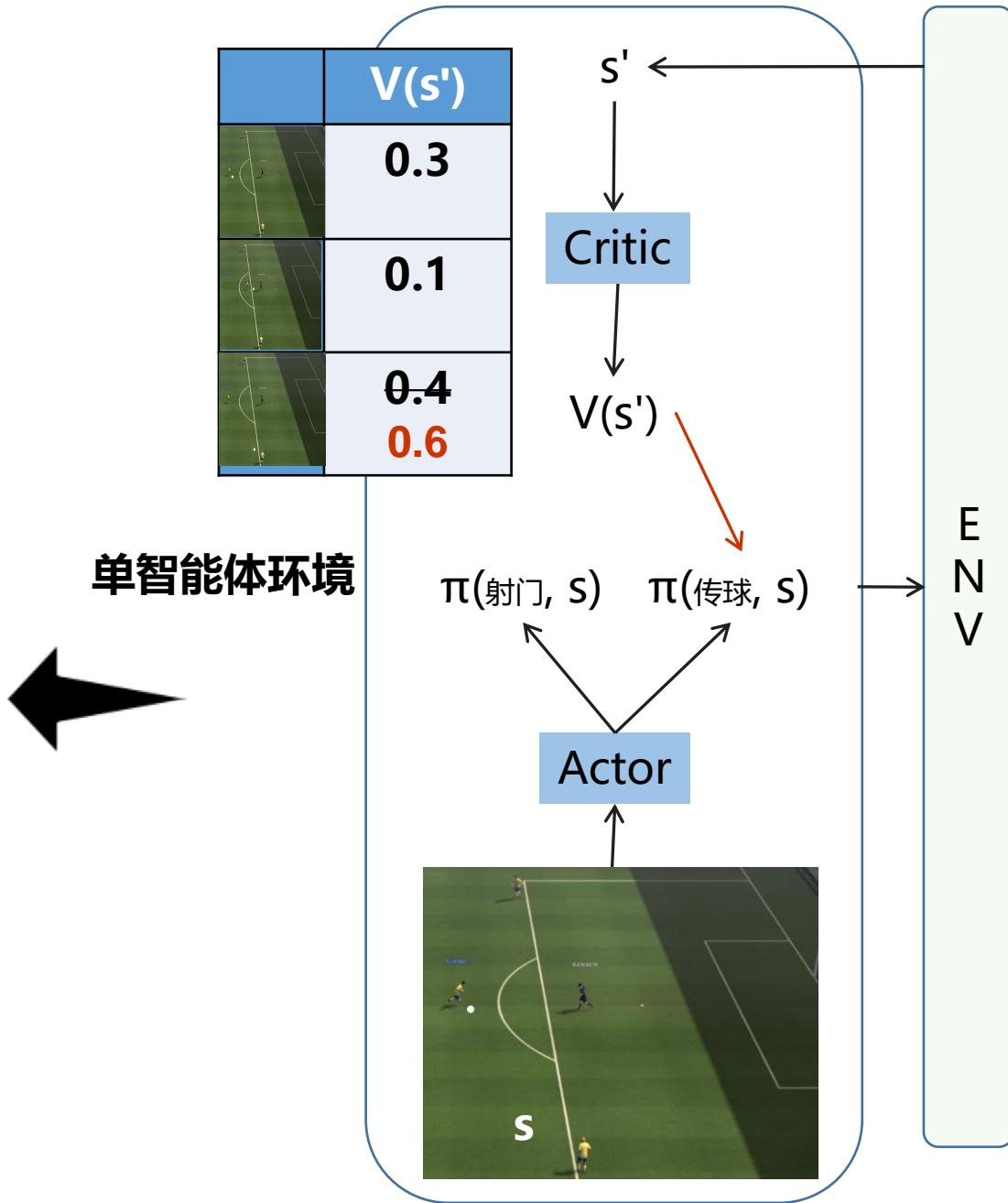
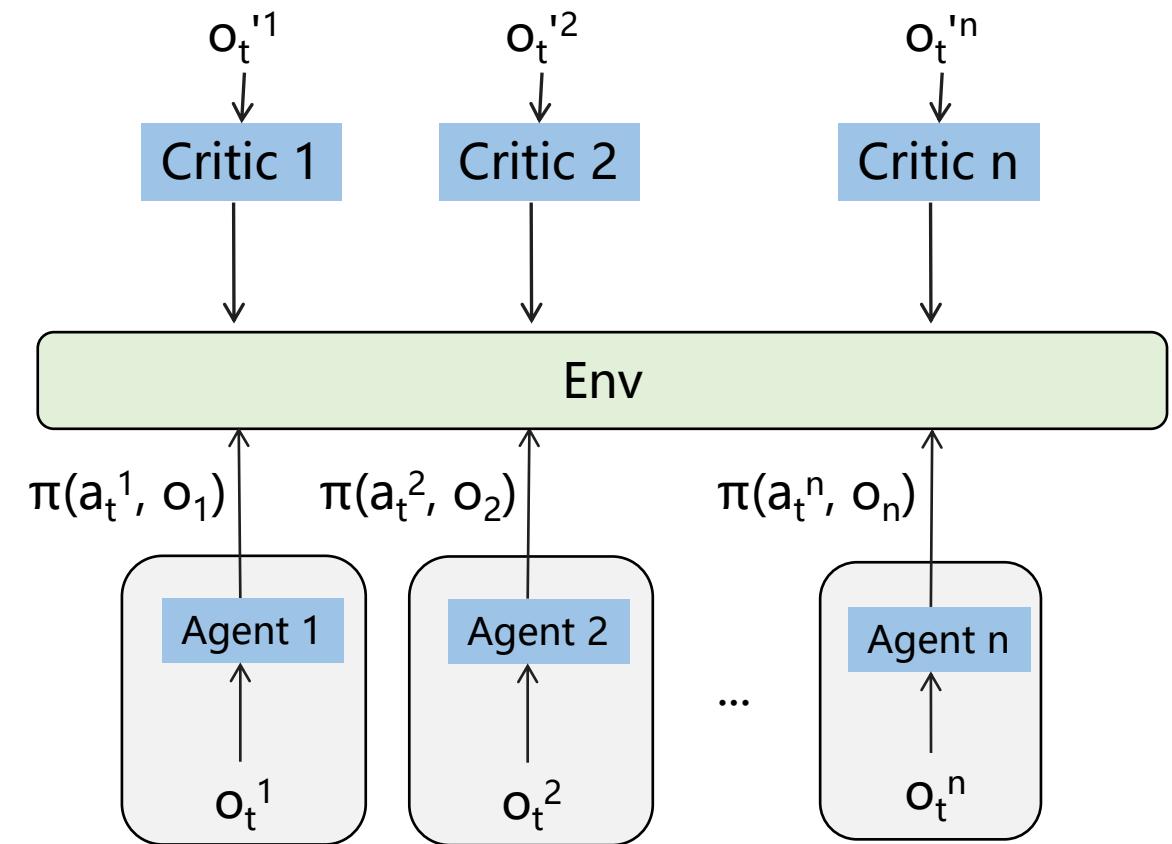
目前基于QMIX的算法仍然是多智能体agent中比较常用的算法。

多智能体系统

- 多智能体强化学习
 - 单智能体深度独立强化学习（去中心化训练）
 - 通信学习（中心化训练）
 - 协同学习（中心化训练）
 - 基于值函数分解
 - 基于Actor-Critic
- 多智能体-大语言模型

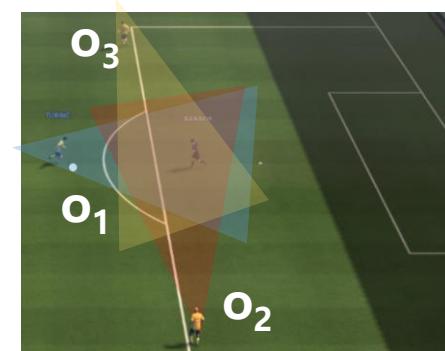
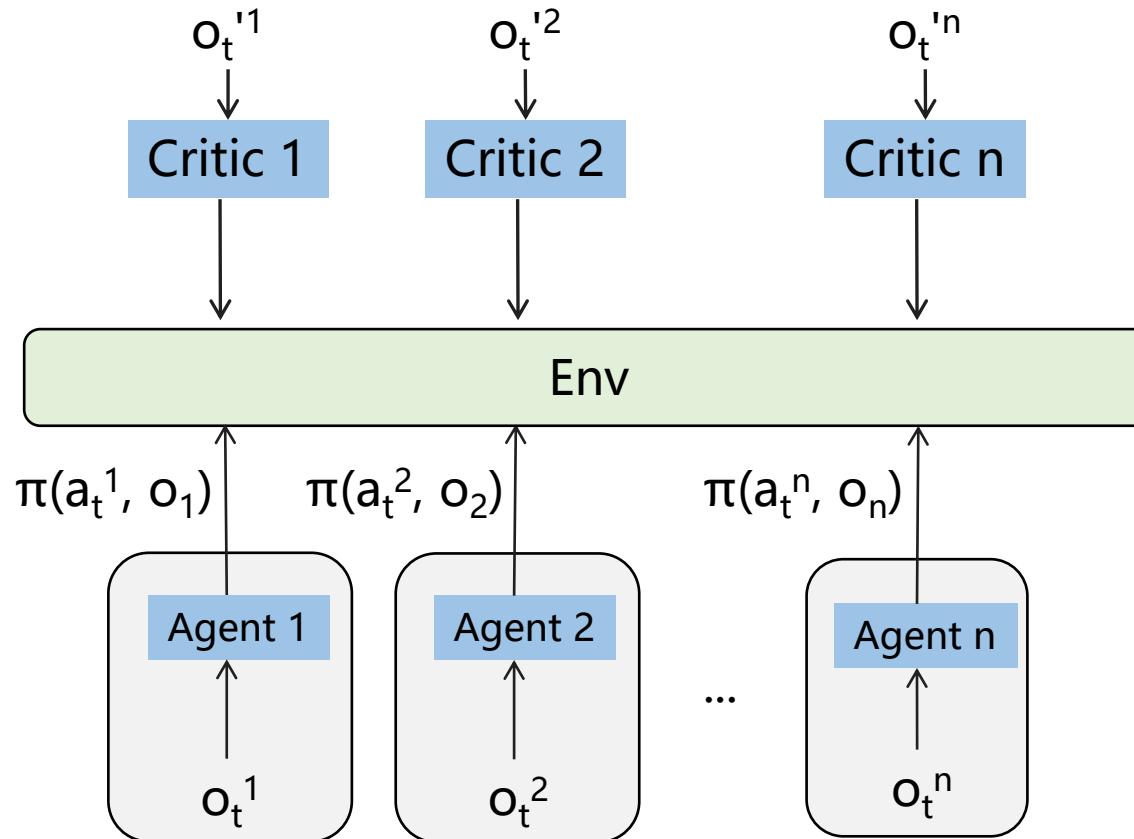
基于Actor-Critic的方法

- Actor学习策略，Critic作为一对指导Actor



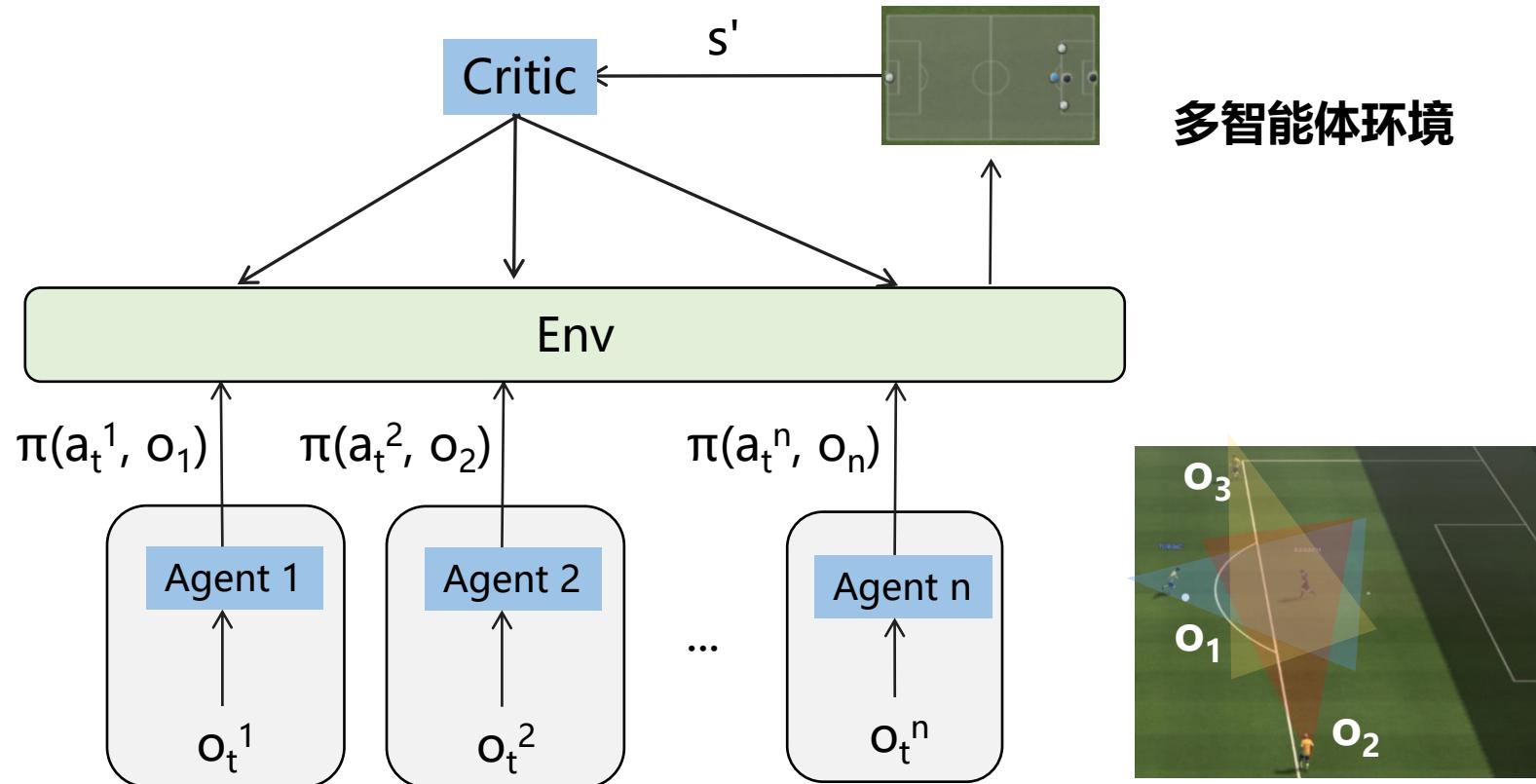
基于Actor-Critic的方法

- Actor学习策略，Critic作为一对指导Actor



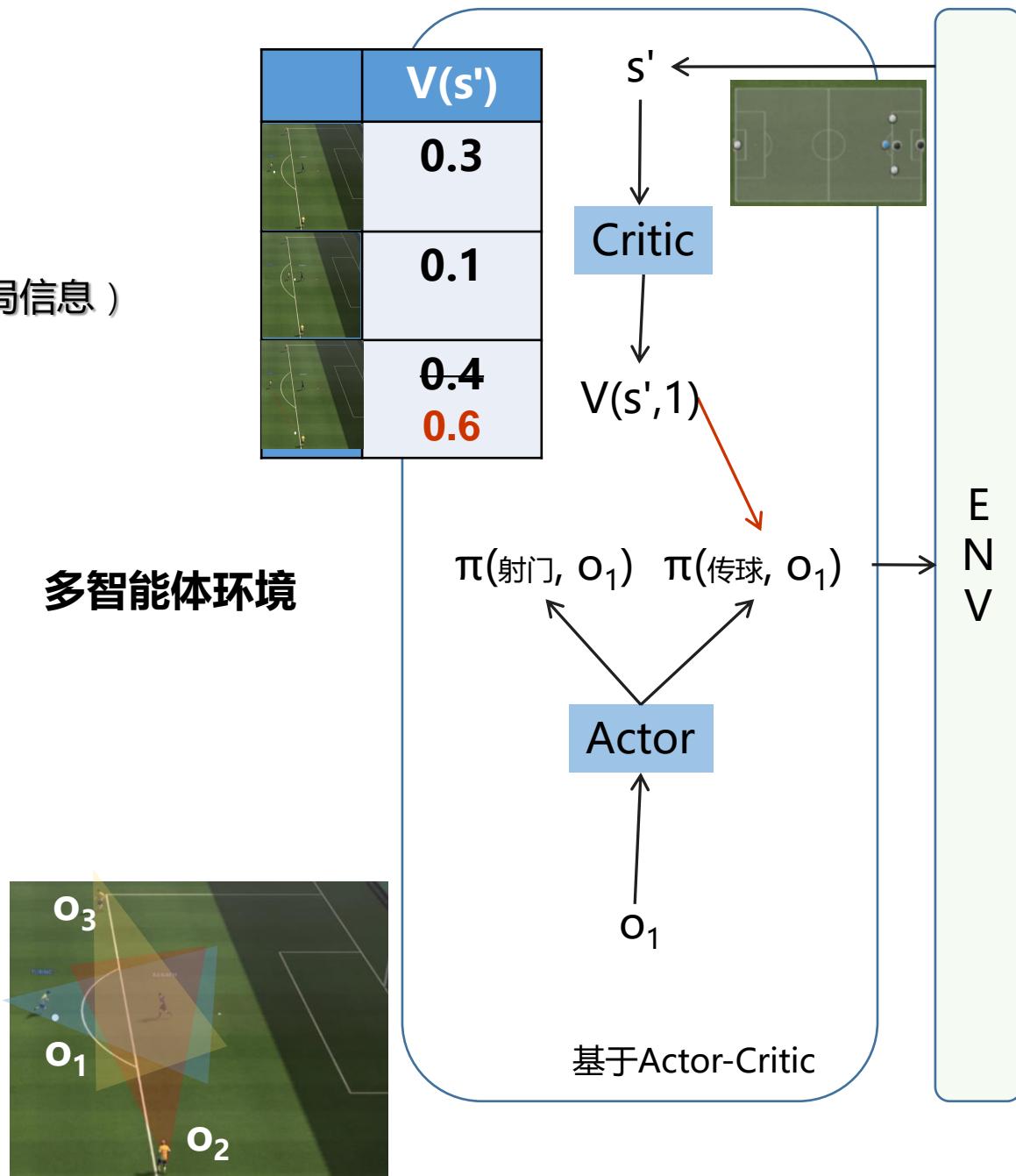
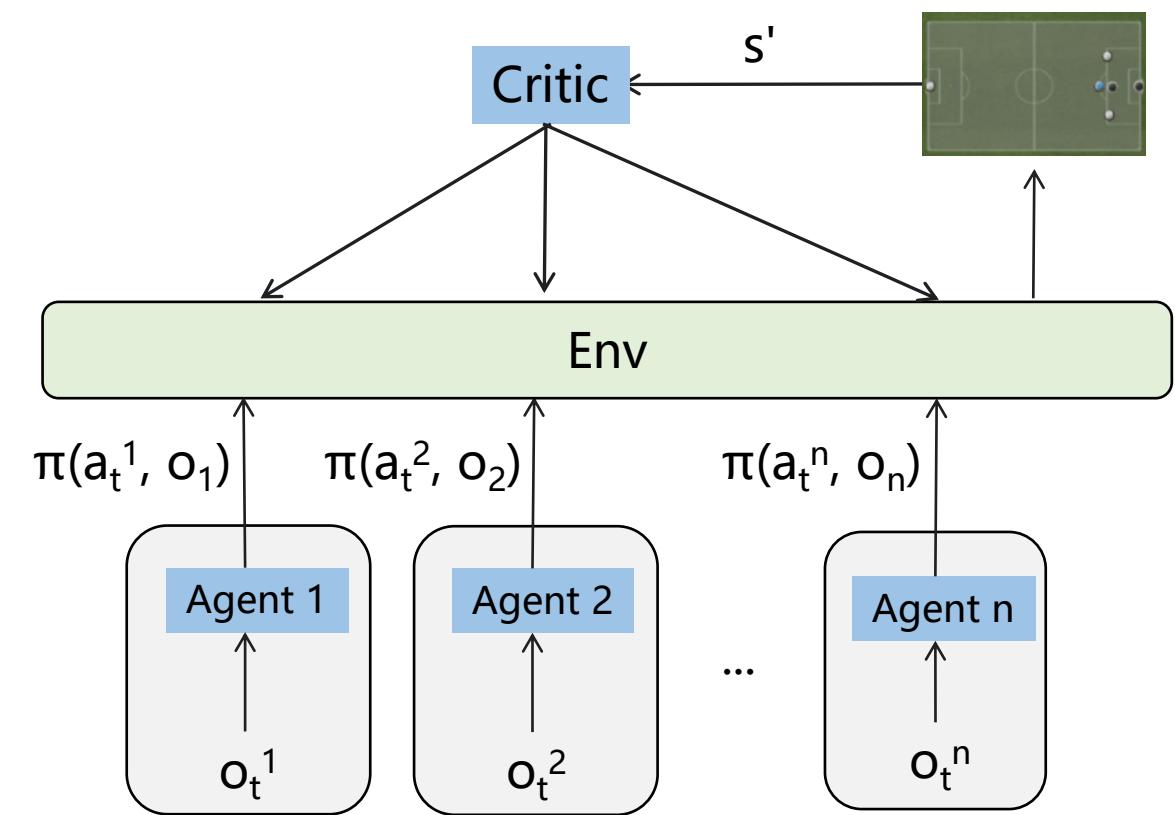
基于Actor-Critic的方法

- 每个agent各一个Actor使用各自的观测进行决策，
- Critic使用状态s（所有人的观测或者系统内置的全局信息）来指导agent
- Critic可以共享网络参数，也可以每个Agent各一个



delete

- 每个agent各一个Actor使用各自的观测进行决策，
- Critic使用状态s（所有人的观测或者系统内置的全局信息）来指导agnet
- Critic可以共享网络参数，也可以每个Agent各一个



基于Actor-Critic的方法

- MADDPG 2017, 基于DDPG算法。
- MAPPO 2022,
 - MAPPO在PPO的基础上加入了针对多智能体设置的tricks,超参数调整简单
 - MAPPO它在诸如星际争霸等任务上都达到了很好的性能。总体来说,MAPPO是迄今为止最成功的多智能体的Actor-Critic算法之一。

MADDPG: Lowe, Ryan, et al. "Multi-agent actor-critic for mixed cooperative-competitive environments." (2017).

MAPPO: Lohse, Oliver, Noah Pütz, and Korbinian Hörmann. "Implementing an online scheduling approach for production with multi agent proximal policy optimization (MAPPO).", 2021.

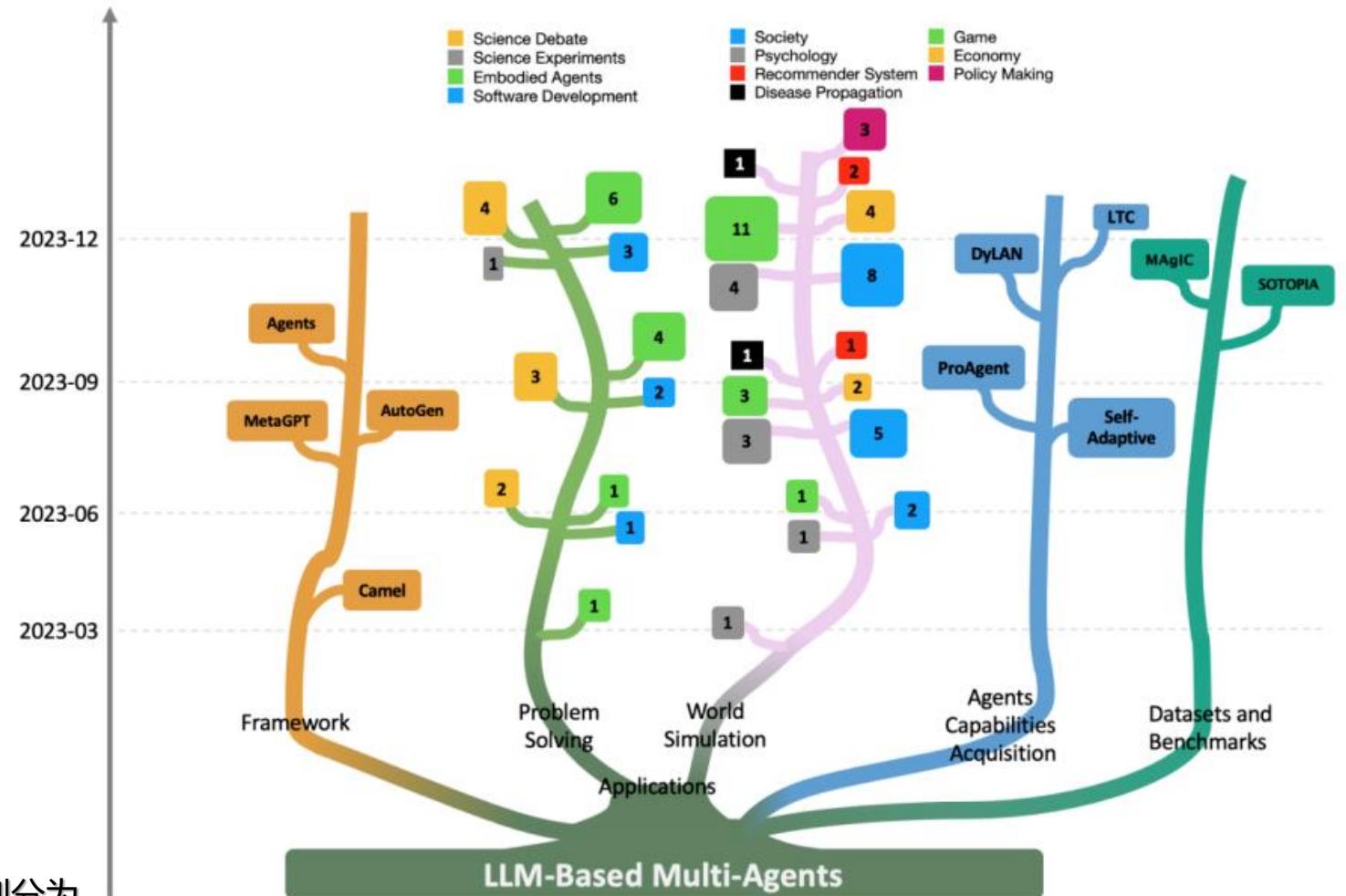
小结

- 通信学习，Agents进行显示通信的以解决非稳定性问题。
 - 但是随着智能体数量增加,需要更多的样本以学会通信完成合作。[]
 - 现存工作主要是在**静态通信**, 实现Agents**动态地选择通信对象,通信内容的工作较少**[]。
- 协作学习虽然也是中心训练到时Agents间没有显式通信
 - 通过在**联合值函数或Critic网络中加入全局信息**使得环境稳定并学会合作与竞争。
- 总体上MARL领域这两年领域内的有价值的文章似乎在变少。比较流行的CTDE框架, 在工业界似乎少有类似的落地场景。
- 目前MARL主要运用在游戏、某些场景用作优化、推荐系统的机制设计。

LLM Multi Agents

- 以前的研究中, 都假设Agent基于简单和启发式的策略函数, 并在**孤立和受限的环境中学习**。
- 人类的学习过程有很大不同, 个体可以从**更复杂、广泛的环境中学习**。
- 随着大型语言模型(LLMs)在广泛的任务中取得了显著的成功, 且LLMs展现出一定的规划和推理能力, 它们被用作自主Agent能自主完成许多任务。
- 目前多智能体(LLM-MA)的研究虽仍处于**早期阶段**, 但是也出现了不少相关工作。

LLM Multi Agents



目前的工作可划分为
使用多Agents解决复杂问题(应用), 多Agents进行世界模拟(应用), Agent的能力获取, 多Agents框架, 数据集和基准这5个方向。

多智能体系统

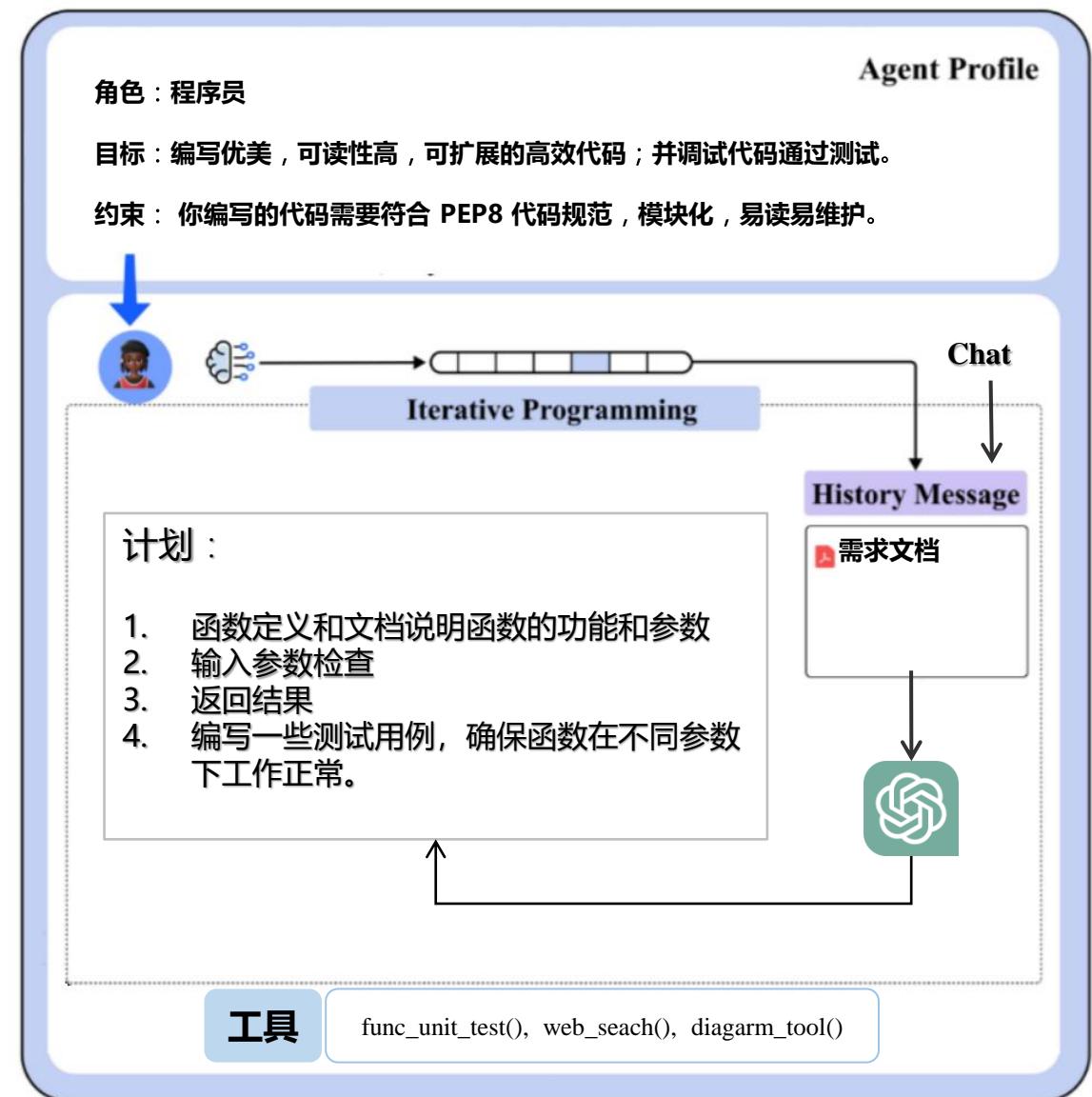
- 多智能体强化学习
 - 单智能体深度强化学习（去中心化训练）
 - 通信学习（中心化训练）
 - 协同学习（中心化训练）
 - 基于值函数
 - 基于Actor-Critic
- 多智能体-大语言模型
 - LLM Agents组件
 - LLM Multi Agents通信
 - Agent能力获取

LLM Multi Agents

如果使用的Agent背后是同一个大模型, 如果一个Agent不能很好地工作, 那么设置多Agents帮助大吗?

LLM Agents组件

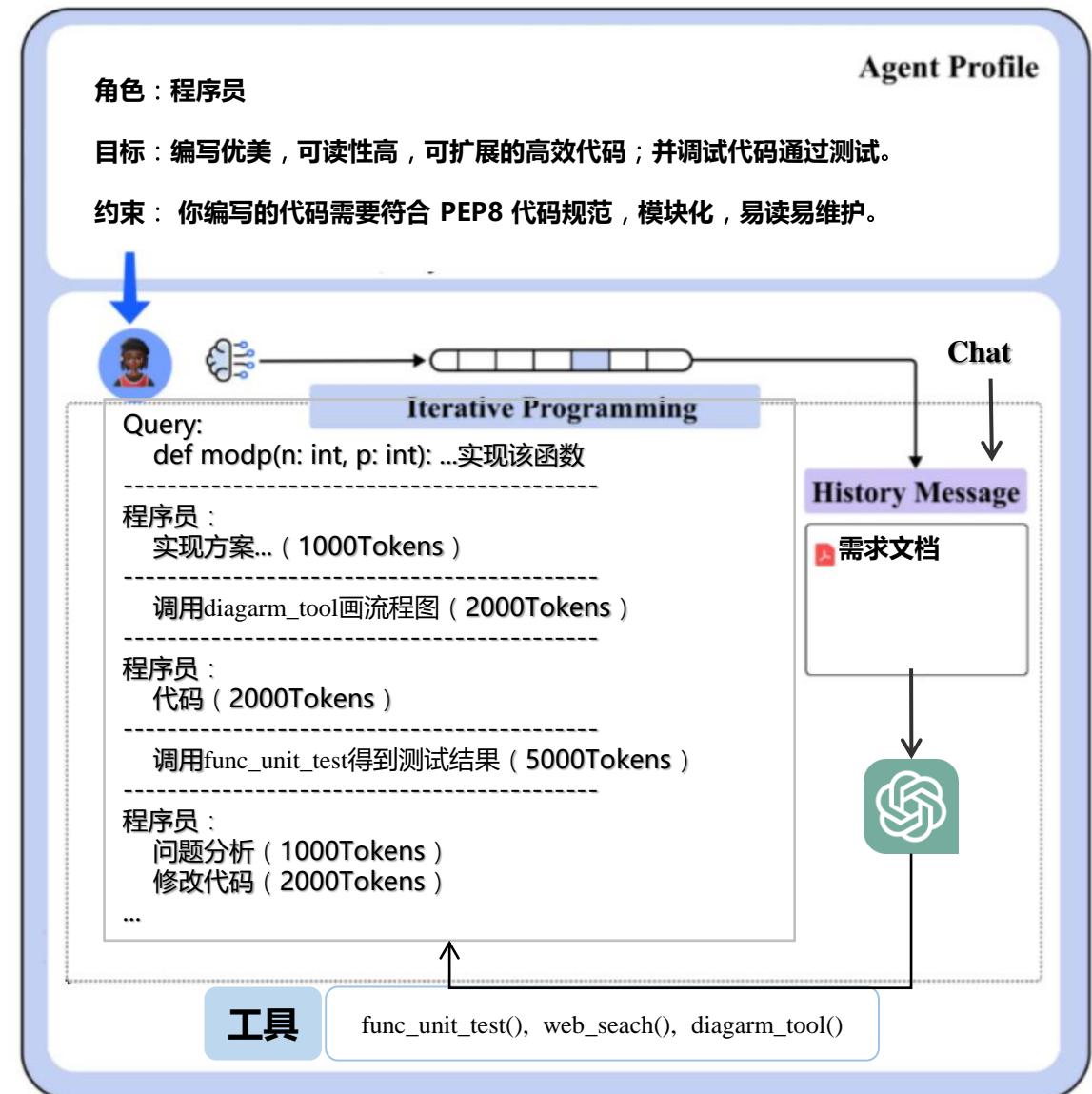
- 内存 (Memory)：
 - 数据库 (长期记忆)
 - 上下文信息 (短期记忆)
- 角色 (Profile)：封装基于领域专业知识的特定技能、行为和工作流程。每个角色都有明确的定义，包括职责、目标和限制。
- 行动 (Actions)：智能体执行的具体任务，使用提供的工具，完成子任务并生成输出；
- 规划 (Planning)：可以要求LLM基于任务，形成一个解决方案。按照计划一步步执行，每走一步并根据环境反馈，进行推理完善解决方案。



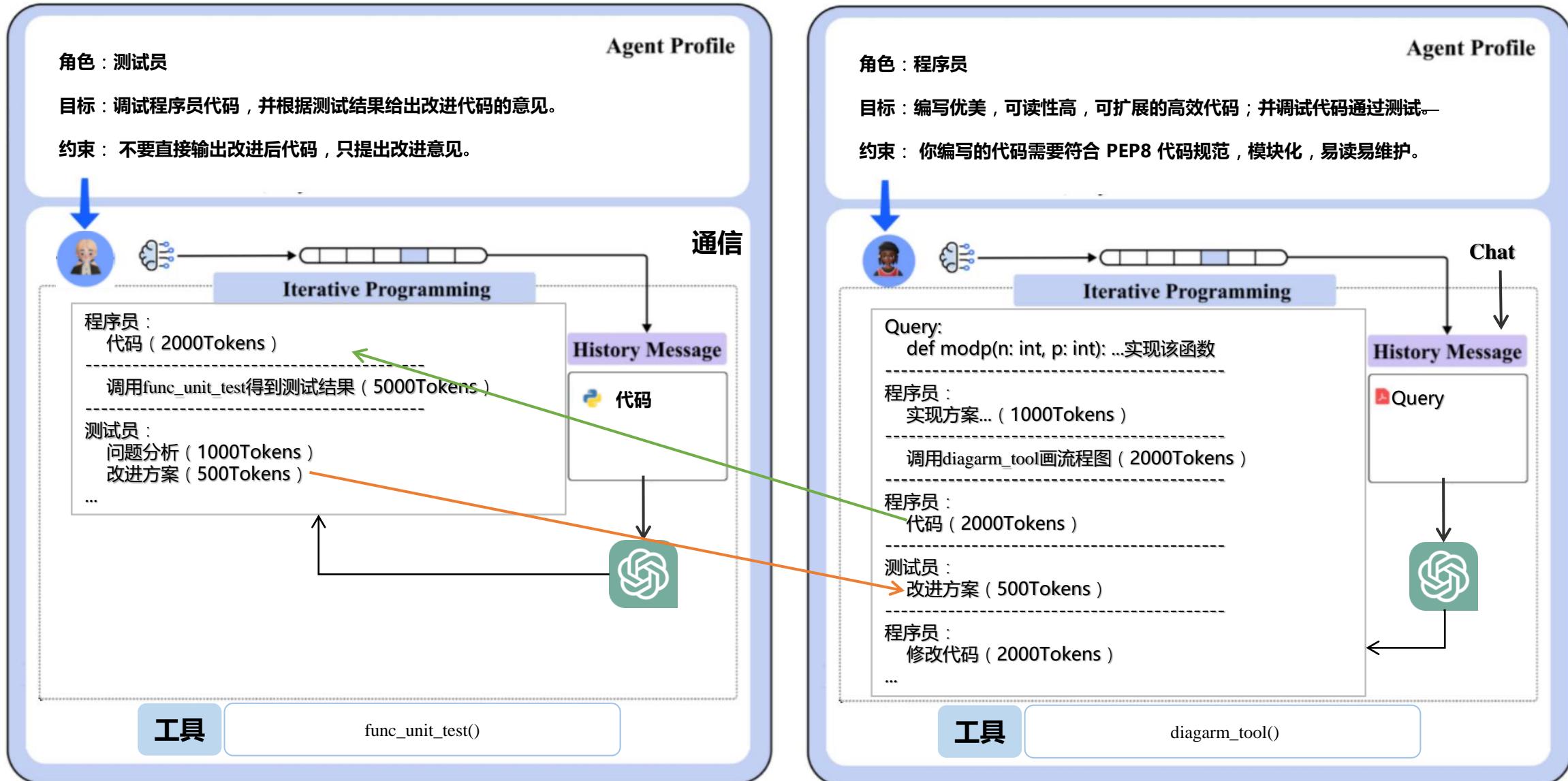
LLM Agents组件

多Agents设计允许您将复杂的问题划分为可处理的工作单元, 这些工作单元可以由专门的Agent负责 :

- 有各自的任务分工, 以及单独的提示 (单独说明和少量示例) 。
 - 记忆中只存各自子任务相关信息。
 - 分组工具, 专注于自己任务相关的工具。
-
- 多个自主Agent相互沟通, 可以解决更动态和复杂的任务。
 - 可以允许你单独测试、评估和改进每个Agent, 而不会影响到整个系统
-



LLM Agents组件



LLM Agents组件

多Agents设计允许您将复杂的问题划分为可处理的工作单元, 这些工作单元可以由专门的Agent负责 :

- 有各自的任务分工, 以及单独的提示 (单独说明和少量示例) 。
- 记忆中只存各自子任务相关信息。
- 分组工具, 专注于自己任务相关的工具。
- **多个自主Agent相互沟通, 可以解决更动态和复杂的任务。**
- **可以允许你单独测试、评估和改进每个Agent, Agent根据反馈动态学习和进化获取所需能力**

多智能体系统

- 多智能体强化学习
 - 单智能体深度强化学习（去中心化训练）
 - 通信学习（中心化训练）
 - 协同学习（中心化训练）
 - 基于值函数
 - 基于Actor-Critic
- 多智能体-大语言模型
 - LLM Agents组件
 - LLM Multi Agents通信
 - Agent能力获取

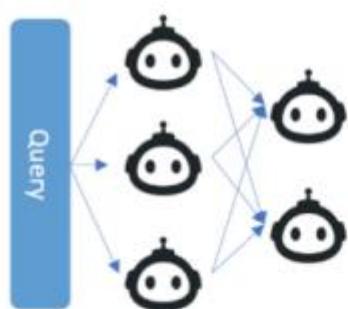
LLM Multi Agents通信

通信范式：

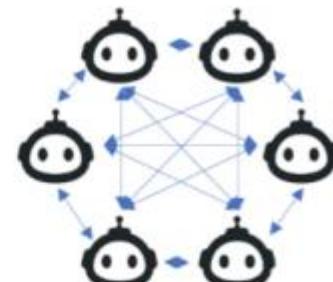
- **合作范式**：Agents共同努力实现共享的目标或目标，通常交换信息以增强集体解决方案。
- **辩论范式**：Agents进行争论性互动时使用，提出并捍卫自己的观点或解决方案，并批评他人的。这种范式适合达成共识或更精炼的解决方案。

LLM Multi Agents通信

通信结构：

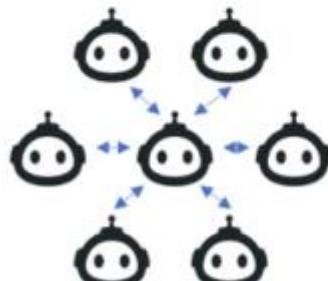


Layered



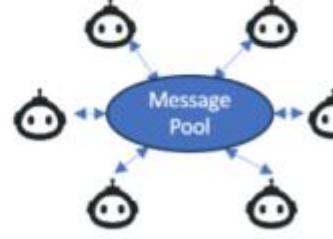
Decentralized

每个agent的每个消息都广播出去



Centralized

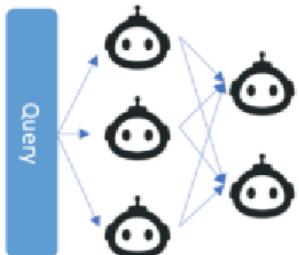
一个agent的作用为中继搜集消息，再分发给有需要的agents



Shared
Message Pool

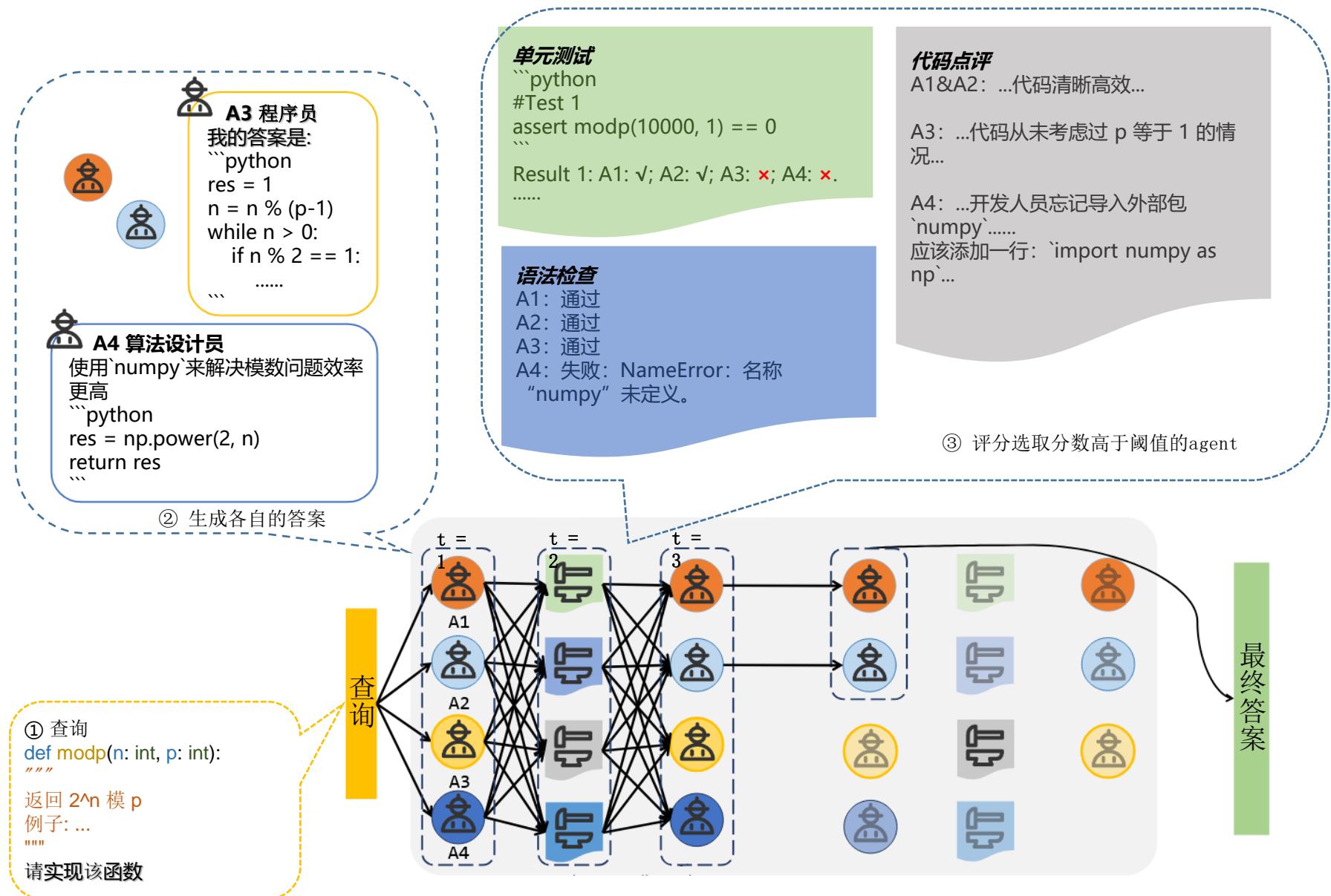
Figure 3: The Agent Communication Structure.

DyLAN :



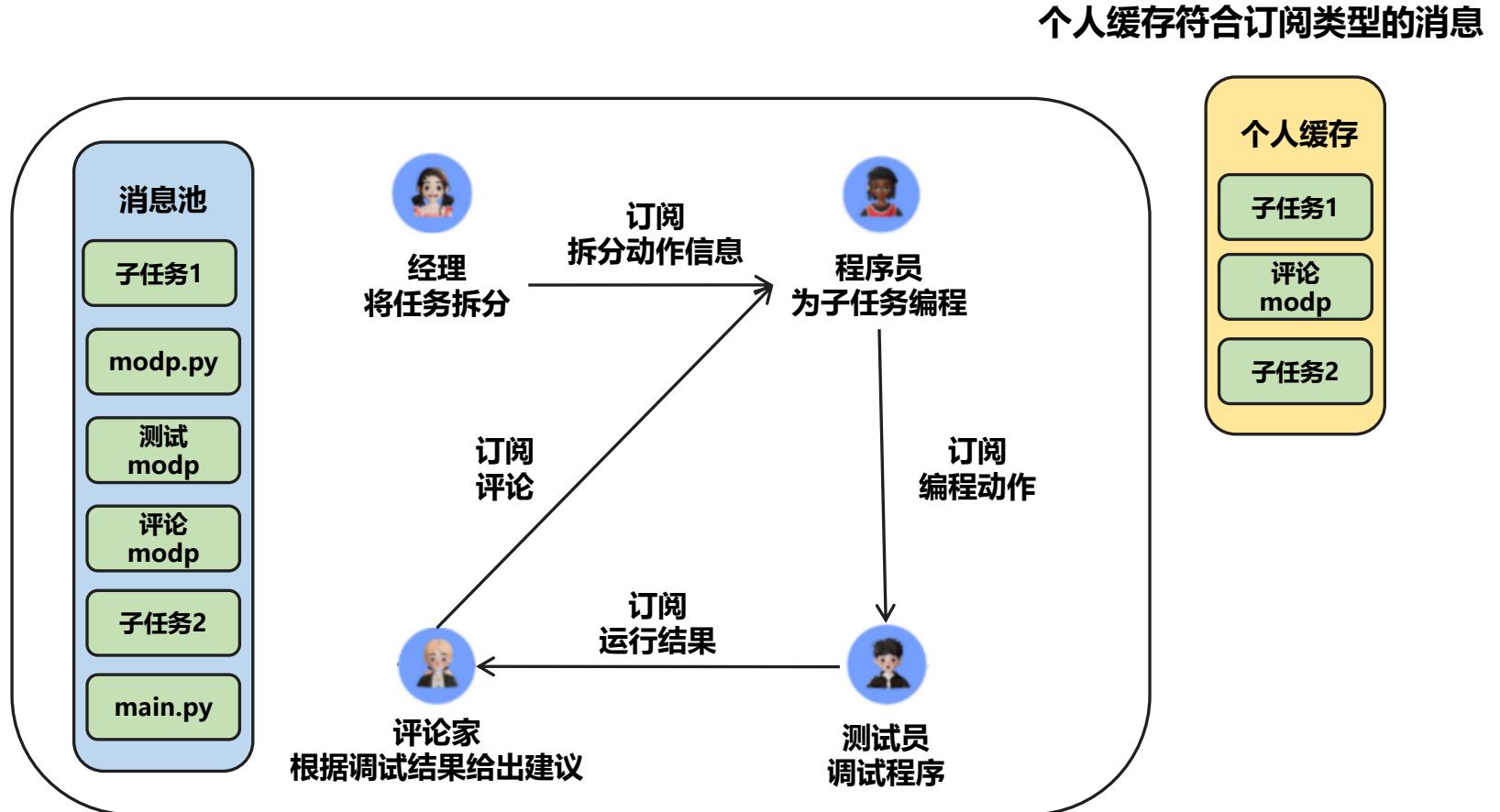
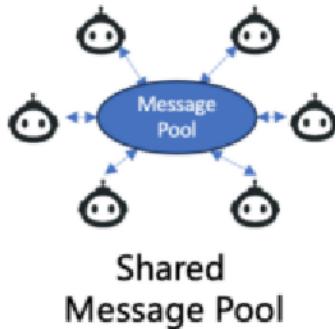
Query

Layered



LLM Multi Agents通信

MetaGPT :



多智能体系统

- 多智能体强化学习
 - 单智能体深度强化学习（去中心化训练）
 - 通信学习（中心化训练）
 - 协同学习（中心化训练）
 - 基于值函数
 - 基于Actor-Critic
- 多智能体-大语言模型
 - LLM Agents组件
 - LLM Multi Agents通信
 - Agent能力获取

Agent能力获取

Agent能力获取指Agent能够动态学习和进化。Agent应该从反馈中学习以增强所需能力,策略。

反馈包括:

1. 来自环境的反馈

- 如软件开发中(从代码解释器那里获得反馈)
- 具身多Agents系统(从现实世界或模拟环境中获得反馈[1])

2. 来自Agents互动的反馈

- 来自其他Agent的论断或来自Agent之间的通信
- 如科学辩论这样的问题解决场景中很常见, Agent通过通信学习批判性地评估和完善结论
 - 在世界模拟场景(如游戏模拟)中, Agent根据其他Agent之间的先前互动学习完善策略。

3. 来自人类的反馈

- 反馈直接来自人类, 保证智能体系统与人类价值观和偏好保持一致
- 这种反馈在大多数"人类集成到特定任务的开发循环流程"的过程中被广泛使用[2]。

Agent能力获取

LLM-MA系统中的Agent主要通过三种主要解决方案增强能力：

1. 记忆

- 大多数LLM-MA系统利用记忆模块来调整Agent的行为。Agent将来自先前互动和反馈的信息存储在它们的记忆中。
- 在执行行动时，它们可以检索相关的、有价值的记忆，特别是那些包含过去类似目标的成功行动的记忆以提升其能力。

1. 自我进化。

- Agent可以通过修改自己(例如改变初始目标和规划策略)来动态自我进化。
- ProAgent它通过预测队友的决策并根据Agent之间的通信日志动态调整每个Agent的策略，促进相互理解和提高协作规划能力。

2. 动态生成。在某些场景中，系统可以在其运行期间即时生成新的Agent。

- 中的系统能够有效地扩展和适应任务，根据任务当前的状态选择或添加新的Agents处理当前任务。

自我进化

自我进化-记忆/修改目标和规划策略:

Agent根据反馈或通信日志

可以通过修改自己(例如改变初始目标和规划策略)来
动态自我进化。



玩家4 (好人)

我才是真正的预言家，我使用我的能力验证了玩家3的身份，发现他是狼人。然而，玩家3却谎称自己是预言家，也就是我的角色。这清楚地表明，玩家3是狼人，他想迷惑村民。

(当玩家3谎称是预言家时，真正的预言家玩家4站出来反对狼人。)



玩家5 (好人)

我投票淘汰玩家3

大多数玩家对玩家3产生怀疑



玩家6 (狼) Thought :

我的思考过程如下：

1. 大多数玩家投票淘汰玩家3，他是我的狼人搭档。如果玩家3被淘汰，我将更难赢得游戏。
2. 但是，如果我投票反对大多数人，可能会引起我的怀疑。
3. 因此，我应该投票淘汰玩家3，以便融入村民中，避免引起怀疑。这是一个艰难的决定，但对于我的生存和赢得游戏来说，这是必要的。



玩家6 (狼)

我投票淘汰玩家3

玩家6 (狼) 权衡了利弊，修改目标和规划策略
决定开始卖队友。



player3



player4



player5



player6

记忆

记忆：

- 将互动和反馈的信息存储在它们的记忆中。
- 下次执行行动时，类似目标的行动时再检索出来加以利用[]。

请总结这次游戏的成功经验，以便后续进行借鉴（不超过200字）

这次游戏的成功经验在于灵活应变和策略调整。当我的狼人搭档被预言家揭露时，我迅速评估了形势，决定与队友划清界限，以避免引起怀疑。我选择顺应大多数玩家的意见，融入好人阵营，降低自身风险。随后，我继续保持低调，利用其他机会逐步削弱好人阵营，最终赢得游戏。关键在于及时调整策略，伪装成好人，避免引起怀疑，同时在合适的时机采取行动。



玩家4(好人)

我才是真正的预言家，我使用我的能力验证了玩家3的身份，发现他是狼人。然而，玩家3却谎称自己是预言家，也就是我的角色。这清楚地表明，玩家3是狼人，他想迷惑村民。

(当玩家3谎称是预言家时，真正的预言家玩家4站出来反对狼人。)



player5

玩家5(好人)

我投票淘汰玩家3

大多数玩家对玩家3产生怀疑



玩家6(狼) Thought :

我的思考过程如下：

1. 大多数玩家投票淘汰玩家3，他是我的狼人搭档。如果玩家3被淘汰，我将更难赢得游戏。
2. 但是，如果我投票反对大多数人，可能会引起我的怀疑。
3. 因此，我应该投票淘汰玩家3，以便融入村民中，避免引起怀疑。这是一个艰难的决定，但对于我的生存和赢得游戏来说，这是必要的。



玩家6(狼)

我投票淘汰玩家3

玩家6(狼人) 权衡了利弊，修改目标和规划策略决定开始卖队友。



player3



player4



player5



player6

自我进化

通信学习(LTC)

- 在多智能体环境中，智能体能够持续进行探索和交互，通过各种通信模式收集轨迹。
- LTC 利用从智能体探索活动中获得的数据来微调智能体。
- 此过程使智能体能够自主适应各自的环境，从而无需人工监督。



玩家4(好人)

我才是真正的预言家，我使用我的能力验证了玩家3的身份，发现他是狼人。然而，玩家3却谎称自己是预言家，也就是我的角色。这清楚地表明，玩家3是狼人，他想迷惑村民。



player5

玩家5(好人)

我投票淘汰玩家3



玩家6(狼) Thought :

我的思考过程如下：

- 大多数玩家投票淘汰玩家3，他是我的狼人搭档。如果玩家3被淘汰，我将更难赢得游戏。
- 但是，如果我投票反对大多数人，可能会引起我的怀疑。
- 因此，我应该投票淘汰玩家3，以便融入村民中，避免引起怀疑。这是一个艰难的决定，但对于我的生存和赢得游戏来说，这是必要的。



玩家6(狼)

我投票淘汰玩家3



player3



player4



player5



player6

狼人reward+1，好人reward-1

自我进化

通信学习(LTC)

- 使用更强的模型进行对话搜集轨迹用于微调

问题：肖恩有五个玩具。圣诞节时，他从爸爸妈妈那里分别得到了两个新玩具。他现在有多少个玩具？

Agent :

肖恩一开始有 5 个玩具。如果他有 2 个新玩具，那么就是 $5+2=7$ 。答案是 7。

GPT4:

不正确。（失败，奖励 -1）

问题：莎拉有 9 支笔，她给鲍勃和汤姆各 3 支笔。莎拉还剩下多少支笔？

Agent :

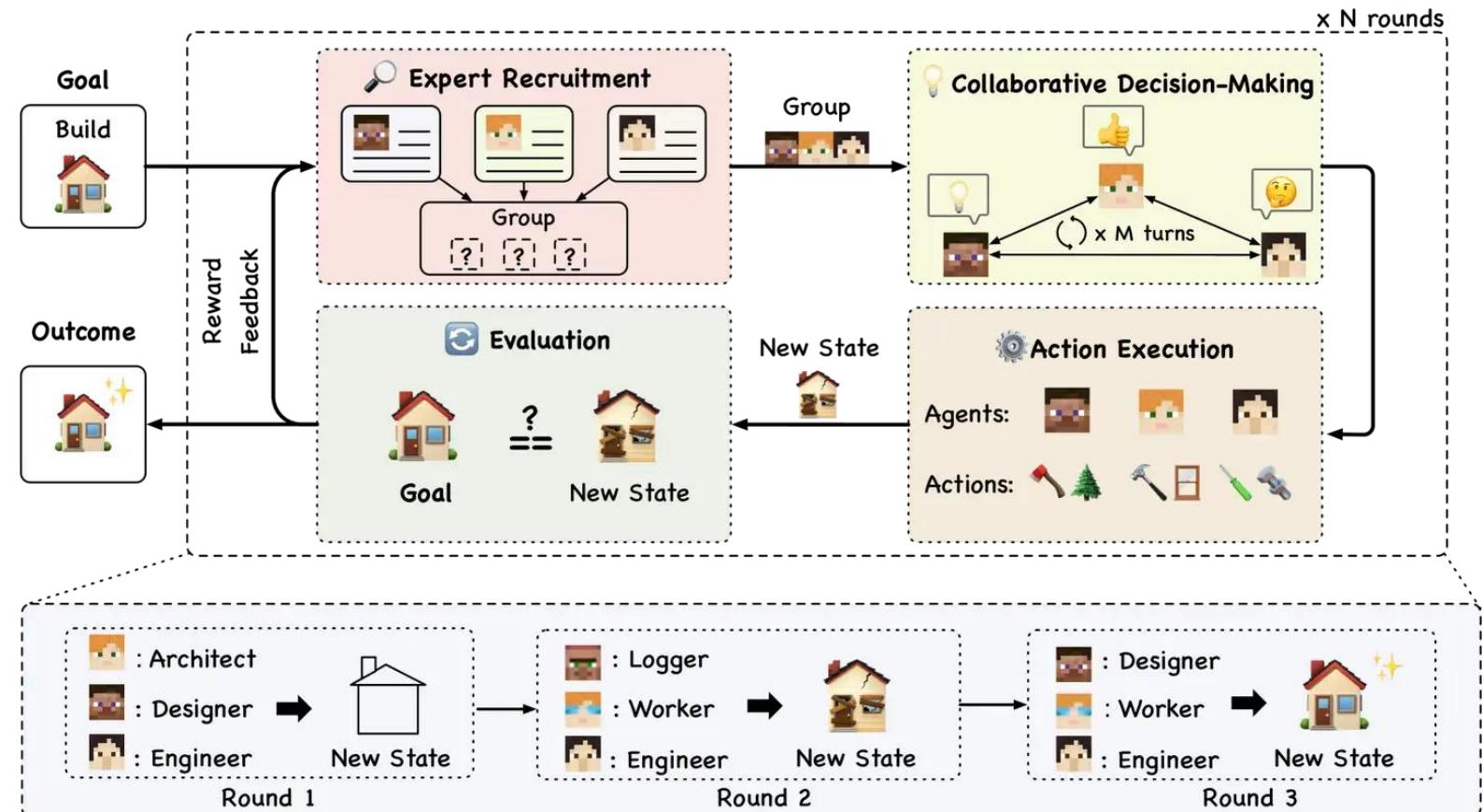
如果他从妈妈那里得到 2 个玩具，从爸爸那里得到 2 个玩具，那么就是 $5+2+2=9$ 。

GPT4:

正确！您已经学会了这类问题。（成功，奖励 +1）

Agent能力获取

动态生成



- Round1 初始时选择工程师，设计师，架构师开始设计房子
- Round2 发现木材不够，选择工程师，伐木工人，工人 收集材料

Agents通信方面,当前的LLM-MA的工作中,主要采用合作、辩论和竞争三种通信范式

- 有分层通信、去中心化通信、中心化通信和共享消息池四种典型通信结构
- 随着LLM-MA系统中运用在更复杂的任务中,代理数量随之增加。设计先进的Agents通信模式变得越来越重要。这些方法旨在优化代理工作流程、针对不同Agents量身定制的任务分配以及代理之间的通信约束,保证代理之间的运作效率,最大限度地减少冲突和冗余通信

Agent能力获取方面,Agent从反馈中学习,通过记忆、自我进化和动态生成等方式调整自身策略以增强能力

- 使用记忆提升Agent的能力。Agent将来自先前互动和反馈的信息存储在它们的记忆中。在执行行动时,它们可以检索相关的、有价值的记忆,以提升其能力。但是如果行动路径过长,可能会存在大量冗余的记忆,如何进一步提炼其中的知识将会是一个挑战
- 自我进化,通过通信和反馈修改Agents目标和规划策略,或则通过通信日志和反馈信息生成数据集用微调LLM以提升Agent的能力