

Mimicking to Dominate: Imitation Learning Strategies for Success in Multiagent Games

Authors: The Viet Bui^[1], Tien Mai^[1], Thanh Hong Nguyen^[2]

^[1]Singapore Management University

^[2]University of Oregon

Introduction

- **Challenges in Multi-Agent Games**

- Dynamic environments
- Strategies and actions of opponents
 - Fully observable MDP: Chess, Go, Habani, etc.
 - Partially observable MDP (POMDP): SMAC, GRF, MPE, etc.



Figure 1. SMAC



Figure 2. Chess

Background

- **Centralized Training and Decentralized Execution (CTDE)**

- Recent works in MARL have focused on CTDE
- Leverage global information to train a centralized critic or joint Q-function
- Face challenges in efficiently and stably learning agent behaviors

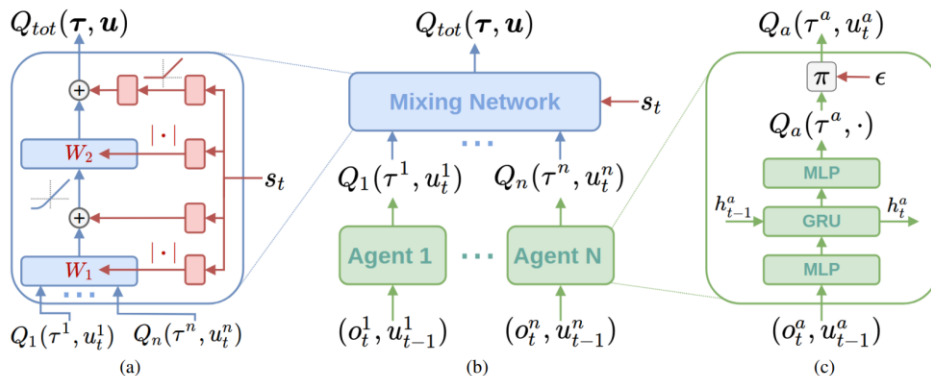


Figure 3. QMIX architecture

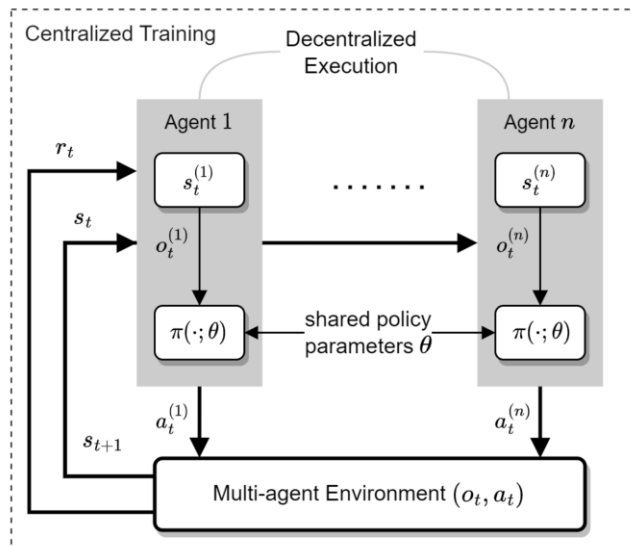


Figure 4. CTDE framework

Background

- **IQ-Learn: A SOTA Imitation Learning Algorithm**

- IQ-Learn reduces the IRL problem to a single optimization over the Q-function:

$$\max_{Q \in \Omega} \min_{\pi \in \Pi} \{J(\pi, Q) = E_{\rho_E} [\phi(Q(s, a) - \gamma E_{s' \sim P(\cdot|s, a)} [V^\pi(s')])] - (1 - \gamma) E_{s_0 \sim \rho_0} [V^\pi(s_0)]\}$$

where:

$$V^\pi(s) = E_{a \sim \pi(\cdot|s)} [Q(s, a) - \log \pi(a|s)]$$

- ϕ is a concave function that defines the statistical divergence between expert and learned policies
- For a fixed Q-function, the policy π is updated to maximize:

$$E_{s \sim D, a \sim \pi(\cdot|s)} [Q(s, a) - \log \pi(a|s)]$$

Multi-Agent POMDP Setting

- The multi-agent POMDP can be represented as a tuple $\{S, \mathcal{N}_\alpha, \mathcal{N}_e, A^\alpha, A^e, P, R\}$, where:
 - S is the global state shared by all agents
 - $\mathcal{N}_\alpha, \mathcal{N}_e$ are the set of ally, enemy agents respectively
 - A^α, A^e are the joint action space of ally, enemy agents respectively
 - P is the transition dynamics
 - R is the reward function
- The objective is to find a policy for the ally agents that maximizes their expected joint reward over time:

$$\max_{\Pi_\alpha} \mathbb{E}_{(A^\alpha, S) \sim \Pi_\alpha} [R(S, A^\alpha)]$$

where:

- $\Pi_\alpha(A^\alpha|S) = \prod_{i \in \mathcal{N}_\alpha} \pi_i^\alpha(a_i^\alpha | o_i^\alpha)$ is the joint policy of ally agents.
- $\pi_i^\alpha(a_i^\alpha | o_i^\alpha)$ is the policy of agent i based on its local observation o_i^α

Challenge

- **Partial Observability**

- Each agent relying only on local observations

- **Decentralized Decision Making**

- Make decisions independently
- Without direct communication

- **Dynamic Environment**

- States change over time
- Requires agents to adapt their strategies

- **Hidden Actions of Opponents**

- Opponent actions are not directly observable

- **We need predict opponent behavior**

- Reduce uncertainty
- Coordinate more effectively
- Improving decision-making
- Improving team coordination
- Improving learning efficiency



Our Solution: Opponent Policy Imitation

- Actions are unobservable → Opponent Next-State Prediction as an IL, where:
 - “Expert” state is a pair $W = (S, A_-^{\alpha})$, A_-^{α} is the joint action of allies in the previous step that led to state S
 - “Expert” action is the next enemy state $S^{e,\text{next}}$
- Adapt to IQ-Learn

$$\max_{\hat{Q}^e} \min_{\hat{\Pi}^e} \left\{ J(\hat{\Pi}^e, \hat{Q}^e) = \sum_{i \in \mathcal{N}_{\alpha}} \mathbb{E}_{(S_i^{e,\text{next}}, w_i^{\alpha}) \sim \rho^{e,\alpha}} \left[\phi \left(\hat{Q}^e(S_i^{e,\text{next}}, w_i^{\alpha}) - \gamma \mathbb{E}_{w_i^{\alpha,\text{next}}} [V_{\Pi}^e(w_i^{\alpha,\text{next}})] \right) \right] - (1 - \gamma) \mathbb{E}_{w_{i0}^{\alpha} \sim P^0, \Pi^{\alpha}} [V_{\Pi}^e(w_{i0}^{\alpha})] \right\}$$

where

$$V_{\Pi}^e(w_i^{\alpha}) = \mathbb{E}_{S_i^{e,\text{next}} \sim \hat{\Pi}^e} [\hat{Q}^e(S_i^{e,\text{next}}, w_i^{\alpha}) - \log \hat{\Pi}^e(S_i^{e,\text{next}} | w_i^{\alpha})]$$

- For a fixed $\hat{Q}^e$, the policy $\hat{\Pi}^e$ is updated by soft actor-critic (SAC)

Our Solution: IMAX-PPO

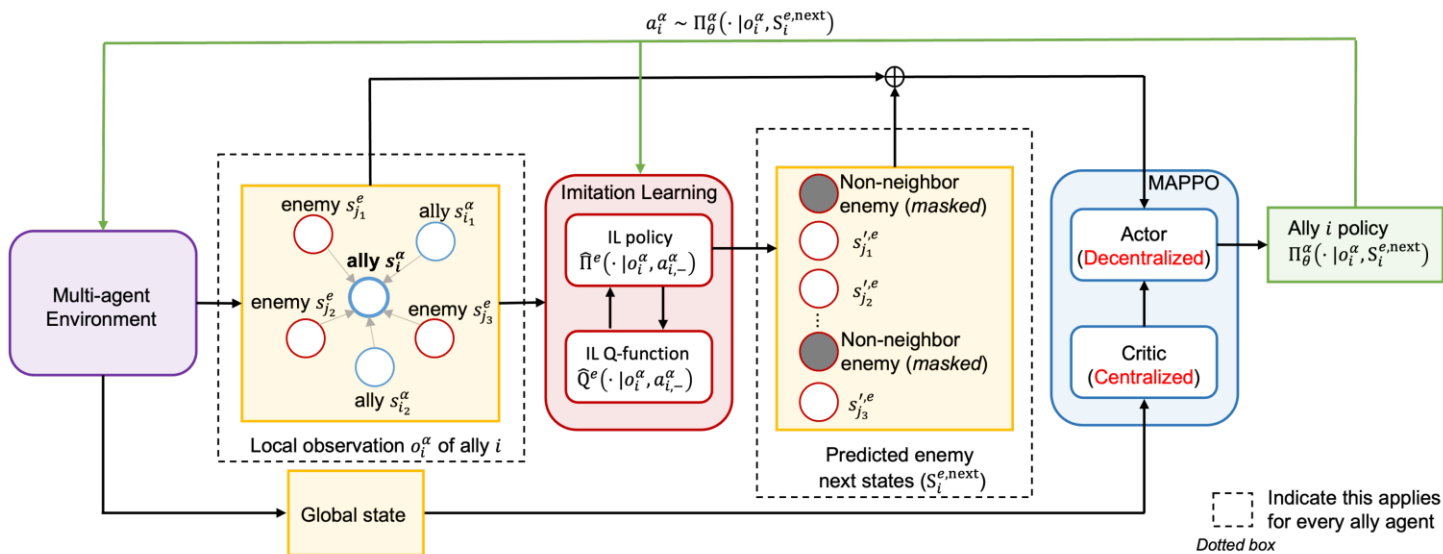


Figure 5. Our IMAX-PPO Framework

Experiments

Tasks	Scenarios	MAPPO	IPPO	QMIX	QPLEX	Sup MAPPO	IMAX-PPO GAIL	InQ
SMAC Protoss	5_vs_5	58.0	54.6	70.2	53.3	71.8	68.1	78.7
	10_vs_10	58.3	58.0	69.0	53.7	67.3	59.6	79.8
	10_vs_11	18.2	20.3	42.5	22.8	36.7	21.3	48.7
	20_vs_20	38.1	44.5	69.7	27.2	71.1	76.3	80.6
	20_vs_23	5.1	4.1	16.5	4.8	21.9	11.8	24.2
SMAC Terran	5_vs_5	52.0	56.2	58.4	70.0	55.8	53.3	69.9
	10_vs_10	58.1	57.3	65.8	66.1	54.1	58.4	72.2
	10_vs_11	28.6	31.0	39.4	41.4	26.9	28.4	53.9
	20_vs_20	52.8	49.6	57.6	23.9	38.6	35.9	65.4
	20_vs_23	11.2	10.0	10.0	7.0	11.2	4.7	17.7
SMAC Zerg	5_vs_5	41.0	37.2	37.2	47.8	52.5	48.6	55.0
	10_vs_10	39.1	49.4	40.8	41.6	57.4	50.6	57.6
	10_vs_11	31.2	26.0	28.0	31.1	38.1	34.8	41.5
	20_vs_20	31.9	31.2	30.4	15.8	44.3	26.7	43.3
	20_vs_23	15.8	8.3	10.1	6.7	13.6	8.2	21.3
Gold Miner	easy	48.9	49.3	57.2	59.8	47.1	54.5	61.8
	medium	40.6	39.5	47.3	50.4	39.4	39.3	55.0
	hard	31.2	31.2	41.7	43.5	31.3	29.7	49.8
GRF	3_vs_1	88.0	82.7	8.1	90.2	96.1	96.4	98.1
	easy	87.8	84.1	16.0	94.9	89.7	64.1	95.0
	hard	77.4	70.9	3.2	95.1	10.7	15.2	97.3

Figure 6. Win-rate percentages of various MARL algorithms across different tasks and scenarios. Higher is better.

Conclusion

- We introduce a novel IL model designed to predict the next moves of opponents in multi-agent games.
- We develop a new MARL algorithm called IMAX-PPO, which integrates our IL model with policy training.
- A comprehensive theoretical analysis is provided, which includes bounds on the impact of the changing policies of allied agents on the IL outcomes.
- Extensive experiments conducted in various challenging game environments, such as SMACv2, Google Research Football, and Gold Miner, demonstrate that the proposed IMAX-PPO algorithm consistently outperforms state-of-the-art MARL algorithms.