O-MAPL:
Offline Multi-agent Preference Learning

- **The Viet Bui**, *Singapore Management University*
- **Tien Anh Mai**, *Singapore Management University*
- **Thanh Hong Nguyen**, *University of Oregon*

ICML
International Conference
On Machine Learning

Introduction

- **Learning from Preferences in MARL:** Most methods first train a separate reward model from preferences, then use that model for policy learning. Misalignment between these phases can degrade performance and cause instability.
- **Our Approach:** Develop a stable, end-to-end framework that learns multi-agent policies directly from **offline preference data**, without needing an explicit reward model.

Background: MaxEnt RL for MARL

We model the cooperative MARL problem as a POMDP:

$$\max_{\pi_{tot}} \mathbb{E}_{\pi_{tot}}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s_t,a_t)-\beta\log\frac{\pi_{tot}(a_t|s_t)}{\mu_{tot}(a_t|s_t)}\right)\right]$$

The optimal policy π^*_{tot} is related to the optimal soft Q-function Q^*_{tot} by:

$$\pi^*_{tot}(a|s)=\mu_{tot}(a|s)\exp\left(\frac{Q^*_{tot}(s,a)-V^*_{tot}(s)}{\beta}\right)$$

Where $V^*_{tot}(s)$ is the optimal soft value function, computed as a log-sum-exp of Q^*_{tot}.

Core Idea: From Rewards to Q-Functions

There's a one-to-one mapping between the reward function $r(s,a)$ and the Q-function $Q_{tot}(s,a)$ via the inverse soft Bellman operator $\mathcal{T}^*$:

$$r(s,a)=(\mathcal{T}^*Q_{tot})(s,a)=Q_{tot}(s,a)-\gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}V_{tot}(s')$$

This unique mapping allows us to reformulate the problem of learning a reward function as learning a Q-function, which is the core idea of our approach.

Preference-based Inverse Q-Learning

Using the **Bradley-Terry model**, the probability of preferring trajectory σ_1 over σ_2 is modeled based on the sum of rewards:

$$P(\sigma_1\succ\sigma_2|Q_{tot})=\frac{\exp\left(\sum_{(s,a)\in\sigma_1}(\mathcal{T}^*Q_{tot})(s,a)\right)}{\exp\left(\sum_{(s,a)\in\sigma_1}(\mathcal{T}^*Q_{tot})(s,a)\right)+\exp\left(\sum_{(s,a)\in\sigma_2}(\mathcal{T}^*Q_{tot})(s,a)\right)}$$

We then maximize the log-likelihood of the preference data over the Q-function. This single-phase approach enhances training stability by integrating reward and policy learning.

Methodology: O-MAPL

O-MAPL is an end-to-end, offline algorithm that learns policies directly from a preference dataset $\mathcal{P}$ of trajectory pairs (σ_1,σ_2), where σ_1 is preferred over σ_2.

Value Factorization

Using local functions (v_i,q_i) and a mixing network $\mathcal{M}_w$:

$$Q^{tot}(s,a)=\mathcal{M}_w[q(s,a)];\ V^{tot}(s)=\mathcal{M}_w[v(s)]$$

Extreme-V loss:

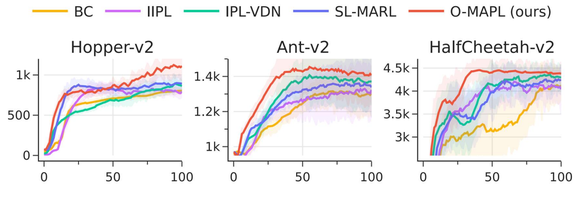
$$\mathcal{J}(v)=\mathbb{E}_{(s,a)\sim\mu_{tot}}\left[\exp\left(\frac{Q^{tot}(s,a)-V^{tot}(s)}{\beta}\right)\right]-\mathbb{E}_{(s,a)\sim\mu_{tot}}\left[\frac{Q^{tot}(s,a)-V^{tot}(s)}{\beta}\right]-1$$

Local Policy Extraction

We extract local policies using *Weighted Behavior Cloning (WBC)*, which offers superior stability and consistency over simpler value-based extraction methods.

$$\max_{\pi_i}\left\{\mathbb{E}_{(s,a)\sim\mu_{tot}}\left[\exp\left(\frac{Q^{tot}(s,a)-V^{tot}(s)}{\beta}\right)\log\pi_i(a_i|s_i)\right]\right\}$$

Experiments & Results



	Tasks	Rule-based					LLM-based				
		BC	IIPL	IPL-VDN	SL-MARL	O-MAPL (ours)	BC	IIPL	IPL-VDN	SL-MARL	O-MAPL (ours)
	2c_vs_64zg	59.6±25.0	60.4±24.7	71.1±22.0	63.5±24.0	**74.4±24.7**	65.6±24.6	60.2±25.9	77.0±21.3	65.2±21.2	**79.5±19.6**
	5m_vs_6m	16.8±18.0	14.3±17.0	16.8±18.0	16.0±18.9	**19.3±19.6**	18.2±18.4	15.0±17.5	18.0±19.2	17.4±19.4	**20.7±20.5**
	6h_vs_8z	0.6±3.8	0.2±2.2	2.5±7.6	1.6±6.8	**4.5±11.0**	0.8±4.3	0.4±3.1	3.5±9.2	3.7±8.9	**6.1±11.2**
	corridor	89.3±15.5	89.8±15.4	93.9±11.6	49.0±22.8	**93.2±13.5**	89.6±15.5	90.6±13.6	**94.5±12.5**	57.6±22.2	**94.5±11.2**
Protoss	5_vs_5	38.1±24.2	31.4±25.2	**54.5±25.9**	49.0±28.2	54.3±24.2	48.4±25.9	41.0±24.2	58.8±24.5	54.3±24.0	**61.5±24.8**
Protoss	10_vs_10	38.7±24.2	28.5±21.8	47.9±27.2	40.6±23.2	**53.7±23.6**	46.3±24.0	41.0±24.4	57.0±23.4	52.5±22.1	**61.1±24.8**
Protoss	10_vs_11	12.7±17.4	12.5±16.5	22.3±21.0	18.6±18.8	**30.7±19.8**	22.7±22.2	15.6±15.9	27.3±24.7	20.9±20.9	**34.4±24.8**
Protoss	20_vs_20	39.8±24.9	35.4±21.5	57.0±24.8	38.7±23.1	**59.8±23.2**	48.4±25.3	43.6±23.6	61.5±22.1	51.8±25.0	**64.5±23.5**
Protoss	20_vs_23	15.2±18.5	9.0±14.2	22.7±21.7	11.1±14.6	**23.4±19.2**	18.0±17.4	9.4±14.7	23.4±21.4	12.1±15.9	**26.4±20.8**

A Sample Prompt to LLMs

You are a helpful and honest judge of good game playing and progress in the StarCraft Multi-Agent Challenge game. Always answer as helpfully as possible, while being truthful.
If you don't know the answer to a question, please don't share false information. I'm looking to have you evaluate a scenario in the StarCraft Multi-Agent Challenge. Your role will be to assess how much the actions taken by multiple agents in a given situation have contributed to achieving victory.

The basic information for the evaluation is as follows.
- **Scenario : 5m_vs_6m**
- **Allied Team Agent Configuration : five Marines (Marines are ranged units in StarCraft 2).**
- **Enemy Team Agent Configuration : six Marines (Marines are ranged units in StarCraft 2).**
- **Situation Description : The situation involves the allied team and the enemy team engaging in combat, where victory is achieved by defeating all the enemies.**
- **Objective : Defeat all enemy agents while ensuring as many allied agents as possible survive.**

* Important Notice : You should prefer the trajectory where our allies' health is preserved while significantly reducing the enemy's health. In similar situations, you should prefer shorter trajectory lengths.

I will provide you with two trajectories, and you should select the better trajectory based on the outcomes of these trajectories. Regarding the trajectory, it will inform you about the final states, and you should select the better case based on these two trajectories.

[Trajectory 1]
1. Final State Information
1) Allied Agents Health : 0.000, 0.000, 0.067, 0.067, 0.000
2) Enemy Agents Health : 0.000, 0.000, 0.000, 0.000, 0.000, 0.040
3) Number of Allied Deaths : 3
4) Number of Enemy Deaths : 5
5) Total Remaining Health of Allies : 0.133
6) Total Remaining Health of Enemies : 0.040
2. Total Number of Steps : 28

[Trajectory 2]
1. Final State Information
1) Allied Agents Health : 0.000, 0.000, 0.000, 0.000, 0.000
2) Enemy Agents Health : 0.120, 0.000, 0.000, 0.000, 0.000, 0.200
3) Number of Allied Deaths : 5
4) Number of Enemy Deaths : 4
5) Total Remaining Health of Allies : 0.000
6) Total Remaining Health of Enemies : 0.320
2. Total Number of Steps : 23

Your task is to inform which one is better between **[Trajectory1]** and **[Trajectory2]** based on the information mentioned above. For example, if **[Trajectory 1]** seems better, output #1, and if **[Trajectory 2]** seems better, output #2. If it's difficult to judge or they seem similar, please output #0.

* Important : Generally, it is considered better when fewer allied agents are killed or injured while inflicting more damage on the enemy. Omit detailed explanations and just provide the answer.