Opponent Modelling for Multi-Objective Multi-Agent Reinforcement Learning in Sequential Settings

Christopher Armel Obame Obiang[1] , **Roxana Rădulescu**[1]
[1]Utrecht University
obiang78@gmail.com, r.t.radulescu@uu.nl

Abstract

Many real-world decision problems involve multiple agents balancing several competing objectives – for example, autonomous vehicles sharing a road must trade off speed, safety and comfort while reacting and adapting to the other drivers. When such agents learn together, each one's perception of the environment is constantly impacted by the others, and standard reinforcement-learning algorithms struggle to perform. In the single-objective case, allowing agents to explicitly model their peers (opponent modelling), or to anticipate how those peers will learn (opponent learning awareness), has emerged as an effective approach. Whether the same ideas help once agents must also reason about multiple objectives has so far only been studied in stateless settings. We take initial steps into the sequential case: we contribute three new methods that bring opponent modelling and opponent learning awareness to multi-objective multi-agent sequential decision-making, and analyse them in mixed-motive and competitive scenarios. We find that while simple opponent modelling has little to no impact on performance, applying opponent learning awareness and meta-learning methods offers a noticeable advantage in competitive environments. Notably, applying meta-learning allowed agents to successively influence the learning of peer agents to their advantage.

1 Introduction

Many real-world problems are distributed, involve multiple stakeholders, and require balancing several competing objectives [Hayes *et al.*, 2022]. For example, smart-grid control must trade off supply and demand across many producers and consumers; journey-planning assistants must balance cost, time, and comfort while accounting for other users' decisions. Handling such problems with a single, monolithic controller is rarely effective [McArthur *et al.*, 2007], and neither single-agent nor single-objective formulations capture their structure.

Reinforcement learning (RL) [Sutton and Barto, 2020] has become the dominant framework for sequential decision-making under uncertainty, and has been extended in two directions that are jointly relevant here. Multi-agent RL (MARL) [Albrecht *et al.*, 2024] studies settings in which several learners share an environment, and must cope with the non-stationarity that arises from their peers co-adapting over time. Multi-objective RL (MORL) [Gábor *et al.*, 1998; Hayes *et al.*, 2022] studies settings in which an agent optimises a vector-valued reward, and provides the support needed to reason explicitly about trade-offs. Multi-Objective Multi-Agent RL (MOMARL) [Rădulescu *et al.*, 2020] sits at the intersection and has attracted growing attention as a principled framing of real-world decision problems [Rădulescu, 2024].

A central challenge in MARL is non-stationarity: as each agent learns, its policy changes, and the dynamics experienced by any individual agent shift in ways that ordinary RL updates do not anticipate. A standard remedy is to equip agents with a model of their peers – opponent modelling [Albrecht and Stone, 2018; He *et al.*, 2016] – or, going further, with awareness of peers' learning process [Foerster *et al.*, 2018a; Zhao *et al.*, 2022]. A complementary line of work applies meta-learning so that an agent can also account for how its own learning trajectory influences peers' trajectories [Kim *et al.*, 2021].

These ideas have been developed almost exclusively in the single-objective setting. The first empirical studies of opponent modelling (OM) and opponent learning awareness (OLA) in the multi-objective case [Zhang *et al.*, 2020; Rădulescu *et al.*, 2022] are restricted to multi-objective normal-form games – stateless interactions. Whether the benefits of these methods transfer to *sequential* multi-objective multi-agent settings, where state, credit assignment and episode length all interact with the multi-objective trade-off, is an open question.

Contributions. We address this gap with the following contributions[1]:

1. We introduce three novel OM/OLA MOMARL algorithms for sequential settings: (i) MO-PPO-OM, which augments MO-PPO with a DRON (Deep Reinforcement Opponent Network)-style module [He *et al.*, 2016]; (ii) MO-POLA, a multi-objective extension of POLA

[1]This work is based on contributions developed in the first author's master thesis [Obame Obiang, 2026].

(Proximal learning with Opponent Learning Awareness) [Zhao *et al.*, 2022] using the loaded-DiCE estimator [Farquhar *et al.*, 2019]; (iii) and MO-META-MAPG, a multi-objective extension of the Meta-Multiagent Policy Gradient [Kim *et al.*, 2021] framework with both own- and peer-learning awareness.

2. We conduct a pairwise-tournament empirical study in two contrasting sequential environments – a mixed-motive gridworld and a zero-sum board game – under both linear and non-linear utility functions.
3. We show that plain opponent modelling does not improve performance over a vanilla baseline, but that OLA and meta-learning confer a clear advantage in mixed-motive settings. In the zero-sum setting, meta-learning is further able to exploit an OLA-only opponent, providing the first such evidence in sequential MOMARL.

2 Background

2.1 Multi-Objective Multi-Agent RL

We formalise our setting as a multi-objective stochastic game (MOSG): $(N, S, A, \mathbf{R}, p)$, where:

- N is a set of $n \geq 2$ agents
- S the state space
- $A = A_1 \times \cdots \times A_n$ the joint action space
- $\mathbf{R} = (\mathbf{R}_1, \ldots, \mathbf{R}_n)$ with $\mathbf{R}_i : S \times A \times S \to \mathbb{R}^d$ a vector-valued reward function for agent i
- $p : S \times A \times S \to [0, 1]$ the transition function

We take a utility-based perspective and assume that each agent i is equipped with a utility function $u_i : \mathbb{R}^d \to \mathbb{R}$ that scalarises its vector return [Roijers *et al.*, 2013]. In this setting, two optimality criteria arise [Hayes *et al.*, 2022]. Under the *scalarised expected return* (SER) criterion,

$$V_u^\pi = u(\mathbb{E}[\textstyle\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_t \mid \pi]) \,, \tag{1}$$

the utility is applied to the expected vector return. Under the *expected scalarised return* (ESR) criterion, the utility is applied to each return realisation and then averaged. The two agree for linear u but diverge for non-linear monotonic u. For this work we focus on SER, which is the more straightforward of the two for extending standard RL approached to the multi-objective case: vector-valued value functions and Bellman backups carry over largely as in the scalar case, with only the final scalarisation step requiring dedicated treatment, whereas ESR generally requires reasoning about the full distribution of returns [Röpke *et al.*, 2023].

When u is non-linear, ordinary policy-gradient estimators do not target $\nabla u(\mathbf{V}^\pi)$ correctly, because u cannot be moved inside the expectation. A practical solution is to condition the policy on *accrued rewards* [Reymond *et al.*, 2023], $\mathbf{R}_t^- = \sum_{k=0}^{t-1} \gamma^k \mathbf{r}_k$, which lets the SER policy gradient be written as a weighted sum of per-objective gradients with weights $\partial u / \partial V^i$ evaluated at $\mathbf{V}^\pi(s_0)$. We adopt this mechanism in all SER-optimising components of our algorithms.

2.2 Opponent Modelling and Learning Awareness

Opponent modelling. DRON – Deep Reinforcement Opponent Network [He *et al.*, 2016] introduced the idea of jointly learning a policy and a representation of the opponent's behaviour through an auxiliary *opponent network*, whose hidden activations condition the agent's value function. This is a simple form of opponent modelling: it predicts what the opponent will do next, but treats the opponent as stationary from one update to the next.

Learning with opponent learning awareness. LOLA [Foerster *et al.*, 2018a] observes that, because peers also learn, an agent can anticipate and shape their next update. A LOLA agent includes a look-ahead term over the opponent's learning step in its own gradient, which empirically recovers reciprocity-based cooperation in partially competitive games. LOLA is sensitive to policy parameterisation, a problem that is particularly pronounced when the opponent's parameters must be *inferred* via opponent modelling [Zhao *et al.*, 2022]. POLA addresses this with proximal operators. The POLA update rule is designed such that parameters with equivalent policies produce equivalent updated policies. To achieve this property, a penalty on policy divergence was added to the update rule. Thus, at each policy update, a POLA agent must solve for:

$$\theta_{t+1}^i(\theta_t^i, \theta_t^{-i}) = \arg\min_{\theta^i} \Big(L^i\big(\pi_{\theta^i}, \pi_{\theta_{t+1}^{-i}(\theta^i, \theta_t^{-i})}\big) + \beta_{\text{out}}\, D(\pi_{\theta_t^i} \| \pi_{\theta^i}) \Big), \tag{2}$$

where θ_t^i refers to the policy parameters of agent i at timestep t, $D(\pi_{\theta_t^i} || \pi_{\theta^i})$ is a shorthand for a general divergence metric defined on the policies, β_{out} is the outer penalty coefficient and $L^i(\pi_{\theta^i}, \pi_{\theta^{-i}}) = -J(\theta^i, \theta^{-i})$ defines the objective of agent i. The opponent's θ_{t+1}^{-i} is defined as:

$$\theta_{t+1}^{-i}(\theta^i, \theta_t^{-i}) = \arg\min_{\theta^{-i}} (L^{-i}(\pi_{\theta^i}, \pi_{\theta^{-i}}) + \beta_{in} D(\pi_{\theta_t^{-i}} || \pi_{\theta^{-i}}) \tag{3}$$

where β_{in} is the inner penalty coefficient. The key property is behavioural invariance: behaviourally equivalent parameterisations produce behaviourally equivalent updates. In practice, POLA is implemented with the DiCE estimator [Foerster *et al.*, 2018b] – or its lower-variance *loaded*-DiCE variant [Farquhar *et al.*, 2019] – to obtain the higher-order derivatives required for the proximal objective.

Meta-Multiagent Policy Gradient (Meta-MAPG). A complementary perspective is that joint policies across successive updates form a Markov chain: the policy at update L depends on the initial joint parameters θ_0 through the whole trajectory of updates. Kim et al. [2021] use this to define a meta-objective over a chain of length L. For any chain step $l < L$, the per-step value gradient $\nabla_{\theta_0^i} V_{\theta_{0:l+1}}^i$ decomposes into three interpretable parts – the current policy, own learning, and peer learning:

$$\nabla_{\theta_0^i} V^i_{\theta_{0:l+1}} = \mathbb{E}\big[G^i(\tau_{\theta_{l+1}})\big(g^{\text{cur}} + g^{\text{own}} + g^{\text{peer}}\big)\big]\,, \quad (4)$$

where the three score terms

$$g^{\text{cur}} = \nabla_{\theta_0^i} \log \pi(\tau_{\theta_0}),$$
$$g^{\text{own}} = \textstyle\sum_{l'=0}^{l} \nabla_{\theta_0^i} \log \pi(\tau_{\theta_{l'+1}} \mid \theta^i_{l'+1}),$$
$$g^{\text{peer}} = \textstyle\sum_{l'=0}^{l} \nabla_{\theta_0^i} \log \pi(\tau_{\theta_{l'+1}} \mid \theta^{-i}_{l'+1})$$

capture, respectively, the influence of the current policy, the agent's *own* subsequent learning, and its *peer's* subsequent learning on the meta-objective. This is strictly more expressive than LOLA/POLA: the agent reasons not only about the peer's next update but about the full induced chain of mutual updates.

3 Related Work

A large body of work has refined opponent modelling with richer architectures – DPIQN/DRPIQN learn latent policy features from raw observations [Hong *et al.*, 2018]; auxiliary-task formulations share representations across actor-critic heads and opponent predictors [Hernandez-Leal *et al.*, 2019]. On the OLA side, COLA improves stability via consistent second-order updates [Willi *et al.*, 2022].

All of the above operate in *single-objective* settings. In the multi-objective case, the only prior empirical studies of opponent modelling and OLA are Zhang et al. [2020] and Rădulescu et al. [2022], which address stateless multi-objective normal-form games using opponent policy reconstruction via conditional action frequencies and Gaussian-process opponent learning estimators respectively. Our work extends this line into *sequential* multi-objective stochastic games, where state transitions, credit assignment and episode length interact with the multi-objective trade-off.

4 Methods

We use a multi-objective variant of PPO [Schulman *et al.*, 2017] (MO-PPO) as the common backbone. MO-PPO shares PPO's clipped surrogate objective but operates on vector returns and scalarises with the utility at the appropriate point under the SER criterion, using accrued rewards [Reymond *et al.*, 2023] when the utility is non-linear. On top of this backbone we build three algorithms of increasing expressivity.

Assumptions. For tractability, we make three assumptions. Agents share a fully observable state and observe peer actions (needed for direct opponent modelling); each agent has access to the *utility function* of its peer (standard in the MO-OLA literature [Rădulescu *et al.*, 2022]), where the peer's objective is otherwise unidentifiable from actions alone); and all agents share a common network architecture and learning rate. We discuss the implications of relaxing these in Section 7.

4.1 MO-PPO-OM: Multi-Objective Opponent Modelling

MO-PPO-OM augments MO-PPO with a DRON-*concat*-style opponent network [He *et al.*, 2016]. The opponent network takes the current state as input and produces a hidden embedding h^{om} that is concatenated into the hidden representations of both the actor and the critic, with an additional fused hidden layer before each output head. The opponent network is trained in a multi-task fashion: a supervised action-prediction loss on the observed peer action, and an indirect end-to-end signal from the MO-PPO actor-critic losses. This algorithm is a simple extension and serves as a baseline check whether ordinary opponent modelling helps, by itself, in MOSGs.

4.2 MO-POLA: Opponent Learning Awareness

MORL policy spaces are large and sensitive to parameterisation, making POLA's behavioural invariance particularly attractive. Under SER with a possibly non-linear utility u_i, we define the agent-i objective as:

$$J^i(\theta^i, \theta^{-i}) = u_i\Big(\mathbb{E}_{\substack{\pi_{\theta^i},\\ \pi_{\theta^{-i}-\Delta\theta^{-i}(\theta^i)}}}\big[\mathbf{G}^i\big]\Big), \quad (5)$$

where $\mathbf{G}^i = \sum_k \gamma^k \mathbf{r}^i_k$ is the vector return. Plugging Equation (5) into (2) yields the multi-objective proximal objective used in MO-POLA, with the symmetric inner objective for θ^{-i}_{t+1} defined analogously. We estimate the proximal objective with loaded DiCE [Farquhar *et al.*, 2019] and approximate the $\arg\min$ by multiple gradient steps on the outer objective that differentiate through a sequence of inner gradient steps. Policy divergence is measured as $D(\pi_{\theta^i_t} \| \pi_{\theta^i}) = \mathbb{E}_s[\mathrm{KL}(\pi_{\theta^i_t}(\cdot|s) \| \pi_{\theta^i}(\cdot|s))]$.

Simulated rollouts are required to train the agent, since the proximal objective must be evaluated under a hypothetical opponent update. These rollouts use the agent's learned model of the peer's policy, obtained from observed actions in real environment rollouts. Algorithm 1 summarises one update.

Algorithm 1 MO-POLA update for agent i

Input: policy parameters θ^i_t, estimated peer parameters $\hat{\theta}^{-i}_t$, learning rates $\alpha_{\text{in}}, \alpha_{\text{out}}$, penalty coefficients $\beta_{\text{in}}, \beta_{\text{out}}$, outer steps M, inner steps K
Initialise $\theta^i \leftarrow \theta^i_t$
for $m = 1, \ldots, M$ **do**
  $\theta^{-i} \leftarrow \hat{\theta}^{-i}_t$
  for $k = 1, \ldots, K$ **do** ▷ inner: peer's anticipated learning
    Simulate trajectories on $s^{\text{in}}_{\leq T}$ with $(\pi_{\theta^i}, \pi_{\theta^{-i}})$
    $\theta^{-i} \leftarrow \theta^{-i} - \alpha_{\text{in}} \nabla_{\theta^{-i}}\big(J^{-i}(\pi_{\theta^i}, \pi_{\theta^{-i}}) + \beta_{\text{in}} D(\pi_{\hat{\theta}^{-i}_t} \| \pi_{\theta^{-i}} | s^{\text{in}}_{\leq T})\big)$
  end for
  Simulate trajectories on $s^{\text{out}}_{\leq T}$ with $(\pi_{\theta^i}, \pi_{\theta^{-i}})$ ▷ outer: agent's update
  $\theta^i \leftarrow \theta^i - \alpha_{\text{out}} \nabla_{\theta^i}\big(J^i(\pi_{\theta^i}, \pi_{\theta^{-i}}) + \beta_{\text{out}} D(\pi_{\theta^i_t} \| \pi_{\theta^i} | s^{\text{out}}_{\leq T})\big)$
end for
Output: θ^i

4.3 MO-Meta-MAPG: Meta-Learning with Self- and Peer-Learning Awareness

The multi-objective meta-objective is defined over a Markov chain of joint policies of length L with a vector meta-value and the SER scalarisation applied after the chain-wise expectation:

$$\max_{\theta_0^i} \mathbb{E}_{p(\theta_0^{-i})}\left[\sum_{l=0}^{L-1} u\Big(\mathbf{V}^i_{\theta_{0:l+1}}(s_0, \theta_0^i)\Big)\right], \tag{6}$$

with $\mathbf{V}^i$ the per-objective meta-value function along the chain. The gradient decomposes over objectives with weights $\partial u/\partial V^{i,n}$, and each per-objective gradient has the same three-part structure as (4). Inner-loop updates use accrued-reward policy gradients for SER under non-linear utilities. Because the peer's parameters are not observed, we replace θ^{-i} with the opponent network's estimate $\hat{\theta}^{-i}$ throughout, and estimate peer losses via DiCE. The resulting procedure, MO-META-MAPG, alternates (i) *meta-training*, which searches for good *initial* parameters θ_0^{i*} that maximise expected adaptation performance, and (ii) *execution*, which applies a single inner-loop update to θ_0^{i*} using real rollouts.

To keep training tractable, meta-training is run for a fixed number of outer iterations (rather than to convergence as in the original Meta-MAPG), and the best iterate found is retained for execution. Algorithms 2 and 3 summarise the two phases.

Algorithm 2 MO-META-MAPG meta-training: optimising the initial parameters of agent i.

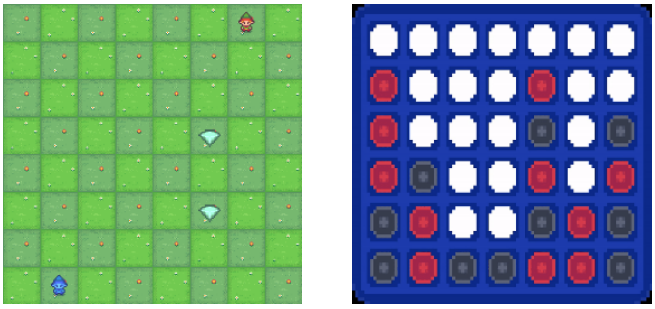
Input: policy parameters θ_t^i, estimated peer parameters $\hat{\theta}_t^{-i}$, inner learning rates α^i, α^{-i}, outer learning rate β, outer steps M, chain horizon L
$\theta_0^i \leftarrow \theta_t^i$; initialise best iterate $(\theta^\star, J^\star) \leftarrow (\theta_0^i, -\infty)$
for $m = 1, \ldots, M$ **do**
  for $l = 0, \ldots, L-1$ **do** ▷ simulated chain of length L
    Sample τ_{θ_l} using $(\pi_{\theta_l^i}, \pi_{\hat{\theta}_l^{-i}})$
    $\theta_{l+1}^i \leftarrow \theta_l^i + \alpha^i \nabla_{\theta_l^i} u\big(\mathbb{E}[\mathbf{G}^i(\tau_{\theta_l})]\big)$
    $\hat{\theta}_{l+1}^{-i} \leftarrow \hat{\theta}_l^{-i} + \alpha^{-i} \nabla_{\hat{\theta}_l^{-i}} u_{-i}\big(\mathbb{E}[\mathbf{G}^{-i}(\tau_{\theta_l})]\big)$
  end for
  $\theta_0^i \leftarrow \theta_0^i + \beta \sum_{l=0}^{L-1} \nabla_{\theta_0^i} u\big(\mathbf{V}^i_{\theta_{0:l+1}}(s_0, \theta_0^i)\big)$ ▷ outer meta-update
  if adaptation performance at θ_0^i exceeds $J^\star$ **then**
    $(\theta^\star, J^\star) \leftarrow (\theta_0^i,$ current performance$)$
  end if
end for
Output: $\theta_0^{i*} \leftarrow \theta^\star$

5 Experimental Setup

We evaluate all four algorithms – MO-PPO (baseline), MO-PPO-OM, MO-POLA, MO-Meta-MAPG – in pairwise tournaments across two environments from the MOMALand library [Felten *et al.*, 2026]. Experiments are run with $n = 2$ agents to isolate the algorithmic effects, under linear and non-linear utilities, with each configuration averaged over 5 random seeds[2]. All methods use actor and critic MLPs with a single 64-unit hidden layer, and opponent networks with the same architecture as the peer actor they estimate; full hyper-parameters are listed in Appendix A.

Algorithm 3 MO-META-MAPG execution: one update for agent i.

Input: meta-trained initial parameters θ_0^{i*}, estimated peer parameters $\hat{\theta}_t^{-i}$, inner learning rate α^i
$\theta_0^i \leftarrow \theta_0^{i*}$
Sample real-environment trajectory τ_θ using $(\pi_{\theta_0^i}, \pi_{\hat{\theta}_t^{-i}})$
$\theta_1^i \leftarrow \theta_0^i + \alpha^i \nabla_{\theta_0^i} u\big(\mathbb{E}[\mathbf{G}^i(\tau_\theta)]\big)$ ▷ single inner-loop adaptation
Output: θ_1^i



(a) MO-ItemGathering: mixed-motive gridworld.

(b) MO-Connect4: board game with per-column objectives.

Figure 1: The two MOMAland environments used in this study

MO-ItemGathering (mixed-motive). A gridworld in which two agents compete to collect coloured items, each colour representing an objective [Källström and Heintz, 2019] (see Figure 1a). Each episode lasts 64 timesteps and uses 3 colours with 3 items per colour randomly placed on an 8×8 grid. The setting is individual-reward, individual-utility, so it is competitive but non-zero-sum: agents can pursue different colour preferences without direct conflict. We use two utility profiles. Linear:

$$u_1 = 0.1x + 0.3y + 0.6z,$$
$$u_2 = 0.3x + 0.6y + 0.1z;$$

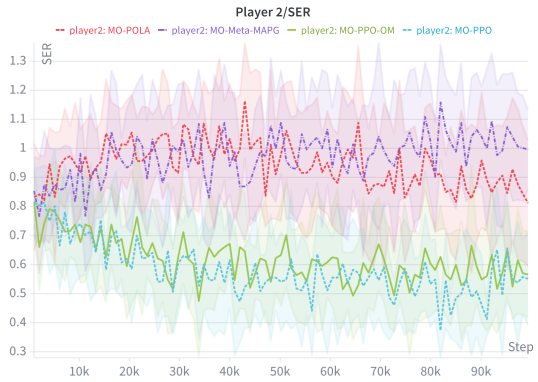
non-linear:

$$u_1 = xz + y^2,$$
$$u_2 = xy + z^2,$$

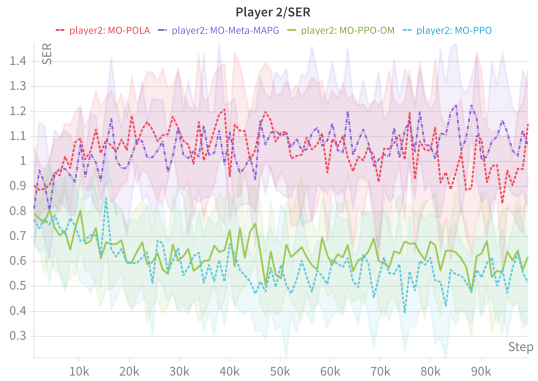
where x, y, z are counts of items collected by the agent of each colour. The linear profile defines a preference ordering; the non-linear profile rewards collecting either a single dominant colour *or* a complementary pair.

MO-Connect4 (zero-sum). A multi-objective extension of the zero-sum board game Connect4 on a 6×7 grid (Figure 1b). Alongside the win/loss outcome, agents receive

[2]Code will be released upon acceptance.

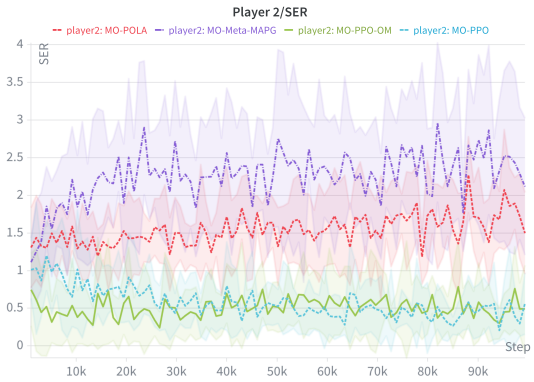


(a) Player 1 (MO-PPO) uses u_1, Player 2 uses u_2.

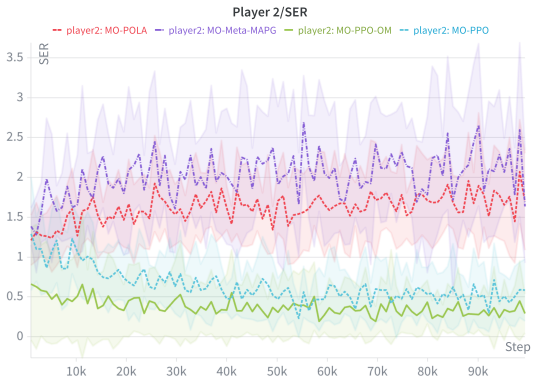


(b) Both players use u_1 (shared utility).

Figure 2: SER in MO-ItemGathering under linear utility, against MO-PPO. MO-Meta-MAPG and MO-POLA substantially outperform MO-PPO-OM and MO-PPO.



(a) Player 1 (MO-PPO) uses u_1, Player 2 uses u_2.



(b) Both players use u_1 (shared utility).

Figure 3: SER in MO-ItemGathering under non-linear utility. The gap between MO-Meta-MAPG and MO-POLA widens, consistent with the hypothesis that meta-learning is especially beneficial when the utility rewards coordinated patterns.

per-column rewards incentivising majority control of specific columns, and a speed bonus for faster wins. The utility functions are designed to induce contrasting playstyles. Linear:

$$u_1 = 0.6x + 0.1(a + b + c + d),$$
$$u_2 = 0.6y + 0.1(c + d + e + f);$$

non-linear:

$$u_1 = (x + 1)(d + e + f),$$
$$u_2 = y(c + 1) + y,$$

where x indicates win/loss, y how quickly the outcome is reached, and $a-f$ are column objectives. u_1 is a patient, left-board player; u_2 is an aggressive, right-board speed player.

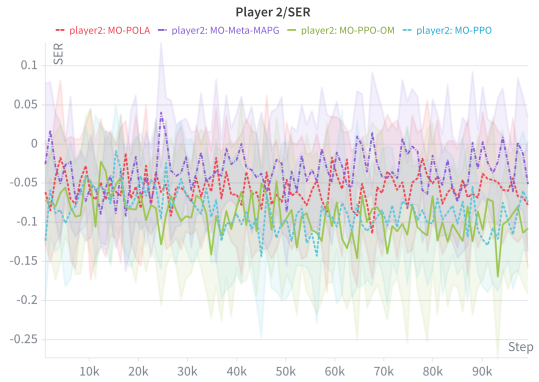
Protocol. For each environment, utility profile, and pair of algorithms, we run a tournament in which each algorithm plays once as agent 1 and once as agent 2, and we additionally run the shared-utility condition in which both agents use the same u. We report Scalarised Expected Return (SER) over training, with shading denoting the ± standard deviation across seeds.

6 Results

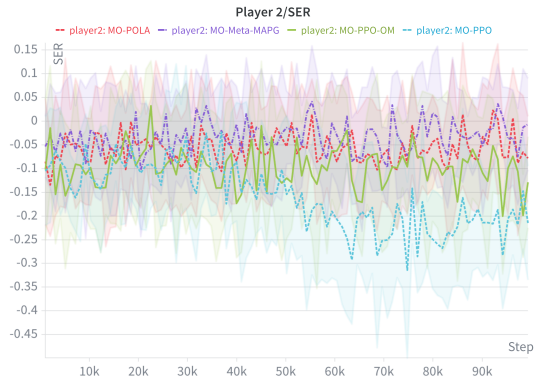
6.1 MO-ItemGathering

Figure 2 shows the SER of each algorithm when evaluated against the MO-PPO baseline under linear utility. MO-Meta-MAPG and MO-POLA converge to an SER of approximately 1.0 and 0.9 respectively, while MO-PPO-OM and MO-PPO plateau around 0.6. The gap is consistent across the shared-utility condition (Figure 2b), confirming that it does not depend on the peer using a complementary colour ranking. Under the non-linear utility (Figure 3), the same qualitative ordering holds, and MO-Meta-MAPG now additionally pulls slightly ahead of MO-POLA – reaching an average SER above 2, compared to roughly 1.5 for the latter. Intuitively, the non-linear utility rewards coordinated patterns of collection, and meta-learning's account of both own and peer learning lets the agent steer the joint trajectory more actively.

Across all other pairings (omitted here for space), the same trend appears: plain opponent modelling provides no measurable benefit over MO-PPO, while OLA and meta-learning both confer a clear advantage.

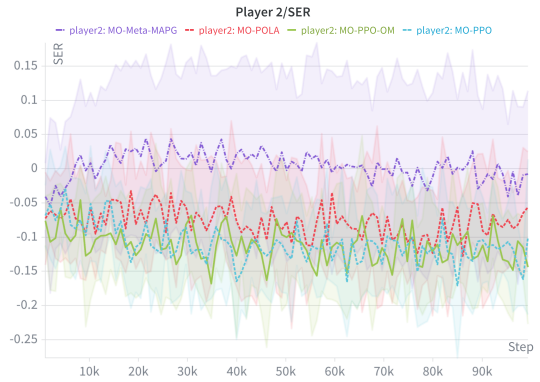


(a) Player 1 (MO-PPO) uses u_1, Player 2 uses u_2.

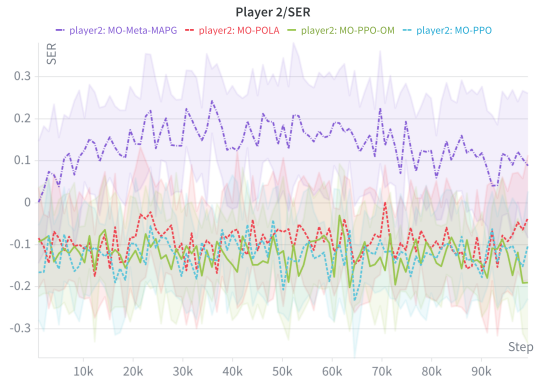


(b) Both players use u_1 (shared utility).

Figure 4: SER in MO-Connect4 under linear utility, against a MO-PPO opponent. In this strictly zero-sum setting, no algorithm dominates.



(a) Player 1 (MO-POLA) uses u_1, Player 2 uses u_2.



(b) Both players use u_1 (shared utility).

Figure 5: SER in MO-Connect4 under linear utility, against a MO-POLA opponent. MO-Meta-MAPG is the only algorithm that consistently attains positive SER in the shared-utility setting (b), indicating active exploitation of the opponent.

6.2 MO-Connect4

The picture in MO-Connect4 is more subtle. When evaluated against MO-PPO or MO-PPO-OM, no algorithm presents a clear gap (Figure 4). Under the strict zero-sum structure of Connect4, both agents have an incentive to explore broadly, making the peer's learning process hard to predict – the regime in which OLA methods are known to struggle [Foerster *et al.*, 2018a].

A more interesting result appears when MO-POLA is the opponent. As shown in Figure 5a, MO-Meta-MAPG consistently achieves a higher SER than the other algorithms (including MO-POLA itself when it plays the agent role). In the shared-utility condition (Figure 5b), only MO-Meta-MAPG sustains a positive SER. The same pattern persists when MO-Meta-MAPG is the opponent under shared non-linear utility (Figure 6b). Against a MO-Meta-MAPG opponent, under non-linear utilities, the MO-PPO and MO-PPO-OM agents are dominated. The MO-Meta-MAPG has an initial advantage, but MO-POLA does eventually catch up (Figure 6a).

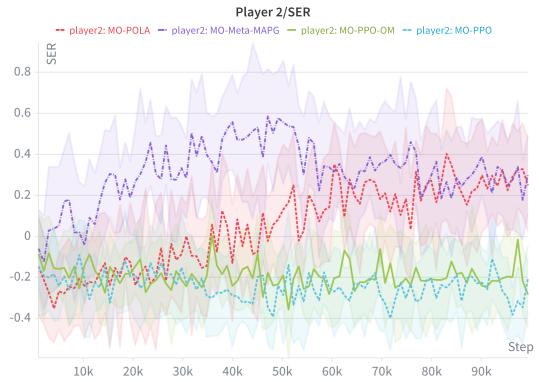
We interpret these asymmetries as evidence that meta-learning is able to shape the learning trajectory of agents to its own advantage, even in a strictly zero-sum game. The meta-agent exploits the very assumption that makes OLA effective in mixed-motive settings – that the opponent is a predictable learner – to steer its opponent into policies that favour the meta-agent's own utility.

7 Discussion

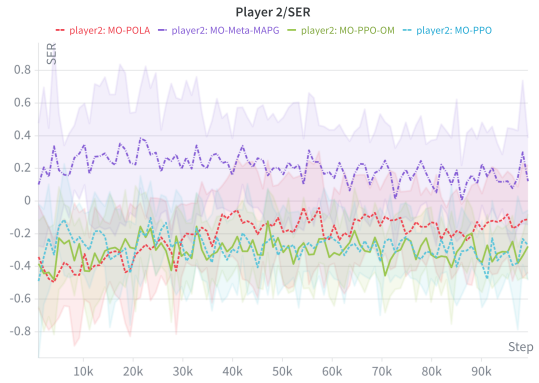
We detail below three main findings of our initial empirical analysis.

Opponent modelling alone is insufficient. MO-PPO-OM performs indistinguishably from MO-PPO across every condition we tested. The DRON-style opponent network does succeed in predicting peer actions, but those predictions become outdated as the peer itself learns – exactly the failure mode that motivates OLA [Foerster *et al.*, 2018a]. This replicates, in the sequential multi-objective regime, the expectation set by the single-objective literature.

Opponent learning awareness helps in mixed-motive MOSGs. Both MO-POLA and MO-Meta-MAPG substantially outperform the baselines in MO-ItemGathering. Under the SER criterion, this is the first empirical evidence that accounting for the peer's learning process – rather than merely its current policy – improves policy quality in sequential



(a) Player 1 (MO-Meta-MAPG) uses u_1, Player 2 uses u_2



(b) Both players use u_1 (shared utility).

Figure 6: SER in MO-Connect4 under non-linear utility, against MO-Meta-MAPG. The exploitation advantage persists, confirming the asymmetry on early- to mid-training but with MO-POLA closing the gap late.

multi-objective settings. The effect is especially pronounced for non-linear utilities, where the coordination structure rewarded by u is itself non-local across objectives.

Meta-learning can exploit an OLA opponent in zero-sum. The most surprising result is the asymmetry between MO-Meta-MAPG and MO-POLA in MO-Connect4. In the strictly zero-sum regime neither OLA nor meta-learning dominates against MO-PPO, but when they are paired against each other meta-learning initially pulls ahead. This is consistent with the intuition that Meta-MAPG's look-ahead over an entire chain of mutual updates lets it model how an OLA agent will react to its own policy shifts, and use that model to steer the peer into suboptimal responses. To our knowledge this is the first sequential-MOMARL evidence of one OLA-family agent exploiting another.

Limitations. Our experiments rely on two assumptions that are standard in the initial OLA/meta-learning literature but nonetheless strong: agents have access to the peer's utility function, and share a common network architecture and learning rate. These assumptions simplify the estimation of opponent updates. We expect the qualitative ordering of methods to persist under weaker assumptions – as long as the opponent network remains expressive enough to track the peer – but quantifying this is important future work. The setting is also limited to two fully-observable agents; generalisation to partial observability and larger agent populations is non-trivial. Finally, we observed in earlier exploratory experiments (omitted here) that team-reward cooperative settings pose an orthogonal challenge: credit assignment interacts with the multi-objective trade-off in ways that opponent-awareness alone does not resolve.

8 Conclusion and Future Work

We presented the first empirical study of opponent modelling and opponent learning awareness in *sequential* multi-objective multi-agent settings, introducing three novel algorithms – MO-PPO-OM, MO-POLA and MO-Meta-MAPG – that extend DRON, POLA and Meta-MAPG to the SER criterion. Across a mixed-motive gridworld and a zero-sum board game, under linear and non-linear utilities, we showed that plain opponent modelling offers no improvement over a MO-PPO baseline, whereas opponent learning awareness provides a clear benefit in mixed-motive interaction, and meta-learning additionally can outperform OLA opponents in zero-sum interaction. These results extend prior findings in multi-objective normal-form games [Rădulescu *et al.*, 2022] to stateful settings, and strengthen the case for designing MOMARL algorithms that reason explicitly about how peers learn.

Several directions are immediate. First, all our algorithms target SER; a corresponding investigation under ESR would broaden their applicability to single-run decision problems. Second, our agents observe the peer's utility function; relaxing this with online utility estimation – possibly using the GP-based approach of [Rădulescu *et al.*, 2022] – is a natural next step. Third, partial observability and larger agent populations remain open. Finally, the cooperative team-reward setting presents an open challenge where the interaction between credit assignment and multi-objective trade-offs deserves dedicated treatment.

References

[Albrecht and Stone, 2018] Stefano V. Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.

[Albrecht *et al.*, 2024] Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press, 2024.

[Farquhar *et al.*, 2019] Gregory Farquhar, Shimon Whiteson, and Jakob Foerster. Loaded DiCE: Trading off bias and variance in any-order score function gradient estimators for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[Felten *et al.*, 2026] Florian Felten, Umut Ucak, Hicham Azmani, Gao Peng, Willem Röpke, Hendrik Baier, Patrick Mannion, Diederik M Roijers, J K Terry, El Ghazali Talbi, Grégoire Danoy, Ann Nowe, and Roxana Rădulescu. MOMAland: A set of benchmarks for multi-objective multi-agent reinforcement learning. *Journal of Data-centric Machine Learning Research*, 2026.

[Foerster *et al.*, 2018a] Jakob Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 122–130. IFAAMAS, 2018.

[Foerster *et al.*, 2018b] Jakob Foerster, Gregory Farquhar, Maruan Al-Shedivat, Tim Rocktäschel, Eric Xing, and Shimon Whiteson. DiCE: The infinitely differentiable monte carlo estimator. In *International Conference on Machine Learning*, pages 1529–1538. PMLR, 2018.

[Gábor *et al.*, 1998] Zoltán Gábor, Zsolt Kalmár, and Csaba Szepesvári. Multi-criteria reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 98, pages 197–205, 1998.

[Hayes *et al.*, 2022] Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

[He *et al.*, 2016] He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. Opponent modeling in deep reinforcement learning. In *International Conference on Machine Learning*, pages 1804–1813, 2016.

[Hernandez-Leal *et al.*, 2019] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. Agent modeling as auxiliary task for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 15, pages 31–37, 2019.

[Hong *et al.*, 2018] Zhang-Wei Hong, Shih-Yang Su, Tzu-Yun Shann, Yi-Hsiang Chang, and Chun-Yi Lee. A deep policy inference Q-network for multi-agent systems. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1388–1396. IFAAMAS, 2018.

[Källström and Heintz, 2019] Johan Källström and Fredrik Heintz. Tunable dynamics in agent-based simulation using multi-objective reinforcement learning. In *Adaptive and Learning Agents Workshop (ALA-19) at AAMAS*, pages 1–7, 2019.

[Kim *et al.*, 2021] Dong-Ki Kim, Miao Liu, Matthew D. Riemer, Chuangchuang Sun, Marwa Abdulhai, Golnaz Habibi, Sebastian Lopez-Cot, Gerald Tesauro, and Jonathan How. A policy gradient algorithm for learning to learn in multiagent reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 5541–5550. PMLR, 2021.

[McArthur *et al.*, 2007] Stephen D. J. McArthur, Euan M. Davidson, Victoria M. Catterson, Aris L. Dimeas, Nikos D. Hatziargyriou, Ferdinanda Ponci, and Toshihisa Funabashi. Multi-agent systems for power engineering applications—part I: Concepts, approaches, and technical challenges. *IEEE Transactions on Power Systems*, 22:1743–1752, 2007.

[Obame Obiang, 2026] Christopher Obame Obiang. Opponent modelling in multi-objective multi-agent decision making. Master's thesis, Utrecht University, Utrecht, the Netherlands, 2026. Available at https://studenttheses.uu.nl/handle/20.500.12932/51073.

[Rădulescu *et al.*, 2020] Roxana Rădulescu, Patrick Mannion, Diederik M. Roijers, and Ann Nowé. Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems*, 34(1):10, 2020.

[Rădulescu *et al.*, 2022] Roxana Rădulescu, Timothy Verstraeten, Yijie Zhang, Patrick Mannion, Diederik M. Roijers, and Ann Nowé. Opponent learning awareness and modelling in multi-objective normal form games. *Neural Computing and Applications*, 34(3):1759–1781, 2022.

[Rădulescu, 2024] Roxana Rădulescu. The world is a multi-objective multi-agent system: Now what? In *ECAI 2024: 27th European Conference on Artificial Intelligence, 19–24 October 2024, Santiago de Compostela, Spain*, pages 32–38. IOS Press, 2024.

[Reymond *et al.*, 2023] Mathieu Reymond, Conor F. Hayes, Denis Steckelmacher, Diederik M. Roijers, and Ann Nowé. Actor-critic multi-objective reinforcement learning for non-linear utility functions. *Autonomous Agents and Multi-Agent Systems*, 37(2):23, 2023.

[Roijers *et al.*, 2013] Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

[Röpke *et al.*, 2023] Willem Röpke, Conor F Hayes, Patrick Mannion, Enda Howley, Ann Nowé, and Diederik M Roi-

jers. Distributional multi-objective decision making. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 5711–5719, 2023.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Sutton and Barto, 2020] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2nd edition, 2020.

[Willi *et al.*, 2022] Timon Willi, Alistair H. Letcher, Johannes Treutlein, and Jakob Foerster. COLA: Consistent learning with opponent-learning awareness. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23804–23831. PMLR, 2022.

[Zhang *et al.*, 2020] Yijie Zhang, Roxana Radulescu, Patrick Mannion, Diederik Roijers, and Ann Nowé. Opponent modelling for reinforcement learning in multi-objective normal form games. In *The 19th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 2080–2082. IFAAMAS, 2020.

[Zhao *et al.*, 2022] Stephen Zhao, Chris Lu, Roger B. Grosse, and Jakob Foerster. Proximal learning with opponent-learning awareness. In *Advances in Neural Information Processing Systems*, volume 35, pages 26324–26336, 2022.

A Implementation Details

A.1 Network Structure

With the exception of MO-PPO-OM, all neural networks used in our experiments consist of a fully connected input layer of 64 units, a single hidden layer of 64 units with Tanh activations, and a fully connected output layer. In MO-PPO-OM, both the actor and the critic instead use two hidden layers: the first 64-unit layer is concatenated with the 64-unit hidden representation produced by the opponent network (giving 128 units in total), which then feeds a second 64-unit hidden layer before the output head. All networks are trained with the Adam optimiser ($\epsilon = 10^{-5}$).

A.2 Hyperparameters

Tables 1–4 list the hyperparameters used for each method. Settings shared across all methods: training is run for 10^5 environment steps with discount factor $\gamma = 1.0$, 4 parallel environments, 256 steps per rollout, 4 minibatches per update and 4 update epochs. The learning rate is linearly annealed over training. Each configuration is evaluated over 30 episodes per checkpoint, and SER curves are averaged over the random seeds.

Table 1: MO-PPO hyperparameters.

Hyperparameter	**Value**
Learning rate	0.01
GAE λ	0.95
Clip coefficient	0.2
Value-loss clipping	yes
Entropy coefficient	0.01
Maximum gradient norm	0.5

Table 2: MO-PPO-OM hyperparameters.

Hyperparameter	**Value**
Learning rate	0.01
GAE λ	0.95
Clip coefficient	0.2
Value-loss clipping	yes
Entropy coefficient	0.01
Maximum gradient norm	0.5
Prediction-loss coefficient	4.0

Table 3: MO-POLA hyperparameters.

Hyperparameter	**Value**
Inner learning rate α_{in}	0.01
Outer learning rate α_{out}	0.01
GAE λ	0.95
Inner steps K	4
Outer steps M	15
Inner penalty coefficient β_{in}	8
Outer penalty coefficient β_{out}	40
Loaded DiCE estimator	yes

Table 4: MO-Meta-MAPG hyperparameters.

Hyperparameter	**Value**
Inner learning rate α_{in}	0.01
Outer learning rate α_{out}	0.01
GAE λ	0.95
Chain horizon L	2
Outer steps M	15
Entropy coefficient	0.01
Maximum gradient norm	0.5