A Regularized Opponent Model with Maximum Entropy Objective

Zheng Tian1⇤ , Ying Wen1⇤ , Zhichen Gong1 , Faiz Punakkath1 , Shihao Zou2 and Jun Wang 1
1University College London
2University of Alberta
{zheng.tian, ying.wen, jun.wang}@cs.ucl.ac.uk

Abstract
In a single-agent setting, reinforcement learning (RL) tasks can be cast into an inference problem by introducing a binary random variable $o$, which stands for the “optimality”. In this paper, we redefine the binary random variable $o$ in multi-agent setting and formalize multi-agent reinforcement learning (MARL) as probabilistic inference. We derive a variational lower bound of the likelihood of achieving the optimality and name it as Regularized Opponent Model with Maximum Entropy Objective (ROMMEO). From ROMMEO, we present a novel perspective on opponent modeling and show how it can improve the performance of training agents theoretically and empirically in cooperative games. To optimize ROMMEO, we first introduce a tabular Q-iteration method ROMMEO-Q with proof of convergence. We extend the exact algorithm to complex environments by proposing an approximate version, ROMMEO-AC. We evaluate these two algorithms on the challenging iterated matrix games and show that they can outperform strong MARL baselines.

1 Introduction
Casting decision making and optimal control as an inference problem have a long history, which dates back to [Kalman, 1960] where the Kalman smoothing is used to solve optimal control in linear dynamics with quadratic cost. Bayesian methods can capture the uncertainties regarding the transition probabilities, the rewards functions in the environment or other agents’ policies. This distributional information can be used to formulate a more structured exploration/exploitation strategy than those commonly used in classical RL, e.g. ϵ-greedy. A common approach in many works [Toussaint and Storkey, 2006; Rawlik et al., 2013; Levine and Koltun, 2013; Abdolmaleki et al., 2018] for framing RL as an inference problem is by introducing a binary random variable $o$ which represents “optimality”. By this way, RL problems are able to lend itself to powerful inference tools [Levine, 2018]. However, the Bayesian approach in a multi-agent environment is less well studied.

In many single-agent works, maximizing entropy is part of a training agent’s objective for resolving ambiguities in inverse reinforcement learning [Ziebart et al., 2008], improving the diversity [Florensa et al., 2017], robustness [Fox et al., 2015] and the compositionality [Haarnoja et al., 2018a] of the learned policy. In Bayesian RL, it often presents in the evidence lower bound (ELBO) for the log likelihood of optimality [Haarnoja et al., 2017; Schulman et al., 2017; Haarnoja et al., 2018b], commonly known as maximum entropy objective (MEO), which encourages the optimal policy to maximize the expected return and long term entropy.

In MARL, there is more than one agent interacting with a stationary environment. In contrast with the single agent environment, an agent’s reward not only depends on the current environment state and the agent’s action but also on the actions of others. The existence of other agents increases the uncertainty in the environment. Therefore, the capability of reasoning about other agents’ belief, private information, behavior, strategy, and other characteristics is crucial. A reasoning model can be used in many different ways, but the most common case is where an agent utilize its reasoning model to help its self decision making [Brown, 1951; Heinrich and Silver, 2016; He et al., 2016; Raileanu et al., 2018; Wen et al., 2019; Tian et al., 2018]. In this work, we use the word “opponent” when referring to another agent in the environment irrespective of the environment’s cooperative or adversarial nature.

In our work, we reformulate the MARL problem into Bayesian inference and derive a multi-agent version of MEO, which we call the regularized opponent model with maximum entropy objective (ROMMEO). Optimizing this objective with respect to one agent’s opponent model gives rise to a new perceptive on opponent modeling. We present two off-policy RL algorithms for optimizing ROMMEO in MARL. ROMMEO-Q is applied in discrete action case with proof of convergence. For the complex and continuous action environment, we propose ROMMEO Actor-Critic (ROMMEO-AC), which approximates the former procedure and extend itself to continuous problems. We evaluate these two approaches on the matrix game and the differential game against strong baselines and show that our methods can outperform all the baselines in terms of the overall performance and speed of convergence.

*The first two authors contributed equally.
2 Method
2.1 Stochastic Games
For an \( n \)-agent stochastic game \([\text{Shapley}, 1953]\), we define a
tuple \((S, A^1, \ldots, A^n, R^1, \ldots, R^n, p, T, \gamma)\), where \( S \) denotes
the state space, \( p \) is the distribution of the initial state, \( \gamma \) is a
discount factor, \( A^i \) and \( R^i = R^i(s, a^i, a^{-i}) \) are the action
space and the reward function for agent \( i \in \{1, \ldots, n\} \) re-
spectively. States are transitioned according to \( T : S \times A \rightarrow S \),
where \( A = \{A^1, \ldots, A^n\} \). Agent \( i \) chooses its action
\( a^i \in A^i \) according to the policy \( \pi^i_\theta (a^i | s) \) parameterized by
\( \theta \) conditioning on some given state \( s \in S \). Let us define
the joint policy as the collection of all agents’ policies \( \pi_\theta \) with \( \theta \)
representing the joint parameter. It is convenient to interpret the joint
policy from the perspective of agent \( i \) such that
\[ \pi^i_\theta = (\pi^i_\theta (a^i | s), \pi_{\theta^{-i}} (a^{-i} | s)), \]
where \( a^{-i} = (a^j)_{j \neq i} \), \( \theta^{-i} = (\theta^j)_{j \neq i} \), and \( \pi_{\theta^{-i}} (a^{-i} | s) \) is a compact representation
of the joint policy of all complementary agents of \( i \). At each
stage of the game, actions are taken simultaneously.

In cooperative games, different agents have the same reward function
\( R^i(s, a^i, a^{-i}) = R(s, a^i, a^{-i}), \forall i \in \{1, \ldots, n\} \). Therefore, each agent’s
objective is to maximize the shared expected return:
\[ \max \eta(\pi_\theta) = \mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^t R(s_t, a^i_t, a^{-i}_t) \right], \]
with \((a^i_t, a^{-i}_t)\) sample from \((\pi^i_\theta, \pi_{\theta^{-i}})\).

2.2 A Variational Lower Bound for Multi-Agent
Reinforcement Learning Problems
We transform the control problem into an inference problem
by introducing a binary random variable \( o^i_t \) which serves as the
indicator for “optimality” for each agent \( i \) at each time
step \( t \). Recall that in single agent problem, reward \( R(s_t, a_t) \)
is bounded, but the achievement of the maximum reward given
the action \( a_t \) is unknown. Therefore, in the single-agent case,
\( o^i_t \) indicates the optimality of achieving the bounded maximum
reward \( r^*_t \). It thus can be regarded as a random variable and we have
\( P(o^i_t = 1 | s_t, a_t) \propto \exp(R(s_t, a_t)) \). Intu-
itively, this formulation dictates that higher rewards reflect a
higher likelihood of achieving optimality, i.e., the case when
\( o^i_t = 1 \). However, the definition of “optimality” in the multi-
agent case is subtly different from the one in the single-agent
situation.

In cooperative multi-agent reinforcement learning (CMARL), to define agent \( i \)’s optimality, we first introduce
the definition of optimum and optimal policy:

**Definition 1.** In cooperative multi-agent reinforcement learning, optimum is a strategy profile \((\pi^{1*}, \ldots, \pi^{n*})\) such that:

\[ \mathbb{E}_{s \sim p, a^1_t \sim \pi^{1*}, a^{-1}_t \sim \pi^{-1}_t} \left[ \sum_{t=1}^{\infty} \gamma^t R(s_t, a^i_t, a^{-i}_t) \right] \]

\[ \geq \mathbb{E}_{s \sim p, a^1_t \sim \pi^{1*}, a^{-1}_t \sim \pi^{-1}_t} \left[ \sum_{t=1}^{\infty} \gamma^t R(s_t, a^i_t, a^{-i}_t) \right] \quad \forall \pi, \]

where \( \pi = (\pi^i, \pi^{-i}) \) and Agent \( i \)’s optimal policy is \( \pi^{i*} \).

In CMARL, a single agent’s “optimality” \( o^i_t \) cannot imply
that it obtains the maximum reward because the reward de-
deps on the joint actions of all agents \((a^i, a^{-i})\). Therefore, we define \( o^i_t = 1 \) only indicates that agent \( i \)’s policy at time
step \( t \) is optimal. The posterior probability of agent \( i \)’s opti-
mality given its action \( a^i_t \) is the probability that the action is sampled from the optimal policy:

\[ P(o^i_t | a^i_t) = P(a^i_t \sim \pi^{i*} | a^i_t) = \pi^{i*} (a^i_t). \] (3)

We also assume that given other players actions \( a^{-i}_t \), the posterior probability of agent \( i \)’s optimality is proportional to its
exponential reward:

\[ P(o^i_t = 1 | s_t, a^i_t, a^{-i}_t) \propto \exp(R(s_t, a^i_t, a^{-i}_t)) \] (4)

The conditional probabilities of “optimality” in both of
CMARL and single-agent case have similar forms. However,
it is worth mentioning that the “optimality” in CMARL has a
different interpretation to the one in single-agent case.

For cooperative games, if all agents play optimally, then
agents can receive the maximum rewards, which is the optim-
um of the games. Therefore, given the fact that other agents
are playing their optimal policies \( o^{-i} = 1 \), the probability
that agent \( i \) also plays its optimal policy
\( P(o^i = 1 | o^{-i} = 1) \) is the probability of obtaining the maximum reward from agent \( i \)’s
perspective. Therefore, we define agent \( i \)’s objective as:

\[ \max \mathcal{J} \Delta \log P(o^i_{1:T} = 1 | o^{-i}_{1:T} = 1) \] (5)

As we assume no knowledge of the optimal policies and
the model of the environment, we treat them as latent variables.
To optimize the observed evidence defined in Eq. 5, there-
fore, we use variational inference (VI) with an auxil-
iary distribution over these latent variables
\( q(o^i_{1:T}, a^{-i}_{1:T}, s_{1:T} | o^i_{1:T} = 1, o^{-i}_{1:T} = 1) \).
Without loss of generality, we here derive the solution
for agent \( i \). We factorize
\( q(o^i_{1:T}, a^{-i}_{1:T}, s_{1:T} | o^i_{1:T} = 1, o^{-i}_{1:T} = 1) \) so as to capture agent \( i \)’s conditional policy on
the current state and opponents actions, and beliefs regarding
opponents actions. This way, agent \( i \) will learn optimal policy,
while also possessing the capability to model opponents’ actions \( a^{-i} \).
Using all modelling assumptions, we may factorize
\( q(o^i_{1:T}, a^{-i}_{1:T}, s_{1:T} | o^i_{1:T} = 1, o^{-i}_{1:T} = 1) \) as:

\[ q(o^i_{1:T}, a^{-i}_{1:T}, s_{1:T} | o^i_{1:T} = 1, o^{-i}_{1:T} = 1) \]

\[ = P(s_1) \prod_t P(s_{t+1} | s_t, a_t) q(o^i_t | a^{-i}_t, s_t, o^i_t = a^{-i}_t = 1) \]

\[ \times q(a^{-i}_t | s_t, o^i_t = a^{-i}_t = 1) \]

\[ = P(s_1) \prod_t P(s_{t+1} | s_t, a_t) \pi(a^i_t | s_t, a^{-i}_t) p(a^{-i}_t | s_t), \]

where we have assumed the same initial and states transitions
as in the original model. With this factorization, we derive a
lower bound on the likelihood of optimality of agent $i$:
\[
\log P(o_{1:T}^i = 1 | o_{1:T}^i = 1) \geq J(\pi, \rho) \triangleq \sum_t \mathbb{E}_{(s_t, a_t^i, a_{t-1}^i) \sim q} [R(s_t, a_t^i, a_{t-1}^i)] + H(\pi(a_t^i | s_t, a_{t-1}^i)) - D_{KL}(\rho(a_t^i | s_t) || P(a_t^i | s_t))
\]
(6)
\begin{equation}
= \sum_t \mathbb{E}_{s_t} [\mathbb{E}_{a_t^i, a_{t-1}^i} \rho(\pi(a_t^i | s_t, a_{t-1}^i) | P(a_t^i | s_t))] - D_{KL}(\rho(a_t^i | s_t) || P(a_t^i | s_t))
\end{equation}
(7)

Written out in full, $\rho(a_t^i | s_t, o_{t-1}^i = 1)$ is agent $i$’s opponent model estimating optimal policies of its agents, $\pi(a_t^i | s_t, a_{t-1}^i, o_{t-1}^i = 1)$, with $a_{t-1}^i, o_{t-1}^i = 1$ is the agent $i$’s conditional policy at optimum ($o_{t-1}^i = o_{t-1}^i = 1$) and $P(a_t^i | s_t, a_{t-1}^i) = 1$ is the prior of optimal policy of opponents. In our work, we set the prior $P(a_t^i | s_t, o_{t-1}^i = 1)$ equal to the observed empirical distribution of opponents’ actions given states. As we are only interested in the case where ($o_{t-1}^i = 1, o_{t-1}^i = 1$), we drop them in $\pi, \rho$ and $P(a_t^i | s_t)$ here and thereafter. $H(\cdot)$ is the entropy function. Eq. 6 is a variational lower bound of $\log P(o_{1:T}^i = 1 | o_{1:T}^i = 1)$ and the derivation is deferred to Appendix B.1.

### 2.3 The Learning of Opponent Model

We can further expand Eq. 6 into Eq. 7 and find that it resembles the maximum entropy objective in single-agent reinforcement learning [Kappen, 2005; Todorov, 2007; Ziebart et al., 2008; Haarnoja et al., 2017]. We denote agent $i$’s expectation of reward $R(s_t, a_t^i, a_{t-1}^i)$ plus entropy of the conditional policy $H(\pi(a^i | s, a^-))$ as agent $i$’s maximum entropy objective (MEO). In the multi-agent version, however, it is worthy of noting that optimizing the MEO will lead to the optimization of $\rho$. This can be counter-intuitive at first sight as opponent behaviour models are normally trained with only past state-action data $(s, a^-)$ to predict opponents’ actions.

However, recall that $\rho(a_{t-1}^i | s_t, o_{t-1}^i = 1)$ is modelling opponents’ optimal policies in our work. Given agent $i$’s policy $\pi^i$ being fixed, optimizing MEO with respect to $\rho$ updates agent $i$’s opponent model in the direction of the higher shared reward $R(s, a^i, a^-)$ and the more stochastic conditional policy $\pi^i(a^i | s, a^-)$, making it closer to the real optimal policies of the opponents. Without any regularization, at iteration $d$, agent $i$ can freely learn a new opponent model $\rho^i_{d+1}$ which is the closest to the optimal opponent policies $\pi_{d+1}^i$ from its perspective given $\pi_{d}^i(a^i | s, a^-)$. Next, agent $i$ can optimize the lower bound with respect to $\pi^i_{d+1}(a^i | s, a^-)$ given $\rho^i_{d+1}$. Then we have an EM-like iterative training and can show it monotonically increases the probability that the opponent model $\rho$ is optimal policies of the opponents. Then, by acting optimally to the converged opponent model $\rho^\infty$, we can recover agent $i$’s optimal policy $\pi^i$.

However, it is unrealistic to learn such an opponent model. As the opponents have no access to agent $i$’s conditional policy $\pi_{d}^i(a^i | s, a^-)$, the learning of its policy can be different from the one of agent $i$’s opponent model. Then the actual opponent policies $\pi_{d+1}^i$ can be very different from agent $i$’s converged opponent model $\rho^\infty$ learned in the above way given agent $i$’s conditional policy $\pi_{d}^i(a^i | s, a^-)$. Therefore, acting optimally to an opponent model far from the real opponents’ policies can lead to poor performance.

The last term in Eq. 7 can prevent agent $i$ building an unrealistic opponent model. The Kullback-Leibler (KL) divergence between opponent model and a prior $D_{KL}(\rho(a_t^i | s_t) || P(a_t^i | s_t))$ can act as a regularizer of $\rho$. By setting the prior to the empirical distribution of opponent past behaviour, the KL divergence penalizes $\rho$ heavily if it deviates from the empirical distribution too much. As the objective in Eq. 7 can be seen as a Maximum Entrophy objective for one agent’s policy and opponent model with regularization on the opponent model, we call this objective as Regularized Opponent Model with Maximum Entropy Objective (ROMMEO).

### 3 Multi-Agent Soft Actor Critic

To optimize the ROMMEO in Eq. 7 derived in the previous section, we propose two off-policy algorithms. We first introduce an exact tabular Q-iteration method with proof of convergence. For practical implementation in a complex continuous environment, we then propose the ROMMEO actor critic ROMMEO-AC, which is an approximation to this procedure.

#### 3.1 Regularized Opponent Model with Maximum Entropy Objective Q-Iteration

In this section, we derive a multi-agent version of Soft Q-iteration algorithm proposed in [Haarnoja et al., 2017] and name our algorithm as ROMMEO-Q. The derivation follows from a similar logic to [Haarnoja et al., 2017], but the extension of Soft Q-learning to MARL is still nontrivial. From this section, we slightly modify the objective in Eq. 7 by adding a weighting factor $\alpha$ for the entropy term and the original objective can be recovered by setting $\alpha = 1$.

We first define multi-agent soft Q-function and V-function respectively. Then we can show that the conditional policy and opponent model defined in Eq. 10 and 11 below are optimal solutions with respect to the objective defined in Eq. 7:

**Theorem 1.** We define the soft state-action value function of agent $i$ as:
\[
Q_{\text{soft}}^{\alpha, \pi}(s_t, a_t^i, a_{t-1}^i) = r_t + \mathbb{E}_{(s_{t+1}, a_{t+1}^i, a_{t+1}^-) \sim q} \sum_{l=1}^{\infty} \gamma^l (r_{t+l}) + \alpha H(\pi(\cdot | s_{t+1}, a_{t+1}^-)) - D_{KL}(\rho(\cdot || P(a_{t+1}^- | s_{t+1})))
\]
and soft state value function as:
\[
V^\alpha(s) = \log \sum_{a^-} P(a^- | s) \left( \sum_{a^i} \exp \left( \frac{1}{\alpha} Q_{\text{soft}}^{\alpha, \pi}(s, a^i, a^-) \right) \right)^\alpha
\]
Then the optimal conditional policy and opponent model for Eq. 6 are:
\[
\pi^\alpha(a^i | s, a^-) = \frac{\exp \left( \frac{1}{\alpha} Q_{\text{soft}}^{\alpha, \pi}(s, a^i, a^-) \right)}{\sum_{a^i} \exp \left( \frac{1}{\alpha} Q_{\text{soft}}^{\alpha, \pi}(s, a^i, a^-) \right)}
\]
and
\[
\rho^∗(a^{−i}|s) = \frac{P(a^{−i}|s)}{\sum_{a_i} \exp\left(\frac{1}{\alpha} Q^\text{soft}_i(s, a^i, a^{−i})\right)} \cdot \exp(V^*(s)).
\] (11)

Proof. See Appendix C.2.

Following from Theorem 1, we can find the optimal solution of Eq. 7 by learning the soft multi-agent Q-function first and recover the optimal policy π∗ and opponent model ρ∗ by Equations 10 and 11. To learn the Q-function, we show that it satisfies a Bellman-like equation, which we name it as multi-agent soft Bellman equation:

**Theorem 2.** We define the soft multi-agent Bellman equation for the soft state-action value function Q^\text{soft}_i(s, a^i, a^{−i}) of agent i as
\[
Q^\text{soft}_i(s, a^i, a^{−i}) = r_i + \gamma E_{s_{t+1}}[V^\text{soft}_{s_{t+1}}].
\] (12)

Proof. See Appendix C.3.

With this Bellman equation defined above, we can derive a solution to Eq. 12 with a fixed point iteration, which we call ROMMEO Q-iteration (ROMMEO-Q). Additionally, We can show that it can converge to the optimal Q^\text{soft}_i and V^\text{soft}_s with certain restrictions as stated in [Wen et al., 2019]:

**Theorem 3.** ROMMEO Q-iteration. In a symmetric game with only one global optimum, i.e. \(E_{\pi^*} [Q^*_i(s)] \geq E_{\pi} [Q^*_i(s)]\), where π∗ is the optimal strategy profile. Let Q^\text{soft}_i(·, ·, ·) and V^\text{soft}_s(·) be bounded and assume
\[
\sum_{a^{−i}} P(a^{−i}|s) \left(\sum_{a^i} \exp\left(\frac{1}{\alpha} Q^\text{soft}_i(s, a^i, a^{−i})\right)\right)^{α} < \infty
\]
and that Q^\text{soft}_i < ∞ exists. Then the fixed-point iteration
\[
Q^\text{soft}_i(s_t, a^i_t, a^{−i}_t) \leftarrow r_t + \gamma E_{s_{t+1}}[V^\text{soft}_{s_{t+1}}],
\] (13)
where V^\text{soft}_s(s_t) \leftarrow \log \sum_{a_t} P(a^i_t|s_t) \times \left(\sum_{a^{−i}_t} \exp\left(\frac{1}{\alpha} Q^\text{soft}_i(s_t, a^i_t, a^{−i}_t)\right)\right)^{α}, \forall s_t, a^i_t, a^{−i}_t, converges to Q^\text{soft}_i and V^\text{soft}_s respectively.

Proof. See Appendix C.3.

3.2 Regularized Opponent Model with Maximum Entropy Objective Actor Critic

The ROMMEO-Q assumes we have the model of the environment and is impractical to implement in high-dimensional continuous problems. To solve these problems, we propose the ROMMEO actor critic (ROMMEO-AC) which is a model-free method. We use neural networks (NNs) as function approximators for the conditional policy, opponent model and Q-function and learn these functions by stochastic gradient. We parameterize the Q-function, conditional policy and opponent model by Q^ω(s, a^i, a^{−i}), π^ω(a^i|s, a^{−i}) and ρ^ω(a^{−i}|s_t) respectively.

Without access to the environment model, we first replace the Q-iteration with Q-learning. Therefore, we can train ω to minimize:
\[
J_\omega(\omega) = E_{(s_t, a^i_t, a^{−i}_t) \sim D} \left[ \frac{1}{2} (Q^\omega(s_t, a^i_t, a^{−i}_t) - R(s_t, a^i_t, a^{−i}_t) - \gamma E_{s_{t+1}} \left[ V^\omega(s_{t+1}) \right]^2 \right].
\] (14)

with
\[
V^\omega(s_{t+1}) = Q^\omega(s_{t+1}, a^i_{t+1}, a^{−i}_{t+1}) - \alpha \log \pi^\omega(a^i_{t+1}|s_{t+1}, a^{−i}_{t+1}) + \log P(a^{−i}_{t+1}|s_{t+1}),
\] (15)
where Q^ω are target functions for providing relatively stable target values. We use a^{−i}_{t+1} denoting the action sampled from agent i’s opponent model ρ(a^{−i}|s_t) and it should be distinguished from a^{−i}_t which is the real action taken by agent i’s opponent. Eq. 15 can be derived from Eq. 10 and 11.

To recover the optimal conditional policy and opponent model and avoid intractable inference steps defined in Eq. 10 and 11 in complex problems, we follow the method in [Haarnoja et al., 2018b] where θ and φ are trained to minimize the KL-divergence:
\[
\mathcal{J}_\pi(\theta) = E_{s_t \sim D, a^i_t \sim \rho} \left[ \mathbb{D}_\text{KL} \left( \pi_\theta(s_t) \bigg| \bigg| \frac{\exp(\frac{1}{\alpha} Q^\omega(s_t, a^i_t, a^{−i}_t))}{Z^\omega(s_t, a^{−i}_t)} \right) \right],
\] (16)
\[
\mathcal{J}_\rho(\phi) = E_{(s_t, a^i_t) \sim D} \left[ \mathbb{D}_\text{KL} \left( \rho(s_t) \bigg| \bigg| \frac{P(s_t)}{\frac{\exp(\frac{1}{\alpha} Q^\omega(s_t, a^i_t, a^{−i}_t))}{\pi_\theta(a^i_t|s_t)}} \right) \right].
\] (17)

By using the reparameterization trick: a^{−i}_t = g_\phi(\epsilon^{−i}_t; s_t) and a^i_t = f_\phi(\epsilon^i_t; s_t, a^{−i}_t), we can rewrite the objectives above as:
\[
\mathcal{J}_\pi(\theta) = E_{s_t \sim D, \epsilon^{−i}_t \sim N, \epsilon^i_t \sim \rho} [\alpha \log \pi_\theta(f_\phi(\epsilon^i_t); s_t, a^{−i}_t)],
\] (18)
\[
\mathcal{J}_\rho(\phi) = E_{(s_t, a^i_t) \sim D, \epsilon^{−i}_t \sim N} [\log \rho_\phi(g_\phi(\epsilon^{−i}_t); s_t)]
\]
\[
- \log P(a^{−i}_t|s_t) - Q^\omega(s_{t+1}, a^i_{t+1}, a^{−i}_{t+1}) + \alpha \log \pi_\theta(a^i_t|s_t, g_\phi(\epsilon^i_t; s_t)) + \alpha \log \pi_\theta(a^{−i}_t|s_t, g_\phi(\epsilon^{−i}_t; s_t)).
\] (19)

The gradient of Eq. 14, 18 and 19 with respect to the corresponding parameters are listed as below:
\[
\nabla_\omega J_\omega(\omega) = \nabla_\omega Q^\omega(s_t, a^i_t, a^{−i}_t) (Q^\omega(s_t, a^i_t, a^{−i}_t))
- R(s_t, a^i_t, a^{−i}_t) - \gamma V^\omega(s_{t+1}),
\] (20)
\[
\nabla_\theta J_\pi(\theta) = \nabla_\theta \alpha \log \pi_\theta(a^{−i}_t|s_t, a^{−i}_t) + \nabla_\theta \phi_\theta(\epsilon^{−i}_t; s_t, a^{−i}_t)
\]
\[
(\nabla_\epsilon^{−i}_t \alpha \log \pi_\theta(a^i_t|s_t, a^{−i}_t) - \nabla_\epsilon^{−i}_t Q^\omega(s_t, a^i_t, a^{−i}_t)),
\] (21)
\[
\nabla_\phi J_\rho(\phi) = \nabla_\phi \log \rho_\phi(\hat{a}^{−i}_t|s_t) + (\nabla_{\hat{a}^{−i}_t} \log \rho_\phi(\hat{a}^{−i}_t|s_t)
\]
\[
- \nabla_{\hat{a}^{−i}_t} Q^\omega(s_t, a^i_t, a^{−i}_t) + \nabla_{\hat{a}^{−i}_t} \alpha \log \pi_\theta(a^i_t|s_t, a^{−i}_t)) \nabla_\phi \phi_\theta(\epsilon^i_t; s_t).
\] (22)

We list the pseudo-code of ROMMEO-Q and ROMMEO-AC in Appendix A.
4 Related Works

In early works, the maximum entropy principle has been used in policy search in linear dynamics [Todorov, 2010; Toussaint, 2009; Levine and Koltun, 2013] and path integral control in general dynamics [Kappen, 2005; Theodorou et al., 2010]. Recently, off-policy methods [Haarnoja et al., 2017; Schulman et al., 2017; Nachum et al., 2017] have been proposed to improve the sample efficiency in optimizing MEO. To avoid complex sampling procedure, training a policy in supervised fashion is employed in [Haarnoja et al., 2018b]. Our work is closely related to this series of recent works because ROMMEO is an extension of MEO to MARL.

A few related works to ours have been conducted in multi-agent soft Q-learning [Wen et al., 2019; Wei et al., 2018; Grau-Moya et al., 2018], where variants of soft Q-learning are applied for solving different problems in MARL. However, unlike previous works, we do not take the soft Q-learning as given and apply it to MARL problems with modifications. In our work, we first establish a novel objective ROMMEO and ROMMEO-Q is only an off-policy method we derive with complete convergence proof, which can optimize the objective. There are other ways of optimizing ROMMEO, for example, the on-policy gradient-based methods, but they are not included in the paper.

There has been substantial progress in combining RL with probabilistic inference. However, most of the existing works focus on the single-agent case. The literature of Bayesian methods in MARL is limited. Among these are methods performing on cooperative games with prior knowledge on distributions of the game model and the possible strategies of others [Chalkiadakis and Boutilier, 2003] or policy parameters and possible roles of other agents [Wilson et al., 2010]. In our work, we assume limited prior knowledge of the environment model, optimal policy, opponents or the observations during the play. In addition, our algorithms are fully decentralized at training and execution, which is more challenging than problems from the centralized training [Foerster et al., 2017; Lowe et al., 2017; Rashid et al., 2018].

In our work, we give a new definition of optimality in CMARL and derive a novel objective ROMMEO. We provide two off-policy RL algorithms for optimizing ROMMEO and the exact version comes with convergence proof. In addition, we provide a natural perspective on opponent modeling in coordination problems: biasing one’s opponent model towards the optimum from its perspective but regularizing it with the empirical distribution of opponent’s real behavior.

5 Experiments

5.1 Iterated Matrix Games

We first present the proof-of-principle result of ROMMEO-Q on iterated matrix games where players need to cooperate to achieve the shared maximum reward. To this end, we study the iterated climbing games (ICG) which is a classic purely cooperative two-player stateless iterated matrix games.

Climbing game (CG) is a fully cooperative game proposed in [Claus and Boutilier, 1998] whose payoff matrix is summarized as follows:

$$ R = \begin{bmatrix} A & B & C \\ \begin{bmatrix} (11, 11) & (-30, -30) & (0, 0) \\ (-30, -30) & (7, 7) & (6, 6) \\ (0, 0) & (0, 0) & (5, 3) \end{bmatrix} \end{bmatrix} $$

It is a challenging benchmark because of the difficulty of convergence to its global optimum. There are two Nash equilibrium $\langle A, A \rangle$ and $\langle B, B \rangle$ but one global optimal $\langle A, A \rangle$. The punishment of miscoordination by choosing a certain action increases in the order of $C \rightarrow B \rightarrow A$. The safest action is $C$ and the miscoordination punishment is the most severe for $A$. Therefore it is very difficult for agents to converge to the global optimum in ICG.

We compare our method to a series of strong baselines in MARL, including Joint Action Learner (JAL) [Claus and Boutilier, 1998], WoLF Policy Hillclimbing (WoLF-PhC) [Bowling and Veloso, 2001], Frequency Maximum Q (FMQ) [Kapetanakis and Kudenko, 2002] and Probabilistic Recursive Reasoning (PR2) [Wen et al., 2019]. ROMMEO-Q-EMP is an ablation study to evaluate the effectiveness of our proposed opponent model learning process, where we replace our opponent model with empirical frequency. Fig. 1a shows the learning curves on ICG for different algorithms. The difference of rewards between ROMMEO-Q and FMQ-c10 may seem small because of the small reward margin between the global optimum and the local one. However, ROMMEO-Q actually outperforms all baselines significantly in terms of converging to the global optimum, which is shown

---

1The experiment code and appendix are available at https://github.com/rommeoijcai2019/rommeo.
in Fig. 1b. To further analyze the opponent modeling described in Sec. 2.3, we visualize the probability of agent \(-i\) taking the optimal action \(A\) estimated by agent \(i\)’s opponent model \(\rho\) and its true policy \(\pi^{-1}\) in Fig. 1c. Agent \(i\)’s opponent model “thinks ahead” of agent \(-i\) and converges to agent \(-i\)’s optimal policy before agent \(-i\) itself converges to the optimal policy. This helps agent \(i\) to respond to its opponent model optimally by choosing action \(A\), which in turn leads to the improvement of agent \(-i\)’s opponent model and policy. Therefore, the game converges to the global optimum. To note, the big drop of \(P(A)\) for both policies and opponent models at the beginning of the training comes from the severe punishment of miscoordination associated with action \(A\).

5.2 Differential Games

We adopt the differential Max of Two Quadratic Game [Wei et al., 2018] for continuous case. The agents have continuous action space of \([-10, 10]\). Each agent’s reward depends on the joint action following the equations: \(r^i(a^1, a^2) = r^2(a^1, a^2) = \max(f_1, f_2)\), where \(f_1 = 0.8 \times \left[-\left(\frac{a^1 + 5}{3}\right)^2 - \left(\frac{a^2 + 5}{4}\right)^2\right], f_2 = 1.0 \times \left[-\left(\frac{a^1 - 5}{3}\right)^2 - \left(\frac{a^2 - 5}{4}\right)^2\right] + 10\). We compare the algorithm with a series of baselines including PR2 [Wen et al., 2019], MASQL [Wei et al., 2018; Grau-Moya et al., 2018], MADDPG [Lowe et al., 2017] and independent learner via DDPG [Lillicrap et al., 2015]. To compare against traditional opponent modeling methods, similar to [Rabinowitz et al., 2018; He et al., 2016], we implement an additional baseline of DDPG with an opponent module that is trained online with supervision in order to capture the latest opponent behaviors, called DDPG-OM. We trained all agents for 200 episodes with 25 steps per episode.

This is a challenging task to most continuous gradient based RL algorithms because gradient update tends to direct the training agent to the sub-optimal point. The reward surface is provided in Fig. 2a; there is a local maximum 0 at \((-5, -5)\) and a global maximum 10 at \((5, 5)\), with a deep valley staying in the middle. If the agents’ policies are initialized to \((0, 0)\) (the red starred point) that lies within the basin of the left local maximum, the gradient-based methods would tend to fail to find the global maximum equilibrium point due to the valley blocking the upper right area.

A learning path of ROMMEO-AC is summarized in Fig. 2a and the solid bright circle on the right corner implies the convergence to the global optimum. The learning curve is presented in Fig. 2b. ROMMEO-AC shows the capability of converging to the global optimum in a limited amount of steps, while most of the baselines can only reach the sub-optimal point. PR2-AC can also achieve the global optimum but requires many more steps to explore and learn. Additionally, fine tuning on the exploration noise or separate exploration stage is required for deterministic RL methods (MADDPG, DDPG, DDPG-OM, PR2-AC), and the learning outcomes of energy-based RL method (MASQL) is extremely sensitive to the annealing scheme for the temperature. In contrast, ROMMEO-AC employs a stochastic policy and controls the exploration level by the weighting factor \(\alpha\). It does not need a separate exploration stage at the beginning of the training or a delicately designed annealing scheme for \(\alpha\).

Furthermore, we analyze the learning path of policy \(\pi\) and modeled opponent policy \(\rho\) during the training, the results are shown in Fig. 2c. The red and orange lines are mean of modeled opponent policy \(\rho\), which always learn to approach the optimal ahead of the policy \(\pi\) (in dashed blue and green lines). This helps the agents to establish the trust and converge to the optimum quickly, which further justifies the effectiveness and benefits of conducting a regularized opponent model proposed in Sec. 2.3.

6 Conclusion

In this paper, we use Bayesian inference to formulate MARL problem and derive a novel objective ROMMEO which gives rise to a new perspective on opponent modeling. We design an off-policy algorithm ROMMEO-Q with complete convergence proof for optimizing ROMMEO. For better generality, we also propose ROMMEO-AC, an actor critic algorithm powered by NNs to solve complex and continuous problems. We give an insightful analysis of the effect of the new learning process of the opponent modeling on agent’s performance in MARL. We evaluate our methods on the challenging matrix game and differential game and show that they can outperform a series of strong base lines. It is worthy of noting that Theorems 1, 2 and 3 only guarantees the convergence to optimal solutions with respect to ROMMEO objective but not the optimum in the game. The achievement of the optimum in the game also relies on the opponent learning algorithm. In our work, we demonstrate that ROMMEO-Q/AC’s convergence to the optimum of the game in self-play setting. The convergence to optimum of games in non-self-play settings will be studied in our future work.