A Regularized Opponent Model with Maximum Entropy Objective

Zheng Tian[1*] , **Ying Wen**[1*] , **Zhichen Gong**[1] , **Faiz Punakkath**[1] , **Shihao Zou**[2] and **Jun Wang** [1]

[1]University College London
[2]University 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 game and differential game respectively 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 [Shapley, 1953], we define a tuple $(\mathcal{S}, \mathcal{A}^1, \ldots, \mathcal{A}^n, R^1, \ldots, R^n, p, \mathcal{T}, \gamma)$, where $\mathcal{S}$ denotes the state space, p is the distribution of the initial state, γ is a discount factor, $\mathcal{A}^i$ and $R^i = R^i(s, a^i, a^{-i})$ are the action space and the reward function for agent $i \in \{1, \ldots, n\}$ respectively. States are transitioned according to $\mathcal{T} : \mathcal{S} \times \mathcal{A}$, where $\mathcal{A} = \{\mathcal{A}^1, \cdots, \mathcal{A}^n\}$. Agent i chooses its action $a^i \in \mathcal{A}^i$ according to the policy $\pi^i_{\theta^i}(a^i|s)$ parameterized by θ^i conditioning on some given state $s \in \mathcal{S}$. Let us define the joint policy as the collection of all agents' policies π_θ with θ representing the joint parameter. It is convenient to interpret the joint policy from the perspective of agent i such that $\pi_\theta = (\pi^i_{\theta^i}(a^i|s), \pi^{-i}_{\theta^{-i}}(a^{-i}|s))$, where $a^{-i} = (a^j)_{j \neq i}$, $\theta^{-i} = (\theta^j)_{j \neq i}$, and $\pi^{-i}_{\theta^{-i}}(a^{-i}|s)$ is a compact representation of the joint policy of all complementary agents of i. At each stage of the game, actions are taken simultaneously. Each agent is presumed to pursue the maximal cumulative reward, expressed as

$$\max \quad \eta^i(\pi_\theta) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t R^i(s_t, a^i_t, a^{-i}_t)\right], \quad (1)$$

with (a^i_t, a^{-i}_t) sample from $(\pi^i_{\theta^i}, \pi^{-i}_{\theta^{-i}})$. In fully cooperative games, we assume there exists at least one joint policy π_θ such that all agents can achieve the maximal cumulative reward with this joint policy.

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_t indicates the optimality of achieving the bounded maximum reward r^*_t. It thus can be regarded as a random variable and we have $P(o_t = 1|s_t, a_t) \propto \exp(R(s_t, a_t))$. Intuitively, this formulation dictates that higher rewards reflect a higher likelihood of achieving optimality, i.e., the case when $o_t = 1$. However, the definition of "optimality" in the multi-agent case is subtlety 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:*

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

$$\geq \mathbb{E}_{s \sim p_s, a^i_t \sim \pi^i, a^{-i}_t \sim \pi^{-i}}\left[\sum_{t=1}^{\infty} \gamma^t R^i(s_t, a^i_t, a^{-i}_t)\right] \quad (2)$$

$$\forall \pi \in \Pi, i \in (1 \ldots n),$$

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

In CMARL, a single agent's "optimality" o^i_t cannot imply that it obtains the maximum reward because the reward depends 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 optimality given its action a^i_t is the probability that the action is sampled from the optimal policy:

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

We also assume that given other players playing optimally $(o^{-i}_t = 1)$, the higher the reward agent i receives the higher the probability of agent i's current policy is optimal $(o^i_t = 1)$:

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

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 optimum 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:

$$\max \quad \mathcal{J} \triangleq \log P(o^i_{1:T} = 1|o^{-i}_{1:T} = 1) \quad (5)$$

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, therefore, we use variational inference (VI) with an auxiliary distribution over these latent variables $q(a^i_{1:T}, a^{-i}_{1:T}, s_{1:T}|o^i_{1:T} = 1, o^{-i}_{1:T} = 1)$. Without loss of generality, we here derive the solution for agent i. We factorize $q(a^i_{1:T}, a^{-i}_{1:T}, s_{1:T}|o^i_{1:T} = 1, o^{-i}_{1:T} = 1)$ 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 $q(a^i_{1:T}, a^{-i}_{1:T}, s_{1:T}|o^i_{1:T} = 1, o^{-i}_{1:T} = 1)$ as:

$$q(a^i_{1:T}, a^{-i}_{1:T}, s_{1:T}|o^i_{1:T} = 1, o^{-i}_{1:T} = 1)$$
$$= P(s_1) \prod_t P(s_{t+1}|s_t, a_t)q(a^i_t|a^{-i}_t, s_t, o^i_t = o^{-i}_t = 1)$$
$$\times q(a^{-i}_t|s_t, o^i_t = o^{-i}_t = 1)$$
$$= P(s_1) \prod_t P(s_{t+1}|s_t, a_t)\pi(a^i_t|s_t, a^{-i}_t)\rho(a^{-i}_t|s_t),$$

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:

$$\log P(o^i_{1:T} = 1 | o^{-i}_{1:T} = 1)$$

$$\geq \mathcal{J}(\pi, \rho) \triangleq \sum_t \mathbb{E}_{(s_t, a^i_t, a^{-i}_t) \sim q}[R^i(s_t, a^i_t, a^{-i}_t)$$

$$+ H(\pi(a^i_t | s_t, a^{-i}_t)) - D_{\text{KL}}(\rho(a^{-i}_t | s_t) || P(a^{-i}_t | s_t))] \quad (6)$$

$$= \sum_t \mathbb{E}_{s_t} [\underbrace{\mathbb{E}_{a^i_t \sim \pi, a^{-i}_t \sim \rho}[R^i(s_t, a^i_t, a^{-i}_t) + H(\pi(a^i_t | s_t, a^{-i}_t))]}_{\text{MEO}}$$

$$- \underbrace{\mathbb{E}_{a^{-i}_t \sim \rho}[D_{\text{KL}}(\rho(a^{-i}_t | s_t) || P(a^{-i}_t | s_t))]]}_{\text{Regularizer of } \rho}. \quad (7)$$

Written out in full, $\rho(a^{-i}_t | s_t, o^{-i}_t = 1)$ is agent i's opponent model estimating *optimal* policies of its opponents, $\pi(a^i_t | s_t, a^{-i}_t, o^i_t = 1, o^{-i}_t = 1)$ is the agent i's conditional policy at optimum ($o^i_t = o^{-i}_t = 1$) and $P(a^{-i}_t | s_t, o^{-i}_t = 1)$ is the prior of optimal policy of opponents. In our work, we set the prior $P(a^{-i}_t | s_t, o^{-i}_t = 1)$ equal to the observed empirical distribution of opponents' actions given states. As we are only interested in the case where ($o^i_t = 1, o^{-i}_t = 1$), we drop them in π, ρ and $P(a^{-i}_t | s_t)$ here and thereafter. $H(\cdot)$ is the entropy function. Eq. 6 is a variational lower bound of $\log P(o^i_{1:T} = 1 | o^{-i}_{1:T} = 1)$ 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 we 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 $R(s_t, a^i_t, a^{-i}_t)$ plus entropy of the conditional policy $H(\pi(a^i | s, a^{-i}))$ 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 ρ*. This can be counter-intuitive at first sight as opponent behaviour models are normally trained with only past state-action data (s, a^{-i}) to predict opponents' actions.

However, recall that $\rho(a^{-i}_t | s_t, o^{-i}_t = 1)$ is modelling opponents' *optimal* policies in our work. Given agent i's policy π^i being fixed, optimizing MEO with respect to ρ updates agent i's opponent model in the direction of the higher shared reward $R(s, a^i, a^{-i})$ and the more stochastic conditional policy $\pi^i(a^i | s, a^{-i})$, 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 ρ^i_{d+1} which is the closest to the optimal opponent policies π^{-i*} from its perspective given $\pi^i_d(a^i | s, a^{-i})$. Next, agent i can optimize the lower bound with respect to $\pi^i_{d+1}(a^i | s, a^{-i})$ given ρ^i_{d+1}. Then we have an EM-like iterative training and can show it monotonically increases the probability that the opponent model ρ is optimal policies of the opponents. Then, by acting optimally to the converged opponent model $\rho^{i\infty}$, we can recover agent i's optimal policy π^{i*}.

However, it is unrealistic to learn such an opponent model. As the opponents have no access to agent i's conditional policy $\pi^i_d(a^i | s, a^{-i})$, the learning of its policy can be different from the one of agent i's opponent model. Then the actual opponent policies π^{-i}_{d+1} can be very different from agent i's converged opponent model $\rho^{i\infty}$ learned in the above way given agent i's conditional policy $\pi^i_d(a^i | s, a^{-i})$. 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 $D_{\text{KL}}(\rho(a^{-i}_t | s_t) || P(a^{-i}_t | s_t))$ can act as a regularizer of ρ. By setting the prior to the empirical distribution of opponent past behaviour, the KL divergence penalizes ρ heavily if it deviates from the empirical distribution too much. As the objective in Eq. 7 can be seen as a *Maximum Entropy 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 we 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 α 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*

$$Q^{\pi^*, \rho^*}_{soft}(s_t, a^i_t, a^{-i}_t) = r_t + \mathbb{E}_{(s_{t+l}, a^i_{t+l}, a^{-i}_{t+l}, ...) \sim q}[\sum_{l=1}^{\infty} \gamma^l (r_{t+l}$$

$$+ \alpha H(\pi^*(a^i_{t+l} | a^{-i}_{t+l}, s_{t+l})) - D_{\text{KL}}(\rho^*(a^{-i}_{t+l} | s_{t+l}) || P(a^{-i}_{t+l} | s_{t+l}))], \quad (8)$$

and soft state value function as

$$V^*(s) = \log \sum_{a^{-i}} P(a^{-i} | s) \left(\sum_{a^i} \exp(\frac{1}{\alpha} Q^*_{soft}(s, a^i, a^{-i})) \right)^{\alpha}, \quad (9)$$

Then the optimal conditional policy and opponent model for Eq. 6 are

$$\pi^*(a^i | s, a^{-i}) = \frac{\exp(\frac{1}{\alpha} Q^{\pi^*, \rho^*}_{soft}(s, a^i, a^{-i}))}{\sum_{a^i} \exp(\frac{1}{\alpha} Q^{\pi^*, \rho^*}_{soft}(s, a^i, a^{-i}))}, \quad (10)$$

and

$$\rho^*(a^{-i}|s) = \frac{P(a^{-i}|s)\left(\sum_{a^i}\exp(\frac{1}{\alpha}Q^*_{soft}(s,a^i,a^{-i}))\right)^\alpha}{\exp(V^*(s))}.$$

(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 $Q^{\pi,\rho}_{soft}(s,a^i,a^{-i})$ of agent i as*

$$Q^{\pi^*,\rho^*}_{soft}(s,a^i,a^{-i}) = r_t + \gamma\mathbb{E}_{(s_{t+1})}[V^*_{soft}(s_{t+1})].$$

(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^*_{soft} and V^*_{soft} 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. $\mathbb{E}_{\pi^*}\left[Q^i_t(s)\right] \geq \mathbb{E}_\pi\left[Q^i_t(s)\right]$, where π^* is the optimal strategy profile. Let $Q_{soft}(\cdot,\cdot,\cdot)$ and $V_{soft}(\cdot)$ be bounded and assume*

$$\sum_{a^{-i}} P(a^{-i}|s)\left(\sum_{a^i}\exp(\frac{1}{\alpha}Q^*_{soft}(s,a^i,a^{-i}))\right)^\alpha < \infty$$

and that $Q^*_{soft} < \infty$ exists. Then the fixed-point iteration

$$Q_{soft}(s_t,a^i_t,a^{-i}_t) \leftarrow r_t + \gamma\mathbb{E}_{(s_{t+1})}[V_{soft}(s_{t+1})],$$

(13)

where $V_{soft}(s_t) \leftarrow \log\sum_{a^{-i}_t} P(a^{-i}_t|s_t) \times \left(\sum_{a^i_t}\exp(\frac{1}{\alpha}Q_{soft}(s_t,a^i_t,a^{-i}_t))\right)^\alpha$, $\forall s_t,a^i_t,a^{-i}_t$, *converges to Q^*_{soft} and V^*_{soft} 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 $Q_\omega(s,a^i,a^{-i})$, $\pi_\theta(a^i_t|s_t,a^{-i}_t)$ and $\rho_\phi(a^{-i}_t|s_t)$ respectively.

Without access to the environment model, we first replace the Q-iteration with Q-learning. Therefore, we can train ω to minimize:

$$\mathcal{J}_Q(\omega) = \mathbb{E}_{(s_t,a^i_t,a^{-i}_t)\sim\mathcal{D}}[\frac{1}{2}(Q_\omega(s_t,a^i_t,a^{-i}_t) \\ - R(s_t,a^i_t,a^{-i}_t) - \gamma\mathbb{E}_{s_{t+1}\sim p_s}[\bar{V}(s_{t+1})])^2], \quad (14)$$

with

$$\bar{V}(s_{t+1}) = Q_{\bar{\omega}}(s_{t+1},a^i_{t+1},\hat{a}^{-i}_{t+1}) - \log\rho_\phi(\hat{a}^{-i}_{t+1}|s_{t+1}) \\ - \alpha\log\pi_\theta(a^i_{t+1}|s_{t+1},\hat{a}^{-i}_{t+1}) + logP(\hat{a}^{-i}_{t+1}|s_{t+1}), \quad (15)$$

where $Q_{\bar{\omega}}$ are target functions for providing relatively stable target values. We use $\hat{a}^{-i}_t$ denoting the action sampled from agent i's opponent model $\rho(a^{-i}_t|s_t)$ 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:

$$\mathcal{J}_\pi(\theta) = \mathbb{E}_{s_t\sim D,a^{-i}_t\sim\rho} \\ \left[D_{\mathrm{KL}}\left(\pi_\theta(\cdot|s_t,\hat{a}^{-i}_t)\,\middle\|\,\frac{\exp(\frac{1}{\alpha}Q_\omega(s_t,\cdot,\hat{a}^{-i}_t))}{Z_\omega(s_t,\hat{a}^{-i}_t)}\right)\right], \quad (16)$$

$$\mathcal{J}_\rho(\phi) = \mathbb{E}_{(s_t,a^i_t)\sim D} \\ \left[D_{\mathrm{KL}}\left(\rho(\cdot|s_t)\,\middle\|\,\frac{P(\cdot|s_t)\left(\frac{\exp(\frac{1}{\alpha}Q(s_t,a^i_t,\cdot))}{\pi_\theta(a^i_t|s_t,\cdot)}\right)^\alpha}{Z_\omega(s_t)}\right)\right]. \quad (17)$$

By using the reparameterization trick: $\hat{a}^{-i}_t = g_\phi(\epsilon^{-i}_t;s_t)$ and $a^i_t = f_\theta(\epsilon^i_t;s_t,\hat{a}^{-i}_t)$, we can rewrite the objectives above as:

$$\mathcal{J}_\pi(\theta) = \mathbb{E}_{s_t\sim D,\epsilon^i_t\sim N,\hat{a}^{-i}_t\sim\rho}[\alpha\log\pi_\theta(f_\theta(\epsilon^i_t;s_t,\hat{a}^{-i}_t)) \\ - Q_\omega(s_t,f_\theta(\epsilon^i_t;s_t,\hat{a}^{-i}_t),\hat{a}^{-i}_t)], \quad (18)$$

$$\mathcal{J}_\rho(\phi) = \mathbb{E}_{(s_t,a_t)\sim D,\epsilon^{-i}_t\sim N}[\log\rho_\phi(g_\phi(\epsilon^{-i}_t;s_t)|s_t) \\ - logP(\hat{a}^{-i}_t|s_t) - Q(s_t,a^i_t,g_\phi(\epsilon^{-i}_t;s_t)) \\ + \alpha\log\pi_\theta(a^i_t|s_t,g_\phi(\epsilon^{-i}_t;s_t))]. \quad (19)$$

The gradient of Eq. 14, 18 and 19 with respect to the corresponding parameters are listed as below:

$$\nabla_\omega\mathcal{J}_Q(\omega) = \nabla_\omega Q_\omega(s_t,a^i_t,a^{-i}_t)(Q_\omega(s_t,a^i_t,a^{-i}_t) \\ - R(s_t,a^i_t,a^{-i}_t) - \gamma\bar{V}(s_{t+1})), \quad (20)$$

$$\nabla_\theta\mathcal{J}_\pi(\theta) = \nabla_\theta\alpha\log\pi_\theta(a^i_t|s_t,\hat{a}^{-i}_t) + \nabla_\theta f_\theta(\epsilon^i_t;s_t,\hat{a}^{-i}_t) \\ (\nabla_{a^i_t}\alpha\log\pi_\theta(a^i_t|s_t,\hat{a}^{-i}_t) - \nabla_{a^i_t}Q_\omega(s_t,a^i_t,\hat{a}^{-i}_t)), \quad (21)$$

$$\nabla_\phi\mathcal{J}_\rho(\phi) = \nabla_\phi\log\rho_\phi(\hat{a}^{-i}_t|s_t) + (\nabla_{\hat{a}^{-i}_t}\log\rho_{\phi^i}(\hat{a}^{-i}_t|s_t) \\ - \nabla_{\hat{a}^{-i}_t}\log P(\hat{a}^{-i}_t|s_t) - \nabla_{\hat{a}^{-i}_t}Q_{\omega^i}(s_t,a^i_t,\hat{a}^{-i}_t) \\ + \nabla_{\hat{a}^{-i}_t}\alpha\log\pi_\theta(a^i_t|s_t,\hat{a}^{-i}_t))\nabla_\phi g_\phi(\epsilon^{-i}_t;s_t). \quad (22)$$

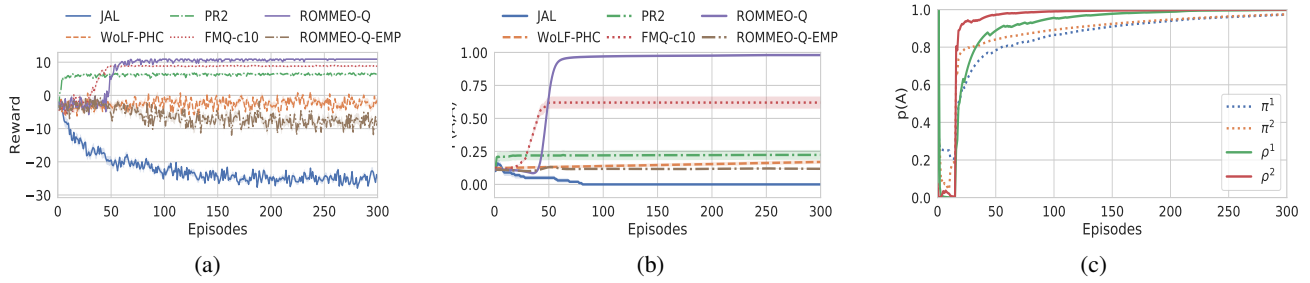We list the pseudo-code of ROMMEO-Q and ROMMEO-AC in Appendix A.

Figure 1: (a): Learning curves of ROMMEO and baselines on ICG over 100 episodes. (b): Probability of convergence to the global optimum for ROMMEO and baselines on ICG over 100 episodes. The vertical axis is the joint probability of taking actions A for both agents. (c): Probability of taking A estimated by agent i's opponent model ρ^i and observed empirical frequency P^i in one trail of training, $i \in \{1, 2\}$.

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 ROM-MEO, 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 very 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 co-ordination 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[1] 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:

$$R = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{ccc} A & B & C \\ \left[\begin{array}{ccc} (11, 11) & (-30, -30) & (0, 0) \\ (-30, -30) & (7, 7) & (6, 6) \\ (0, 0) & (0, 0) & (5, 3) \end{array} \right] \end{array}. \quad \text{It is}$$

a challenging benchmark because of the difficulty of convergence to its global optimum. There are two Nash equilibrium (A, A) and (B, B) but one global optimal (A, A). The punishment of miscoordination by choosing a certain action increases in the order of $C \to B \to A$. 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

---

[1]The experiment code and appendix are available at https://github.com/rommeoijcai2019/rommeo.
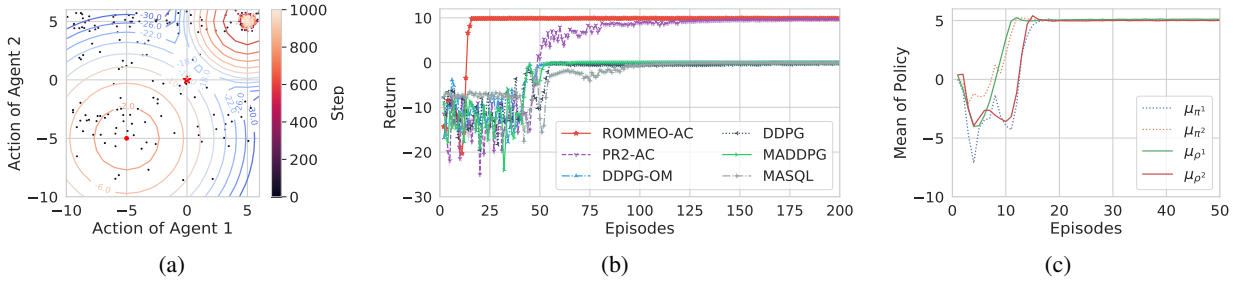
Figure 2: Experiment on Max of Two Quadratic Game. (a) Reward surface and learning path of agents. Scattered points are actions taken at each step; (b) Learning curve of ROMMEO and baselines. (c) Mean of agents' policies π and opponent models ρ.

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 ρ^i and its true policy π^{-i} 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: $r^1\left(a^1, a^2\right) = r^2\left(a^1, a^2\right) = \max\left(f_1, f_2\right)$, where $f_1 = 0.8 \times [-(\frac{a^1+5}{3})^2 - (\frac{a^2+5}{3})^2]$, $f_2 = 1.0 \times [-(\frac{a^1-5}{1})^2 - (\frac{a^2-5}{1})^2] + 10$. 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 suboptimal 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 α. It does not need a separate exploration stage at the beginning of the training or a delicately designed annealing scheme for α.

Furthermore, we analyze the learning path of policy π and modeled opponent policy ρ during the training, the results are shown in Fig. 2c. The red and orange lines are mean of modeled opponent policy ρ, which always learn to approach the optimal ahead of the policy π (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.

References

[Abdolmaleki *et al.*, 2018] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Rémi Munos, Nicolas Heess, and Martin A. Riedmiller. Maximum a posteriori policy optimisation. *CoRR*, abs/1806.06920, 2018.

[Bowling and Veloso, 2001] Michael Bowling and Manuela Veloso. Rational and convergent learning in stochastic games. In *IJCAI*, San Francisco, CA, USA, 2001.

[Brown, 1951] George W. Brown. Iterative solution of games by fictitious play. In *AAPA*. New York, 1951.

[Chalkiadakis and Boutilier, 2003] Georgios Chalkiadakis and Craig Boutilier. Coordination in multiagent reinforcement learning: A bayesian approach. In *AAMAS*, pages 709–716, New York, NY, USA, 2003. ACM.

[Claus and Boutilier, 1998] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. AAAI '98/IAAI '98, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.

[Florensa *et al.*, 2017] Carlos Florensa, Yan Duan, and Pieter Abbeel. Stochastic Neural Networks for Hierarchical Reinforcement Learning. *arXiv e-prints*, April 2017.

[Foerster *et al.*, 2017] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual Multi-Agent Policy Gradients. *arXiv e-prints*, page arXiv:1705.08926, May 2017.

[Fox *et al.*, 2015] Roy Fox, Ari Pakman, and Naftali Tishby. Taming the Noise in Reinforcement Learning via Soft Updates. *arXiv e-prints*, page arXiv:1512.08562, December 2015.

[Grau-Moya *et al.*, 2018] Jordi Grau-Moya, Felix Leibfried, and Haitham Bou-Ammar. Balancing Two-Player Stochastic Games with Soft Q-Learning. *arXiv e-prints*, page arXiv:1802.03216, February 2018.

[Haarnoja *et al.*, 2017] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *CoRR*, abs/1702.08165, 2017.

[Haarnoja *et al.*, 2018a] Tuomas Haarnoja, Vitchyr Pong, Aurick Zhou, Murtaza Dalal, Pieter Abbeel, and Sergey Levine. Composable Deep Reinforcement Learning for Robotic Manipulation. *arXiv e-prints*, page arXiv:1803.06773, March 2018.

[Haarnoja *et al.*, 2018b] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv e-prints*, page arXiv:1801.01290, January 2018.

[He *et al.*, 2016] He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. Opponent modeling in deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1804–1813, New York, New York, USA, 20–22 Jun 2016. PMLR.

[Heinrich and Silver, 2016] Johannes Heinrich and David Silver. Deep reinforcement learning from self-play in imperfect-information games. *CoRR*, abs/1603.01121, 2016.

[Kalman, 1960] Rudolf Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of basic Engineering*, 82:35–45, 01 1960.

[Kapetanakis and Kudenko, 2002] Spiros Kapetanakis and Daniel Kudenko. Reinforcement learning of coordination in cooperative multi-agent systems. In *Eighteenth National Conference on Artificial Intelligence*, Menlo Park, CA, USA, 2002.

[Kappen, 2005] Hilbert J. Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11), 2005.

[Levine and Koltun, 2013] Sergey Levine and Vladlen Koltun. Variational policy search via trajectory optimization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *NIPS '13*. 2013.

[Levine, 2018] Sergey Levine. Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review. *arXiv e-prints*, page arXiv:1805.00909, May 2018.

[Lillicrap *et al.*, 2015] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[Lowe *et al.*, 2017] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *arXiv e-prints*, page arXiv:1706.02275, June 2017.

[Nachum *et al.*, 2017] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the Gap Between Value and Policy Based Reinforcement Learning. *arXiv e-prints*, page arXiv:1702.08892, February 2017.

[Rabinowitz *et al.*, 2018] Neil C Rabinowitz, Frank Perbet, H Francis Song, Chiyuan Zhang, SM Eslami, and Matthew Botvinick. Machine theory of mind. *arXiv preprint arXiv:1802.07740*, 2018.

[Raileanu *et al.*, 2018] Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. *CoRR*, abs/1802.09640, 2018.

[Rashid *et al.*, 2018] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *arXiv e-prints*, page arXiv:1803.11485, March 2018.

[Rawlik *et al.*, 2013] Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference (extended abstract). IJCAI '13, pages 3052–3056, 2013.

[Schulman *et al.*, 2017] John Schulman, Xi Chen, and Pieter Abbeel. Equivalence Between Policy Gradients and Soft Q-Learning. *arXiv e-prints*, page arXiv:1704.06440, April 2017.

[Shapley, 1953] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

[Theodorou *et al.*, 2010] Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. A generalized path integral control approach to reinforcement learning. *J. Mach. Learn. Res.*, 11, December 2010.

[Tian *et al.*, 2018] Zheng Tian, Shihao Zou, Tim Warr, Lisheng Wu, and Jun Wang. Learning Multi-agent Implicit Communication Through Actions: A Case Study in Contract Bridge, a Collaborative Imperfect-Information Game. *arXiv e-prints*, page arXiv:1810.04444, October 2018.

[Todorov, 2007] Emanuel Todorov. Linearly-solvable markov decision problems. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *NIPS '19*. 2007.

[Todorov, 2010] Emanuel Todorov. Policy gradients in linearly-solvable mdps. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *NIPS*. 2010.

[Toussaint and Storkey, 2006] Marc Toussaint and Amos Storkey. Probabilistic inference for solving discrete and continuous state markov decision processes. ICML '06, pages 945–952, New York, NY, USA, 2006. ACM.

[Toussaint, 2009] Marc Toussaint. Robot trajectory optimization using approximate inference. ICML '09, pages 1049–1056, New York, NY, USA, 2009. ACM.

[Wei *et al.*, 2018] Ermo Wei, Drew Wicke, David Freelan, and Sean Luke. Multiagent soft q-learning. *AAAI*, 2018.

[Wen *et al.*, 2019] Ying Wen, Yaodong Yang, Rui Luo, Jun Wang, and Wei Pan. Probabilistic recursive reasoning for multi-agent reinforcement learning. In *ICLR*, 2019.

[Wilson *et al.*, 2010] Aaron Wilson, Alan Fern, and Prasad Tadepalli. Bayesian policy search for multi-agent role discovery. AAAI'10, pages 624–629. AAAI Press, 2010.

[Ziebart *et al.*, 2008] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI '08*, 2008.

A   Algorithms

Algorithm 1 Multi-agent Soft Q-learning

Result: policy π^i, opponent model ρ^i

Initialization:
Initialize replay buffer $\mathcal{M}$ to capacity M.
Initialize $Q_{\omega^i}(s, a^i, a^{-i})$ with random parameters ω^i, $P(a^{-i}|s)$ arbitrarily, set γ as the discount factor.
Initialize target $Q_{\bar{\omega}^i}(s, a^i, a^{-i})$ with random parameters $\bar{\omega}^i$, set C the target parameters update interval.

while not converge **do**

   Collect experience

   For the current state s_t compute the opponent model $\rho^i(a_t^{-i}|s_t)$ and conditional policy $\pi^i(a_t^i|s_t, a_t^{-i})$ respectively from:

$$\rho^i(a_t^{-i}|s_t) \propto P(a_t^{-i}|s_t) \left(\sum_{a_t^i} \exp(\frac{1}{\alpha} Q_{\omega^i}(s_t, a_t^i, a_t^{-i})) \right)^{\alpha},$$

$$\pi^i(a_t^i|s_t, \hat{a}_t^{-i}) \propto \exp(\frac{1}{\alpha} Q_{\omega^i}(s_t, a_t^i, \hat{a}_t^{-i})).$$

   Compute the marginal policy $\pi^i(a_t^i|s_t)$ and sample an action from it:

$$a_t^i \sim \pi^i(a_t^i|s_t) = \sum_{a^{-i}} \pi^i(a_t^i|s_t, a_t^{-i}) \rho(a_t^{-i}|s_t).$$

   Observe next state s_{t+1}, opponent action a_t^{-i} and reward r_t^i, save the new experience in the reply buffer:

$$\mathcal{M} \leftarrow \mathcal{M} \cup \{(s_t, a_t^i, a_t^{-i}, s_{t+1}, r_t^i)\}.$$

   Update the prior from the replay buffer:

$$P(a_t^{-i}|s_t) = \frac{\sum_{m=1}^{|\mathcal{M}|} \mathbb{I}(s = s_t, a^{-i} = a_t^{-i})}{\sum_{m=1}^{|\mathcal{M}|} \mathbb{I}(s = s_t)} \forall s_t, a_t^{-i} \in \mathcal{M}.$$

   Sample a mini-batch from the replay buffer:

$$\{s_t^{(n)}, a_t^{i,(n)}, a_t^{-i,(n)}, s_{t+1}^{(n)}, r_t^{(n)}\}_{n=1}^N \sim \mathcal{M}.$$

  Update $Q_{\omega^i}(s, a^i, a^{-i})$:

  for each tuple $(s_t^{(n)}, a_t^{i,(n)}, a_t^{-i,(n)}, s_{t+1}^{(n)}, r_t^{(n)})$ **do**

   Sample $\{a^{-i,(n,k)}\}_{k=1}^K \sim \rho$, $\{a^{i,(n,k)}\}_{k=1}^K \sim \pi$.
   Compute empirical $\bar{V}^i(s_{t+1}^{(n)})$ as:

$$\bar{V}^i(s_{t+1}^{(n)}) = \log \left(\frac{1}{K} \sum_{k=1}^K \frac{\left(P^{\frac{1}{\alpha}}(a^{-i,(n,k)}|s_{t+1}^{(n)}) \exp(\frac{1}{\alpha} Q_{\bar{\omega}^i}(s_{t+1}^{(n)}, a^{i,(n,k)}, a^{-i,(n,k)})) \right)^{\alpha}}{\pi(a^{i,(n,k)}|s_{t+1}^{(n)}, a^{-i,(n,k)}) \rho(a^{-i,(n,k)}|s_{t+1}^{(n)})} \right).$$

   Set

$$y^{(n)} = \begin{cases} r_t^{(n)} & \text{for terminal } s_{t+1}^{(n)} \\ r_t^{(n)} + \gamma \bar{V}^i(s_{t+1}^{(n)}) & \text{for non-terminal } s_{t+1}^{(n)} \end{cases}$$

   Perform gradient descent step on $(y^{(n)} - Q_{\omega^i}(s_{t+1}^{(n)}, a^{i,(n)}, a^{-i,(n)}))^2$ with respect to parameters ω^i
   Every C gradient descent steps, reset target parameters:

$$\bar{\omega}^i \leftarrow \omega$$

  end for
end while
Compute converged π^i **and** ρ^i

Algorithm 2 Multi-agent Variational Actor Critic

Result: policy π_{θ^i}, opponent model ρ_{ϕ^i}
Initialization:
Initialize parameters θ^i, ϕ^i, ω^i, ψ^i for each agent i and the random process $\mathcal{N}$ for action exploration.
Assign target parameters of joint action Q-function: $\bar{\omega}^i \leftarrow \omega$.
Initialize learning rates $\lambda_V, \lambda_Q, \lambda_\pi, \lambda\phi, \alpha$, and set γ as the discount factor.
for Each episode $d = (1, \ldots, D)$ **do**
  Initialize random process $\mathcal{N}$ for action exploration.
  for each time step t **do**
    For the current state s_t, sample an action and opponent's action using:
    $\hat{a}_t^{-i} \leftarrow g_{\phi^{-i}}(\epsilon^{-i}; s_t)$, where $\epsilon_t^{-i} \sim \mathcal{N}$,
    $a_t^i \leftarrow f_{\theta^i}(\epsilon^i; s_t, \hat{a}_t^{-i})$, where $\epsilon_t^i \sim \mathcal{N}$.
    Observe next state s_{t+1}, opponent action a_t^{-i} and reward r_t^i, save the new experience in the replay buffer:

$$\mathcal{D}^i \leftarrow \mathcal{D}^i \cup \{(s_t, a_t^i, a_t^{-i}, \hat{a}_t^{-i}, s_{t+1}, r_t^i)\}.$$

    Update the prior from the replay buffer:

$$\psi^i = \arg\max \mathbb{E}_{\mathcal{D}^i}[-P(a^{-i}|s)\log P_{\psi^i}(a^{-i}|s)]$$

    Sample a mini-batch from the reply buffer:

$$\{s_t^{(n)}, a_t^{i,(n)}, a_t^{-i,(n)}, \hat{a}_t^{-i,(n)}, s_{t+1}^{(n)}, r_t^{(n)}\}_{n=1}^N \sim \mathcal{M}.$$

    For the state $s_{t+1}^{(n)}$, sample an action and opponent's action using:
    $\hat{a}_{t+1}^{-i,(n)} \leftarrow g_{\phi^{-i}}(\epsilon^{-i}; s_{t+1}^{(n)})$, where $\epsilon_{t+1}^{-i} \sim \mathcal{N}$,
    $a_{t+1}^{i,(n)} \leftarrow f_{\bar{\theta}^i}(\epsilon^i; s_{t+1}^{(n)}, \hat{a}_{t+1}^{-i,(n)})$, where $\epsilon_{t+1}^i \sim \mathcal{N}$.
    $\bar{V}^i(s_{t+1}^{(n)}) = Q_{\bar{\omega}}(s_{t+1}^{(n)}, a_{t+1}^{i,(n)}, \hat{a}_{t+1}^{-i,(n)}) - \alpha \log \pi_{\theta^i}(a_{t+1}^{i,(n)}|s_{t+1}^{(n)}, \hat{a}_{t+1}^{-i,(n)}) - \log \rho_{\phi^i}(\hat{a}_{t+1}^{-i,(n)}|s_{t+1}^{(n)}) + \log P_{\psi^i}(\hat{a}_{t+1}^{-i,(n)}|s_{t+1}^{(n)})$.
    Set

$$y^{(n)} = \begin{cases} r_t^{(n)} & \text{for terminal } s_{t+1}^{(n)} \\ r_t^{(n)} + \gamma \bar{V}^i(s_{t+1}^{(n)}) & \text{for non-terminal } s_{t+1}^{(n)} \end{cases}$$

$$\nabla_{\omega^i}\mathcal{J}_Q(\omega^i) = \nabla_{\omega^i}Q_{\omega^i}(s_t^{(n)}, a_t^{i,(n)}, a_t^{-i,(n)})(Q_{\omega^i}(s_t^{(n)}, a_t^{i,(n)}, a_t^{-i,(n)}) - y^{(n)})$$

$$\nabla_{\theta^i}\mathcal{J}_\pi(\theta^i) = \nabla_{\theta^i}\alpha \log \pi_{\theta^i}(a_t^{i,(n)}|s_t^{(n)}, \hat{a}_t^{-i,(n)})$$
$$+ (\nabla_{a_t^{i,(n)}}\alpha \log \pi_{\theta^i}(a_t^{i,(n)}|s_t^{(n)}, \hat{a}_t^{-i,(n)}) - \nabla_{a_t^{i,(n)}}Q_\omega(s_t^{(n)}, a_t^{i,(n)}, \hat{a}_t^{-i,(n)}))\nabla_\theta f_{\theta^i}(\epsilon_t^i; s_t^{(n)}, \hat{a}_t^{-i,(n)})$$

$$\nabla_{\phi^i}\mathcal{J}_\rho(\phi^i) = \nabla_{\phi^i}\log \rho_{\phi^i}(\hat{a}_t^{-i,(n)}|s_t^{(n)})$$
$$+ (\nabla_{\hat{a}_t^{-i,(n)}}\log \rho_{\phi^i}(\hat{a}_t^{-i,(n)}|s_t^{(n)}) - \nabla_{\hat{a}_t^{-i,(n)}}\log P(\hat{a}_t^{-i,(n)}|s_t^{(n)}) - \nabla_{\hat{a}_t^{-i,(n)}}Q_{\omega^i}(s_t^{(n)}, a_t^{i,(n)}, \hat{a}_t^{-i,(n)})$$
$$+ \nabla_{\hat{a}_t^{-i,(n)}}\alpha \log \pi_{\theta^i}(a^{i,(n)}|s_t^{(n)}, \hat{a}_t^{-i,(n)}))\nabla_{\phi^i}g_{\phi^i}(\epsilon_t^{-i}; s_t^{(n)})$$

    Update parameters:
    $\omega^i = \omega^i - \lambda_Q \nabla_{\omega^i}\mathcal{J}_Q(\omega^i)$
    $\theta^i = \theta^i - \lambda_\pi \nabla_{\theta^i}\mathcal{J}_\pi(\theta^i)$
    $\phi^i = \phi^i - \lambda_{\phi^i} \nabla_{\phi^i}\mathcal{J}_\rho(\phi^i)$
  end for
  Every C gradient descent steps, reset target parameters:

$$\overline{\omega^i} = \beta\omega^i + (1-\beta)\overline{\omega^i}$$

  .
end for

B Variational Lower Bounds in Multi-agent Reinforcement Learning

B.1 The Lower Bound of The Log Likelihood of Optimality

We can factorize $P(a_{1:T}^i, a_{1:T}^{-i}, s_{1:T}|o_{1:T}^{-i})$ as :

$$P(a_{1:T}^i, a_{1:T}^{-i}, s_{1:T}|o_{1:T}^{-i}) = P(s_1)\prod_t P(s_{t+1}|s_t, a_t)P(a_t^i|a_t^{-i}, s_t, o_t^{-i})P(a_t^{-i}|s_t, o_t^{-i}), \tag{23}$$

where $P(a_t^i|a_t^{-i}, s_t, o_t^{-i})$ is the conditional policy of agent i when other agents $-i$ achieve optimality. As agent i has no knowledge about rewards of other agents, we set $P(a_t^i|a_t^{-i}, s_t, o_t^{-i}) \propto 1$.

Analogously, we factorize $q(a_{1:T}^i, a_{1:T}^{-i}, s_{1:T}|o_{1:T}^i, o_{1:T}^{-i})$ as:

$$q(a_{1:T}^i, a_{1:T}^{-i}, s_{1:T}|o_{1:T}^i, o_{1:T}^{-i}) = P(s_1)\prod_t P(s_{t+1}|s_t, a_t)q(a_t^i|a_t^{-i}, s_t, o_t^i, o_t^{-i})q(a_t^{-i}|s_t, o_t^i, o_t^{-i}) \tag{24}$$

$$= P(s_1)\prod_t P(s_{t+1}|s_t, a_t)\pi(a_t^i|s_t, a_t^{-i})\rho(a_t^{-i}|s_t), \tag{25}$$

where $\pi(a_t^i|a_t^{-i}, s_t)$ is agent 1's conditional policy at optimum and $\rho(a_t^{-i}|s_t)$ is agent 1's model about opponents' optimal policies.

With the above factorization, we have:

$$\log P(o_{1:T}^i|o_{1:T}^{-i})$$

$$= \log \sum_{a_{1:T}^i, a_{1:T}^{-i}, s_{1:T}} P(o_{1:T}^i, a_{1:T}^i, a_{1:T}^{-i}, s_{1:T}|o_{1:T}^{-i}) \tag{26}$$

$$\geq \sum q(a_{1:T}^i, a_{1:T}^{-i}, s_{1:T}|o_{1:T}^i, o_{1:T}^{-i})\log\frac{P(o_{1:T}^i, a_{1:T}^i, a_{1:T}^{-i}, s_{1:T}|o_{1:T}^{-i})}{q(a_{1:T}^i, a_{1:T}^{-i}, s_{1:T}|o_{1:T}^i, o_{1:T}^{-i})} \tag{27}$$

$$= \mathbb{E}_{(a_{1:T}^i, a_{1:T}^{-i}, s_{1:T}\sim q)}[\sum_{t=1}^T \log P(o_t^i|o_t^{-i}, s_t, a_t^i, a_t^{-i}) + \cancel{\log P(s_1)} + \cancel{\sum_{t=1}^T \log P(s_{t+1}|s_t, a_t^i, a_t^{-i})} \tag{28}$$

$$\cancel{-\log P(s_1)} - \cancel{\sum_{t=1}^T \log P(s_{t+1}|s_t, a_t^i, a_t^{-i})} \tag{29}$$

$$-\sum_{t=1}^T \log\pi(a_t^i|s_t, a_t^{-i}) - \sum_{t=1}^T \log\frac{\rho(a_t^{-i}|s_t)}{P(a_t^{-i}|s_t, o_t^{-i})} + \sum_{t=1}^T \log P(a_t^i|s_t, a_t^{-i}, o_t^{-i})] \tag{30}$$

$$= \mathbb{E}_{(a_{1:T}^i, a_{1:T}^{-i}, s_{1:T}\sim q)}[\sum_{t=1}^T R^i(s_t, a_t^i, a_t^{-i}) - \log\pi(a_t^i|s_t, a_t^{-i}) - \log\frac{\rho(a_t^{-i}|s_t)}{P(a_t^{-i}|s_t, o_t^{-i})} + \log 1] \tag{31}$$

$$= \sum_t \mathbb{E}_{(s_t, a_t^i, a_t^{-i})\sim q}[R^i(s_t, a_t^i, a_t^{-i}) + H(\pi(a_t^i|s_t, a_t^{-i})) - D_{KL}(\rho(a_t^{-i}|s_t)||P(a_t^{-i}|s_t, o_t^{-i}))]. \tag{32}$$

C Multi-Agent Soft-Q Learning

C.1 Soft Q-Function

We define the soft state-action value function $Q_{soft}^{\pi,\rho}(s, a, a^{-i})$ of agent i in a stochastic game as:

$$Q_{soft}^{\pi,\rho}(s_t, a_t^i, a_t^{-i})$$

$$= r_t + \mathbb{E}_{(s_{t+l}, a_{t+l}^i, a_{t+l}^{-i}, \ldots)\sim q}[\sum_{l=1}^\infty \gamma^l(r_{t+l} + \alpha H(\pi(a_{t+l}^i|a_{t+l}^{-i}, s_{t+l})) - D_{KL}(\rho(a_{t+l}^{-i}|s_{t+l})||P(a_{t+l}^{-i}|s_{t+l})))] \tag{33}$$

$$= \mathbb{E}_{(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i})}[r_t + \gamma(\alpha H(\pi(a_{t+1}^i|s_{t+1}, a_{t+1}^{-i})) - D_{KL}(\rho(a^{-i}|s_{t+1})||P(a^{-i}|s_{t+1})) + Q_{soft}^{\pi,\rho}(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i}))] \tag{34}$$

$$= \mathbb{E}_{(s_{t+1}, a_{t+1}^{-i})}[r_t + \gamma(\alpha H(\pi(\cdot|s_{t+1}, a_{t+1}^{-i})) - D_{KL}(\rho(a^{-i}|s_{t+1})||P(a^{-i}|s_{t+1})) + \mathbb{E}_{a_{t+1}^i\sim\pi}[Q_{soft}^{\pi,\rho}(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i})])] \tag{35}$$

$$= \mathbb{E}_{(s_{t+1})}[r_t + \gamma(\mathbb{E}_{a_{t+1}^{-i}\sim\rho, a_{t+1}^i\sim\pi}[\alpha H(\pi(a_{t+1}^i|s_{t+1}, a_{t+1}^{-i}))] - D_{KL}(\rho(\cdot|s_{t+1})||P(\cdot|s_{t+1}))]$$

$$+ \mathbb{E}_{a_{t+1}^{-i}\sim\rho, a_{t+1}^i\sim\pi}[Q_{soft}^{\pi,\rho}(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i})])], \tag{36}$$

Then we can easily see that the objective in Eq. 7 can be rewritten as:

$$\mathcal{J}(\pi, \phi) = \sum_t \mathbb{E}_{(s_t, a_t^i, a_t^{-i}) \sim (p_s, \pi, \rho)}[Q_{soft}^{\pi,\rho}(s_t, a_t^i, a_t^{-i}) + \alpha H(\pi(a_t^i|s_t, a_t^{-i})) - D_{\mathrm{KL}}(\rho(a_t^{-i}|s_t)||P(a_t^{-i}|s_t))], \qquad (37)$$

by setting $\alpha = 1$.

C.2 Policy Improvement and Opponent Model Improvement

Theorem 4. *(Policy improvement theorem) Given a conditional policy π and opponent model ρ, define a new conditional policy $\tilde{\pi}$ as*

$$\tilde{\pi}(\cdot|s, a^{-i}) \propto \exp(\frac{1}{\alpha}Q_{soft}^{\pi,\rho}(s, \cdot, a^{-i})), \forall s, a^{-i}. \qquad (38)$$

Assume that throughout our computation, Q is bounded and $\sum_{a^i} Q(s, a^i, a^{-i})$ is bounded for any s and a^{-i} (for both π and $\tilde{\pi}$). Then $Q_{soft}^{\tilde{\pi},\rho}(s, a^i, a^{-i}) \geq Q_{soft}^{\pi,\rho}(s, a^i, a^{-i}) \forall s, a$.

Theorem 5. *(Opponent model improvement theorem) Given a conditional policy π and opponent model ρ, define a new opponent model $\tilde{\rho}$ as*

$$\tilde{\rho}(\cdot|s) \propto \exp(\sum_{a^i} Q_{soft}^{\pi,\rho}(s, a^i, \cdot)\pi(a^i|\cdot, s) + \alpha H(\pi(s)) + \log P(\cdot|s)), \forall s, a^i. \qquad (39)$$

Assume that throughout our computation, Q is bounded and $\sum_{a^{-i}} \exp(\sum_{a^i} Q(s, a^i, a^{-i})\pi(a^i|s, a^{-i}))$ is bounded for any s and a^i (for both ρ and $\tilde{\rho}$). Then $Q_{soft}^{\pi,\tilde{\rho}}(s, a^i, a^{-i}) \geq Q_{soft}^{\pi,\rho}(s, a^i, a^{-i}) \forall s, a$.

The proof of Theorem 4 and 5 is based on two observations that:

$$\alpha H(\pi(\cdot|s, a^{-i})) + \mathbb{E}_{a^i \sim \pi}[Q_{soft}^{\pi,\rho}(s, a^i, a^{-i})] \leq \alpha H(\tilde{\pi}(\cdot|s, a^{-i})) + \mathbb{E}_{a^i \sim \tilde{\pi}}[Q_{soft}^{\pi,\rho}(s, a^i, a^{-i})], \qquad (40)$$

and

$$\mathbb{E}_{a_{t+1}^{-i} \sim \rho, a_{t+1}^i \sim \pi}[\alpha H(\pi(a_{t+1}^i|s_{t+1}, a_{t+1}^{-i}))] - D_{\mathrm{KL}}(\rho(\cdot|s_{t+1})||P(\cdot|s_{t+1}))] + \mathbb{E}_{a_{t+1}^{-i} \sim \rho, a_{t+1}^i \sim \pi}[Q_{soft}^{\pi,\rho}(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i})] \quad (41)$$

$$\leq \mathbb{E}_{a_{t+1}^{-i} \sim \tilde{\rho}, a_{t+1}^i \sim \pi}[\alpha H(\pi(a_{t+1}^i|s_{t+1}, a_{t+1}^{-i}))] - D_{\mathrm{KL}}(\tilde{\rho}(a_{t+1}^{-i}|s_{t+1})||P(\cdot|s_{t+1})) + \mathbb{E}_{a_{t+1}^{-i} \sim \rho, a_{t+1}^i \sim \pi}[Q_{soft}^{\pi,\rho}(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i})].$$
$$(42)$$

First, we notice that

$$\alpha H(\pi(\cdot|s, a^{-i})) + \mathbb{E}_{a^i \sim \pi}[Q_{soft}^{\pi,\rho}(s, a^i, a^{-i})] = -\alpha D_{\mathrm{KL}}(\pi(\cdot|s, a^{-i})||\tilde{\pi}(\cdot|s, a^{-i})) + \alpha \log \sum_{a^i} \exp(\frac{1}{\alpha}Q_{soft}^{\pi,\rho}(s, a^i, a^{-i})). \quad (43)$$

Therefore, the LHS is only maximized if the KL-Divergence on the RHS is minimized. This KL-Divergence is minimized only when $\pi = \tilde{\pi}$, which proves the Equation 40.

Similarly, we can have

$$\mathbb{E}_{a^{-i} \sim \rho, a^i \sim \pi}[\alpha H(\pi(a^i|s, a^{-i}))] - D_{\mathrm{KL}}(\rho(\cdot|s)||P(\cdot|s))]) + \mathbb{E}_{a^{-i} \sim \rho, a^i \sim \pi}[Q_{soft}^{\pi,\rho}(s, a^i, a^{-i})]$$

$$= -D_{\mathrm{KL}}(\rho(\cdot|s)||\tilde{\rho}(\cdot|s)) + \log \sum_{a^{-i}} \exp(\sum_{a^i} Q^{\pi,\rho}(s, a^i, a^{-i})\pi(a^i|s, a^{-i}) + \alpha H(\pi(\cdot|s, a^i)) + \log P(a^{-i}|s)), \qquad (44)$$

which proves the Equation 42.

With the above observations, the proof of Theorem 4 and 5 is completed by as follows:

$$Q_{soft}^{\pi,\rho}(s_t, a_t^i, a_t^{-i})$$

$$= \mathbb{E}_{(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i})}[r_t + \gamma(\alpha H(\pi(a_{t+1}^i|s_{t+1}, a_{t+1}^{-i})) - D_{\mathrm{KL}}(\rho(a_{t+1}^{-i}|s_{t+1})||P(a_{t+1}^{-i}|s_{t+1})) + Q_{soft}^{\pi,\rho}(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i}))] \tag{45}$$

$$= \mathbb{E}_{(s_{t+1}, a_{t+1}^{-i})}[r_t + \gamma(\alpha H(\pi(\cdot|s_{t+1}, a_{t+1}^{-i})) - D_{\mathrm{KL}}(\rho(a_{t+1}^{-i}|s_{t+1})||P(a_{t+1}^{-i}|s_{t+1})) + \mathbb{E}_{a_{t+1}^i \sim \pi}[Q_{soft}^{\pi,\rho}(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i})])] \tag{46}$$

$$\leq \mathbb{E}_{(s_{t+1}, a_{t+1}^{-i})}[r_t + \gamma(\alpha H(\tilde{\pi}(\cdot|s_{t+1}, a_{t+1}^{-i})) - D_{\mathrm{KL}}(\rho(a_{t+1}^{-i}|s_{t+1})||P(a_{t+1}^{-i}|s_{t+1})) + \mathbb{E}_{a_{t+1}^i \sim \tilde{\pi}}[Q_{soft}^{\pi,\rho}(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i})])] \tag{47}$$

$$= \mathbb{E}_{(s_{t+1})}[r_t + \gamma(\mathbb{E}_{a_{t+1}^{-i} \sim \rho, a_{t+1}^i \sim \pi}[\alpha H(\tilde{\pi}(a_{t+1}^i|s_{t+1}, a_{t+1}^{-i}))] - D_{\mathrm{KL}}(\rho(\cdot|s_{t+1})||P(\cdot|s_{t+1}))$$
$$+ \mathbb{E}_{a_{t+1}^{-i} \sim \rho, a_{t+1}^i \sim \pi}[Q_{soft}^{\pi,\rho}(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i})])] \tag{48}$$

$$\leq \mathbb{E}_{(s_{t+1})}[r_t + \gamma(\mathbb{E}_{a_{t+1}^{-i} \sim \tilde{\rho}, a_{t+1}^i \sim \pi}[\alpha H(\tilde{\pi}(a_{t+1}^i|s_{t+1}, a_{t+1}^{-i}))] - D_{\mathrm{KL}}(\tilde{\rho}(\cdot|s_{t+1})||P(\cdot|s_{t+1}))$$
$$+ \mathbb{E}_{a_{t+1}^{-i} \sim \tilde{\rho}, a_{t+1}^i \sim \pi}[Q_{soft}^{\pi,\rho}(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i})])] \tag{49}$$

$$= \mathbb{E}_{(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i}) \sim \tilde{q}}[r_t + \gamma(\alpha H(\tilde{\pi}(a_{t+1}^i|s_{t+1}, a_{t+1}^{-i})) - D_{\mathrm{KL}}(\tilde{\rho}(a^{-i}|s_{t+1})||P(a^{-i}|s_{t+1})) + r_{t+1})$$
$$+ \gamma^2 \mathbb{E}_{(s_{t+2}, a_{t+2}^{-i})}[\alpha H(\pi(\cdot|s_{t+2}, a_{t+2}^{-i})) - D_{\mathrm{KL}}(\rho(a_{t+2}^{-i}|s_{t+2})||P(a_{t+2}^{-i}|s_{t+2})) + \mathbb{E}_{a_{t+2}^i \sim \pi}[Q_{soft}^{\pi,\rho}(s_{t+2}, a_{t+2}^i, a_{t+2}^{-i})]]]] \tag{50}$$

$$\leq \mathbb{E}_{(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i})}[r_t + \gamma(\alpha H(\tilde{\pi}(a_{t+1}^i|s_{t+1}, a_{t+1}^{-i})) - D_{\mathrm{KL}}(\tilde{\rho}(a^{-i}|s_{t+1})||P(a^{-i}|s_{t+1})) + r_{t+1})$$
$$+ \gamma^2 \mathbb{E}_{(s_{t+2}, a_{t+2}^{-i})}[\alpha H(\pi(\cdot|s_{t+2}, a_{t+2}^{-i})) - D_{\mathrm{KL}}(\rho(a_{t+2}^{-i}|s_{t+2})||P(a_{t+2}^{-i}|s_{t+2})) + \mathbb{E}_{a_{t+2}^i \sim \tilde{\pi}}[Q_{soft}^{\pi,\rho}(s_{t+2}, a_{t+2}^i, a_{t+2}^{-i})]]]] \tag{51}$$

$$\vdots$$

$$\leq r_t + \mathbb{E}_{(s_{t+l}, a_{t+l}^i, a_{t+l}^{-i}, \dots) \sim \tilde{q}}[\sum_{l=1}^{\infty} \gamma^l(r_{t+l} + \alpha H(\tilde{\pi}(a_{t+l}^i|a_{t+l}^{-i}, s_{t+l})) - D_{\mathrm{KL}}(\tilde{\rho}(a_{t+l}^{-i}|s_{t+l})||P(a_{t+l}^{-i}|s_{t+l}))] \tag{52}$$

$$= Q_{soft}^{\tilde{\pi},\tilde{\rho}}(s_t, a_t^i, a_t^{-i}). \tag{53}$$

With Theorem 4 and 5 and the above inequalities, we can see that, if we start from an arbitrary conditional policy π_0 and an arbitrary opponent model ρ_0 and we iterate between policy improvement as

$$\pi_{i+1}(\cdot|s, a^{-i}) \propto \exp(\frac{1}{\alpha}Q_{soft}^{\pi_t,\rho_t}(s, \cdot, a^{-i})), \tag{54}$$

and opponent model improvement as

$$\rho_{t+1}(\cdot|s) \propto \exp(\sum_{a^i} Q_{soft}^{\pi_{t+1},\rho_t}(s, a^i, \cdot)\pi_{t+1}(a^i|\cdot, s) + \alpha H(\pi_{t+1}(s)) + \log P(\cdot|s)), \tag{55}$$

then $Q_{soft}^{\pi_t,\rho_t}(s, a^i, a^{-i})$ can be shown to increase monotonically. Similar to [Haarnoja *et al.*, 2017], we can show that with certain regularity conditions satisfied, any non optimal policy and opponent model can be improved this way and Theorem 1 is proved.

C.3 Soft Bellman Equation

As we show in Appendix C.2, when the training converges, we have:

$$\pi^*(a^i|s, a^{-i}) = \frac{\frac{1}{\alpha}\exp(Q^*(s, a^i, a^{-i}))}{\sum_{a^i}\exp(\frac{1}{\alpha}Q^*(s, a^i, a^{-i}))}, \tag{56}$$

and

$$\rho^*(a^{-i}|s) = \frac{\exp(\sum_{a^i}Q^*(s, a^i, a^{-i})\pi^*(a^i|s, a^{-i}) + \alpha H(\pi^*(a^i|s, a^{-i})) + \log P(a^{-i}|s))}{\sum_{a^{-i}}\exp(\sum_{a^i}Q^*(s, a^i, a^{-i})\pi^*(a^i|s, a^{-i}) + \alpha H(\pi^*(a^i|s, a^{-i})) + \log P(a^{-i}|s))}$$
$$= \frac{P(a^{-i}|s)\left(\sum_{a^i}\exp(Q_{soft}^*(s, a^i, a^{-i}))\right)^{\alpha}}{\exp(V^*(s))}, \tag{57}$$

where the equality in Eq. 57 comes from substituting π^* with Eq. 56 and we define the soft sate value function $V_{soft}^{\pi,\rho}(s)$ of agent i as:

$$V_{soft}^{\pi,\rho}(s_t) = \log \sum_{a_t^{-i}} P(a_t^{-i}|s_t) \left(\sum_{a_t^i} \exp \left(\frac{1}{\alpha} Q_{soft}^{\pi,\rho}(s_t, a_t^i, a_t^{-i}) \right) \right)^{\alpha}. \tag{58}$$

Then we can show that

$$Q_{soft}^{\pi^*,\rho^*}(s, a^i, a^{-i})$$
$$= r_t + \gamma \mathbb{E}_{s' \sim p_s}[(\mathbb{E}_{a_{t+1}^{-i} \sim \rho, a_{t+1}^i \sim \pi}[\alpha H(\pi(a_{t+1}^i|s_{t+1}, a_{t+1}^{-i}))] - D_{KL}(\rho(\cdot|s_{t+1})||P(\cdot|s_{t+1}))]$$
$$+ \mathbb{E}_{a_{t+1}^{-i} \sim \rho, a_{t+1}^i \sim \pi}[Q_{soft}^{\pi,\rho}(s_{t+1}, a_{t+1}^i, a_{t+1}^{-i})])]$$
$$= r_t + \gamma \mathbb{E}_{s' \sim p_s}[V^*(s')]. \tag{59}$$

We define the soft value iteration operator $\mathcal{T}$ as:

$$\mathcal{T}Q(s, a^i, a^{-i}) = R(s, a^i, a^{-i}) + \gamma \mathbb{E}_{s' \sim p_s} \left[\log \sum_{a^{-i'}} P(a^{-i'}|s') \left(\sum_{a^{i'}} \exp \left(\frac{1}{\alpha} Q(s', a^{i'}, a^{-i'}) \right) \right)^{\alpha} \right]. \tag{60}$$

In a symmetric fully cooperative game with only one global optimum, we can show as done in [Wen *et al.*, 2019], the operator defined above is a contraction mapping. We define a norm on Q-values $\|Q_1^i - Q_2^i\| \triangleq \max_{s,a^i,a^{-i}} |Q_1^i(s, a^i, a^{-i}) - Q_2^i(s, a^i, a^{-i})|$. Let $\varepsilon = \|Q_1^i - Q_2^i\|$, then we have:

$$\log \sum_{a^{-i'}} P(a^{-i'}|s') \left(\sum_{a^{i'}} \exp \left(\frac{1}{\alpha} Q_1(s', a^{i'}, a^{-i'}) \right) \right)^{\alpha} \leq \log \sum_{a^{-i'}} P(a^{-i'}|s') \left(\sum_{a^{i'}} \exp \left(\frac{1}{\alpha} Q_2(s', a^{i'}, a^{-i'}) + \varepsilon \right) \right)^{\alpha}$$

$$= \log \sum_{a^{-i'}} P(a^{-i'}|s') \left(\sum_{a^{i'}} \exp \left(\frac{1}{\alpha} Q_2(s', a^{i'}, a^{-i'}) \right) \exp(\varepsilon) \right)^{\alpha}$$

$$= \log \sum_{a^{-i'}} P(a^{-i'}|s') \exp(\varepsilon)^{\alpha} \left(\sum_{a^{i'}} \exp \left(\frac{1}{\alpha} Q_2(s', a^{i'}, a^{-i'}) \right) \right)^{\alpha}$$

$$= \alpha\varepsilon + \log \sum_{a^{-i'}} P(a^{-i'}|s') \left(\sum_{a^{i'}} \exp \left(\frac{1}{\alpha} Q_2(s', a^{i'}, a^{-i'}) \right) \right)^{\alpha}. \tag{61}$$

Similarly, $\log \sum_{a^{-i'}} P(a^{-i'}|s') \left(\sum_{a^{i'}} \exp \left(\frac{1}{\alpha} Q_1(s', a^{i'}, a^{-i'}) \right) \right)^{\alpha} \geq -\alpha\varepsilon + \log \sum_{a^{-i'}} P(a^{-i'}|s') \left(\sum_{a^{i'}} \exp \left(\frac{1}{\alpha} Q_2(s', a^{i'}, a^{-i'}) \right) \right)^{\alpha}$. Therefore $\|\mathcal{T}Q_1^i - \mathcal{T}Q_2^i\| \leq \gamma\varepsilon = \gamma \|Q_1^i - Q_2^i\|$, where $\alpha = 1$.