

Variational Autoencoders for Opponent Modeling in Multi-Agent Systems

Georgios Papoudakis

School of Informatics
The University of Edinburgh
Edinburgh Centre for Robotics
g.papoudakis@ed.ac.uk

Stefano V. Albrecht

School of Informatics
The University of Edinburgh
Edinburgh Centre for Robotics
s.albrecht@ed.ac.uk

Abstract

Multi-agent systems exhibit complex behaviors that emanate from the interactions of multiple agents in a shared environment. In this work, we are interested in controlling one agent in a multi-agent system and successfully learn to interact with the other agents that have fixed policies. Modeling the behavior of other agents (opponents) is essential in understanding the interactions of the agents in the system. By taking advantage of recent advances in unsupervised learning, we propose modeling opponents using variational autoencoders. Additionally, many existing methods in the literature assume that the opponent models have access to opponent’s observations and actions during both training and execution. To eliminate this assumption, we propose a modification that attempts to identify the underlying opponent model using only local information of our agent, such as its observations, actions, and rewards. The experiments indicate that our opponent modeling methods achieve equal or greater episodic returns in reinforcement learning tasks against another modeling method.

1 Introduction

In recent years, several promising works [Mnih et al., 2015, Schulman et al., 2015a, Mnih et al., 2016] have arisen in deep reinforcement learning (RL), leading to fruitful results in single-agent scenarios. In this work, we are interested in using single-agent RL in multi-agent systems, where we control one agent and the other agents (opponents) in the environment have fixed policies. The agent should be able to successfully interact with a diverse set of opponents as well as generalize to new unseen opponents. One effective way to address this problem is opponent modeling. The opponent models output specific characteristics of the opponents based on their trajectories. By successfully modeling the opponents, the agent can reason about opponents’ behaviors and goals and adjust its policy to achieve optimal outcomes. There is a rich literature of modeling opponents in multi-agent systems [Albrecht and Stone, 2018].

Several recent works have proposed learning opponent models using deep learning architectures [He et al., 2016, Raileanu et al., 2018, Grover et al., 2018a, Rabinowitz et al.,

2018]. In this work, we focus on learning opponent models using Variational Autoencoders (VAEs) [Kingma and Welling, 2014]. VAE are generative models that are commonly used for learning representations of data, and various works use them in RL for learning representations of the environment [Igl et al., 2018, Ha and Schmidhuber, 2018, Zintgraf et al., 2019]. We first propose a VAE for learning opponent representations in multi-agent systems based on the opponent trajectories.

A shortcoming of this approach and most opponent modeling methods, as will be discussed in Section 2, is that they require access to opponent’s information, such as observations and actions, during training as well as execution. This assumption is too limiting in the majority of scenarios. For example, consider Poker, in which agents do not have access to the opponent’s observations. Nevertheless, during Poker, humans can reason about the opponent’s behaviors and goals using only their local observations. For example, an increase in the table’s pot could mean that the opponent either holds strong cards or is bluffing. Based on the idea that an agent can reason about an opponent models using its own observations, actions, and rewards in a recurrent fashion, we propose a second VAE-based architecture. The encoder of the VAE learns to represent opponent models conditioned on only local information removing the requirement to access the opponents’ information during execution.

We evaluate our proposed methodology using a toy example and the commonly used Multi-agent Particle Environment [Mordatch and Abbeel, 2017]. We evaluate the episodic returns that RL algorithms can achieve. The experiments indicate that opponent modeling without opponents’ information can achieve comparable or higher average returns, using RL, than models that access the opponent’s information.

2 Related Work

Learning Opponent Models. In this work, we are interested in opponent modeling methods that use neural networks to learn representations of the opponents. He et al. [2016] proposed an opponent modeling method that learns a modeling network to reconstruct the opponent’s actions given the opponent observations. Raileanu et al. [2018] developed an

algorithm for learning to infer opponents’ goals using the policy of the controlled agent. Hernandez-Leal et al. [2019] used auxiliary tasks for modeling opponents in multi-agent reinforcement learning. Grover et al. [2018a] proposed an encoder-decoder method for modeling the opponent’s policy. The encoder learns a point-based representation of different opponents’ trajectories, and the decoder learns to reconstruct the opponent’s policy given samples from the embedding space. Additionally, Grover et al. [2018a] introduce an objective to separate embeddings of different agents into different clusters.

$$d(z_+, z_-, z) = \frac{1}{(1 + e^{|z-z_-|_2 - |z-z_+|_2})^2} \quad (1)$$

where z_+ and z_- are embeddings of the same agent from two different episodes and embedding z_- is generated from the episode of a different agent. Zheng et al. [2018] use an opponent modeling method for better opponent identification and multi-agent reinforcement learning. Rabinowitz et al. [2018] proposed the Theory of mind Network (TomNet), which learns embedding-based representations of opponents for meta-learning. Tacchetti et al. [2018] proposed *Relation Forward Model* to model opponents using graph neural networks. A common assumption among these methods, which our work aims to eliminate, is that access to opponents trajectories is available during execution.

Representation Learning in Reinforcement Learning.

Another related topic that has received significant attention is representation learning in RL. Using unsupervised learning techniques to learn low-dimensional representations of the MDP has led to significant improvement in RL. Ha and Schmidhuber [2018] proposed a VAE-based and a forward model to learn state representations of the environment. Hausman et al. [2018] learned tasks embeddings and interpolated them to solve harder tasks. Igl et al. [2018] used a VAE for learning representation in partially-observable environments. Gupta et al. [2018] proposed a model which learns Gaussian embeddings to represent different tasks during meta-training and manages to quickly adapt to new task during meta-testing. The work of Zintgraf et al. [2019] is closely related, where Zintgraf et al. proposed a recurrent VAE model, which receives as input the observation, action, reward of the agent, and learns a variational distribution of tasks. Rakelly et al. [2019] used representations from an encoder for off-policy meta-RL. Note that all these works have been applied for learning representations of tasks or properties of the environments. In contrast, our approach is focused on learning representations of opponents.

3 Background

3.1 Reinforcement Learning

Markov Decision Processes (MDPs) are commonly used to model decision making problems. An MDP consists of the set of states \mathbb{S} , the set of actions \mathbb{A} , the transition function, $P(s'|s, a)$, which is the probability of the next state, s' , given the current state, s , and the action, a , and the reward function, $r(s', a, s)$, that returns a scalar value conditioned on two consecutive states and the intermediate action. A policy function is used to choose an action given a state, which

can be stochastic $\mathbf{a} \sim \pi(\mathbf{a}|\mathbf{s})$ or deterministic $\mathbf{a} = \mu(\mathbf{s})$. Given a policy π , the state value function is defined as $V(\mathbf{s}_t) = \mathbb{E}_\pi[\sum_{i=t}^H \gamma^{i-t} r_t | s = \mathbf{s}_t]$ and the state-action value (Q-value) $Q(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{E}_\pi[\sum_{i=t}^H \gamma^{i-t} r_t | s = \mathbf{s}_t, a = \mathbf{a}_t]$, where $0 \leq \gamma \leq 1$ is the discount factor and H is the finite horizon of the episode. The goal of RL is to compute the policy that maximizes state value function V , when the transition and the reward functions are unknown.

There is a large number of RL algorithms; however, in this work, we focus on two actor-critic algorithms; the synchronous Advantage Actor-Critic (A2C) [Mnih et al., 2016, Dhariwal et al., 2017] and the Deep Deterministic Policy Gradient (DDPG) [Silver et al., 2014, Lillicrap et al., 2015]. DDPG is an off-policy algorithm, using an experience replay for breaking the correlation between consecutive samples and target networks for stabilizing the training [Mnih et al., 2015]. Given an actor network with parameters θ and a critic network with parameter ϕ , the gradient updates are performed using the following update rules.

$$\begin{aligned} & \min_{\phi} \frac{1}{2} \mathbb{E}_B[(r + \gamma \cdot Q_{\text{target}, \phi'}(s', \mu_{\text{target}, \theta'}(s')) - Q_{\phi}(s, \mathbf{a}))^2] \\ & \min_{\theta} -\mathbb{E}_B[Q_{\phi}(s, \mu_{\theta}(s))] \end{aligned}$$

On the other hand, A2C is an on-policy actor-critic algorithm, using parallel environments to break the correlation between consecutive samples. The actor-critic parameters are optimized by:

$$\min_{\theta, \phi} \mathbb{E}_B[-\hat{A} \log \pi_{\theta}(\mathbf{a}|\mathbf{s}) + \frac{1}{2}(r + \gamma V_{\phi}(s') - V_{\phi}(s))^2]$$

where the advantage term, \hat{A} , can be computed using the Generalized Advantage Estimation (GAE) [Schulman et al., 2015b].

3.2 Variational Autoencoders

Consider samples from a dataset $\mathbf{x} \in \mathbb{X}$ which are generated from some hidden (latent) random variable \mathbf{z} based on a generative distribution $p_{\mathbf{u}}(\mathbf{x}|\mathbf{z})$ with unknown parameter \mathbf{u} and a prior distribution on the latent variables, which we assume is a Gaussian with $\mathbf{0}$ mean and unit variance $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$. We are interested in approximating the true posterior $p(\mathbf{z}|\mathbf{x})$ with a variational parametric distribution $q_{\mathbf{w}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{w})$. Kingma and Welling [2014] proposed the Variational Autoencoders (VAE) to learn this distribution. Starting from the Kullback-Leibler (KL) divergence from the approximate to the true posterior $D_{\text{KL}}(q_{\mathbf{w}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x}))$, the lower bound on the evidence $\log p(\mathbf{x})$ is derived as:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z} \sim q_{\mathbf{w}}(\mathbf{z}|\mathbf{x})}[\log p_{\mathbf{u}}(\mathbf{x}|\mathbf{z})] - D_{\text{KL}}(q_{\mathbf{w}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

The architecture consists of an encoder which receives a sample \mathbf{x} and generates the Gaussian variational distribution $p(\mathbf{z}|\mathbf{x}; \mathbf{w})$. The decoder receives a sample from the Gaussian variational distribution and reconstructs the initial input \mathbf{x} . The architecture is trained using the reparameterization trick Kingma and Welling [2014]. Higgins et al. [2017] proposed β -VAE, where a parameter $\beta \geq 0$ is used to control

the trade-off between the reconstruction loss and the KL-divergence.

$$L(\mathbf{x}; \mathbf{w}, \mathbf{v}) = \mathbb{E}_{z \sim q_w(z|\mathbf{x})} [\log p_u(\mathbf{x}|z)] - \beta D_{\text{KL}}(q_w(z|\mathbf{x}) \| p(z))$$

4 Approach

4.1 Problem Formulation

We consider a modified Markov Game [Littman, 1994], which consists of N agents $\mathbb{I} = \{1, 2, \dots, N\}$, a set of states \mathbb{S} , the joint action space $\mathbb{A} = \mathbb{A}_1 \times \mathbb{A}_{-1}$, the transition function $P : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \rightarrow \mathbb{R}$ and the reward function $r : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \rightarrow \mathbb{R}^N$. We consider partially observable settings, where each agent i has access only to its local observation \mathbf{o}_i and reward r_i . Additionally, two sets of pretrained opponents are provided, $\mathbb{T} = \{\mathbb{I}_{-1,m}\}_{m=1}^{m=M_T}$ and $\mathbb{G} = \{\mathbb{I}_{-1,m}\}_{m=1}^{m=M_G}$, which are responsible for providing the joint action \mathbb{A}_{-1} . Note that by opponent we refer to $\mathbb{I}_{-1,m}$, which consists of one or more agents, independently from the type of the interactions. Additionally, we assume that the sets \mathbb{T} and \mathbb{G} are disjoint. At the beginning of each episode, we sample a pretrained opponent from the set \mathbb{T} during training or from \mathbb{G} during testing. Our goal is to train agent 1 using RL, to maximize the average return against opponents from the training set, \mathbb{T} , and generalize to opponents sampled from the test \mathbb{G} .

$$\arg \max_{\theta} \mathbb{E}_{\pi} [\mathbb{E}_{\mathbb{T}} [\sum_t \gamma^t r_t]] \quad (2)$$

Note, that when we refer to agent 1 we drop the subscript.

4.2 Variational Autoencoder with Access to Opponent's Information

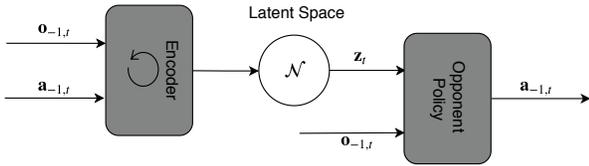


Figure 1: Diagram of the proposed VAE architecture

We assume a number of K provided episode trajectories for each pretrained opponent $j \in \mathbb{T}$, $E^{(j)} = \{\tau_{-1}^{(j,k)}\}_{k=0}^{k=K-1}$, where $\tau_{-1}^{(j,k)} = \{\mathbf{o}_{-1,t}, \mathbf{a}_{-1,t}\}_{t=0}^{t=H}$, and $\mathbf{o}_{-1,t}, \mathbf{a}_{-1,t}$ are the observations and actions of the opponent at the time step t in the trajectory. These trajectories are generated from the opponents in set \mathbb{T} , which are represented in the latent space from the variable z and for which we assume there exists an unknown model $p_u(\tau_{-1}|z)$. Our goal is to approximate the unknown posterior, $p(z|\tau_{-1})$, using a variational Gaussian distribution $\mathcal{N}(\mu, \Sigma; \mathbf{w})$ with parameters \mathbf{w} . We consider using a β -VAE for the sequential task:

$$L(\tau_{-1}; \mathbf{w}, \mathbf{u}) = \mathbb{E}_{z \sim q_w(z|\tau_{-1})} [\log p_u(\tau_{-1}|z)] - \beta D_{\text{KL}}(q_w(z|\tau_{-1}) \| p(z))$$

We can further subtract the discrimination objective (equation 1) that was proposed by Grover et al. [2018a]. Since the discrimination objective is always non-negative, we derive and optimize a lower bound as:

$$L(\tau_{-1}; \mathbf{w}, \mathbf{u}) \geq \mathbb{E}_{z \sim q_w(z|\tau_{-1})} [\log p_u(\tau_{-1}|z)] - \beta D_{\text{KL}}(q_w(z|\tau_{-1}) \| p(z)) - \lambda d(\mathbb{E}(z_+), \mathbb{E}(z_-), \mathbb{E}(z))$$

The discrimination objective receives as input the mean of the variational Gaussian distribution, produced by three different trajectories. Despite the less tight lower bound, the discrimination objective will separate the opponents in the embedding space, which could potentially lead to higher episodic returns. At each time step t , the recurrent encoder network generates a latent sample z_t , which is conditioned on the opponent's trajectory $\tau_{-1,1:t}$, until time step t . The KL divergence can be written as:

$$D_{\text{KL}}(q_w(z|\tau_{-1}) \| p(z)) = \sum_{t=1}^H D_{\text{KL}}(q_w(z_t|\tau_{-1,1:t}) \| p(z_t))$$

The lower bound consist of the reconstruction loss of the trajectory which involves the observation and actions of the opponent. The opponent's action, at each time step depends on its observation and the opponent's policy, which is represented by the latent variable z . We use a decoder that consists of fully-connected layers, however, a recurrent network can be used, if we instead assume that the opponent decides its actions based on the history of its observations. Additionally, the observation at each time step depends only on the dynamics of the environment and the actions of the agents and not on the identity of the opponent. Therefore, the reconstruction loss factorizes as:

$$\begin{aligned} \log p_u(\tau_{-1}|z) &= \sum_{t=1}^H \log p_u(\mathbf{a}_{-1,t} | \mathbf{o}_{-1,t}, z_t) \\ & p_u(\mathbf{o}_{-1,t} | \mathbf{s}_{t-1}, \mathbf{a}_{t-1}, \mathbf{a}_{-1,t-1}) \\ & \propto \sum_{t=1}^H \log p_u(\mathbf{a}_{-1,t} | \mathbf{o}_{-1,t}, z_t) \end{aligned} \quad (3)$$

From the equation above, we observe that the loss is the reconstruction of the opponent's policy given the current observation and a sample from the latent variable. Figure 1 illustrates the diagram of the VAE.

4.3 Variational Autoencoder without Access to Opponent's Information

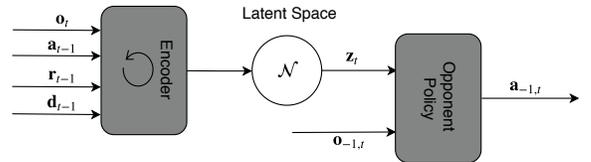


Figure 3: Diagram of the proposed VAE architecture using local information

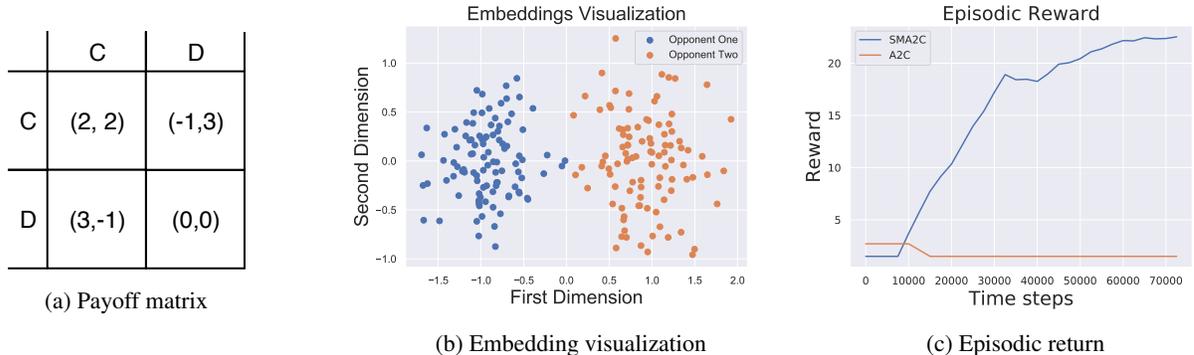


Figure 2: Payoff matrix, embedding visualization and episodic return during training in the toy example.

In Sections 1 and 2, it was noted that most agent modeling methods assume access to opponent’s observations and actions is available both during training and execution. To eliminate this assumption, we propose a VAE that uses a parametric variational distribution which is conditioned on the observation-action-reward triplet of the agent under our control and a variable d indicating whether the episode has terminated; $q_w(z|\tau = (o, a, r, d))$. Specifically, our goal is to approximate the true posterior that is conditioned on opponent’s information, with a variational distribution that only depends on local information. The use of this local information in a recurrent fashion has been successfully used in meta-RL settings [Wang et al., 2016, Duan et al., 2016]. We start by computing the KL divergence between the two distributions:

$$D_{\text{KL}}(q_w(z|\tau)||p(z|\tau_{-1})) = \mathbb{E}_{z \sim q_w(z|\tau)}[\log q_w(z|\tau) - \log p(z|\tau_{-1})]$$

By following the works of Kingma and Welling [2014] and Higgins et al. [2017] and using the Jensen inequality, the VAE objective can be written as:

$$L(\tau, \tau_{-1}; w, v) = \mathbb{E}_{z \sim q_w(z|\tau)}[\log p_u(\tau_{-1}|z)] - \beta D_{\text{KL}}(q_w(z|\tau)||p(z)) \quad (4)$$

The reconstruction loss factorizes exactly similar to equation 3. From equation 4, it can be seen that the variational distribution only depends on locally available information. Since during execution only the encoder is required to generate the opponent’s model, this approach removes the assumption that access to the opponent’s observations and actions is available during execution. Figure 3 presents the diagram of the VAE.

4.4 Reinforcement Learning Training

We use the latent variable z augmented with the agent’s observation to condition the policy of our agent, which is optimized using RL. Consider the augmented observation space $\mathbb{O}' = \mathbb{O} \times \mathbb{Z}$, where \mathbb{O} is the original observation space of our agent in the Markov game, and \mathbb{Z} is the representation space of the opponent models. The advantage of learning the policy on \mathbb{O}' compared to \mathbb{O} is that the policy can adapt to different $z \in \mathbb{Z}$.

After training the variational autoencoder described in Section 4.2, we use it to train our agent against the opponents in the set \mathbb{T} . We use the DDPG [Lillicrap et al., 2015] algorithm for this task. At the beginning of each episode, we sample an opponent from the set \mathbb{T} . The agent’s input is the local observation and a sample from the variational distribution. We refer to this as OMDDPG (Opponent Modeling DDPG).

We optimize the second proposed VAE method jointly with the policy of the controlled agent. We use the A2C algorithm, similarly to the meta-learning algorithm RL² [Wang et al., 2016, Duan et al., 2016]. In the rest of this paper, we refer to this as SMA2C (Self Modeling A2C). The actor’s and the critic’s input is the local observation and a sample from the latent space. We back-propagate the gradient from both the actor and the critic loss to the parameters of the encoder. Therefore, the encoder’s parameters are shaped to maximize both the VAE’s objective as well as the discounted sum of rewards.

5 Experiments

5.1 Toy Example

We will first provide a toy example to illustrate SMA2C. We consider the classic repeated game of prisoner’s dilemma with a constant episode length of 25 time steps. We control one agent, and the opponent uses one of two possible policies. The first policy always defects, while the second opponent policy follows a tit-for-tat policy. At the beginning of the episode, one of the two opponent policies is randomly selected. We train SMA2C against the two possible opponents. The agent that we control has to identify the correct opponent, and the optimal policy, it can achieve, is to defect against opponent one and collaborate with opponent two. Figure 2 shows the payoff matrix, the embedding space at the last time step of the episode, and the episodic return that SMA2C and A2C achieve during training. Note that, based on the payoff matrix, the optimal average episodic return that can be achieved is 24.5.

5.2 Experimental Framework

To evaluate the proposed methods in more complex environments, we used the Multi-agent Particle Environment (MPE)

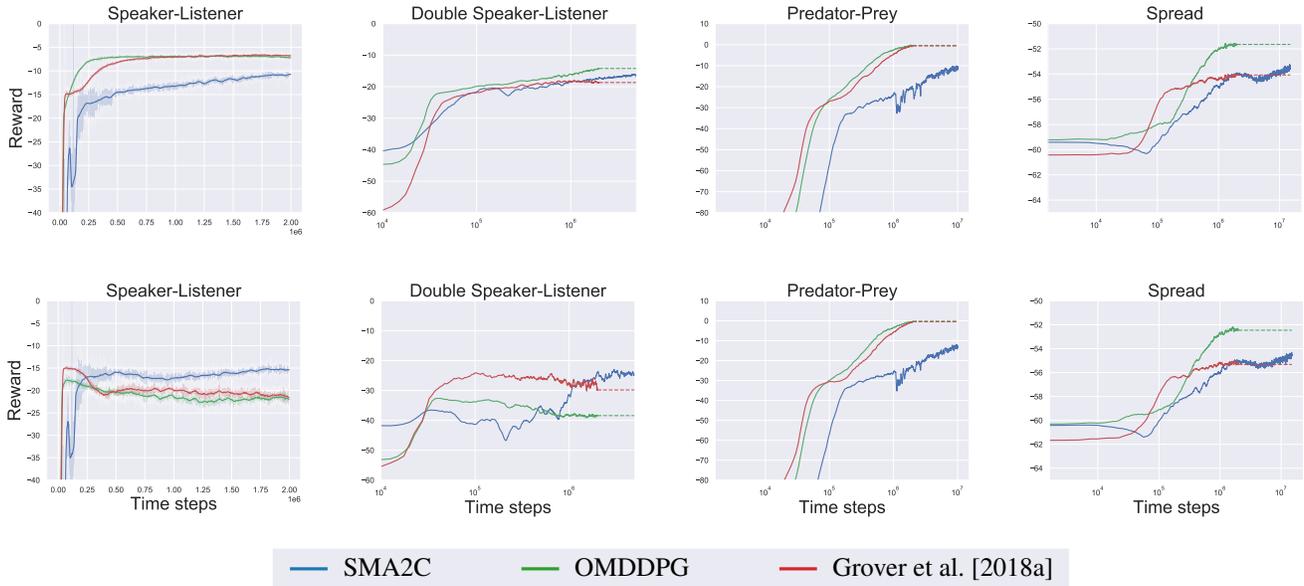


Figure 4: Episodic return during training against the opponents from \mathbb{T} (top row) and \mathbb{G} (bottom row). Four environments are evaluated; speaker-listener (first column), double speaker-listener (second column), predator-prey (third column) and spread (fourth column). For the double speaker-listener, predator-prey and spread environments the x-axis is in logarithmic scale.

[Mordatch and Abbeel, 2017], which provides several different multi-agent environments. The environments have continuous observation, discrete action space, and fixed-length episodes of 25 time steps. Four environments are used for evaluating the proposed methodology; speaker-listener, double-speaker listener, predator-prey, and spread. During the experiments, we evaluated the two proposed algorithms OMDDPG and SMA2C and the modeling method of Grover et al. [2018a] combined with DDPG [Lillicrap et al., 2015].

In all the environments, we pretrain ten different opponents, where five are used for training and five for testing. In the speaker-listener environment, we control the listener, and we create ten different speakers using different communication messages for different colors. In the double speaker-listener, which consists of two agents that have to be both listener and speaker simultaneously, we control the first agent. We create a diverse set of opponents that have different communication messages similar to speaker-listener, while they learn to navigate using the MADDPG algorithm [Lowe et al., 2017], with different initial random seeds. In the predator-prey environment, we control the prey and pretrain the three other agents in the environment using MADDPG with different initial parameters. Similarly, in spread, we control one of the agents, while the opponents are pretrained using MADDPG.

We use agent generalization graphs [Grover et al., 2018b] to evaluate the generalization of the proposed methods. We evaluate two types of generalizations. First, we evaluate the episodic returns against the opponents that are used for training, \mathbb{T} , which Grover et al. [2018b] call "weak generalization". Secondly, we evaluate against unknown opponents from the set \mathbb{G} , which is called "strong generalization".

5.3 Reinforcement Learning Performance

We evaluate the proposed opponent modeling methods in RL settings. In Figure 4, the episodic returns for the three methods in all four environments are presented.

Every line corresponds to the average return over five runs with different initial seeds, and the shadowed part represents the 95% confidence interval. We evaluate the models every 1000 episodes for 100 episodes. During the evaluation, we sample an embedding from the variational distribution at each time step, and the agent follows the greedy policy. The hyperparameters for all the experiments in Figure 4 were optimized on weak generalization scenarios, against opponents from the set \mathbb{T} .

OMDDPG is an upper baseline for SMA2C achieving higher returns in all environments during weak generalization. However, OMDDPG, as well as the method by Grover et al. [2018a], tend to overfit and perform poorly during strong generalization in the speaker-listener and double speaker-listener environment. SMA2C achieves higher returns than the method proposed by Grover et al. [2018a] in more than half of the scenarios.

5.4 Evaluation of the Representations

To evaluate the representations created from our models we will estimate the Mutual Information (MI) between the variational distribution ($q(z|\tau)$ or $q(z|\tau_{-1})$) and the prior on the opponents' identities, which is uniform. This is a common way to estimate the quality of the representation [Chen et al., 2018, Hjelm et al., 2019]. To estimate the MI, we use the Mutual Information Neural Estimation (MINE) [Belghazi et al., 2018]. Note that the upper bound of the MI, the entropy of the uniform distribution, in our experiments is

Table 1: MI estimations using MINE in the double speaker-listener and the predator-prey environment of the embeddings at the 15th, 20th and 25th time step of the trajectory.

| Algorithm \ Time step | Double speaker-listener | | | Predator-prey | | |
|-----------------------|-------------------------|------|------|---------------|------|------|
| | 15 | 20 | 25 | 15 | 20 | 25 |
| OMDDPG | 0.86 | 0.81 | 0.86 | 0.87 | 0.86 | 0.85 |
| SMA2C | 0.73 | 0.72 | 0.7 | 1.27 | 1.04 | 1.39 |
| Grover et al. [2018a] | 1.16 | 1.21 | 1.20 | 1.29 | 1.38 | 1.41 |

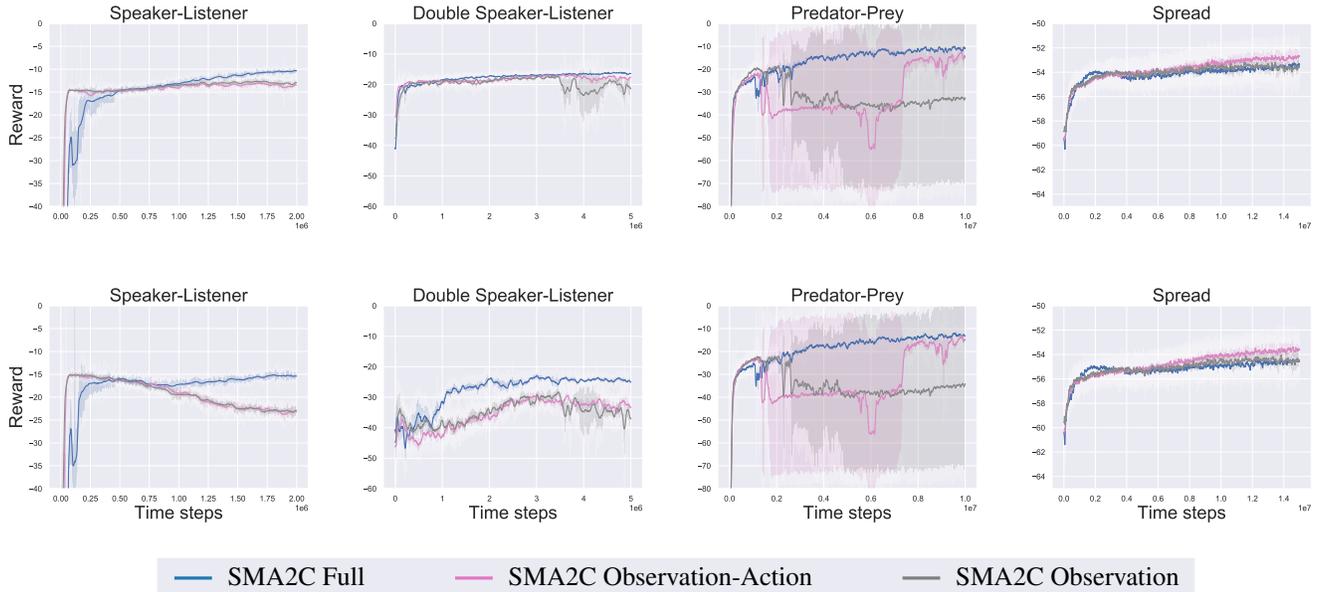


Figure 5: Ablation on the episodic returns for different inputs in the VAE of SMA2C for weak (top row) and strong (bottom row) generalization in all four environments.

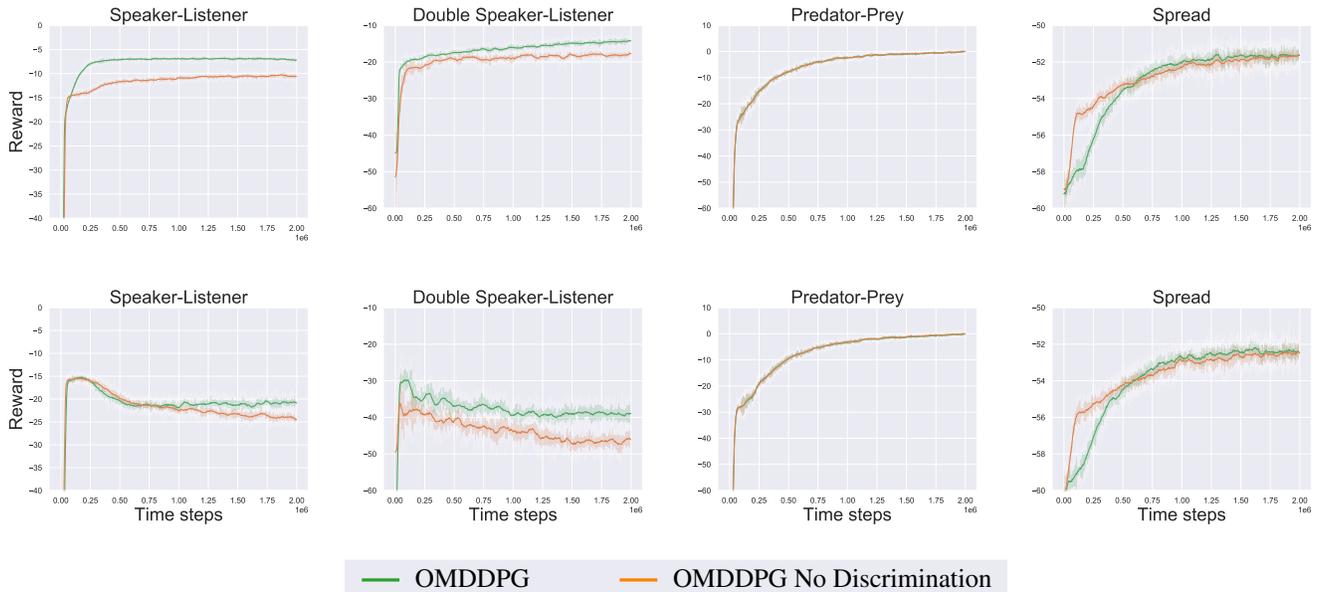


Figure 6: Ablation study on the importance of the discrimination objective. Episodic returns in all four environments for weak (top row) and strong (bottom row) generalization.

1.61. We gather 200 trajectories against each opponent in \mathbb{T} , where 80% of them are used for training and the rest for testing.

From Table 1, we observe that the method of Grover et al. [2018a] achieves significantly higher values of MI. We believe that the main reason behind this is the discrimination objective that implicitly increases MI. This is apparent in the MI values of OMDDPG as well. SMA2C manages to create opponent representations, based only on the local information of our agent, that have statistical dependency with the opponent identities. Additionally, based on Figure 4, we observe that the value of MI does not directly correlate to the episodic returns in RL tasks.

5.5 Ablation Study on SMA2C Inputs

We perform an ablation study to assess the performance requirements of SMA2C. Our proposed method utilizes the observation, action, reward, and termination sequence of our agent to generate the opponent’s model. We use different combinations of inputs in the encoder and compare the episodic returns. In Figure 5, the average episode return is presented for three different cases; SMA2C, SMA2C with only observations and actions and SMA2C with only observations. From Figure 5, we observe that during weak generalization the performance of SMA2C without using the reward and the termination variable is comparable to SMA2C in all environments. However, during strong generalization the performance SMA2C without using the reward and the termination variable drops in the speaker-listener and double speaker-listener environment. SMA2C using only the observations for modeling has similar performance to SMA2C that uses both the observations and actions of our agent in all environments except the predator-prey.

5.6 Ablation Study on Discrimination Objective

Another element of this work is the utilization of the discrimination objective proposed by Grover et al. [2018a] in the VAE loss. To better understand how the opponent separation in the embedding space is related to RL performance. Figure 6 shows the episodic return during the training for the OMDDPG with and without the discrimination objective. Using the discrimination objective has a significant impact on the episodic returns in the speaker-listener and the double speaker-listener environment. From Figure 6 we observe that the discrimination objective increases the performance of OMDDPG in the speaker-listener and double speaker-listener environment for weak and strong generalization.

6 Conclusion

To conclude this work, we proposed two methods for opponent modeling in multi-agent systems using variational autoencoders. First, we proposed OMDDPG, a VAE-based method that uses the common assumption that access to opponents’ information is available during execution. The goal of this work is to motivate opponent modeling without access to opponent’s information. The core contribution of this work is SMA2C, which learns representations without requiring access to opponent’s information during execution.

To the best of our knowledge this is the first study showing that effective opponent modeling can be achieved without requiring access to opponent observations. In the future, we would like to research how these models can be used for non-stationary opponents. In particular, we plan on investigating two scenarios; the first is multi-agent deep RL, where different agents are learning concurrently leading to non-stationarity in the environment [Papoudakis et al., 2019], which prevents the agents from learning optimal policies. Secondly, we would like to explore whether the proposed models can deal with opponents that try to deceive it and exploit the controlled agent [Ganzfried and Sandholm, 2011, 2015].

References

- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.
- John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, 2015a.
- Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, 2016.
- Stefano V Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 2018.
- He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. Opponent modeling in deep reinforcement learning. In *International Conference on Machine Learning*, 2016.
- Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. *arXiv preprint arXiv:1802.09640*, 2018.
- Aditya Grover, Maruan Al-Shedivat, Jayesh K Gupta, Yura Burda, and Harrison Edwards. Learning policy representations in multiagent systems. *International Conference on Machine Learning*, 2018a.
- Neil C Rabinowitz, Frank Perbet, H Francis Song, Chiyan Zhang, SM Eslami, and Matthew Botvinick. Machine theory of mind. *International Conference on Machine Learning*, 2018.
- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014.
- Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for pomdps. In *International Conference on Machine Learning*, 2018.
- David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

- Luisa Zintgraf, Maximilian Igl, Kyriacos Shiarlis, Anuj Mahajan, Katja Hofmann, and Shimon Whiteson. Variational task embeddings for fast adaptation in deep reinforcement learning. *International Conference on Learning Representations Workshop on Structure & Priors in Reinforcement Learning*, 2019.
- Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. *arXiv preprint arXiv:1703.04908*, 2017.
- Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor. Agent modeling as auxiliary task for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2019.
- Yan Zheng, Zhaopeng Meng, Jianye Hao, Zongzhang Zhang, Tianpei Yang, and Changjie Fan. A deep bayesian policy reuse approach against non-stationary agents. In *Advances in Neural Information Processing Systems*, pages 954–964, 2018.
- Andrea Tacchetti, H Francis Song, Pedro AM Mediano, Vinicius Zambaldi, Neil C Rabinowitz, Thore Graepel, Matthew Botvinick, and Peter W Battaglia. Relational forward models for multi-agent learning. *arXiv preprint arXiv:1809.11044*, 2018.
- Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. *International Conference on Learning Representations*, 2018.
- Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. In *Advances in Neural Information Processing Systems*, 2018.
- Kate Rakelly, Aurick Zhou, Deirdre Quillen, Chelsea Finn, and Sergey Levine. Efficient off-policy meta-reinforcement learning via probabilistic context variables. *International Conference on Machine Learning*, 2019.
- Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. <https://github.com/openai/baselines>, 2017.
- David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, 2014.
- 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.
- John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.
- Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*, 2017.
- Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings 1994*. Elsevier, 1994.
- Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.
- Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. RI^2 : Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.
- Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 2017.
- Aditya Grover, Maruan Al-Shedivat, Jayesh K Gupta, Yuri Burda, and Harrison Edwards. Evaluating generalization in multiagent systems using agent-interaction graphs. In *International Conference on Autonomous Agents and Multiagent Systems*, 2018b.
- Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2018.
- R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *International Conference on Learning Representations*, 2019.
- Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.
- Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V Albrecht. Dealing with non-stationarity in multi-agent deep reinforcement learning. *arXiv preprint arXiv:1906.04737*, 2019.
- Sam Ganzfried and Tuomas Sandholm. Game theory-based opponent modeling in large imperfect-information games. In *International Conference on Autonomous Agents and Multiagent Systems*, 2011.
- Sam Ganzfried and Tuomas Sandholm. Safe opponent exploitation. *ACM Transactions on Economics and Computation (TEAC)*, 2015.