Multi-Agent Actor-Critic with Hierarchical Graph Attention Network

Heechang Ryu, Hayong Shin, Jinkyoo Park*
Industrial & Systems Engineering, KAIST, Republic of Korea
{rhc93, hyshin, jinkyoo.park}@kaist.ac.kr

Abstract

Most previous studies on multi-agent reinforcement learning focus on deriving decentralized and cooperative policies to maximize a common reward and rarely consider the transferability of trained policies to new tasks. This prevents such policies from being applied to more complex multi-agent tasks. To resolve these limitations, we propose a model that conducts both representation learning for multiple agents using hierarchical graph attention network and policy learning using multi-agent actor-critic. The hierarchical graph attention network is specially designed to model the hierarchical relationships among multiple agents that either cooperate or compete with each other to derive more advanced strategic policies. Two attention networks, the inter-agent and inter-group attention layers, are used to effectively model individual and group level interactions, respectively. The two attention networks have been proven to facilitate the transfer of learned policies to new tasks with different agent compositions and allow one to interpret the learned strategies. Empirically, we demonstrate that the proposed model outperforms existing methods in several mixed cooperative and competitive tasks.

Introduction

In nature, the battle for dominance rights or desirable territories between individuals is a typical phenomenon (Smith and Price 1973). Occasionally, individuals in the same group cooperate to compete against enemy groups. They can gain stronger immunity against predators (Ugelvig and Cremer 2007) or powerful forces to overpower preys (Powell and Clark 2004). The cooperation and competition among agents are also important modeling paradigms in various engineering systems, such as smart grids (Dall'Anese, Zhu, and Giannakis 2013), logistics (Ying and Dayong 2005; Cao et al. 2013), and distributed vehicles/robots (Corke, Peterson, and Rus 2005; Fax and Murray 2004; Matignon et al. 2012). To control such complex systems composed of many interacting components, researchers have studied multi-agent reinforcement learning (MARL) for a long time.

Recently, reinforcement learning (RL) combined with deep neural networks has achieved human-level or higher performances in challenging games (Mnih et al. 2015; Silver et al. 2016; 2017). The advances in RL and deep learning have led great interests in MARL, hoping that it can resolve complex and large scale problems. The majority of MARL algorithms have focused on deriving decentralized policies for conducting a collaborative task. For the collection of actions individually determined by decentralized policies to be coordinated, it is important to impose consensus when deriving the decentralized policies. To achieve this, under the concept of centralized training and decentralized execution (CTDE), MARL learns centralized critics for multiple agents and derives decentralized actors using partial gradient from the centralized critics. Depending on the information available in the execution phase, CTDE approach can be further categorized into "learning-for-consensus approach", where each agent determines decentralized action solely based on its local observation (Lowe et al. 2017; Foerster et al. 2018; Zhang et al. 2018; Iqbal and Sha 2019), and "learning-to-communicate approach", where each agent employs communication schemes (i.e., interchanges messages) during the execution phase (Sukhbaatar, Fergus, and others 2016; Peng et al. 2017; Jiang and Lu 2018; Singh, Jain, and Sukhbaatar 2019; Das et al. 2019).

Although a variety of MARL algorithms have been proposed, unresolved issues still exist in MARL when it is applied to realistic environments. First, in terms of the generality of the modeling framework, most models focus on deriving pure cooperation or competition among multiple agents by forcing all agents to seek a shared common reward rather than modeling the relationships among heterogeneous agents in a mixed cooperative-competitive task (issue in modeling flexibility). In addition, the models are limited for modeling a large number of agents owing to the curse of dimensionality in modeling centralized critics (issue in scalability). This is connected with a more fundamental limitation, in which the trained model cannot be transferred to different tasks with different numbers of agents having different goals (issue in transferability).

We herein propose a model, called **H**ierarchical graph **A**ttention-based **M**ulti-**A**gent actor-critic (HAMA), that

*Corresponding author

conducts both representation learning for multi-agent system and policy learning using multi-agent actor-critic in end-to-end learning. HAMA employs a hierarchical graph neural network to effectively model the inter-agent relationships in each group of agents and inter-group relationships among groups. HAMA additionally employs inter-agent and inter-group attentions to adaptively extract the state-dependent relationships among multiple agents, which is proven to be effective for helping policies to adjust their high-level strategies (e.g., cooperate or compete). The combination of hierarchical graph neural networks with two distinct attention layers, which we refer to as a **H**ierarchical **G**raph **A**ttention ne**T**work (HGAT), effectively processes the local observation of each agent into a single embedding vector, an information-condensed and contextualized state representation for each individual agent. HAMA sequentially uses the embedding vector for each agent to compute the individual critic and actor for deriving decentralized policies.

We empirically demonstrate that HAMA outperforms existing MARL algorithms in four different game scenarios. Furthermore, we demonstrate that the policies trained by HAMA in a small-scale game with a small number of agents can be applied directly to control a large number of agents in a new game. Finally, we demonstrate that inter-agent and inter-group attentions can be used to interpret the derived policy and decision-making process.

Related Work

We categorize existing MARL studies into two categories; learning-for-consensus approach and learning-to-communicate approach depending on how the consensus among multiple agents is derived.

Learning-for-Consensus Approaches. Learning-for-consensus approaches in MARL focus on deriving decentralized policies (actors) for agents, each of which maps a local observation for an agent to an individual action for it. To make such individually chosen actions be coordinated to conduct collaborative tasks, these approaches first construct a centralized critic for either a global reward or individual reward and use the centralized critic to derive the decentralized actor. MADDPG (Lowe et al. 2017) has extended DDPG (Lillicrap et al. 2015) to multi-agent settings for mixed cooperative-competitive environments. COMA (Foerster et al. 2018) constructs a centralized critic and computes an agent-specific advantage function to derive a decentralized actor. FDMARL (Zhang et al. 2018) has proposed a distributed learning approach for each agent to learn a global critic using its local reward and the transferred critic parameters from the networked neighboring agents. Because these models directly use the state or observation in constructing critic or actor networks, it is difficult to apply such models to a large-scale environment or transfer them to new environments. As a way to resolve the scalability issue, the concepts of graph neural network (Gori, Monfardini, and Scarselli 2005; Scarselli et al. 2009; Battaglia et al. 2018) and attention network have been employed to effectively represent the global state and accordingly centralized critics. For example, MAAC (Iqbal and Sha 2019) employs the attention network and graph neural network to model a centralized critic, from which decentralized actors are derived using soft actor-critic (Haarnoja et al. 2018). In addition, DGN (Jiang, Dun, and Lu 2018) applies a graph convolutional network to model a centralized Q-function for each agent using a deep Q-network (Mnih et al. 2015).

Learning-to-Communicate Approaches. Another method to achieve consensus among decentralized policies in cooperative environments is using communication among agents. In this framework, each agent learns how to transmit messages to other agents and process the messages received from other agents to determine an individual action. During the centralized training, such message generating and processing procedures are learned to induce cooperation among agents. During the execution phase, agents exchange the messages to determine their actions. CommNet (Sukhbaatar, Fergus, and others 2016) uses a large single neural network to process all the messages transmitted by all agents globally, and the processed message is used to guide all agents to cooperate. BiCNet (Peng et al. 2017) has proposed a communication channel in the form of bi-directional recurrent network to accommodate messages from any number of agents, thus resolving the scalability issue. To effectively specify the communication structure while considering the relative relationships among agents, ATOC (Jiang and Lu 2018) has proposed a communication channel in a bi-directional LSTM with an attention layer. The attention layer in ATOC enables each agent to process the messages from other agents differently depending on their state-dependent importance. Similar to ATOC, IC3Net (Singh, Jain, and Sukhbaatar 2019) has been proposed to actively select and mask messages from other agents during communication by applying gating function in the message aggregation step. TarMAC (Das et al. 2019) has proposed a targeted communication protocol to determine whom to communicate with and what messages to transmit using attention networks. Such communication-based methods use attention networks to learn the communication structure/protocol effectively. Although the communication helps each agent to use extensive information in its individual decision making, this approach requires a separate communication channel and a well-established communication environment for message exchange. Furthermore, some communication-based methods can be applied to only cooperative tasks because it does not make sense to receive messages from competitive agents while playing a game.

Novelties of HAMA. As introduced, the graph neural network and attention network structures have been widely employed (1) to model a critic for scalable learning in learning-for-consensus approach, and (2) to model communication structure in learning-to-communicate approach. HAMA, our proposed model, employs the HGAT to embrace the merits of employing a graph representation in both learning-for-consensus and learning-to-communicate approaches. The proposed HGAT capturing enhanced relative inductive biases (Battaglia et al. 2018) enables HAMA to model both centralized critic and decentralized actor (1) that can be scal-

able and transferable and (2) that can effectively utilize the contextualized state representation. In particular, because it learns how to represent the partial observation through graph embedding, it is different from other communication approaches that utilize the messages processed by other agents. Therefore, our approach is considered to be a kind of learning-for-consensus approaches and can be applied for mixed cooperative and competitive environments easily.

Background

Partially Observable Markov Game (POMG). A POMG is an extension of partially observable Markov decision process to a game with multiple agents. A POMG for N agents is defined as follows: $s \in \mathcal{S}$ denotes the global state of the game; $o_i \in \mathcal{O}_i$ denotes a local observation that agent i can acquire; $a_i \in \mathcal{A}_i$ is an action for agent i. The reward for agent i is computed as a function of state s and joint action $\mathbf{a}$ as $r_i : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \mapsto \mathbb{R}$. The state evolves to the next state according to the state transition model $\mathcal{T} : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \mapsto \mathcal{S}$. The initial state is determined by the initial state distribution $\rho : \mathcal{S} \mapsto [0, 1]$. The agent i aims to maximize its discounted return $R_i = \sum_{t=0}^{T} \gamma^t r_i^t$, where $\gamma \in [0, 1]$ is a discount factor.

Multi-Agent Deep Deterministic Policy Gradient (MADDPG). Deterministic policy gradient (DPG) (Silver et al. 2014) aims to directly derive a deterministic policy, $a = \mu(s; \theta)$, that maximizes the expected return $\mathcal{J}(\theta) = \mathbb{E}_{s\sim\rho^\mu, a\sim\mu_\theta}[R] \approx \mathbb{E}_{s\sim\rho^\mu, a\sim\mu_\theta}[Q^\mu(s, a; \phi)]$, where $Q^\mu(s, a; \phi) = \mathbb{E}_{s'}[r(s, a) + \gamma\mathbb{E}_{a'\sim\mu}[Q^\mu(s', a')]$. The parameter θ of $\mu(s; \theta)$ is subsequently optimized by the gradient of $\mathcal{J}(\theta)$: $\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{s\sim\mathcal{D}}[\nabla_\theta \mu(s; \theta) \nabla_a Q^\mu(s, a; \phi)|_{a=\mu(s;\theta)}]$. The $\mathcal{D}$ is an experience replay buffer that stores (s, a, r, s') samples. Deep deterministic policy gradient (DDPG), an actor-critic model based on DPG, uses deep neural networks to approximate the critic and actor of each agent.

MADDPG is a multi-agent extension of DDPG for deriving decentralized policies for the POMG. MADDPG comprises the individual Q-network and policy network for each agent. The Q-network for agent i is learned by minimizing the loss: $\mathcal{L}(\phi_i) = \mathbb{E}_{\mathbf{o},\mathbf{a},r,\mathbf{o}'\sim\mathcal{D}}[(Q_i^\mu(\mathbf{o}, \mathbf{a}; \phi_i) - y_i)^2]$, where $\mathbf{o} = (o_1, \ldots, o_N)$ and $\mathbf{a} = (a_1, \ldots, a_N)$ are, respectively, the observations and actions of all agents, and $y_i = r_i(s, \mathbf{a}) + \gamma Q_i^{\mu'}(\mathbf{o}', \mathbf{a}'; {\phi_i}')|_{a'_j=\mu'(o'_j);\theta')}$. The policy network $\mu_i(o_i; \theta_i)$ of agent i is optimized using the gradient: $\nabla_{\theta_i} \mathcal{J}(\theta_i) = \mathbb{E}_{\mathbf{o},\mathbf{a}\sim\mathcal{D}}[\nabla_{\theta_i} \mu_i(o_i; \theta_i) \nabla_{a_i} Q_i^\mu(\mathbf{o}, \mathbf{a}; \phi_i)|_{a_i=\mu_i(o_i;\theta_i)}]$.

Graph Attention Network (GAT). The GAT (Veličković et al. 2017) is an effective model to process structured data that is represented as a graph. The GAT has proposed a way to compute the node-embedding vector of graph nodes by aggregating node embeddings h_j from neighboring nodes $\{j \in \mathcal{N}_i\}$ that are connected to the target node i as $h'_i = \sigma(\sum_{j\in\mathcal{N}_i} \alpha_{ij} \mathbf{W} h_j)$. The attention weight $\alpha_{ij} = \text{softmax}_j(e_{ij})$, where $e_{ij} = a(\mathbf{W}h_i, \mathbf{W}h_j)$, quantifies the importance of node j to node i in computing node-embedding value h'_i.

Methods

HAMA comprises a representation learning framework for processing the state represented as a graph and a multi-agent actor-critic network for deriving decentralized policies for the agents. As shown in Figure 1, HAMA represents the game state as a graph and computes for each agent the node-embedding vector that compactly summarizes each agent's status in relation with other groups of agents and environment. The computed node-embedding vector for each agent is subsequently used to compute the Q-value and action in an actor-critic framework.

State Representation Using HGAT

We propose HGAT, a network stacking multiple GATs hierarchically, that processes each agent's local observation into a high-dimensional node-embedding vector to represent the hierarchical inter-agent and inter-group relationships of each agent.

Agent Clustering. The first step in representation learning is to cluster all the agents into distinct groups C^k using prior knowledge or data. For pure cooperative tasks, all the agents can be categorized into a single group. If the target task involves competition between two groups, we can cluster the agents into two groups. In addition, we can cluster into a group the agents that do not execute any actions but participate in the game (i.e., terrain components or obstacles). In this study, we assume that the agents can be easily clustered into K groups using prior knowledge on the agents, which implies that HAMA utilizes enhanced relative inductive biases regarding the group relationships.

Node-Embedding Using GAT in Each Cluster. Agent i has the local observation $o_i = \{s_j | \ j \in V(i)\}$ where s_j is the local state of agent j, and $V(i)$ specifies the visual range of agent i. The visual range can be specified depending on environment settings so that agent i can observe the agents within a certain distance. Thus, our agent can observe nearby agents as a partial observation. Agent i computes the different node-embedding vectors $\bar{h}_i^k$ for different groups $k = 1, ..., K$ to summarize the individual relationships between agent i and agents from different groups. To compute $\bar{h}_i^k$, agent i first computes embedding $h_{ij}^k = f_M^k(s_i, s_j; w_M^k)$ between itself and agents in $j \in C^k \cap V(i)$ and computes the aggregated embedding $\bar{h}_i^k = \sum_{j\in C^k\cap V(i)} \alpha_{ij}^k h_{ij}^k$. The inter-agent attention weight α_{ij}^k quantifies the importance of the embedding h_{ij}^k from agent j to agent i. The inter-agent attention weight is computed as softmax $\alpha_{i,\cdot}^k \propto \exp(e_{i,\cdot}^k)$ where $e_{ij}^k = f_\alpha^k(s_i, s_j; w_\alpha^k)$. The attention can be extended to multiple attention heads (Vaswani et al. 2017), but the current study employs only plain and classical attention networks. It is noteworthy that agent i computes embedding h_{ij}^k by processing its own observation on other agents; therefore, the other agents are not required to send messages to agent

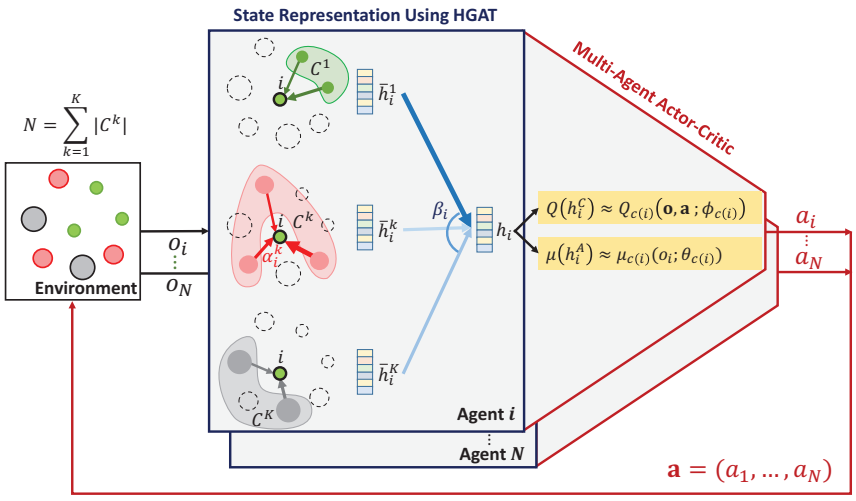

Figure 1: Overview of HAMA.

i, unlike other learning-to-communicate approaches that require agents to exchange their hidden vectors.

Hierarchical State Representation Using Multi-Graph Attention. This step aggregates the group-level node-embedding vectors $\bar{h}_i^1, ..., \bar{h}_i^K$ of agent i for the information-condensed and contextualized state representation of agent i as $h_i = \sum_{k=1}^{K} \beta_i^k \bar{h}_i^k$ while considering the relationships between agent i and the groups of other agents. The inter-group attention weight β_i^k guides which group agent i should focus more on to achieve its objective. For example, if β_i^k is large for the same group which agent i belongs to, it implies that agent i focuses on cooperating with the agents in the same group. Otherwise, agent i would focus more on competing with agents from different groups. The inter-group attention weight is computed as softmax $\beta_i = (\beta_i^1, ..., \beta_i^K) \propto \exp(q_i)$ where $q_i = [q_i^1, \ldots, q_i^K] = f_\beta([\bar{h}_i^1, \ldots, \bar{h}_i^K]; w_\beta)$. The hierarchical state representation is particularly useful when considering mixed cooperative-competitive games where each agent or group possesses their own objectives, which will be empirically shown by various experimental results in this study. The embedding and attention functions in this study comprise a two-layered MLP with 256 units and ReLUs.

Multi-Agent Actor-Critic

The proposed method uses the embedding vectors h_i^C and h_i^A of agent i to compute, respectively, the individual Q-value $Q_{c(i)}(\mathbf{o}, \mathbf{a}) \approx Q_{c(i)}(h_i^C; \phi_{c(i)})$ and determine the action $a_i = \mu_{c(i)}(o_i) \approx \mu_{c(i)}(h_i^A; \theta_{c(i)})$, where $c(i)$ is the group to which agent i belongs. Note that the embedding vectors h_i^C and h_i^A are computed separately using two different HGATs; computing h_i^C requires a joint action $\mathbf{a}$ in the training phase under CTDE. Additionally, agents in the same group share the actor and critic networks for generalization.

Compared to using raw observation as an input for the critic and actor network (Lowe et al. 2017), using node-embedding vectors computed from HGAT as inputs offers the following advantages: (1) a node-embedding vector can be computed by considering the hierarchical relationships among agents, i.e., relative inductive biases, thus providing contextualized state representation; (2) it is scalable to a large number of agents as the dimension of a node-embedding vector does not change with the number of agents; and (3) HGAT enables the learned policy to be used in environments of any agent or group size, i.e., the property that transfer learning aims to achieve.

The training of HAMA is similar to that of MADDPG. The shared critic Q_k for agent i in group k is trained to minimize the loss $\mathcal{L}$:

$$\mathcal{L}(\phi_k) = \mathbb{E}_{\mathbf{o},\mathbf{a},r_i,\mathbf{o}' \sim \mathcal{D}}[(Q_k^\mu(\mathbf{o}, \mathbf{a}; \phi_k) - y_i)^2],$$
$$y_i = r_i + \gamma Q_k^{\mu'}(\mathbf{o}', \mathbf{a}'; \phi_k')|_{a_i' = \mu'(o_i'; \theta')} \quad (1)$$

where $Q^{\mu'}$ and μ' are, respectively, the target critic and actor networks for stable learning with delayed parameters (Lillicrap et al. 2015). In CTDE framework, the joint observation and action are assumed to be available for training. The shared actor μ_k for agent i in group k is then trained using gradient ascent algorithm with the gradient computed as:

$$\nabla_{\theta_k} \mathcal{J}(\theta_k) =$$
$$\mathbb{E}_{\mathbf{o},\mathbf{a} \sim \mathcal{D}}[\nabla_{\theta_k} \mu_k(o_i; \theta_k) \nabla_{a_i} Q_k^\mu(\mathbf{o}, \mathbf{a}; \phi_k)|_{a_i = \mu_k(o_i; \theta_k)}] \quad (2)$$

where a_i is the action of agent i in $\mathbf{a}$. During the training, the joint observation $\mathbf{o}$ and joint action $\mathbf{a}$ are used, whereas, during the execution, only the learned policy $\mu_k(o_i; \theta_k) \approx \mu_{c(i)}(h_i^A; \theta_{c(i)})$ is used with the embedding vector h_i^A computed using only local observation o_i of agent i.

Experiments

Figure 2 shows the environments we use to evaluate the performances of the proposed and baseline MARL algorithms. It includes cooperative environments that have been widely used in existing studies (Lowe et al. 2017; Jiang and

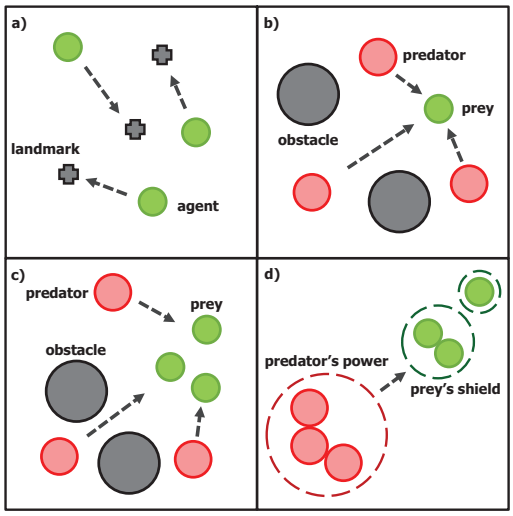

Figure 2: Illustrations of the experimental environments, including a) *Cooperative Navigation*, b) *3 vs. 1 Predator-Prey*, c) *3 vs. 3 Predator-Prey*, and d) *The More-The Stronger*.

Lu 2018) as well as mixed cooperative-competitive environments extended from well-known environments. As baseline algorithms for comparing the performances, we consider MADDPG and MAAC because they belong to learning-for-consensus approaches and are designed to process only local observation during the execution phase as HAMA does. These algorithms are more general for mixed cooperative-competitive games where communication is not always possible. For the cooperative navigation, we additionally consider ATOC, one of learning-to-communicate approaches, as a baseline because this game is fully cooperative; thus, the communication-based method can be naturally considered. All the performance measures are obtained by executing the trained policies with 3 different random seeds on 200 episodes. Regarding the visual range of the agent, we assume that each agent observes up to three nearest neighboring agents per each group with relative positions and velocities in all the experiment settings.

Cooperative Navigation

First, the proposed model is validated in the cooperative navigation, where only cooperation among agents exists. In the game, all the agents, which are homogeneous, are required to reach one of the landmarks without colliding with each other. Each episode starts with n randomly generated agents and landmarks and ends after 25 timesteps. During an episode, each agent receives $-d$, the distance to the nearest landmark, as a reward. In addition, each agent receives an additional reward, -1, whenever it collides with other agents during navigation. It is an optimal strategy that each agent occupies its distinct landmark.

Figure 3 compares the normalized mean penalties of four different MARL algorithms (instead of representing results with rewards that are negative, we present the results in terms of penalty as a negative reward since it is more intuitive for this game). A smaller mean penalty implies closer to the nearest landmark and fewer collisions with other agents. The penalty is averaged over every 10,000 steps until reaching 3 million steps.

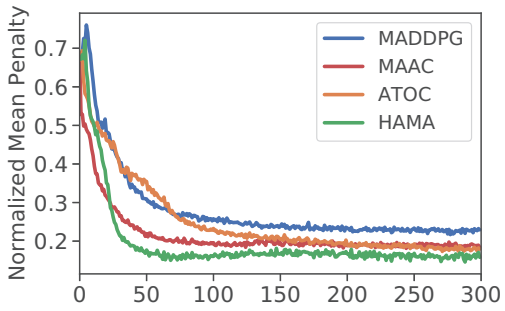

Figure 3: Penalties during training on cooperative navigation with 3 agents.

Table 1: The mean and standard deviation of penalties in cooperative navigation.

n	MADDPG	MAAC	ATOC	HAMA
3	0.22±0.010	0.19±0.012	0.17±0.007	**0.15**±0.008
Transfer learning using the policy trained with $n = 3$				
50	-	-	0.09 (70)	**0.05** (**90**)
100	-	-	0.07 (81)	**0.04** (**96**)

In training, as shown in Figure 3, HAMA converges fast to the lowest value. This is possible because HAMA effectively represents the state of each agent by considering their relative positions and velocities through HGAT. Table 1 compares the normalized mean penalties that the agents obtain during testing with 200 episodes with the trained models. ATOC achieves smaller mean penalty than MADDPG and MAAC. Because ATOC employs an active communication scheme based on attention network, it can effectively derive the cooperative behavior among agents. Our model has a lower mean penalty than ATOC. This indicates that the cooperative strategies trained by HAMA can effectively induce coordination among agents even without having active communication among agents.

Due to the use of a shared actor with efficient state representation, the trained policies by HAMA and ATOC can be applied to the cooperative game with any number of agents/landmarks, whereas the policies trained by MADDPG and MAAC cannot be transferred. The performance of transfer learning is also summarized in Table 1. When the policies trained by 3 agents are used to play the game with 50 and 100 agents, HAMA has the lower average penalty and higher percentages of landmark occupation (provided in the parenthesis in the table) by the participating agents. Note that when we conduct the transfer learning experiments, we reduce the size of agents (25 times smaller) to have a large number of agents in the same environment, where each agent can observe three nearest agents and three landmarks.

Table 2: The mean and standard deviation of scores for predators in 3 vs. 1 predator-prey game.

predator	prey ($n = 1$)		
($n = 3$)	MADDPG	MAAC	HAMA
MADDPG	0.30±0.02	0.23±0.02	0.07±0.01
MAAC	0.35±0.02	**0.39**±0.03	0.07±0.01
HAMA	**0.45**±0.03	**0.39**±0.03	**0.16**±0.01

3 vs. 1 Predator-Prey

A predator-prey game consists of two groups of agents competing with each other, along with obstacles that participate in the game but do not take an action. The goal of three homogeneous predators is to capture one prey, while the goal of the prey is to escape from the predators. For the predators to capture the prey, they need to cooperate with each other because of their slower speed and acceleration compared with those of the prey. Each predator gets a positive reward, $+10$, when it catches the prey, and the prey receives a negative reward, -10, when it is caught by a predator. When the prey leaves a certain zone, the prey receives a negative reward to prevent it from leaving farther. It is noteworthy that each agent seeks to maximize their accumulated rewards, which results in the competition between predators and prey.

We compare the performance of HAMA with two other models as baselines: MADDPG and MAAC. Each model is trained while self-playing (i.e., predators and prey are trained with the same model), and the trained policies are validated while having the trained policies compete with other policies trained by different models. Table 2 summarizes the average scores that a predator can obtain per step in an episode. The results indicate that the predators trained by HAMA have higher or similar scores than MADDPG and MAAC when competing with the prey trained by other models. Similarly, the prey trained by HAMA results in the lowest score when competing with the predators trained by different models. Note that when HAMA plays the role of a single prey where no cooperation is required, it still performs best in defending itself from the predators because it effectively configures the relationships with the predators and obstacles by using HGAT.

3 vs. 3 Predator-Prey

The next game we consider is 3 vs. 3 predator-prey game, a variant of the original 3 vs. 1 predator-prey game. The game rules are identical to those of 3 vs. 1 predator-prey game. In this game, if a predator recaptures a prey that has already been captured, neither reward occurs. Instead, each predator receives an additional reward, $+10 * t_r$, when the predators capture all preys, where t_r is the number of remaining timesteps in the episode, and the game ends. Although the game is similar to that of the original predator-prey game, the optimal strategy of the agents is no longer clear because more diverse and complex strategic interactions occur among the two groups of agents. For example, a predator can choose to either cooperate with other predators to chase a prey or to capture a prey individually if the prey is nearby. In addition to MADDPG and MAAC, we consider two heuristic strategies for the predators. In Heuristic 1, all the predators chase the same prey that has not been captured yet. In Heuristic 2, each predator chases the prey closest to the predator.

Table 3: The mean and standard deviation of scores for predators in 3 vs. 3 predator-prey game.

predator	prey ($n = 3$)		
($n = 3$)	MADDPG	MAAC	HAMA
Heuristic1	0.35±0.07	0.15±0.10	0.005±0.001
Heuristic2	0.72±0.10	0.30±0.14	0.01±0.001
MADDPG	1.18±0.13	1.05 ±0.22	0.02±0.01
MAAC	0.65±0.20	0.33±0.13	0.07±0.04
HAMA	**6.33**±0.10	**3.36**±0.34	**1.19**±0.09

Table 4: The mean and standard deviation of scores for different architectures in 3 vs. 3 predator-prey game.

predator	prey ($n = 3$)
($n = 3$)	HG-IAGA (HAMA)
SG-IAA (MAAC)	0.07±0.04
HG-NA	0.57±0.07
HG-IAA	1.03±0.08
HG-IGA	0.37±0.06
HG-IAGA (HAMA)	**1.19**±0.09

Table 3 compares the results of the game when the predators and preys, each of which is trained by self-playing, compete against each other. As shown in the table, the predators trained by HAMA achieve the highest scores against the preys trained by all other algorithms. Similarly, the preys trained by HAMA defend the best against the predators trained by all different algorithms, including the two heuristic strategies. The performance of HAMA is incomparably superior to those of other methods in both roles of predator and prey. This is remarkable in that HAMA and MAAC achieve a similar performance in the 3 vs. 1 predator-prey game where the only strategy of the predators is to cooperate to capture a unique prey. Meanwhile, in the 3 vs. 3 predator-prey game, each predator can choose from various strategies, such as cooperating with other predators or chasing prey individually. When chasing prey, a predator can also choose which prey to chase. The superior performance of HAMA is possible because it learns to represent better the hierarchical relationships among agents in the dynamic game owing to the relative inductive biases imposed by HGAT.

We validate our hypothesis on the success of HAMA's strategy by conducting an ablation study. Table 4 summarizes the performances of the following variant models:

- SG-IAA: Single-Graph & Inter-Agent Attention
- HG-NA: Hierarchical-Graph & No Attention
- HG-IAA: Hierarchical-Graph & Inter-Agent Attention

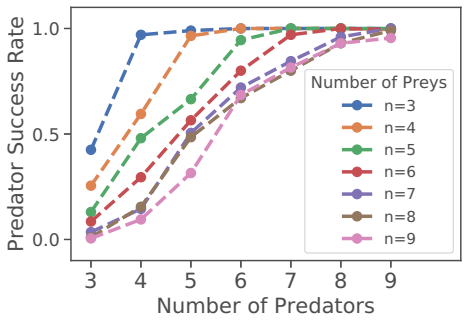

Figure 4: Results of transfer learning with the trained policies by HAMA in 3 vs. 3 predator-prey game. The success rate of predators is the rate of episodes in which predators capture all preys.

- HG-IGA: Hierarchical-Graph & Inter-Group Attention
- HG-IAGA (HAMA): Hierarchical-Graph & Inter-Agent & Inter-Group Attention

Note that the SG-IAA has a similar architecture with MAAC. Compared to the SG-IAA, a hierarchical graph attention architecture always scores higher regardless of whether attention is used. We assume that this effectiveness is due to the use of enhanced relative inductive biases regarding both the agent-level interactions and the group-level interactions. In comparing the role of attention, when both attentions are considered, the HG-IAGA outperforms the others. The combination of hierarchical graph structure and specially designed attentions is a key factor that induces the superior performance of HAMA.

Transfer Learning. In general, when the number of predators is large and the number of preys is small, the predators have a higher chance to win the game (i.e., capture all the preys within a single episode). As shown in Figure 4, this general trend is well realized when the predator and prey policies trained by HAMA in the 3 vs. 3 predator-prey game are transferred to play an m vs. n predator-prey game. It shows that the success rate of the predators is close to 1 when m (number of predators) $> n$ (number of preys).

Interpreting Strategies. We explain why a certain action of the agent is induced at a certain state by analyzing and interpreting the inter-agent and inter-group attention weights in HAMA. In Figure 5, the blue, red, and gray circles represent the predators, preys, and obstacles, respectively. The plots in each row show how each predator agent, which is represented by the blue circle with a black outline, attends other agents in the same and different groups over time. The width of the arrow indicates the magnitude of the attention weight α_{ij}^k on the agent the arrow is pointing out in each group. The blue and red bars at the top of each figure indicate the magnitudes of inter-group attention weights β_i^k to the predator ($k = 1$) and the prey groups ($k = 2$), respectively. The black arrow indicates the agent's action (i.e., direction and speed). From the predator's perspective, the attention to predators and preys can be interpreted as the attention to cooperation and competition. Figure 5 depicts the situation where predator 1 and 3 increase the cooperative attention (i.e., attention to the same group) over time to jointly chase the prey into a corner of the box. Meanwhile, predator 2 attempts to catch one prey, whose strategy is indicated by the attention to the competition (i.e., attention to the different group).

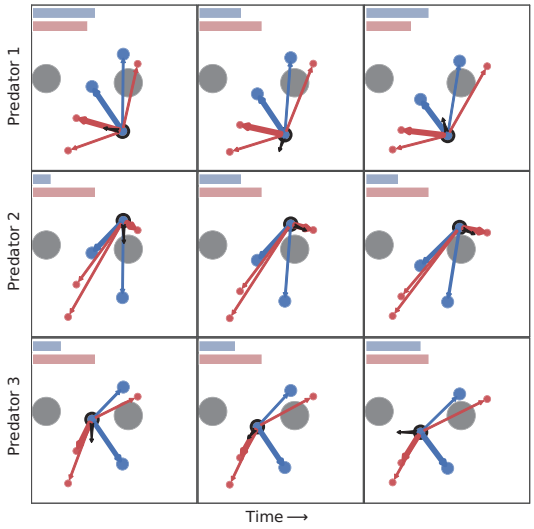

Figure 5: Reasoning on the strategy of HAMA.

The More-The Stronger

The more-the stronger game keeps the framework of the 3 vs. 3 predator-prey game. The additional game rule in this game is that when the preys are clustered together, only a group of predators whose size is equal to or larger than that of the clustered preys can capture the preys. For example, one predator can capture one prey by itself, but three-gathered predators are required to capture three-gathered preys. HAMA outperforms other models in this game.

Conclusions

We herein proposed a multi-agent actor-critic model based on state representation by a hierarchical graph attention network. Empirically, we demonstrated that the learned model outperformed other MARL models on a variety of cooperative and competitive multi-agent environments. In addition, the proposed model has been proven to facilitate the transfer of learned policies to new tasks with different agent compositions and allow one to interpret the learned strategies.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2017R1A2B4006290) and by the Technology Innovation Program (or Industrial Strategic Technology Development Program, 10067705, Development of AI-based Smart Construction System for 20% Cost Saving) funded by the Ministry of Trade, Industry & Energy (MI, Korea).

References

Battaglia, P. W.; Hamrick, J. B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Cao, Y.; Yu, W.; Ren, W.; and Chen, G. 2013. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics* 9(1):427–438.

Corke, P.; Peterson, R.; and Rus, D. 2005. Networked robots: Flying robot navigation using a sensor net. In *Robotics research. The eleventh international symposium*, 234–243. Springer.

Dall'Anese, E.; Zhu, H.; and Giannakis, G. B. 2013. Distributed optimal power flow for smart microgrids. *IEEE Trans. Smart Grid* 4(3):1464–1475.

Das, A.; Gervet, T.; Romoff, J.; Batra, D.; Parikh, D.; Rabbat, M.; and Pineau, J. 2019. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning*, 1538–1546.

Fax, J. A., and Murray, R. M. 2004. Information flow and cooperative control of vehicle formations. *IEEE transactions on automatic control* 49(9):1465–1476.

Foerster, J. N.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Gori, M.; Monfardini, G.; and Scarselli, F. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, 729–734. IEEE.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 1856–1865.

Iqbal, S., and Sha, F. 2019. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, 2961–2970.

Jiang, J., and Lu, Z. 2018. Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems*, 7254–7264.

Jiang, J.; Dun, C.; and Lu, Z. 2018. Graph convolutional reinforcement learning for multi-agent cooperation. *arXiv preprint arXiv:1810.09202*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, O. P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 6379–6390.

Matignon, L.; Jeanpierre, L.; Mouaddib, A.-I.; et al. 2012. Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In *AAAI*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529.

Peng, P.; Wen, Y.; Yang, Y.; Yuan, Q.; Tang, Z.; Long, H.; and Wang, J. 2017. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*.

Powell, S., and Clark, E. 2004. Combat between large derived societies: a subterranean army ant established as a predator of mature leaf-cutting ant colonies. *Insectes Sociaux* 51(4):342–351.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2009. The graph neural network model. *IEEE Transactions on Neural Networks* 20(1):61–80.

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms. In *ICML*.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *Nature* 550(7676):354.

Singh, A.; Jain, T.; and Sukhbaatar, S. 2019. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *International Conference on Learning Representations*.

Smith, J. M., and Price, G. R. 1973. The logic of animal conflict. *Nature* 246(5427):15.

Sukhbaatar, S.; Fergus, R.; et al. 2016. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, 2244–2252.

Ugelvig, L. V., and Cremer, S. 2007. Social prophylaxis: group interaction promotes collective immunity in ant colonies. *Current Biology* 17(22):1967–1971.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Ying, W., and Dayong, S. 2005. Multi-agent framework for third party logistics in e-commerce. *Expert Systems with Applications* 29(2):431–436.

Zhang, K.; Yang, Z.; Liu, H.; Zhang, T.; and Başar, T. 2018. Fully decentralized multi-agent reinforcement learning with networked agents. *arXiv preprint arXiv:1802.08757*.