https://doi.org/10.20965/jaciii.2024.p0273

Research Paper:

Effective Action Learning Method Using Information Entropy for a Single Robot Under Multi-Agent Control

Yuma Uemura, Riku Narita, and Kentarou Kurashige

Muroran Institute Technology
27-1 Mizumoto-cho, Muroran, Hokkaido 050-8585, Japan
E-mail: kentarou@ieee.org

[Received December 20, 2022; accepted October 17, 2023]

Robots that learn to perform actions using reinforcement learning to should be able to learn not only static environments, but also environmental changes. Heterogeneous multi-agent reinforcement learning (HMARL) was developed to perform an efficient search, with multiple agents mounted on a single robot to achieve tasks quickly. Responding to environmental changes using normal reinforcement learning can be challenging. However, HMARL does not consider the use of multiple agents to address environmental changes. In this study, we filtered the agents in HMARL using information entropy to realize a robot capable of maintaining high task achievement rates in response to environmental changes.

Keywords: reinforcement learning, multi-agent

1. Introduction

In recent years, the demand for robots has increased [1, 2], and studies have called for robots that can operate in complex environments [3–5]. However, it is often impossible to know what will happen in a complex environment in advance and therefore difficult to design robot controls accordingly. In such situations, reinforcement learning allows the robot to learn autonomously and adapt its behavior to the environment without having to design its controls in advance [6–8]. Reinforcement learning is expected to play an active role in disaster sites and industrial fields, where designing control systems in advance may be difficult [9, 10].

However, reinforcement learning is difficult to adapt to changes in an environment. When the environment changes, relearning is required, and it takes time to adapt the behavior to the changed environment. To cope with such changes, transfer learning, which transfers learned knowledge when the environment changes, has been developed [11, 12]. Transfer learning is a method of adapting an existing learned model to another. This method allows the transfer of knowledge even when the environment changes, thereby accelerating the adaptation. However, this method may result in the transfer of inappropriate knowledge, and correcting incorrect knowledge can increase learning time so it takes longer than without transfer. Therefore, we propose heterogeneous multi-agent reinforcement learning (HMARL) for single robots as a method to use learned knowledge without transition learning [13].

Multi-agent reinforcement learning is a learning method that uses multiple agents [14, 15]. Similarly, HMARL uses multiple agents in different learning spaces. In general, many studies, in which there are multiple robots and the learning space between robots is heterogeneous, have used HMARL [16, 17]. This study aimed to improve the efficiency of task accomplishment through cooperative behavior between robots in different learning spaces. There have also been studies that used the HMARL for the action selection of a single robot [13]. Unlike systems that focus on cooperative behavior among robots, HMARL systems focus on the action selection of a single robot. The HMARL system for a single robot comprises of multiple agents with different learning spaces on a single robot. These agents learn and accumulate knowledge based on the states recognized by the robot. In addition, each agent suggests the robot's next action. HMARL for a single robot adopts one of several proposed action choices, and the robot acts according to that choice. This method allows for exploration and task accomplishment using the knowledge of multiple agents. When the environment changes, the robot does not relearn but uses the agent's action choices that were learned before the environment changed. This reduces the time required for relearning. Each agent learns and chooses actions independently, and the robot chooses actions among them. Thus, there is no transfer of knowledge between the agents. Thus, there is no inhibition in relearning owing to inappropriate knowledge transfer.

However, there are problems with HMARL in a single robot. In this paper, we consider HMARL on a single robot, as in the previous study [13], and we propose a method to solve the problems of the previous studies. The problems are as follows. In prior research, when a robot was asked to accomplish a task, it adopted the action choices of the agent with the most advanced learning and acted accordingly. The learning progress of each agent was determined based on the temporal-difference (TD) error. If the TD error was small, the agent was considered to be making learning progress; if the TD error was large,

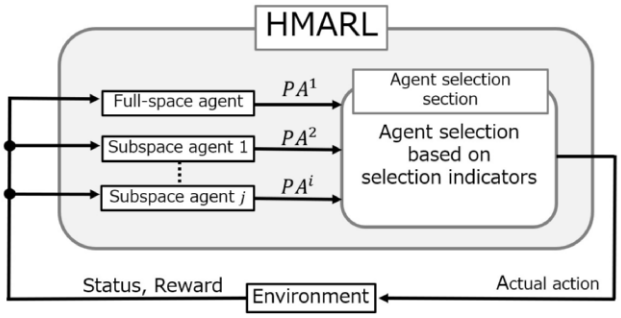

Fig. 1. Conceptual diagram of HMARL.

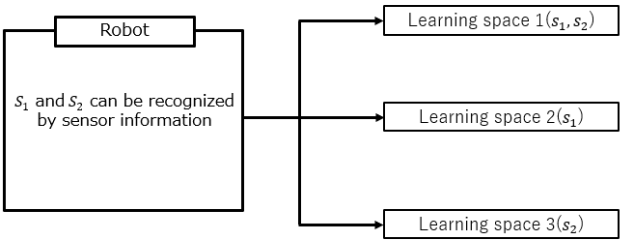

Fig. 2. Different state recognition methods.

the agent was considered as not making progress. However, the TD error is an update of the Q value, and it is not possible to determine the type of action the agent is performing from the TD error. Even if an agent's action selection method is approximately random, if the agent's TD error is small, the agent's action selection is judged to be aimed at achieving the task. This results in near-random selection of agents action suggestions regardless of the intention to complete a task. Task accomplishment by random actions is difficult and its efficiency is reduced. Therefore, the system used in the previous study needs a measure to determine whether each agent's choice of action is random.

In this paper, we propose a method to exclude agents with near-random action selection from the system, in addition to the methods used in previous studies. This prevents the robot from performing random actions when it wants to accomplish a task and improves its performance. The proposed method determines whether the action selection of an agent is random by measuring the information entropy of the action selection probability of each agent. Information entropy is used to avoid adopting the action choices of agents that suggest near-random actions. This prevents the robot from acting randomly when it wants to accomplish a task and improves its ability to accomplish the task compared to existing studies. The purpose of this study is to improve the efficiency of robot task accomplishment by excluding agents based on information entropy, in addition to the methods used in previous studies.

2. HMARL in a Single Robot

2.1. Heterogeneous Multi-Agent Reinforcement Learning

HMARL is a method for efficiency learning behaviors using multiple agents with different properties in a single robot. A conceptual diagram of the HMARL system is shown in **Fig. 1**.

An HMARL system includes one full-space agent and multiple subspace agents. Agent type is denoted by i. All agents select their own actions, which are defined as provisional actions (PA^i). The agent-selection procedure determines the *Actual action* to be output to the environment in the PA^i. All the agents learn based on the *Status* and *Reward* obtained from the *Actual action*. In HMARL, the learning rate varies depending on the method used to select the *Actual action*. Therefore, the design of the agent-selection process is important in HMARL. To create agents with different characteristics, each agent was trained to recognize different states. An image of the different state recognition methods is shown in **Fig. 2**.

For example, as shown in the figure, the robot states, s_1 and s_2, are recognized by the sensor attached to the robot. The learning spaces that can be created based on different states are learning space 1, which learns based on s_1 and s_2, learning space 2, which learns based on s_1, and learning space 3, which learns based on s_2. The learning space in which the states recognized by the robot and the created learning space match is the learning space of the full-space agent. The other learning spaces are those of the subspace agents. By learning from different states in this manner, each agent learns differently and chooses different actions. The learning space of a subspace agent is smaller than that of a full-space agent. This enables them to learn faster than a full-space agent can individually if a solution can be obtained by a subspace agent in the given environment. In addition, the HMARL system can use the action selection of subspace agents to accomplish the task, compared with a single agent that randomly selects actions for a new state to visit. Therefore, it is possible to explore and accomplish tasks more efficiently than when using a single agent. Thus, in the HMARL system, each agent learns based on different states and outputs its own action-selection probability. HMARL uses an agent-selection method to determine the action selection probability to use. Details of the agent-selection method are described in the next section.

2.2. Agent-Selection Method

In HMARL, the learning efficiency varies considerably depending on the method of selecting the *Actual action* to be executed among multiple *PA*s. In previous studies, *Actual action* was selected by measuring and determining the learning progress of full-space agents. In reinforcement learning, an untrained agent performs exploratory behavior to acquire knowledge. An agent that has learned more chooses actions based on the information it has obtained with an awareness of task accomplishment. Using this property, HMARL selects the *PA* of the least-trained

agent among the *PA*s and performs an exploratory action if the learning of the full-space agents has not progressed. When the learning of the full-space agents has advanced, the most advanced agent in *PA* is selected to accomplish the task. This enables efficient non-random search and precise task accomplishment. The learning progress of all the spatial agents is measured using the absolute value of the TD error. The absolute value of the TD error approaches zero as the learning progresses, and this property is used to measure the learning progress. The learning progress is calculated from the weighted average of the TD errors of the full-space agents using Eqs. (1) and (2), where μ_{δ_t} is the weighted average of the TD errors of the full-space agents, with weights set by a constant α.

$$\mu_{\delta_t} = (1-\alpha)\mu_{\delta_{t-1}} + \alpha|\delta_t| \quad \ldots\ldots\ldots (1)$$

$$L_t = 1 - \frac{\mu_{\delta_t}}{\mu_{\delta_{max}}} \quad \ldots\ldots\ldots\ldots (2)$$

In addition, δ_t is the TD error at time t for full-space agents. The learning progress L_t is obtained by dividing the calculated weighted average by the largest ever TD error $\mu_{\delta_{max}}$. For example, L_t of 0.3 means that the HMARL system is 30% ahead in learning, and L_t of 0.7 means that it is 70% ahead. In the previous study, the probability of selecting the least or most-trained agent varied with L_t calculated from Eq. (2). For example, if L_t is 0.3, the system selects the least-trained agent with a probability of 30% and uses the most-trained agent with a probability of 70%. When L_t is low, the utilization of the least advanced training *PA*, which performs more exploratory actions increases. When L_t is high, it increases the utilization of the advanced learning *PA* that performs the tasks. To determine the most and least-trained agents, the HMARL is designed with a value that represents the learning progress of each agent.

In reinforcement learning, the behavior of an untrained agent with a large TD error is considered the most exploratory behavior because it is considered highly novel with a large amount of information available. Conversely, actions that are highly learned with small TD errors utilize known knowledge with low novelty. Therefore, the degree of exploratory behavior was calculated based on the change in TD error for each agent.

$$\Delta I_t^i = \left|\delta_t^i\right| - \frac{1}{N}\sum_{k=1}^{N}\left|\delta_t^k\right| \quad \ldots\ldots\ldots (3)$$

$$I_{t+1}^i \leftarrow I_t^i + \Delta I_t^i \quad \ldots\ldots\ldots\ldots (4)$$

Equations (3) and (4) are used as indicators of agents making exploratory action choices, where i is the agent number, N is the total number of agents, and t indicates the time. Eq. (3) is used to determine the difference between the mean absolute values of the TD error for each agent. This determines whether the agent is making learning progress compared with other agents. The difference ΔI_t^i is used as the updated amount of I_t^i, which represents each agent's exploratory action selection, and is updated using Eq. (4). The action choice of the agent with the largest I_t^i value is the most exploratory action. The action choice of the agent with the lowest I_t^i value is the action with the highest knowledge use. The selection of actions focuses on exploration in the early stages of learning by varying the probability of the agent being selected depending on the learning progress of the system. In the second, the selection of actions focuses on knowledge use. This enables efficient non-random search and precise task accomplishment. However, when HMARL is used in a changing environment, there may be cases where agents with few recognizable states cannot learn effectively. The *PA* of an agent that cannot learn is likely near random. However, in the agent-selection method provided in the previous study, a *PA* that is close to random behavior cannot be determined from magnitude of the TD error and may be selected. There is a concern that selecting a *PA* that is close to such random behavior may result in an inefficient search and an inability to accomplish the task, owing to continued random behavior. In this study, we propose a method to remove the *PA*s that are close to random behavior from the potential options before performing agent selection. Specific methods for discriminating *PA* that perform random actions and for narrowing down the agents to be excluded from the agent-selection part are provided in Section 3.

3. Improving Learning Performance by Filtering Agents Using Information Entropy

The use of agents with random behavior in HMARL presents problems that render otherwise inefficient searches and tasks impossible. In this study, we distinguish agents that perform random actions by measuring the information entropy and narrowing down the list of agents that will perform the action. This allowed us to propose an extension of the HMARL to agents performing random actions. In our method, the robot has an agent with a learning space formed based on all sensor information and multiple agents with learning spaces created for each sensor. Each of these agents is trained in parallel and outputs its own calculation of the action value and selection. The proposed system calculates the information entropy of the agents based on the action value output from each agent and narrows down the agents to be put into action. Subsequently, it inputs the actions selected by the agent into the environment, acquires the states and rewards from the environment, and updates the learning space of each agent.

3.1. Improving Learning Performance by Filtering Out Agents that Perform Random Actions

In conventional reinforcement learning, when the environment changes, the results of learning in an environment similar to the past cannot be utilized, and learning must continue until an optimal behavior is achieved. In the previous study, an HMARL model trained from each of several mounted sensors selected an action for each sen-

sor. This allowed the system to use the knowledge learned from each sensor when there is a change in the environment, unlike conventional reinforcement learning. However, as the environment changes, the action choices of agents with few recognizable states may become more random. There are some concerns that selecting actions from agents performing random actions may result in inefficient search and inability to accomplish a task. In the agent selection in the previous study, the selection was based on the amount of change in the TD error, and it was not possible to determine the agent's choice of action. Therefore, agents whose action selection is close to random should be excluded from agent selection options. In this study, whether each agent's action choice is random is calculated by information entropy, and agents performing random actions are excluded from the *Actual action* choices. This allows agents whose action selection is not random to perform the action, thereby taking advantage of the subspace agents.

3.2. Filtering Agents Using Information Entropy

This method uses information entropy in terms of each agent's action selection probability as a measure of the randomness of the action selection. The information entropy is a measure of whether the probability of an event is random or biased. The information entropy is high in the random case and close to zero in the biased case. This method uses information entropy as a measure of the randomness of the action selection as the agent's action selection probability. The randomness of each agent's action selection changes with their learning progress. Therefore, the randomness of an agent's action selection must consider their learning progress. This method calculates the information entropy of each agent and its average value, and removes the agent action choices that are above the average from *Actual action*. Because each agent learns in parallel in HMARL, we design a dynamically changing criterion that considers the learning progress using the average value of each agent as the criterion. This eliminates agents that are closer to random behavior than other agents even if they have been trained the same number of times. We calculate the information entropy from Eqs. (5) and (6), respectively. Eq. (5) is a measure called the softmax method, which is used for agent action selection here. The selection probability of action a in the agent state s is obtained from the action selection probability $\pi(s,a)$. In addition, $Q(s,a)$ is the action value of action a in state s, providing higher selection probabilities to actions with higher action values. n is the number of actions that can be taken in state s and τ is a parameter called temperature, which is a real number within the range $\tau > 0$.

$$\pi(s,a) = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_{b=1}^{n} e^{\frac{Q(s,b)}{\tau}}} \quad \ldots\ldots\ldots\ldots (5)$$

The probability of selecting an action for each agent is normalized from 0 to 1 using Eq. (6) to determine whether the probability of selecting an action is random based on $\pi(s,a)$ calculated from Eq. (5). $H_i(s)$ is the information entropy of agent i in state s and $\pi(s,a_j)$ is the selection probability of action a_j in state s calculated using the softmax method.

$$H_i(s) = -\frac{1}{\log n} \sum_{j=1}^{n} \pi(s,a_j) \log \pi(s,a_j) \quad \ldots (6)$$

By using Eq. (6), the presence or absence of bias in each agent's action selection probability can be expressed as a value between zero and one. n is the number of actions that can be performed in the state s. This method calculates the average information entropy of all agents using Eq. (7). N is the total number of agents. An agent with above-average information entropy is highly random, and vice versa. We include $H_i(s) \leq \bar{H}(s)$ agents in the agent selection process. We select agents with above-average information entropy $\bar{H}(s) \leq H_i(s)$ by relative evaluation, excluding them in the agent selection.

$$\bar{H}(s) = \frac{1}{N} \sum_{k=1}^{N} H_k(s) \quad \ldots\ldots\ldots\ldots (7)$$

Using this method, agents with are close to random behavior can be distinguished. This prevents *Actual action* from approaching a random action, allowing efficient search and task accomplishment.

4. Simulation Experiment Assuming a Trash-Pickup Task

In this section, we examine whether narrowing down agents using information entropy improves the efficiency with which a robot can accomplish tasks. We compared the results of ordinary reinforcement learning with those of HMARL using the proposed method and HMARL without the proposed method. A simulation experiment was conducted in an environment that simulated a robot picking up trash from a factory. We compared the total amount of trash collected to determine whether the task efficiency improved. We compared the amount of garbage collected by changing the hyper parameters for qualitative evaluation of the proposed method. We also compared the number of times that each agent was used in the previous study and the proposed method to confirm whether the exclusion was due to information entropy.

4.1. Experimental Setup

In this experiment, a robot was asked to pick up trash in a computational environment that simulated a factory environment. A schematic of the experimental setup is shown in **Fig. 3**.

The experimental environment was designed to simulate the environment in a manufacturing facility in which garbage was generated around machines during production. In this environment, in addition to the static production machines, garbage-collecting robots equipped with

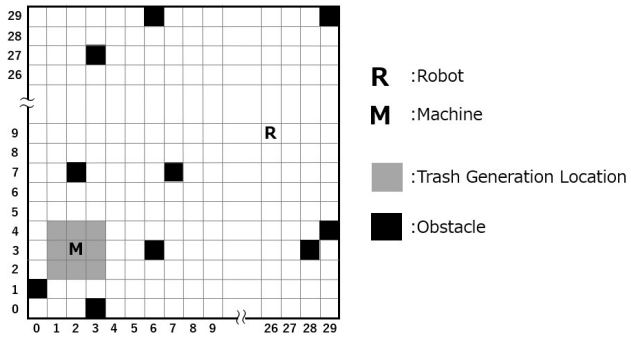

Fig. 3. Overview of simulation environment.

multiple sensors were present. The robots recognized their state based on onboard sensors and learned to collect debris that fell around the machines using reinforcement learning. The environment was a 30×30 grid map, and obstacles were placed around the map in addition to the machines. Thirty obstacles were placed on the map. The machines produced sound and trash, which were generated in a one-square area around the machine. The robot was equipped with position and sound sensors. The position sensor recognizes the x-axis and y-axis position based on the robot's movement history. The sound sensor could recognize the loudness and direction of incoming sound.

The robot could perform four types of actions: up, down, left, and right movements, and each action could move the robot one square in one direction. The robot could not move in the direction of walls or installed obstacles. The agents were created as full-space agents, with the training space created based on information from the position and sound sensors, as well as the position and sound agents created for each sensor. The position agent received the robot's position as a state and output an action selection probability, whereas the sound agent received the direction and loudness of the sound as a state and output another action selection probability. The proposed method measured the information entropy of three action selection probabilities to determine which agents to exclude. The full-space agent received the robot's position, loudness, and direction of the incoming sound as states, and output the action selection probabilities. When learning using normal reinforcement learning, only full-space agents selected actions.

The reward was 1 if the robot picked up the trash, and -0.1 if it failed to do so. The experiment comprised 5,000 trials, with each trial consisting of 1,000 actions. The environment randomly changed the positions of the robot and the machine in each trial. The obstacles were placed randomly at the beginning of the experiment and did not change as the trials progressed. The environmental settings of the experiment are shown in **Table 1**, and the experimental settings are shown in **Table 2**.

4.2. Results and Discussion

A comparison of the cumulative amounts of collected trash is shown in **Fig. 4**. The total amount of trash collected is shown in **Table 3**. The amount of trash collected when the hyperparameter softmax τ was changed in the proposed method is shown in **Table 4**.

In the previous study, the total amount of trash collected was 257,440 g. The total amount of trash collected by a single agent was 321,504 g. The total amount of trash collected using the proposed method was 726,207 g.

Next, **Fig. 5** shows the cumulative number of times that each agent was used in the previous HMARL, and **Fig. 6** shows the cumulative number of times that each agent was used in the proposed HMARL. The results show that the proposed method had relatively high use of sound agents up to about 1,500 trials, but after that, all spatial agents were used. In comparison, the previous study used the location agent the most frequently.

Table 1. Environment settings for the experiments with three types of agents.

Action	up, down, left, right
Size of environment	30×30
Initial robot position	Randomly
Initial position of the machine	Randomly
Obstacle location	Fixed
Trash generation locations	Around the machine

Table 2. Experimental setup for the experiments with three types of agents.

Number of experiments	1
Number of attempts	5,000
Learning timing	Per action
Definition of one trial	1,000 actions
Learning methods	Q-learning
Action selection methods	Softmax method
Softmax temperature variable τ	0.1
Learning rate α	0.1
Discount rate γ	0.9
Number of agents used	3

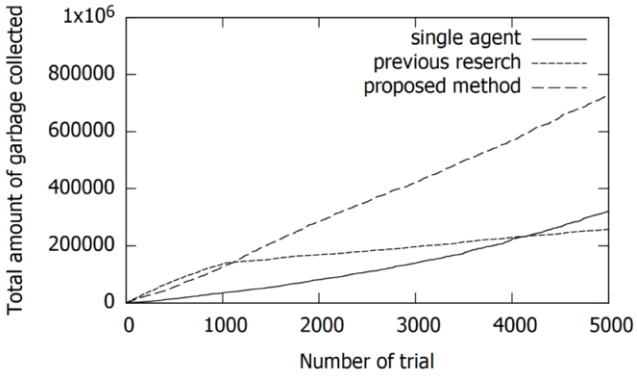

Fig. 4. Total amount of trash collected with three types of agents.

Table 3. Total amount of trash collected by each method with three types of agents.

Method	Previous	Single	Proposed
Amount of garbage	257,440 g	321,504 g	726,207 g

Table 4. Total amount of trash collected by changed τ.

τ	0.1	0.2	0.4
Amount of garbage	726,207 g	595,864 g	403,065 g

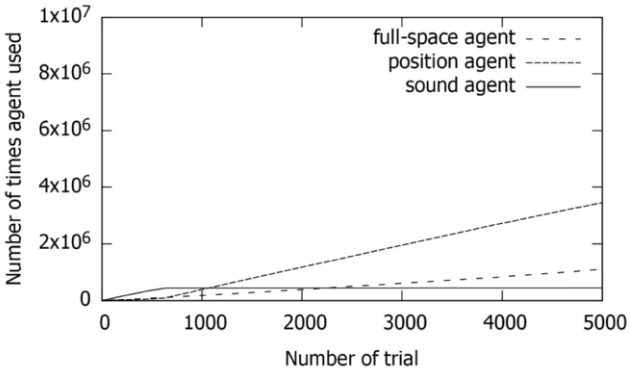

Fig. 5. Number of times the three types agents is used in the previous research.

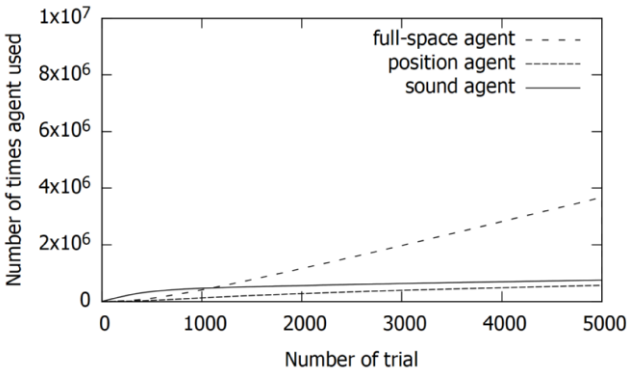

Fig. 6. Number of times the three types agents is used in the proposed method.

In this study, we conducted comparative experiments with training using only a single agent, training with HMARL without the proposed method, and training with HMARL using the proposed method. This resulted in a change in the total amount of trash collected. This occurred because, in the case of full-space agents only, there were many spaces where Q values had not yet accumulated. Therefore, learning was necessary. However, HMARL does not learn from a state where Q values have not been accumulated. Rather, it efficiently accomplishes the task by using the knowledge of multiple agents, and the amount of garbage collected is considered to have changed. A comparison of the results between the proposed method and the previous method showed a large difference in the final amount of trash collected. We attribute this result to the agent selection method based on the TD error of the previous method being unable to determine which agents were performing random actions. As a result, accomplishing the task became difficult. In approximately 4,000 trials, the total amount of trash collected was overtaken by the single agent, and the final result was the least amount of trash collected. When the value of hyperparameter τ was varied, the amount of trash collected decreased as the value of τ was increased. In action selection using softmax, increasing the value of τ can emphasize the low probability. This may have made the robot's action selection more exploratory and reduced its task accomplishment rate. We observed the cumulative number of times each HMARL agent was used and confirmed that in approximately around 1,500 trials, when the number of trials was low, the sound agent's action choices were frequently adopted, and the knowledge of the agent created for each sensor was used. We believe that this occurred because we used the knowledge of the agent created for each sensor trained in the previous trial, rather than the knowledge of the full-space agent for which Q values had not yet been accumulated. After 1,500 trials, all spatial agents were trained and could start trials with accumulated Q values; therefore, the number of trials used was considered to have exceeded the knowledge of the sound agent. In comparison, the HMARL in the previous study had the highest value for the number of times that the location agent was used. This confirms the exclusion of agents due to information entropy in comparison with the results of the proposed method. This result indicates that the action selection probability of the location agent was closer to a random action than those of the other agents. The number of times that the location agent was used in the previous study increased, even when the number of trials increased. These results indicate that, in previous studies, when agents wanted to accomplish a task after learning had progressed, they selected an action using an agent that suggested near-random action. Therefore, the task accomplishment rate of the previous study was lower than that of the proposed method.

5. Experiments with More Agents

In this experiment, agents were added to the experimental environment described in Section 4. A temperature sensor was added to the robot to create a temperature and light agent. These two agents were not involved in the trash pick-up task. We examined how this addition would affect the learning efficiency of the conventional HMARL and verified how the learning efficiency varied with the proposed method.

5.1. Summary of Experiments with Four Types of Agents

In this experiment, a temperature agent was added to the full-space agent, sound agent, and position agent de-

Table 5. Environment settings for the experiments with four types of agents.

Action	up, down, left, right
Size of environment	30×30
Initial robot position	Randomly
Initial position of the machine	Randomly
Obstacle location	Fixed
Trash generation locations	Around the machine
Environment temperature	24°C

scribed in Section 4, and HMARL was performed using four types of agents. The environment was a 30×30 grid map, on which the machines and obstacles were placed. The machines produced sound and debris in a one-square area around the machine. Thirty obstacles were placed on the map. A temperature sensor was added to the group of sensors used in the robot, as described in Section 4. The temperature of the environment was constant at 24°C, and the temperature agent was always aware of the constant conditions and proceeded with its learning. The robot could perform four types of actions: including up, down, left, and right movements, and each action could move the robot one square in one direction. The robot could not move in the direction of walls or installed obstacles. The agent created a full-space agent with all sensor information, a position agent created for each sensor, a sound agent, and a temperature agent. The full-space agent received all the sensor information recognized by the robot as a state and output the action selection probabilities. The position agent received the robot's position as a state and output an action selection probability. The sound agent received the direction and loudness of the sound as a state and output an action selection probability. The temperature agent received the temperature of the environment as a state and output an action selection probability. The proposed method measured the information entropy of four action selection probabilities to determine which agents to exclude.

The reward was 1 if the robot picked up the trash, and -0.1 if it failed to do so. The experiment comprised 5,000 trials, with each trial consisting for 1,000 actions. The environment randomly changed the positions of the robot and the machine in each trial. The obstacles were placed randomly at the beginning of the experiment and did not change as the trials progressed. The environmental settings of the experiment are shown in **Table 5**, and the experimental settings are shown in **Table 6**.

5.2. Results of Comparative Experiments with Four Types of Agents

In this section, we examine whether narrowing down agents by information entropy improves the robot's task accomplishment efficiency. We compared the results of ordinary reinforcement learning with those of HMARL using the proposed method and HMARL without the proposed method. We compared the total amount of trash collected to determine whether the task efficiency improved. By observing the number of times that each agent was used, we can confirm the change in the number of times that the agent was used owing to the information entropy. A comparison of the cumulative amounts of trash collected is shown in **Fig. 7**. The final total amount of collected trash is shown in **Table 7**.

Table 6. Experimental setup for the experiments with four types of agents.

Number of experiments	1
Number of attempts	5,000
Learning timing	Per action
Definition of one trial	1,000 actions
Learning methods	Q-learning
Action selection methods	Softmax method
Softmax temperature variable τ	0.1
Learning rate α	0.1
Discount rate γ	0.9
Number of agents used	4

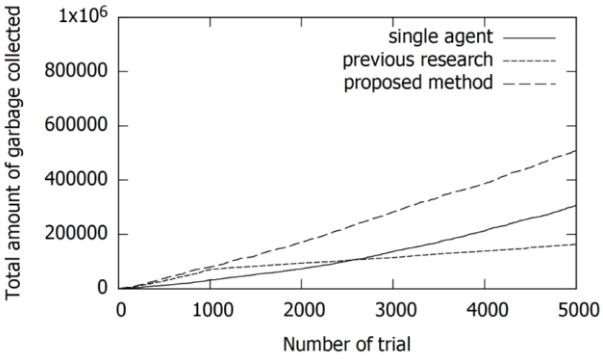

Fig. 7. Total amount of trash collected with four types of agents.

Table 7. Total amount of trash collected by each method with four types of agents.

Method	Previous	Single	Proposed
Amount of garbage	306,973 g	163,535 g	509,606 g

In the previous study, the total amount of collected trash was 306,973 g. The total amount of trash collected from a single agent was 163,535 g. The total amount of trash collected by the proposed method was 509,606 g. Next, the numbers of times that each agent was used in the previous research and the proposed method are shown in **Figs. 8** and **9**. As a result, both increased the number of times that the position agent was used, which was not necessary to accomplish the task.

In this study, we conducted comparative experiments with training using only a single agent, training with HMARL without the proposed method, and training with HMARL using the proposed method. As a result, the HMARL-based method performed worse when the num-

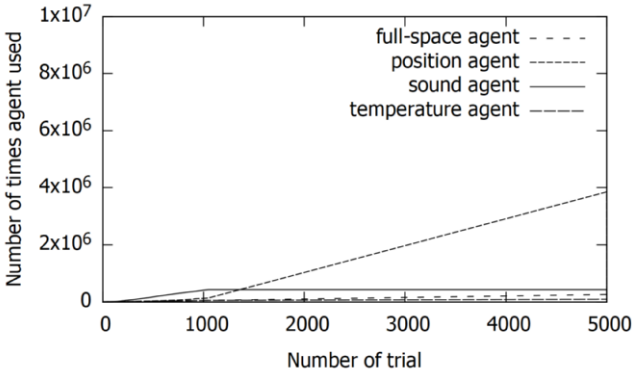

Fig. 8. Number of times the four types agents is used in the previous research.

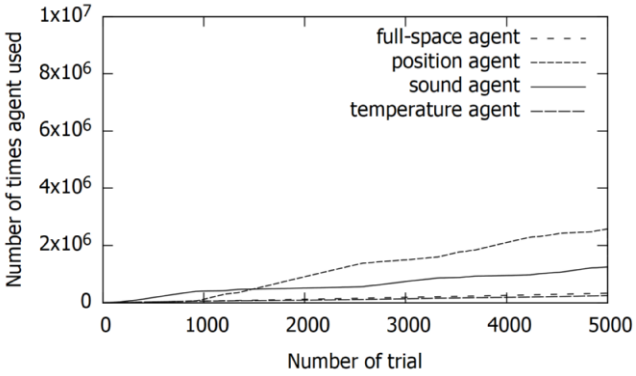

Fig. 9. Number of times the four types agents is used in the proposed method.

ber of agents was increased to four than when the number of agents was three. However, no significant changes were observed in the total amount of trash collected using a single agent. The addition of agents that were not aware of the state change resulted in poor results because of the use of their knowledge by agents that were not required to accomplish the task. However, the proposed method performed better than the methods used in the previous studies because it can exclude agents that have not yet learned to function as expected and the action selection of which is close to random. The task completion rate of the single agent did not change because we only added a state axis that always received only constant values; therefore, the learning space did not increase and no significant difference in learning performance occurred. In terms of the number of times that each agent was used, the position agent was used most frequently in both cases. The reason for this may be that the average information entropy, the criterion for exclusion, increased as the number of agents that is not required to accomplish the task. This is thought to have caused a decrease in the learning performance because agents with near-random behavior that should have been excluded were included in the agent-selection section. In the next experiment, the number of agents was increased to five.

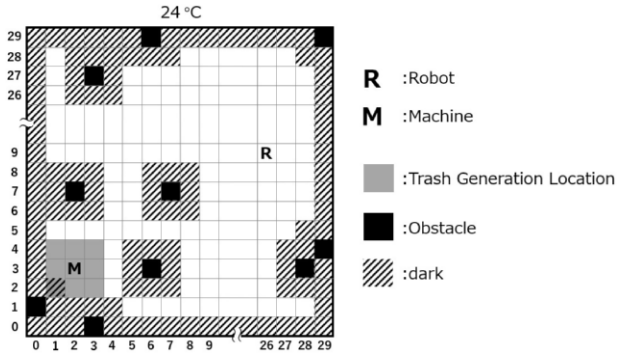

Fig. 10. Simulation environment.

5.3. Summary of Experiments with Five Different Agents

In this experiment, the robot was equipped with an illuminance sensor to create a light agent, and five types of agents were used. The environment was a 30×30 grid map, on which machines and obstacles were placed. The machines produced sound and debris, which was generated in one-square around the machines. Thirty obstacles were placed on the map. The temperature of the environment was held constant at 24°C. The light agent learned in a discrete environment where it received a value of 200 lx for one square around walls and obstacles and 300 lx in other conditions. An overview of the environment is shown in **Fig. 10**.

The robot could perform four types of actions: up, down, left, and right movements, and each action could move the robot one square in one direction. The robot could not move in the direction of walls or installed obstacles. Agents were created as full-space agents, and position agents, sound agents, temperature agents, and light agents were created for each sensor. The full-space agent received all the sensor information recognized by the robot as a state and output the action selection probabilities. The position agent received the position of the robot as a state and output an action selection probability. The sound agent received the direction and loudness of the sound as a state and output an action selection probability. The temperature agent received the temperature of the environment as a state and output an action selection probability. The light agent received the illuminance of the environment as a state and output an action selection probability. The proposed method measured the information entropy of these five action selection probabilities to determine which agents to exclude.

The reward was 1 if the robot picked up the trash, and -0.1 if it failed to do so. The experiment comprised 5,000 trials, with each trial consisting for 1,000 actions. The environment randomly varied the positions of the robot and machine in each trial. The obstacles were placed randomly at the beginning of the experiment and did not change as the trials progressed. The environmental settings of the experiment are shown in **Table 8**, and the experimental settings are shown in **Table 9**.

Table 8. Environment settings for the experiments with five types of agents.

Action	up, down, left, right
Size of environment	30×30
Initial robot position	Randomly
Initial position of the machine	Randomly
Obstacle location	Fixation
Trash generation locations	Around the machine
Environment temperature	24°C
Illuminance [dark, bright]	[200 lx, 300 lx]

Table 9. Experimental setup for the experiments with five types of agents.

Number of experiments	1
Number of attempts	5,000
Learning timing	Per action
Definition of one trial	1,000 actions
Learning methods	Q-learning
Action selection methods	Softmax method
Softmax temperature variable τ	0.1
Learning rate α	0.1
Discount rate γ	0.9
Number of agents used	5

5.4. Results of Comparative Experiments with Five Types of Agents

In this section, we examine whether filtering by information entropy improved the robot's task accomplishment efficiency. We compared the results of ordinary reinforcement learning with those of HMARL using the proposed method and HMARL without the proposed method. We compared the results in terms of the total amount of trash collected to determine whether the task efficiency improved. By observing the number of times that each agent was used, we confirmed the change in the number of times that the agent was used owing to the information entropy. A comparison of the cumulative amounts of collected trash is shown in **Fig. 11**. The final total amount of collected trash is shown in **Table 10**.

In the previous study, the total amount of trash collected was 309,477 g. The total amount of trash collected by a single agent was 138,882 g. The total amount of trash collected using the proposed method was 409,517 g.

Next, the number of times that each agent was used in a method developed in previous research and in the proposed method is shown in **Figs. 12** and **13**. Both methods resulted in very high values for the use of position agents.

In this study, we conducted comparative experiments with training using only a single agent, training with HMARL without the proposed method, and training with HMARL using the proposed method. As a result, the amount of trash collected by HMARL was lower than that collected by the four agents. However, the proposed

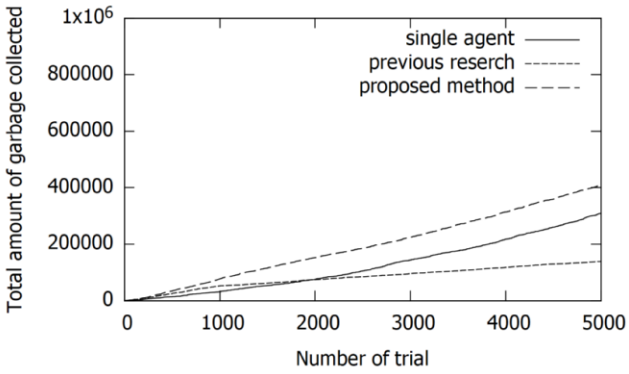

Fig. 11. Total amount of trash collected with five types of agents.

Table 10. Total amount of trash collected by each method with five types of agents.

Method	Previous	Single	Proposed
Amount of garbage	309,477 g	138,882 g	409,517 g

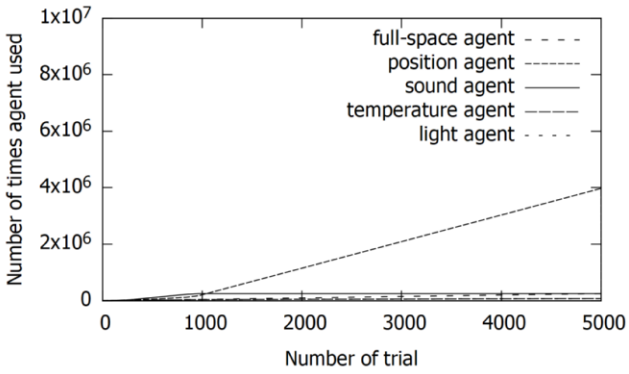

Fig. 12. Number of times the five types agents is used in the previous research.

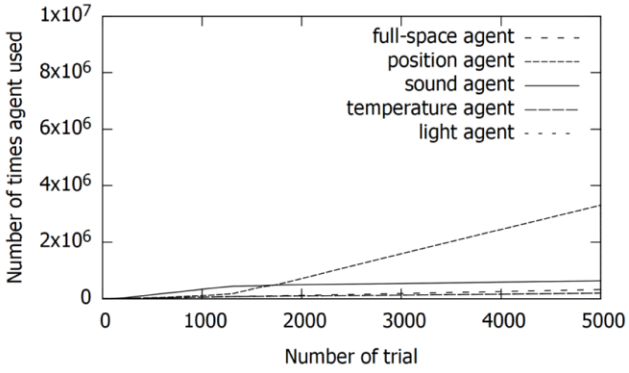

Fig. 13. Number of times the five types agents is used in the proposed method.

method reduced the decrease in the amount of trash collected. This allowed for maximum trash collection compared with single agents and previous studies. For a single agent, the learning performance did not change significantly because the learning space used did not increase as much as when there were four agents. The reason the training space used did not increase is that the state of the added illumination sensor was determined from the information provided by the existing position sensor. The change in agent usage showed a significant increase in the use of location agents compared with the use of four agents. The reason for this may be that the number of agents whose random behavior was not necessary to accomplish the task increased compared to the four types of agents. This increased the average information entropy of all the agents, which is the criterion for an agent to be excluded by the proposed method. An increase in the mean increased the criteria for agents whose action choices were close to random, making it difficult to narrow down the agents that should be excluded. This also caused a performance degradation in the HMARL system of the proposed method.

6. Conclusion

In this paper, we proposed a method to improve the task accomplishment efficiency using HMARL in a single robot. In HMARL, agents with few recognizable states may act randomly, thereby inhibiting efficient search and task accomplishments. However, in previous studies, the agent-selection method for actual actions was calculated from the TD error, and the problem of selecting agents without being able to judging whether each agent's action selection was close to random was present. Therefore, in this study, we confirmed an improvement in task accomplishment efficiency by filtering agents using information entropy before agent selection. In Section 4, we showed that the proposed method significantly improved the degree of task accomplishment and demonstrated its usefulness.

In Section 5, we described a similar experiment conducted in an environment with multiple agents. We increased the number of agents to four and five and checked the changes in task achievement efficiency and the number of times that the agents were used. Consequently, the task efficiency of the proposed method was the best for all four or five task types because it reduced the number of agents that were unnecessary to accomplish the task. However, we confirmed that the task efficiency in both the previous study and the proposed method decreased as the number of agents increased.

In future studies, we plan to improve the method for filtering agents using information entropy. The proposed method calculates the information entropy of each agent and its average to filter out the agents that are unnecessary for accomplishing task. However, it is conceivable that the average information entropy could also increase with the number of agents that are not required to accomplish the task. This increases the number of criteria for narrowing the list and may result in the selection of agents that are unnecessary for task accomplishment in the agent-selection process. The experimental results in Section 4 also confirmed the number of times that an agent was used, which had been excluded when there were three types of agents. This indicates that we cannot simply create a criterion to reduce the average information entropy. A unique evaluation value should be set for each agent to determine whether it should be included in agent selection.

Further research could include simulations, as well as practical experiments. Using actual equipment, we would aim to verify whether the proposed method can handle disturbances and noise in an environment that is not capable of receiving values as programmed. We would also like to prepare several single robots proposed in this study and verify the performance of the emphasized actions among them. This allows for a comparison with existing multi-agent systems and HMARL using open-source benchmarks. We would like to clarify the advantages and disadvantages of each method and compare them to develop our methods.

References:

[1] T. Hashimoto, X. Tao, T. Suzuki, T. Kurose, Y. Nishikawa, and Y. Kagawa, "Decision Making of Communication Robots Through Robot Ethics," J. Adv. Comput. Intell. Intell. Inform., Vol.25, No.4, pp. 467-477, 2021. https://doi.org/10.20965/jaciii.2021.p0467

[2] Y. Yamazaki, M. Ishii, T. Ito, and T. Hashimoto, "Frailty Care Robot for Elderly and its Application for Physical and Psychological Support," J. Adv. Comput. Intell. Intell. Inform., Vol.25, No.6, pp. 944-952, 2021. https://doi.org/10.20965/jaciii.2021.p0944

[3] H. Zhang, Y. Wang, J. Zheng, and J. Yu, "Path planning of industrial robot based on improved RRT algorithm in complex environments," IEEE Access, Vol.6, pp. 53296-53306, 2018. https://doi.org/10.1109/ACCESS.2018.2871222

[4] D. Kragic, J. Gustafson, H. Karaoguz, P. Jensfelt, and R. Krug, "Interactive, Collaborative Robots: Challenges and Opportunities," Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI), pp. 18-25, 2018. https://doi.org/10.24963/ijcai.2018/3

[5] K.N. McGuire, C. De Wagter, K. Tuyls, H. J. Kappen, and G. C. H. E. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," Science Robotics, Vol.4, No.35, 2019. https://doi.org/10.1126/scirobotics.aaw9710

[6] Y. Tsurumine, Y. Cui, E. Uchibe, and T. Matsubara, "Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation," Robotics and Autonomous Systems, Vol.112, pp. 72-83, 2019. https://doi.org/10.1016/j.robot.2018.11.004

[7] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," IEEE Robotics and Automation Letters, Vol.4, No.2, pp. 610-617, 2019. https://doi.org/10.1109/LRA.2019.2891991

[8] K. Rao, C. Harris, A. Irpan, S. Levine, J. Ibarz, and M. Khansari, "RL-CycleGAN: Reinforcement learning aware simulation-to-real," 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 11154-11163, 2020. https://doi.org/10.1109/CVPR42600.2020.01117

[9] H. X. Pham, H. M. La, D. Feil-Seifer, and L. Van Nguyen, "Reinforcement learning for autonomous UAV navigation using function approximation," 2018 IEEE Int. Symp. on Safety, Security, and Rescue Robotics (SSRR), 2018. https://doi.org/10.1109/SSRR.2018.8468611

[10] Y. Tang, M. Chen, C. Wang, L. Luo, J. Li, G. Lian, and X. Zou, "Recognition and localization methods for vision-based fruit picking robots: A review," Frontiers in Plant Science, Vol.11, 2020. https://doi.org/10.3389/fpls.2020.00510

[11] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," J. of Machine Learning Research, Vol.10, pp. 1633-1685, 2009.

[12] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," Proc. of the IEEE, Vol.109, No.1, pp. 43-76, 2020. https://doi.org/10.1109/JPROC.2020.3004555

[13] R. Narita, T. Matsushima, and K. Kurashige, "Efficient exploration by switching agents according to degree of convergence of learning on heterogeneous multi-agent reinforcement learning in single robot," 2021 IEEE Symp. Series on Computational Intelligence (SSCI), 2021. https://doi.org/10.1109/SSCI50451.2021.9659982
[14] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," Proc. of the 36th Int. Conf. on Machine Learning (PMLR), Vol.97, pp. 2961-2970, 2019.
[15] L. Liang, H. Ye, and Y. L. Geoffrey, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," IEEE J. on Selected Areas in Communications, Vol.37, No.10, pp. 2282-2292, 2019. https://doi.org/10.1109/JSAC.2019.2933962
[16] Y. Rizk, M. Awad, and E. W. Tunstel, "Cooperative heterogeneous multi-robot systems: A survey," ACM Computing Surveys, Vol.52, No.2, Article No.29, 2019. https://doi.org/10.1145/3303848
[17] Y. Yang, H. Modares, D. C. Wunsch, and Y. Yin, "Leader–follower output synchronization of linear heterogeneous systems with active leader using reinforcement learning," IEEE Trans. on Neural Networks and Learning Systems, Vol.29, No.6, pp. 2139-2153, 2018. https://doi.org/10.1109/TNNLS.2018.2803059

Name:
Yuma Uemura

Affiliation:
Division of Information and Electronic Engineering, Muroran Institute of Technology

Address:
27-1 Mizumoto-cho, Muroran, Hokkaido 050-8585, Japan

Brief Biographical History:
2022 Received B.S. degree from Information and Electronic Engineering, Muroran Institute of Technology
2022- Division of Information and Electronic Engineering, Muroran Institute of Technology

Main Works:
• Reinforcement learning.

Name:
Riku Narita

Affiliation:
Division of Information and Electronic Engineering, Muroran Institute of Technology

Address:
27-1 Mizumoto-cho, Muroran, Hokkaido 050-8585, Japan

Brief Biographical History:
2021 Received B.S. degree from Information and Electronic Engineering, Muroran Institute of Technology
2021- Division of Information and Electronic Engineering, Muroran Institute of Technology

Main Works:
• Reinforcement learning.

Name:
Kentarou Kurashige

ORCID:
0000-0001-9364-1269

Affiliation:
Muroran Institute of Technology

Address:
27-1 Mizumoto-cho, Muroran, Hokkaido 050-8585, Japan

Brief Biographical History:
2002 Received Ph.D. degree from Nagoya University
2002-2005 Research Associate, Fukuoka University
2005-2015 Research Associate, Muroran Institute of Technology
2015- Associate Professor, Muroran Institute of Technology

Main Works:
• Self-generation of reward in reinforcement learning, decision making under multi objective, and motivation model for machine learning.

Membership in Academic Societies:
• Institute of Electrical and Electronics Engineers (IEEE), Senior Member
• The Robotics Society of Japan (RSJ)
• The Japanese Society for Artificial Intelligence (JSAI)
• Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)