IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 May 2024, accepted 10 June 2024, date of publication 18 June 2024, date of current version 26 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3416179

RESEARCH ARTICLE

Mask-Attention A3C: Visual Explanation of Action–State Value in Deep Reinforcement Learning

HIDENORI ITAYA[1], (Graduate Student Member, IEEE), TSUBASA HIRAKAWA[2], (Member, IEEE), TAKAYOSHI YAMASHITA[2], (Member, IEEE), HIRONOBU FUJIYOSHI[3], (Member, IEEE), AND KOMEI SUGIURA[4], (Member, IEEE)

[1]Department of Computer Science, Chubu University, Kasugai-shi, Aichi 487-8501, Japan
[2]Center for Mathematical Science and Artificial Intelligence, Chubu University, Kasugai-shi, Aichi 487-8501, Japan
[3]Department of Robotics, Chubu University, Kasugai-shi, Aichi 487-8501, Japan
[4]Department of Computer Science, Keio University, Yokohama, Kanagawa 223-8522, Japan

Corresponding author: Hidenori Itaya (itaya@mprg.cs.chubu.ac.jp)

This work was supported by the New Energy and Industrial Technology Development Organization (NEDO) under Project JPNP20006.

ABSTRACT Deep reinforcement learning (DRL) can learn an agent's optimal behavior from the experience it gains through interacting with its environment. However, since the decision-making process of DRL agents is a black-box, it is difficult for users to understand the reasons for the agents' actions. To date, conventional visual explanation methods for DRL agents have focused only on the policy and not on the state value. In this work, we propose a DRL method called Mask-Attention A3C (Mask A3C) to analyze agents' decision-making by focusing on both the policy and value branches, which have different outputs. Inspired by the Actor-Critic method, our method introduces an Attention mechanism that applies mask processing to the feature map of the policy and value branches using mask-attention, which is a heat-map representation of the basis for judging the policy and state values. We also propose the introduction of a Mask-attention Loss to obtain highly interpretable mask-attention. By introducing this loss function, the agent learns not to gaze at regions that do not affect its decision-making. Our evaluations with Atari 2600 as a video game strategy task and robot manipulation as a robot control task showed that visualizing the mask-attention of an agent during its action selection facilitates the analysis of the agent's decision-making. We also investigated the effect of Mask-attention Loss and confirmed that it is useful for analyzing agents' decision-making. In addition, we showed that these mask-attentions are highly interpretable to the user by conducting a user survey on the prediction of the agent's behavior.

INDEX TERMS Deep reinforcement learning, explainable AI, visual explanation, video games, robot manipulation.

I. INTRODUCTION

The real world is a complex environment made up of many diverse factors. Deep reinforcement learning (DRL) is attracting interest as a technique for deriving optimal behavior for agents in this complex environment. DRL agents have achieved a high performance in various control tasks [1], [2], [3], [4], [5], [6], but DRL has a black-box problem in that it is very difficult for users to understand the agent's decision-making. There are two main components to this problem. First, it is difficult to know what kind of data was used for learning, that is, what kind of experience the agent had, because the data is collected through interactions between the agent and the environment. Second, the internal processing of the network to calculate the agent's behavior is complex and the basis for the behavior is unknown. The second issue is a particularly major obstacle when it comes to applying DRL agents to real-world environments. For example, if a DRL agent that can control an air conditioner to a comfortable room temperature suddenly raises the

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo.

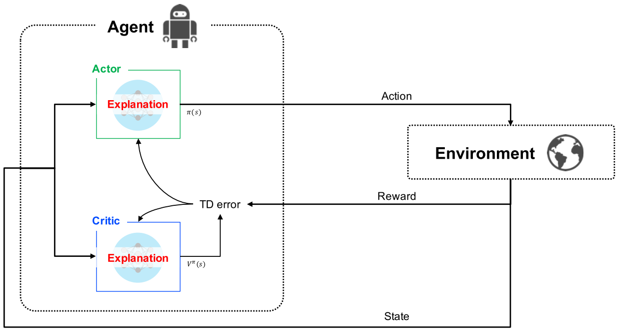


FIGURE 1. Overview of Mask-attention A3C.

temperature, users will feel uneasy if they do not know why the temperature was raised. As another example, if we want to learn how to play a game from a DRL agent that can obtain high scores, we cannot learn if we do not know the basis for its decision-making. Therefore, understanding an agent's decision-making is crucial for demonstrating the reliability of a DRL agent.

Visual explanation analyzes network output factors by using an attention map to highlight important regions in the input image. In DRL, research efforts are underway to introduce this visual explanation technique for enabling an understanding of DRL agents' decision-making. There are two main approaches in this regard. The first is to post-process trained DRL models to analyze their decisions. This approach utilizes the inputs/outputs, network weights, and gradient information of the learned DRL model to reveal its decision-making process. Therefore, the network structure of the DRL model is the same as before, but post-processing is required on the learned DRL model. Also, this method calculates an attention map, which indicates the factors affecting the output of the DRL model, but the calculation of the attention map is computationally expensive. The second approach makes the network structure of the DRL model an interpretable structure in advance. This method clarifies the decision-making of DRL models by designing the network structure as an interpretable structure during the construction phase of the DRL model. Compared to the first approach, it is less computationally expensive because it does not require any post-processing of the learned DRL model, and only forward propagation is used to compute the attention map. Therefore, research that analyzes agents' decision-making using methods with interpretable structures has become the mainstream.

When designing DRL models as interpretable structures, many visual explanation methods have been proposed that focus on and analyze the policy of the DRL models [7], [8], [9], [10], [11], [12], [13], [14]. This is because the policy is direct output values that represent the agent's decision-making. On the other hand, there are no studies dealing with agent analysis with a focus on state value. In DRL, the state value represents the expected value of the revenue per episode, and DRL based on the policy gradient method learns strategies to maximize this state value. In other words, the state value is as important as the policy in analyzing agents' decision-making. Therefore, we need to develop an explainable algorithm for DRL that focuses on the state value as well as the policy. Thus, the objects to be explained in our study are policy and state value in the agent model (reference Figure 1).

Toward an explainable algorithm for policy and state value, we focus on the Actor-Critic-based DRL algorithm [15], [16], [17], [18], [19], [20]. We propose Mask-Attention A3C (Mask A3C), a visual explanation method for deep reinforcement learning. Mask A3C visually analyzes the decision-making process of the agent by calculating and visualizing mask-attention, which is an attention map of policy and state value during the agent's action selection. The purpose of the Mask-attention Loss is to improve the interpretability of the mask-attention to the user by limiting the gazing area related to the agent's decision-making. Since the policy and state value in this method are learned by considering the mask-attention, the agent's performance can be improved.

This paper is a revised version of the research results presented in reference [21], with improved methods and additional experiments. In the previous report, the mask-attention which indicates the agent's decision-making was obtained, but this also indicates some areas that are irrelevant to the agent's output, and thus has low interpretability for the user. In response to this problem, this paper introduces mask-attention loss to obtain highly interpretable mask-attentions. We confirmed the effectiveness of our method in robot manipulation, which requires high control performance. Also, we evaluated the interpretability of mask-attention to the user by investigating the user's prediction of the agent's behavior using teaching with mask-attention. The experiments were extended accordingly.

The four contributions of this paper:

- We focus on the Actor-Critic method and propose a visual explanation method in DRL based on policy and state value. This enables the analysis of the agent model's decision-making process from two different perspectives.
- Our attention module, called mask-attention, which expresses the region of gazing at the output and is easily obtained by a forward pass. We also introduce Mask-attention Loss, which restricts gazing to regions that do not affect the agent's decision-making, to improve the interpretability of mask-attention.
- Through experiments using video games and robot manipulation, we clarify the decision-making process of the agent by utilizing mask-attention. The implementation of the attention mechanism considers mask-attention when outputting the agent's control values and improves the agent's performance by highlighting relevant regions.
- We evaluated the interpretability of mask-attention by investigating users' predictions of agent behavior

using mask-attention. Our study provided a highly interpretable map of the agent model's behavior to the user.

II. RELATED WORKS

We describe our research on the visual explanation of agent models in deep reinforcement learning, which is most relevant to our study. We also briefly describe the field of image recognition, where visual explanation of models has been actively studied.

A. VISUAL EXPLANATIONS FOR AGENT MODEL

Research attempting to analyze the decision-making of DRL agents can be classified into direct approaches that focus on the agent model and indirect approaches that focus on the states, *etc.*, observed by the agent. Indirect approaches are analysis methods based on state-space clustering, mask-based sensitivity analysis, *etc.* [8], [22], and [23]. On the other hand, the direct approach analyzes what part of the input information the agent model gazes at, thereby providing a clearer understanding of the agent's decision-making process. The direct approach is described below.

The policy of DRL agent model can be classified as Value-based, Policy-based, or Actor-Critic-based depending on the learning method. Value-based DRL utilizes a neural network to represent the action value function $Q(a|s;\theta)$ and updates the network parameter θ through TD learning to obtain the optimal policy [24], [25], [26]. Value-based DRL selects agent actions based on the action value function. Therefore, the analysis of agents' decision-making for action-value models has been addressed. Sorokin et al. introduced the attention mechanism to the action-value model [7]. They analyzed the action value function $Q(s, a;)$ by using deep recurrent Q-Network [27], a value-based DRL method using RNN, as an action value model and introducing the attention mechanism before LSTM. Zahavy et al. analyzed the state recognition of DRL agents based on the feature values of the action value model acquired through training [8]. They analyzed the agents' decision-making by clustering the state space using manually generated features. Zhang et al. guided the gazing area of the agent model by utilizing human gaze information while playing video games [9]. They trained a model that reproduces the human gaze in a heatmap by supervised learning and then augmented the input values of the action value model with this gaze information.

Policy-based DRL utilizes a neural network to directly represent the policy and learns the optimal policy $\pi(a|s;\theta)$ by updating the network parameters θ using the policy gradient method [17], [18], [28]. Policy-based DRL uses policy models to select agent actions, so the analysis of agents' decision-making is conducted on policy model. Manchin et al. visualized agent behavior as an attention map by implementing self-attention in a policy model [13]. They applied proximal policy optimization [18] as the policy model.

Actor-Critic-based DRL consists of Actor, which outputs policy $\pi(a|s;\theta)$, and Critic which outputs state value $V(s;\theta)$ [29]. The state value $V(s;\theta)$ is the expected value of the reward in the current state s and represents how good the current state is. In Actor-Critic-based DRL, the Actor selects and executes the current action according to the policy, the probability distribution of the action, and Critic evaluates Actor using the state value. Each network parameter is updated in parallel, with the Actor using the policy gradient method and the Critic using the TD error. Asynchronous Advantage Actor-Critic (A3C) [15] is a typical Actor-Critic-based DRL method. This method combines distributed DRL [30], [31], [32], asynchronous parameter updates in distributed learning, and advantage learning that considers rewards several steps ahead. A3C uses multiple environments to generate the training data (i.e., experiences) in parallel to obtain a high-performance agent in a short training time. Actor-Critic-based DRL uses an Actor model to select agent actions, so the analysis of agents' decision-making for the Actor model is underway. Greydanus et al. calculated perturbed images with a Gaussian filter applied using the gradient information of the agent model and obtained saliency maps from the perturbed images [10]. Since this method is a bottom-up approach, backpropagation is required to obtain the saliency map. Weitkamp et al. apply a bottom-up gazing area calculation method based on Grad-CAM [33], the visual explanation method in image recognition to the Actor model [11]. This method requires back-propagation to compute the attention map, similar to the Greydanus et al. method. Shi et al. emphasized task-related regions related to the agent's decision-making by generating a fine-grained attention mask in the agent model [14]. They focused on the actor model and showed agent decision-making by generating an attention mask to highlight task-related regions. Mott et al. obtain two Attentions ("what" and "where") by using query-based attention in the actor model [12]. This method requires a major change in the network architecture (key, value, *etc.*) because of the need to generate a query for the attention.

Actor-Critic-based DRL uses state values, the output of the Critic model, to learn policy. Therefore, not only the Actor but also the Critic are considered to contribute to the agent's decision-making. However, conventional methods focus only on the Actor model, i.e., policy, and do not consider the state value. In this study, we implement an attention mechanism for both Actor and Critic, and obtain a mask-attention that is an attention map for the policy and state values. We also propose Mask-attention Loss, which prevents agents from gazing at regions that do not affect their decisions. By introducing Mask-attention Loss into the learning process, we obtain a highly interpretable mask-attention for the agent's decision-making. By visualizing the mask-attention during inference, we can analyze the agent's decisions obtained from the learning process in terms of a visual explanation of the policy and state value.

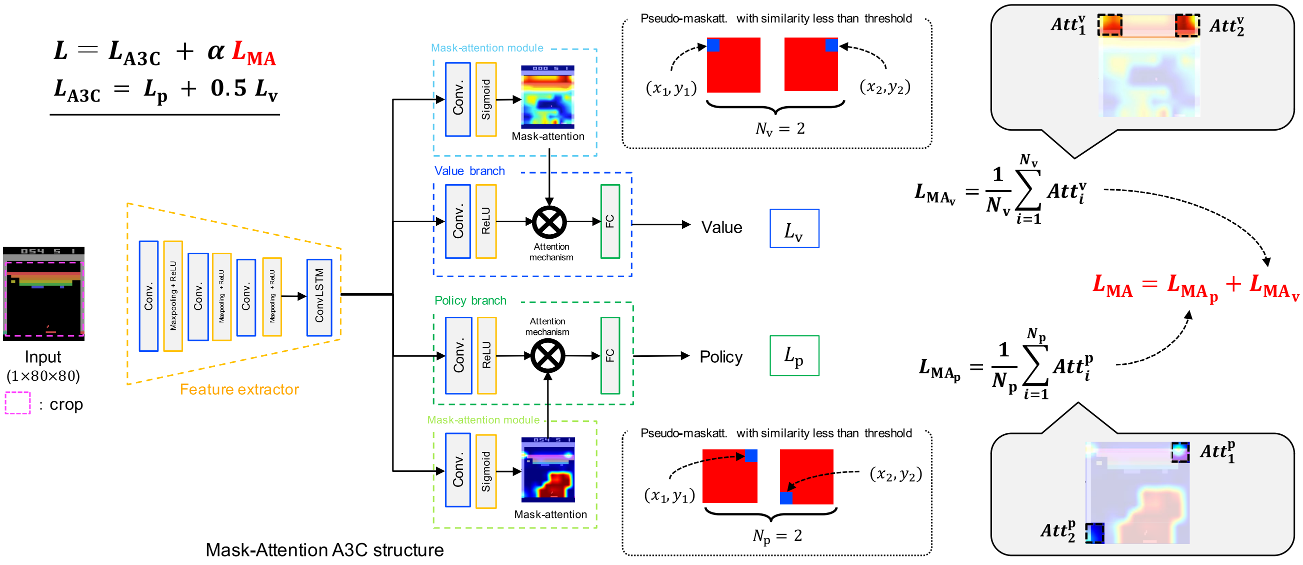


FIGURE 2. Overview of Mask-Attention A3C with Mask-attention Loss.

B. VISUAL EXPLANATIONS FOR IMAGE RECOGNITION MODEL

In the field of image recognition, the internal processing of recognition models is complex, and these models suffer from the black-box problem. To address this problem, several works have attempted to analyze the reasons for the judgments about the inference results of recognition models. Proposals in this vein include visual explanation methods based on feature maps of CNN that constitute recognition models [33], [34], [35], [36], [37], visual explanation methods based on perturbations to the input image [38], [39], [40], and visual explanation method that incorporates an explanatory mechanism into the model [41], [42]. Fukui et al. proposed the Attention Branch Network (ABN), which applies the attention map to the attention mechanism [43]. This method visually explains the reason for the model's decisions by using the attention map during training, and at the same time, improves the recognition accuracy. These works demonstrate that using an attention map in the inference process of a model can improve recognition accuracy in image recognition.

III. MASK-ATTENTION A3C

To clarify the basis for decision-making in agent models, we consider that two perspectives within the framework of deep reinforcement learning are important: policy and state value. Therefore, we focus on A3C, a typical distributed DRL method of the Actor-Critic, and propose Mask-Attention A3C (Mask A3C), which enables the interpretation of agent model decisions by incorporating an attention mechanism in Actor and Critic. Mask A3C introduces an attention mechanism to the policy branch (Actor) and the value branch (Critic). In this way, we obtain mask-attention, which is an attention map that shows the gazing area associated with the output of each branch. In addition, by introducing Mask-attention Loss, which restricts the agents not to gaze at unnecessary regions that do not affect their decision-making, we obtain the mask-attention that highly explains the agents' decision-making. Mask-attention Loss generates a pseudo mask-attention by introduces specific perturbations to the mask-attention, and identifies unwanted regions based on fluctuations in output values when using the pseudo mask-attention. By visually presenting the two mask-attentions, a detailed understanding of the agent's decision-making is achieved. Furthermore, by introducing the attention mechanism, this approach improves the agent's performance by considering mask-attention when learning to estimate policy and state value.

A. OVERVIEW OF MASK A3C STRUCTURE

The network structure of Mask A3C is shown in Figure 2 left. Mask A3C consists of a feature extractor, output branches (policy and value) with an attention mechanism. The details of each component of Mask A3C are described below.

1) FEATURE EXTRACTOR

This module calculates a feature map $F_{fe}(\mathbf{s}_t)$ from a given state $\mathbf{s}_t$ using a convolutional layer and recurrent neural network (RNN). The state at time t is defined as $\mathbf{s}_t$, where $\mathbf{s}_t$ in this task is an image. Mnih et al. reported that the use of LSTM in A3C can consider temporal information of the input state and significantly improve the agent's performance [15]. However, since LSTM cannot consider the spatial information of the input image, mask-attention cannot be computed when using LSTM in Mask A3C. Therefore, we use convolutional LSTM (ConvLSTM) [44] as RNN, which can consider spatio-temporal information. The extracted feature maps are input to the policy branch and the value branch, the mask-attention module.

2) MASK-ATTENTION MODULE

This module generates mask-attention for the policy and state value: $M_{\mathrm{v}}(\mathbf{s}_t)$ for the value branch and $M_{\mathrm{p}}(\mathbf{s}_t)$ for the policy branch. The mask-attention is generated by applying a sigmoid function to the feature map $F_{fe}(\mathbf{s}_t)$ with a convolution layer of $1 \times 1\times$ # of channels. This mask-attention is then input to each output branch.

3) OUTPUT BRANCHES WITH ATTENTION MECHANISM

The policy branch has the role of Actor and outputs policy, and the value branch has the role of Critic and outputs state value. The input to each branch is the feature map $F_{\mathrm{fe}}(\mathbf{s}_t)$ extracted by the feature extractor. Each branch receives $F_{\mathrm{fe}}(\mathbf{s}_t)$ and applies convolutional layers and ReLU to compute new mid-layer feature maps $F_v(\mathbf{s}_t)$ and $F_p(\mathbf{s}_t)$ respectively. These mid-layer feature maps and mask-attention are then used by the attention mechanism for each output branch. The attention mechanism performs mask processing using mask-attention on the mid-layer feature maps for each branch. This masking process can emphasize the regions that contribute to the optimal behavior and state value. The feature maps $F'_{\mathrm{v}}(\mathbf{s}_t)$ and $F'_{\mathrm{p}}(\mathbf{s}_t)$ of each branch after mask processing are calculated as follows

$$F'_{\mathrm{v}}(\mathbf{s}_t) = F_{\mathrm{v}}(\mathbf{s}_t) \cdot M_{\mathrm{v}}(\mathbf{s}_t), \tag{1}$$

$$F'_{\mathrm{p}}(\mathbf{s}_t) = F_{\mathrm{p}}(\mathbf{s}_t) \cdot M_{\mathrm{p}}(\mathbf{s}_t), \tag{2}$$

where $M(\mathbf{s}_t)$ is the mask-attention. By inputting the masked feature maps $F'_{\mathrm{p}}(\mathbf{s}_t)$ and $F'_{\mathrm{v}}(\mathbf{s}_t)$ to the output layer, policy and state value are obtained. Using the masked feature map, the agent focuses on the highlighted regions and selects the optimal action.

B. MASK-ATTENTION LOSS

We introduce Mask-attention Loss in training, which restricts the agent from gazing at unnecessary regions that do not affect its decision-making. This enables us to obtain a mask-attention that focuses only on regions that contribute to the policy and the state value. These unnecessary regions are identified from the variation of output values in each branch using pseudo mask-attention (pseudo-maskatt), where only one pixel is set to 0 and all others are set to 1. The size of the pseudo-maskatts are the same as that of mask-attention. Also, pseudo-maskatt is created for all positions by shifting the pixel positions with value 0 — that is, the number of pseudo-maskatts is the same as the size of the pseudo-maskatt. An example of the Mask-attention Loss calculation is shown in Figure 2 right.

The flow of calculating the mask-attention loss is as follows.

1) Calculate the mask-attention and the output value (policy or state value) at each branch by inputting the state observed from the environment.

TABLE 1. Experiment details of video game strategy task.

Comparison methods	A3C, Mask A3C, Policy Mask A3C, Value Mask A3C, and Mask A3C MaskattL	
Training conditions	number of worker	30
	optimizer	Adam
	global steps	1.0×10^8
	learning rate	0.0001
	discount rate	0.99
	termination condition of an episode	reached 1.0×10^4 step or the end of 1 game
	skip frame	4
	threshold value for select pseudo-maskatt	0.1
	start step of Mask-attention Loss	0.8×10^8

2) Calculate the policy and state value when pseudo-maskatt is used for the attention weight of the attention mechanism. Here, the input values are the same as in 1).
3) Calculate the degree of difference is calculated between the output values (policy and state value) of 1) and 2). The difference degree of the policy is calculated using Kullback-Leibler (KL) divergence, as the policy is probability-distributed. As for the difference degree of the state value, it is calculated using the L1 norm. Each degree of difference $\mathrm{Dif}_{\mathrm{p}}$, $\mathrm{Dif}_{\mathrm{v}}$ is calculated as

$$\mathrm{Dif}_{\mathrm{p}} = \left|\mathrm{KL}\left(\pi(\mathbf{s}_t, M_{\mathrm{p}}(\mathbf{s}_t)) \parallel \pi(\mathbf{s}_t, M_{\mathrm{pseudo}})\right)\right|_1, \tag{3}$$

$$\mathrm{Dif}_{\mathrm{v}} = |V(\mathbf{s}_t, M_{\mathrm{v}}(\mathbf{s}_t)) - V(\mathbf{s}_t, M_{\mathrm{pseudo}})|_1, \tag{4}$$

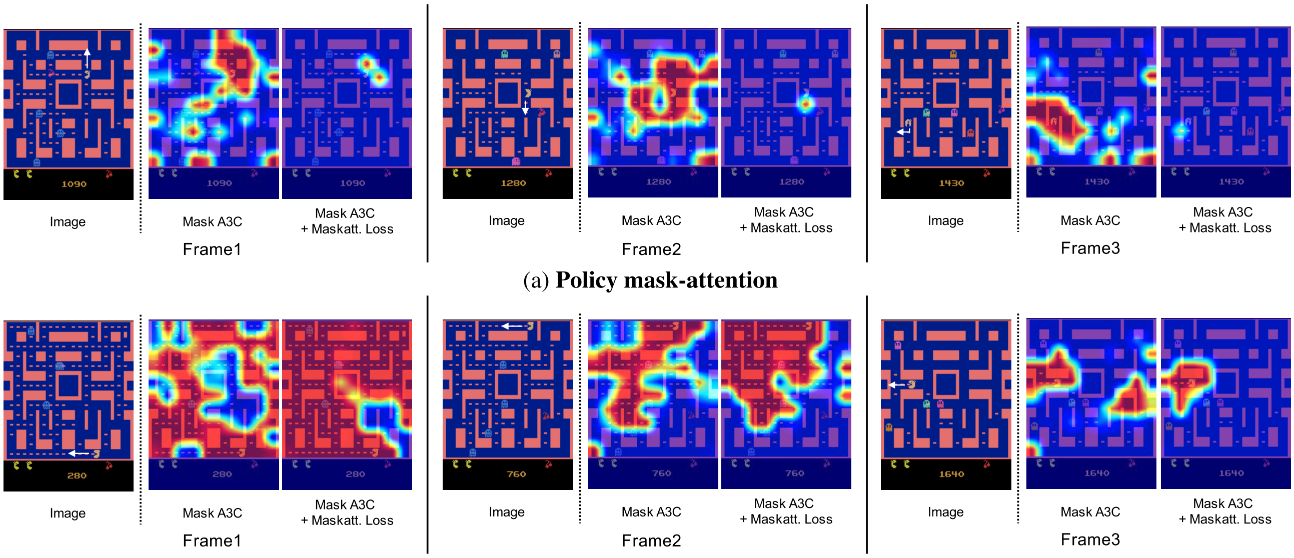
where $\mathbf{s}_t$ is the input state at time t, $M(\mathbf{s}_t)$ is the mask-attention, $\pi(\mathbf{s}_t, \cdot)$ is the policy, $V(\mathbf{s}_t, \cdot)$ is the state value, and $\mathrm{KL}(\cdot \parallel \cdot)$ is the KL divergence. The lower the value of Dif, the less the output value changes between mask-attention and pseudo-maskatt. That is, when Dif is low, the region corresponding to the pixel value 0 in pseudo-maskatt is an unnecessary region that does not affect the agent's decision-making.
4) Select pseudo-maskatt with low degree of difference by thresholding.
5) Calculate the mask-attention loss that restricts the DRL model from gazing at the attention weight of the mask-attention corresponding to the pixel value 0 of the pseudo-maskatt selected in 4). Mask-attention Loss L_{MA} is calculated as follows:

$$L_{\mathrm{MA}} = L_{\mathrm{MA_p}} + L_{\mathrm{MA_v}}, \tag{5}$$

$$L_{\mathrm{MA_p}} = \frac{1}{N_{\mathrm{p}}} \sum_{i=0}^{N_{\mathrm{p}}} Att_i^{\mathrm{p}}, \tag{6}$$

$$L_{\mathrm{MA_v}} = \frac{1}{N_{\mathrm{v}}} \sum_{i=0}^{N_{\mathrm{v}}} Att_i^{\mathrm{v}}, \tag{7}$$

where $Att^{\mathrm{p}}, Att^{\mathrm{v}}$ are the attention weights of the mask-attention corresponding to the pixel value 0 of the pseudo-maskatt for the policy branch and the value branch. Also, $N_{\mathrm{p}}, N_{\mathrm{v}}$ is the number of $Att^{\mathrm{p}}, Att^{\mathrm{v}}$.



(a) **Policy mask-attention**

(b) **Value mask-attention**

FIGURE 3. Comparison of mask-attention by Mask-attention Loss: The arrow shows the direction of travel of Pac-Man.

6) The other loss functions are the same as those of A3C. Therefore, add the Mask-attention Loss calculated in 5) to the A3C loss function during training. When adding Mask-attention Loss, the learning rate α is multiplied and the scale of the loss is adjusted. This learning rate is a hyperparameter.

By introducing this loss into the training process, Mask-attention approximates the value of 0 for regions that don't contribute to the output value. In other words, Mask-attention Loss has the effect of making mask-attention a map that shows only the regions that contribute to the output. This improves the interpretability of mask-attention and promotes user understanding of the agent's behavior.

IV. EXPERIMENTS

In this section, we describe experiments using a video game strategy task to evaluate the effectiveness of Mask A3C and Mask-attention Loss. In addition, we confirm that the effects in these video games are equally effective in robot manipulation tasks, where deep reinforcement learning has been successful. To confirm that mask-attention is highly interpretable to users, we conduct a questionnaire investigation of users' predictions of the agent's behavior. Sec. IV-A - IV-G describe experiments with a video game strategy task, and Sec. IV-H describes a robot manipulation task. Sec. IV-I describes a questionnaire experiment about the interpretability of mask-attention for users to understand the agent's decision-making.

A. EXPERIMENTAL DETAILS OF THE VIDEO GAME STRATEGY TASK

Experiments were conducted using the OpenAI gym game tasks [45] to evaluate the explanation of the agent's decision-making and the effectiveness of the performance improvement by Mask A3C. Six video games were used: Breakout (BO), Ms. Pac-Man (MP), Seaquest (SQ), Space Invaders (SI), Beamrider (BR), and Fishing Derby (FD).

Table 1 shows the details of our experiments. In comparison methods, Policy Mask A3C and Value Mask A3C refer to a Mask A3C in which the attention mechanism is implemented on one side of the branch (i.e., policy branch or value branch). Mask A3C MaskattL refers to a Mask A3C with mask-attention loss. In training conditions, Mask-attention Loss in Mask A3C MaskattL was introduced after 0.8×10^8. In other words, the same loss function as that of A3C and Mask A3C was used for learning until the number of global steps was 0.8×10^8. In the early training phase, the agent model is unclear about which regions contribute to the output. Therefore, we introduced Mask-attention Loss in the late training phase to facilitate learning that excludes unnecessary gazing areas after the agent model's gazing areas have been clarified. The threshold value for extracting pseudo-maskatts with a low degree of difference was set to 0.1 in the Mask-attention Loss calculation.

We utilized the following five evaluation metrics.

- Effect of Mask-attention Loss on mask-attention
- Visualization comparison of gazing areas using previous studies
- Analysis of agent's decision-making process through visualization of mask-attention
- Comparison on Atari 2600 game scores
- Comparison by score reduction in inverted mask-attention
- Agent reaction to the new state from mask-attention's point of view

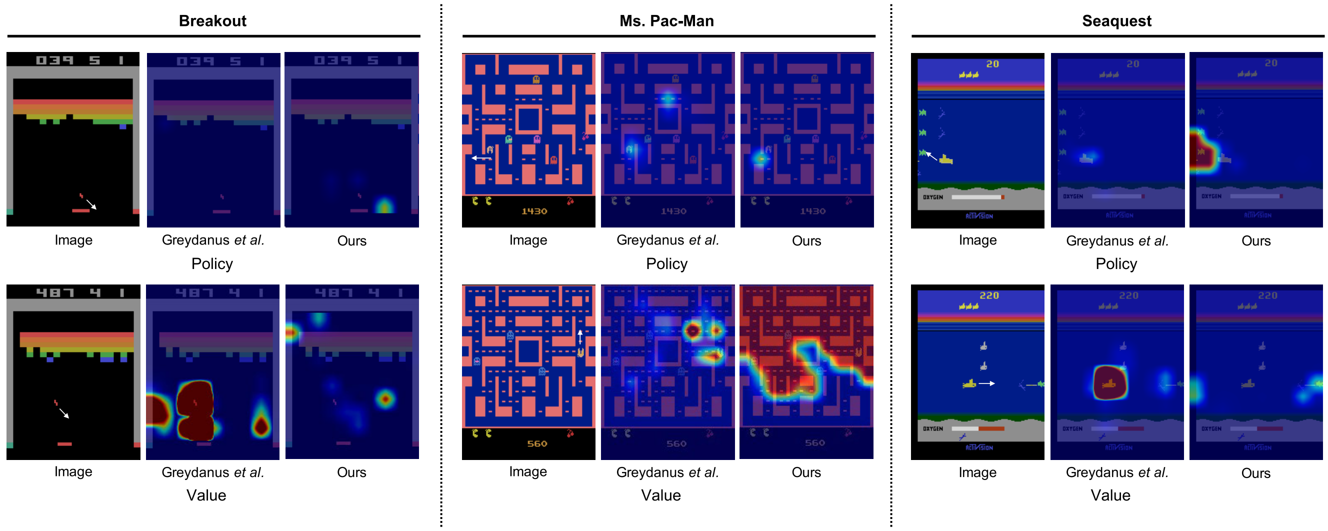


FIGURE 4. Visualization comparison of gazing areas with previous studies: Ours is a visualization of mask-attention with Mask A3C MaskattL.

1) DETAILS OF AGENT MODEL IMPLEMENTATION

We describe the details of the agent model used in the video game experiment. The input and output are a grayscale image of the game screen and control commands for the current state of each game. The input grayscale image is resized to 80 × 80. The feature extractor is constructed with three convolutional layers (maxpooling and ReLU are applied after the convolution process) and ConvLSTM. The convolutional layers are two layers with 32 output dimensions and one layer with 64 output dimensions. Also, the output dimension of the hidden state of ConvLSTM is 64. The policy branch consists of one convolutional layer (ReLU is applied after convolution process), one fully connected layer (FC), and a softmax function. Here, layers are a convolutional layer with 32 output dimensions and a FC with the same number of output units as the number of agent actions. The value branch consists of one convolutional layer (ReLU is applied after the convolution process) with 32 output dimensions and one FC with one output unit. The A3C in this experiment has the same as the network structure as Mask A3C without the attention mechanism.

B. EFFECT OF MASK-ATTENTION LOSS ON MASK-ATTENTION

Figure 3 shows a visualization example of mask-attention in the same frame in MP for the Mask A3C and the Mask A3C MaskattL. In (a), it is evident that Mask A3C MaskattL focuses more on Pac-Man's direction of movement compared to Mask A3C. As the task in MP is to control which direction the agent moves, we can conclude that Mask A3C MaskattL is strongly gazing at Pac-Man's direction of travel. In (b), we can see that Mask A3C and Mask A3C MaskattL are both focused on the cookie, which is the source of the score. Mask A3C MaskattL shows broader coverage of the remaining cookies on the screen compared to Mask A3C.

TABLE 2. Calculation time of gazing area per frame and model size in Breakout: A3C shows the inference time for calculating policy and state values. NVIDIA RTX A6000 was used for the measurements.

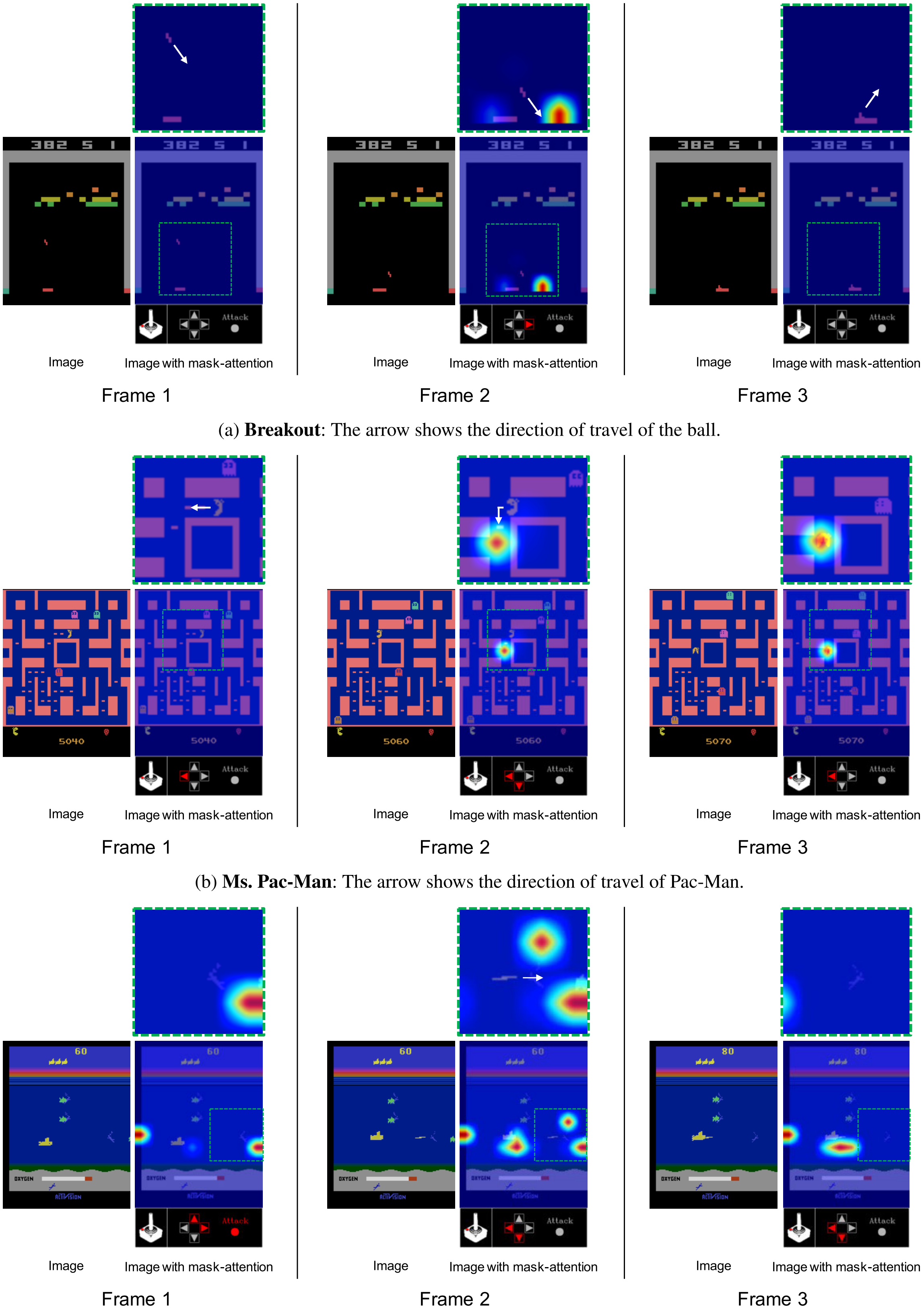
method	calculate time [second]	model parameters
A3C [15]	7.46×10^{-4}	343.909k
Greydanus *et al.* [10]	3.18×10^{-1}	343.909k
Ours	9.31×10^{-4}	344.039k

As these results show, we can confirm that the introduction of Mask-attention Loss limits the gazing area for both the policy and state value. In addition, by limiting the gazing area, the user can obtain the mask-attention that indicates only a specific area. Therefore, we can obtain the mask-attentions that are highly interpretable to the agent's decision-making for the user.

C. VISUALIZATION COMPARISON OF GAZING AREAS USING PREVIOUS STUDIES

We confirm that the mask-attention calculated by our method is useful by comparing visual explanations for the agent model using previous and ours studies. Also, we confirm that our method can calculate the gazing area at low cost by verifying the calculate time of the gazing area and the number of parameters of the agent model. Here, we use the perturbation-based Greydanus et al. method [10] as a previous study of visual explanation methods for agent models. Greydanus et al.'s method is a saliency calculation method to generate saliency maps for interpreting agent's action selection, and generates perturbed images by perturbing specific pixels in the input image. The saliency map is calculated from the importance of the perturbed pixels based on the fluctuation of the output values when these perturbed images are used as input to the agent model.

Figure 4 shows a visualization of the gazing area for previouss and ours studies. The Greydanus et al. method is a

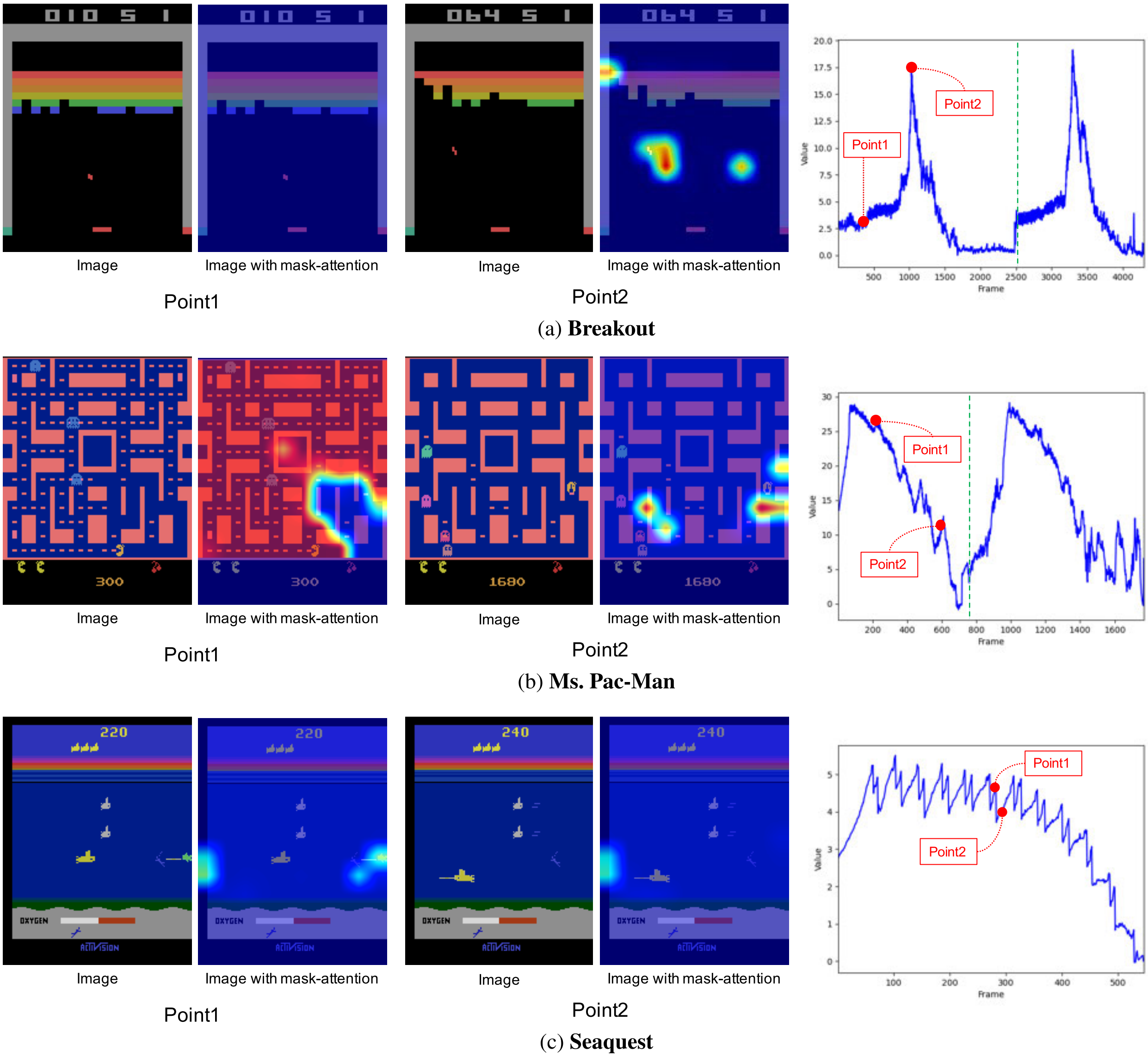


(a) **Breakout**: The arrow shows the direction of travel of the ball.

(b) **Ms. Pac-Man**: The arrow shows the direction of travel of Pac-Man.

(c) **Seaquest**: The arrow in Frame 2 represents a torpedo, which is attacking a submarine.

FIGURE 5. Visualization example of mask-attention in policy branch: This is a visualization of mask-attention with Mask A3C MaskattL. The controller of "Image with mask-attention" is the action chosen by the DRL agent in the current state.



(a) **Breakout**

(b) **Ms. Pac-Man**

(c) **Seaquest**

FIGURE 6. Visualization example of mask-attention in value branch: This is a visualization of mask-attention with Mask A3C MaskattL. Examples of mask-attention at two points where the state value changes significantly are shown. The graph shows the transition of state value, where the dashed line in the graph indicates the transition to the next stage in each game.

map that focuses on characteristic objects such as balls, pac-man, and submarines (Breakout value, Ms. Pac-Man policy, Seaquest value, *etc.*). On the other hand, Ours is a map that focuses on objects and areas affected by the agent's behavior, such as the ball, pac-man's direction of movement, and fish that are the target of the submarine's attack (Breakout policy, Ms. Pac-Man policy, *etc.*). Furthermore, compared to Ours, Greydanus et al.'s approach sometimes shows maps with no gazing area (Breakout policy) or with gazing in irrelevant areas to the agent's behavior (Ms. Pac-Man policy).

Table 2 shows the calculation time of the gazing area and the number of parameters of the agent model for each method. Here, A3C is the agent model without visual explanation method and is the standard value for each method. Ours calculates mask-attention at the same time as inferring the policy and state value, so the mask-attention module increases the calculate time and model parameter. However, the Mask-attention module has a simple structure that consists of a Conv. 1 × 1 and a Sigmoid function, which allows for real-time inference of policy and state values, and gazing area calculation. On the other hand, Greydanus et al. method is not an approach to the agent model structure, so the model parameter is the same value compared to A3C. However, this method calculates the gazing area using the perturbation image, which significantly increases the calculate time.

From these results, it can be seen that our method can visualize the gazing areas that indicate areas related to the agent's behavior by introducing a simple structure called mask-attention module, which significantly reduces

TABLE 3. Max and mean scores between 100 episodes in Atari 2600 games: Scores of models with the highest mean score among five trials in each method are shown. Bold text indicates the highest score of the max / mean score in each game.

Att. mech.		Maskatt.	BO		MP		SQ		SI		BR		FD	
Policy	Value	loss	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean
			864	**662.0**	5380	4573.3	2760	2728.2	19505	18531.8	34748	28341.1	41	32.1
✓			**864**	595.8	6330	4833.8	**2820**	2784.0	19860	19102.8	32604	**28495.3**	41	37.5
	✓		**864**	606.9	4830	4044.5	**2820**	**2786.4**	19675	18537.8	**35108**	28205.7	43	36.1
✓	✓		**864**	650.5	**9750**	**6716.3**	2780	2748.6	20510	**19508.1**	31440	26113.1	47	**39.3**
✓	✓	✓	**864**	573.7	7140	5681.9	1860	1835.2	**20700**	19099.3	31200	24685.1	**63**	**39.3**

the calculation time of the gazing areas compared to the Greydanus et al. method.

D. ANALYSIS OF AGENT'S DECISION-MAKING PROCESS THROUGH VISUALIZATION OF MASK-ATTENTION

Figures 5 and 6 show visualization examples of the mask-attention with Mask A3C MaskattL for BO, MP, and SQ. The following is an explanation of the mask-attention shown in these figures.

1) BREAKOUT (BO)

BO is a game in which the player controls a paddle to hit the ball back and destroy blocks. The agent (i.e., the paddle) has three actions: No-op, Left, and Right. Figure 5a shows the mask-attention for the policy in BO. In Frame 1, the ball is moving toward the right side of the paddle. In Frame 2, when the ball is approaching the paddle, the DRL model infers Right by gazing in the direction of the ball. In Frame 3, the paddle moves to the region gazed at in Frame 2 and returns the ball. Hence, we see that the agent controls the paddle in accordance with the direction of the ball's travel. Figure 6a shows the mask-attention for the state value in BO. At Point 1, the agent shows no gazing area, correlating with a low state value on the graph. The agent at Point 2 gazes at the left end of the block, and the graph shows that its state value is high. Note that at Point 2, the left end of the block has fewer blocks. In BO, getting the ball to the space above the blocks is one of the major factors to obtain a high score. From these results, we can see that the value branch recognizes the importance of getting the ball to the space above the blocks.

2) MS. PAC-MAN (MP)

MP is a game that controls the player to collect scattered cookies while avoiding enemies. The agent (i.e., Ms. Pac-Man) has nine actions: No-op, Up, Down, Left, Right, Up+Left, Up+Right, Down+Left, and Down+Right. Figure 6a shows the mask-attention for the policy in MP. In Frame 1, the agent selects Left and Pac-Man moves to the left. In Frame 2, Pac-Man reaches a crossroads and the agent selects Down+Left to gaze at the cookie below Pac-Man. In Frame 3, Pac-Man moves to the point where the agent was gazing at in frame 2 and acquires the cookie. Thus, the agent is controlling Pac-Man to move toward the cookie. Figure 6b shows the mask-attention for the state value in MP. At Point 1, the agent gazes at the whole screen because the game is just starting. In contrast, the gazing area shrinks at Point 2 as the number of cookies decreases. In addition, from Point 1 to Point 2, the state value also decreases as the number of cookies on the screen decreases. These results indicate that the agents recognize the cookies as a scoring source.

3) SEAQUEST (SQ)

SQ is a game in which the player controls a submarine to rescue divers while destroying enemy submarines and fish. The agent (i.e., the submarine) has six actions: No-op, Up, Down, Left, Right, and Attack. Figure 5c shows the mask-attention for the policy in SQ. In Frame 1, the agent is gazing at the fish that appeared from the right side of the screen and selects Attack. In Frame 2, the agent's beam is directed toward the fish that the agent was gazing at in Frame 1. In Frame 3, the agent is no longer gazing at the fish and defeats it. These results indicate that the agent recognizes the fish as soon as it appears and controls the submarine to destroy it. Figure 6c shows the mask-attention for the state value in SQ. At Point 1 (just before destroying the fish), the agent is gazing at the fish. At Point 2 (the state value has decreased), the fish gazed at in Point 1 has been destroyed and the gazing area for the fish has disappeared. These results indicate that the agent recognizes that destroying fish is an important factor in SQ.

4) DISCUSSION

We confirmed that two different mask-attentions (policy and state value) can be obtained by implementing an attention mechanism in the output branch of the Actor-Critic based DRL method. We confirmed that the mask-attention of policy directly indicates the region that contributes to the action selected by the agent. This is because policy represents the probability distribution of the agents' possible actions in the current state. On the other hand, we confirmed that the mask-attention of the state value indicates the region that represents the main characteristics of the game. This is because the state value represents the expected value of return in the current state. Here, the return is the sum of the rewards in the episode. By using mask-attention from these two viewpoints, we have shown that it is possible to clarify the agent's decision-making.

E. COMPARISON ON ATARI 2600 GAME SCORES

The max and mean scores between evaluation of 100 episodes for each comparison methods in Atari 2600 games are shown in the Table 3. In BO, we see that A3C has the highest score in terms of mean score, but in terms of max score, Mask A3C

TABLE 4. Game scores with and without inverting the gaze area of the policy mask-attention: The checkmark for the Inverse att. indicates whether the gazing area of mask-attention is inverted or not. The random indicates the score when the action is randomly selected.

Att. mech.		Maskatt.	Inverse	BO		MP		SQ		SI		BR		FD	
Policy	Value	loss	att.	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean
✓				864	595.8	6630	4833.8	2820	2784.0	19860	19102.8	32604	28495.3	41	37.5
			✓	4	2.2	290	268.9	280	158.2	805	306.9	4996	1554.2	−49	−75.7
✓	✓			864	650.5	9750	6716.3	2780	2748.6	20510	19508.1	31440	26113.1	47	39.3
			✓	18	5.2	810	446.0	220	160.6	535	311.4	2216	1156.8	−57	−84.4
✓	✓	✓		864	573.7	7140	5681.9	1860	1835.2	20700	19099.3	31200	24685.1	63	39.3
			✓	2	0.9	1080	671.5	520	222.0	1150	488.6	660	232.8	−43	−63.4
random				5	1.2	1080	247.8	300	82.8	460	142.1	852	356.5	−85	−93.1

has the highest score (864) possible in BO for all methods. Since BO is a simple game without external factors (only balls and blocks are affected), Mask A3C is not likely to score significantly lower than A3C. In MP, SI, and FD, it scored higher than A3C by introducing the mask-attention in the policy branch. This is because the mask-attention emphasizes objects that contribute to the agent's action selection (e.g. cookies and enemies in MP, defensive walls and invaders in SI, fishes closest to the player in FD). BR has enemies that can be defeated to score points and enemies that should be avoided, and these enemies are quite similar in appearance. Emphasis on the target by mask-attention alone is not sufficient to capture the differences in the detailed characteristics of the enemies. Therefore, we consider that there was no significant difference in the scores for any of the methods. SQ's score varied greatly depending on whether the agent was able to acquire the actions to attack the fish or to replenish oxygen. As shown in the mask-attention visualization example in Sec. IV-D, the agent is gazing at the fish and attacking them, but is not gazing at the oxygen gauge at the bottom of the screen. Since mask-attention alone is not enough to make the agent gaze at the oxygen gauge, there was no significant difference in the scores for any of the methods. The mean score of the methods with Mask-attention Loss decreased compared to Mask A3C (BO, MP, SQ, BR). On the other hand, it can improved or achieved the same score for SI and FD compared to A3C. From these results, we consider that although the Mask-attention Loss limits the gazing area and makes it difficult to consider the surrounding information of the gazing target, the mask-attention loss improves the score compared to A3C and achieves a higher interpretability than the Mask A3C.

F. COMPARISON BY SCORE REDUCTION IN INVERTED MASK-ATTENTION

We investigate whether mask-attention indicates the gazing areas of an agent's decision-making process. In this experiment, we focus on the mask-attention of the policy branch to confirm its contribution to the agent's action selection. We assumed that mask-attention indicates the reasoning behind the agent's decision-making process, and evaluated the game score by inverting the mask-attention gazing areas (i.e., making regions with high attention values low and those with low values high). If the game score obtained by the agent does not change when mask-attention is inverted (indicating no effect of mask-attention on the agent's action selection), then the mask-attention is not considered to contribute to the agent's action selection. In contrast, if the game score decreases (indicating mask-attention does affect the agent's action selection), mask-attention is considered to have a significant contribution to the agent's action selection. In this method, the mask-attention of the policy branch in the trained agent model is inverted to create a map, and the attention mechanism uses this map to select actions. By comparing scores with and without mask-attention inversion, we confirm whether the mask-attention is effective as a visual explanation for the agent's decision-making. The mask-attention $M(\cdot)$ gazing area inverted maps $M_{\text{inverse}}(\cdot)$ were created as follows.

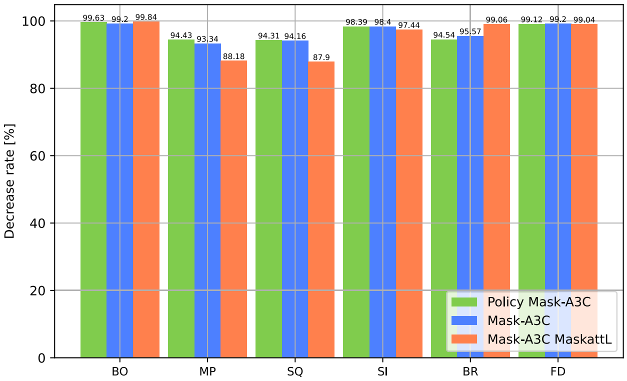
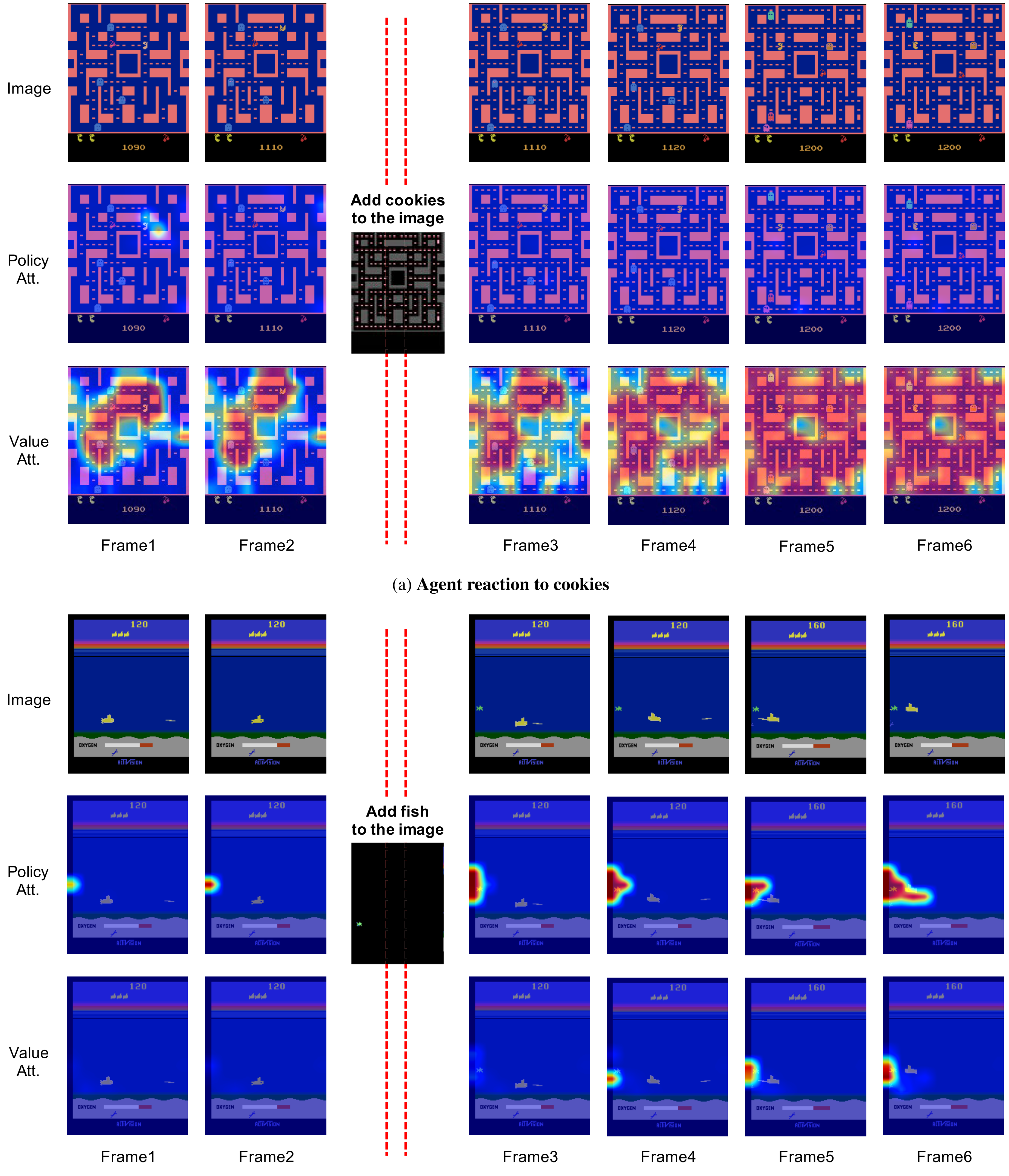


FIGURE 7. Decrease rate of mean score due to inverse policy mask-attention: The mean score = average scores between 100 episodes.

$$M_{\text{inverse}}(\mathbf{s}_t) = 1 - M(\mathbf{s}_t), \tag{8}$$

where $\mathbf{s}_t$ is the state (grayscale image of the game screen).

Table 4 shows a comparison of scores with and without mask-attention inversion, while Figure 7 shows the decrease rate of scores due to the mask-attention inversion. As shown in Table 4, in all games, the score is significantly lower when mask-attention is inverted. Especially in BO and BR, the scores after the mask-attention inversion are lower than random in models with Mask-attention Loss. In MP, SQ, SI, and FD, as shown in Figure 7, the mean score decreased by more than 85%, similar to the other games. Therefore, we can confirm that the mask-attention inversion significantly reduces the score. From this result, we can



(a) **Agent reaction to cookies**

(b) **Agent reaction to fishes**

FIGURE 8. Agent's reaction to new states with mask-attention visualization: This is a visualization of mask-attention with Mask A3C MaskattL.

conclude that the mask-attention gazing areas in the policy branch are the areas that contribute to the agent's action selection for achieving a high score. Additionally, since the decrease rates of Mask A3C and Mask A3C MaskattL are

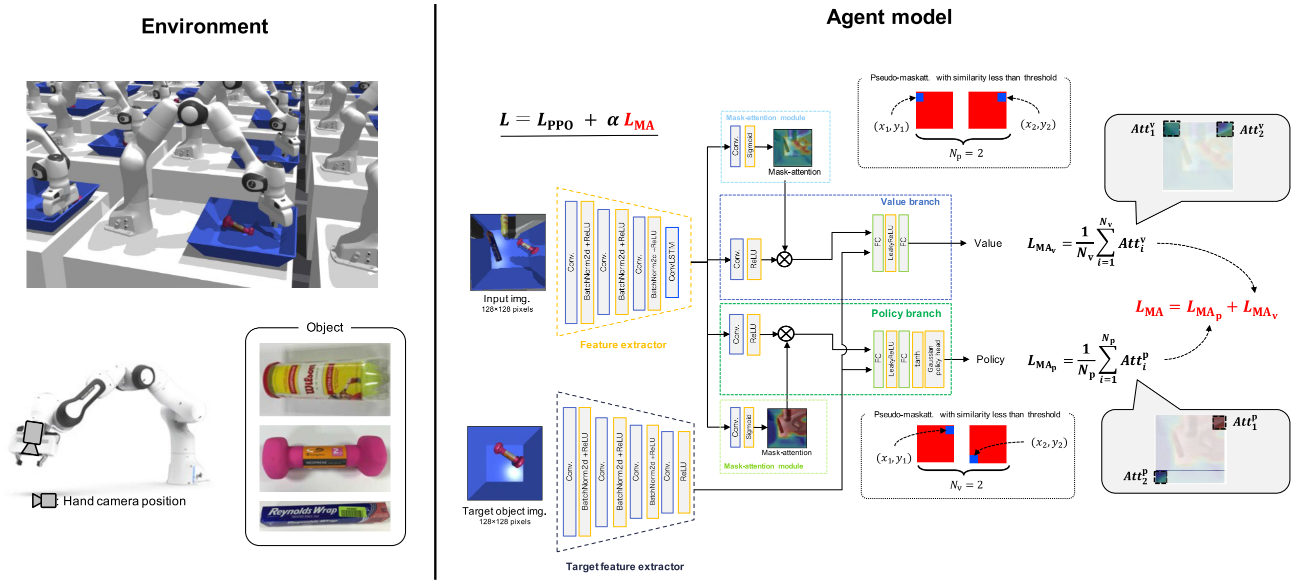


FIGURE 9. Overview of our methods in robot manipulation.

similar, we can conclude that the gazing areas limited by Mask-attention Loss are effective for analyzing the agent's decision-making. Thus, by limiting the gazing area with Mask-attention Loss, the agent's decision-making is more clearly indicated by the mask-attention.

G. AGENT REACTION TO THE NEW STATES FROM MASK-ATTENTION'S POINT OF VIEW

We investigated the effects on mask-attention and the agent's decision-making when there is an unexpected change in the agent's gazing object described in Sec. IV-D. If a change in the gazing object significantly affects the agent's behavior and mask-attention, the object is considered to be an important factor contributing to the agent, and the agent's gazing object expressed by the mask-attention is considered to be correct. In this experiment, cookies (MP) and fishes (SQ) were used as targets. The agent model was Mask A3C MaskattL with Mask-attention Loss used for visualization in Sec.IV-D. As an experimental method, cookies (in the case of MP) or fish (in the case of SQ) were added to the input image to the agent model at unexpected frames during the evaluation. In MP, the frame to add a cookie is when the agent finishes acquiring half of the cookies on the screen. In SQ, the frame to add the fish is the frame in which the fish does not exist. After adding each object, we investigated changes in the agent's behavior and mask-attention.

Figure 8a shows the change in agent behavior and mask-attention after adding cookies in MP. As seen from the value mask-attention, the frame before the cookies is added, the agent is gazing at the remaining cookies on the screen. In contrast, after Frame 2, the agent attends to all cookies, including the added ones. These results indicate that MP's cookies are objects that contribute significantly to the agent's behavior.

TABLE 5. Experiment details of robot manipulation task.

Comparison methods	PPO, Mask PPO, Mask PPO MaskattL	
Training conditions	optimizer	Adam
	training steps	2.5×10^7
	learning rate	0.0002
	discount rate	0.99
	termination condition of an episode	12 step
	threshold value for select pseudo-maskatt	0.01
	start step of Mask-attention Loss	2.0×10^7

Figure 8b shows the change in the agent's behavior and mask-attention when a fish is added in SQ. As can be seen, the policy mask-attention in Frames 1 and 2 indicates that the agent is gazing at the left side of the screen where the fish appears, and the policy mask-attention after Frame 2 indicates that agent is strongly gazing at the fish. The value mask-attention in Frames 1 and 2 indicates that there is no gazing area, and after Frame 2, the agent is strongly focused on the added fish. These results indicate that the fish in SQ are objects that contribute significantly to the agents' behavior.

In SQ, the policy mask-attention is strongly focused on the fish in Frame 3, the frame immediately after the fish is added. In other words, the policy mask-attention is considered to be immediately affected by the addition of the fish. In contrast, the value mask-attention in SQ does not pay attention to the fish in Frame 3, but then gradually begins to do so. Similarly, MP's mask-attention does not pay attention to all cookies in Frame 3 (the frame immediately after the cookie is added), and then gradually pays more attention to the cookies. These results indicate that the policy mask-attention and the value mask-attention have different effects on mask-attention when the gazing object

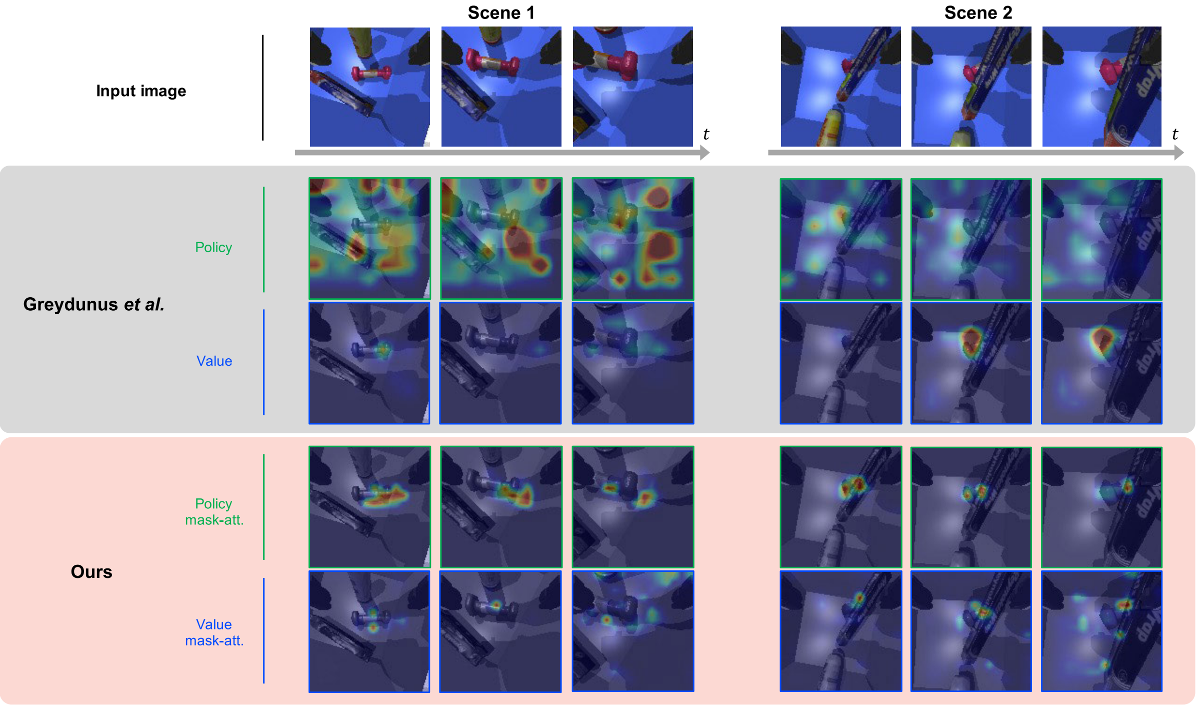


FIGURE 10. Visualization example of mask-attention in robot manipulation: Ours is a visualization of mask-attention with Mask PPO MaskattL.

changes. In other words, these mask-attentions have different knowledge about the agent's decision-making. The effect of policy mask-attention on the change of the gazing object was immediate, whereas the effect of value mask-attention was several frames later. Therefore, we conclude that policy mask-attention indicates regions that contribute to the agent's current behavior, and value mask-attention indicates regions that are related to game characteristics considering time series information.

H. APPLICATION TO ROBOT MANIPULATION TASKS

In this section, we describe experiments with robot manipulation tasks to evaluate the effectiveness of our method.

1) EXPERIMENTAL DETAILS OF THE ROBOT MANIPULATION TASK

Figure 9 shows an overview of the robot manipulation task and agent model. This experiment targets robot manipulation using Isaac-gym [46], a physics simulation environment developed by NVIDIA for reinforcement learning research. This task is to grasp a target object from among multiple objects using Franka Emika's Panda, a single-armed robot. The objects are dumbbells, plastic wrap, and a tennis ball case from the ARC2017 RGBD Dataset [47]. These objects are randomly positioned in the tray for each episode. The target grasping objects in this task are dumbbells. The control target (agent) is a Panda end-effector, and a camera is placed in front of the end-effector, with the hand viewpoint image as input information for the agent. Here, the end-effector automatically lowers as soon as the episode starts and reaches the tray in 12 steps. After 12 steps, the end-effector is automatically closed, and then it is raised. In other words, the goal of this task is to control the end-effector to grasp dumbbells within 12 steps. The actions that can be selected by the agent model are to move the end-effector forward or backward with respect to the tray, to move left or right, and to rotate the end-effector forward or backward; these actions are continuous values. These actions are performed only during the descent of the end-effector. The reward is $+1$ only when the dumbbells, the target grasping objects, are grasped and lifted above a certain height. The training of the agent model utilizes Proximal Policy Optimization (PPO) [18], an Actor-Critic based DRL algorithm that has achieved high performance in the robot control task.

Table 5 shows the details of this experiment. Here, the comparative method Mask PPO refers to the proposed method that introduces the mask-attention module in the structure described below, and Mask PPO MaskattL means Mask PPO with Mask-attention Loss. As shown in the training conditions, the Mask-attention Loss of Mask PPO MaskattL was introduced after 2.0×10^7. In other words, the same loss function as in PPO and Mask PPO was used until the training step reached 2.0×10^7. The reason for introducing Mask-attention Loss from the late training phase is the same as for the video game task (see Sec. IV-A). The threshold for selecting pseudo-maskatts for the Mask-attention Loss calculation was set to 0.01. We utilized the following two evaluation metrics.

TABLE 6. Average grasping success rate during 1,000 episodes in robotic manipulation.

Mask-att. module	Mask-att. Loss	grasping rate [%]
		54.58
✓		55.91
✓	✓	56.18

- Visual explanation of the agent model with mask-attention
- Evaluation by grasping success rate of target object

a: DETAILS OF AGENT MODEL IMPLEMENTATION

We describe the details of the agent model used in our robot manipulation task experiments. The inputs are the hand viewpoint image and the target grasping object image from the camera attached to the end-effector, and the outputs are the control values of the end-effector. The input RGB images of the hand viewpoint and the target grasping object are resized to 128 × 128. The feature extractor is a module for extracting features from the hand viewpoint image and consists of three convolution layers (Batch normalization and ReLU are applied after convolution processing) and a ConvLSTM. The convolution layer consists of one layer with 16 output dimensions, one with 32 output dimensions, and one with 64 output dimensions. The output dimension of the hidden state in ConvLSTM is 64. The target feature extractor is a module for extracting the features of the target grasping object image, and has the same structure as the feature extractor (except for ConvLSTM) in addition to introducing a convolution layer with 32 output dimensions (ReLU is applied after the convolution processing). The value branch consists of a convolution layer with 32 output dimensions (ReLU is applied after convolution), two fully connected layers (FC), and LeakyReLU [48] (applied between FC). The policy branch consists of the same modules as the value branch, in addition to a Hyperbolic tangent function (tanh) and a Gaussian policy head that generates a Gaussian distribution with the value of tanh applied as the mean value. The FC before tanh has the same number of output units as the number of agent actions. In each branch, the agent learns actions considering the target grasping object by concatenating the feature vectors extracted by the target feature extractor with the input values before the first FC. The PPO in this experiment is the same as the above structure without the mask-attention module.

2) VISUAL EXPLANATION OF THE AGENT MODEL WITH MASK-ATTENTION

We confirm the effectiveness of mask-attention in robot manipulation tasks. Here, mask-attention is a map visualized by Mask PPO MaskattL with Mask-attention Loss. An example of mask-attention visualization is shown in Figure 10. Scene 1 shows the entire dumbbell, which is the object to be grasped, appearing in the image, and Scene 2 shows the dumbbell under plastic wrap, partially occluded. From the policy mask-attention of Ours in Scene 1, we can see that the agent consistently focuses on the dumbbells and ignores other objects. This means that the agent correctly recognizes the dumbbell as the object to be grasped, and controls the grasping of only the dumbbell by not gazing at any objects other than the dumbbell. The value mask-attention of Ours in Scene 1 also shows the same focus on the dumbbells. In addition, in the Value mask-attention, the agent strongly focuses on the handle of the dumbbell. This means that the agent is recognizing the important area for grasping the dumbbell. These mask-attention trends in Scene 1 are also confirmed in Scene 2, where occlusion occurs.

Compared to Greydunus et al.'s method, we can confirm that in the Policy visualization example, Greydunus et al. is noisy, whereas Ours accurately captures the target grasped object. In the Value visualization example, both Greydunus et al. and Ours are gazing at the target grasped object, but Ours is capturing a more localized region of the object.

These results indicate that the agent correctly recognizes the target grasping object from multiple objects despite occlusion, understands the important areas for grasping (dumbbell handle, etc.), and controls the end-effector.

3) EVALUATION BY GRASPING SUCCESS RATE OF TARGET OBJECT

To confirm the effect of the mask-attention module and mask-attention loss on the control performance of the agent in the robot manipulation task. In this experiment, we confirm the effectiveness by the grasping success rate of the dumbbell to be grasped. The grasping success rates with and without Mask-attention module and Mask-attention Loss are shown in Table 6. The grasping success rate is the average of 1,000 episodes of evaluation of the trained model. The success rate of the model without the mask-attention modules (PPO) was 54.58, while the success rate with mask-attention modules was 55.91, an improvement of 1.33 pt. The success rate of the model with the mask-attention module (Mask PPO) was 55.91, while the success rate of the Mask PPO with Mask-attention Loss was 56.18, an improvement of 0.27 pt. These results indicate that the introduction of the Mask-attention module and Mask-attention Loss has improved the performance of the robot by recognizing the areas that are important for optimal robot control.

From the above visual explanation of mask-attention and the evaluation of grasping success rate in robot manipulation, we confirmed that our method is effective not only in 2D environments such as video game strategies, but also in 3D environments such as robot control.

I. INTERPRETABILITY EVALUATION OF MASK-ATTENTION BASED ON AGENT BEHAVIOR PREDICTION BY USERS

In this section, we evaluate the interpretability of mask-attention by investigating whether mask-attention is a map that can be understood for users.

TABLE 7. Average correct answer rate and "unknown" answer rate to the agent's behavior prediction questions: w/o attention is teaching with RGB images only.

teaching methods to users	task		all task	unknown
	games	robot		
w/o attention	49.60	22.94	42.94	3.67
Greydanus *et al.* [10]	32.74	12.94	27.79	8.67
Ours	73.33	44.11	66.02	2.50

1) VERIFICATION METHOD

In this experiment, we verify whether users can understand the agent's behavior when taught using mask-attention, our proposed method. As a comparison, we also verify the teaching method without a gazing area and the teaching method with a saliency map based on Greydanus et al. method [10]. The teaching method without a gazing area is teaching using only RGB images, which are the agent's input information. In this experiment, a total of 51 participants were divided into three groups based on the teaching method, and they answered two types of questions. The first is a question about predicting the agent's behavior. This question asks the participants to predict the agent's behavior in a given frame from the map shown in the teaching method for each group, and to answer the question in the choice format. The answer choices are the actions that the agent could take in the task, and "unknown" is added. The participants choose "unknown" if they cannot determine the agent's behavior from the given information. The second is a question about the prediction of the target grasping object. This question asks the agent to predict which of three objects (dumbbells, plastic wrap, tennis ball case) it is trying to grasp, and to answer the question in the choice format. This is the question for the robot manipulation task, and will be answered after the behavior of agent prediction question in the robot manipulation. The verification process is described below.

1) The 51 participants were divided into 3 groups (17 participants per group) based on the teaching method. Each group is described below.
   - w/o attention: No indication of gazing area, teaching by RGB image.
   - Greydanus et al.: Teaching with saliency map of Greydanus et al..
   - Ours: Teaching with mask-attention of our method introducing Mask-attention Loss.
2) The participants are prompted to understand what the task is by watching a video with a description of each task. In order to limit themselves to understanding the task without teaching, we use a demonstration video with random control (i.e., we don't use demonstrations by trained agents).
3) The participants answer the agent's behavior prediction questions (10 questions per task, 30 questions for all tasks) indicated by the teaching method for each group, in a choice format.

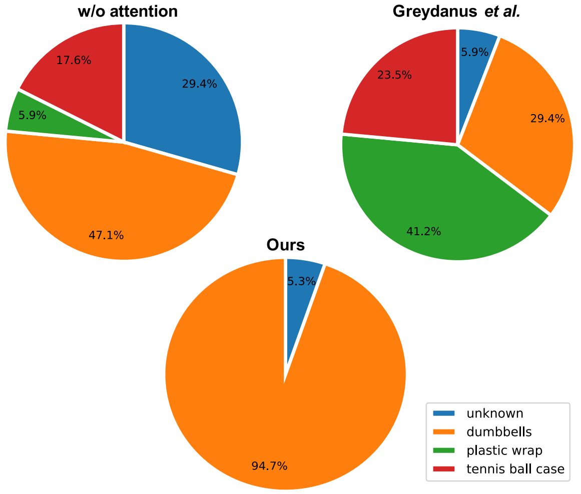


FIGURE 11. Details of the answers to the prediction questions of the grasps target object.

4) The participants perform all tasks 2) and 3) (video game: 3 tasks, robot manipulation: 1 task). After answering all the questions in the robot manipulation task, the participants answer the prediction questions for the target grasping object.
5) We analyze the interpretability of each teaching method based on the correct answer rate of the agent's behavior prediction questions and the correct answer rate of the grasped object prediction questions by participants in each group.

A high percentage of correct answers to these questions indicates that the user can understand the agent's behavior from the map used for teaching. In other words, the map is highly interpretable to the user.

2) VERIFICATION RESULTS

We analyze the interpretability of the maps used for each teaching method from the correct answer rates to the two types of questions. Table 7 shows the average correct answer rate and the percentage of "unknown" answers for question about agent behavior prediction. A high average correct answer rate indicates that the map used for teaching is a map that correctly represents the agent's behavior, and a low percentage of "unknown" answers indicates that the map is easy for the user to read the agent's behavior. Therefore, it can be considered that the visual explanation method of the agent with a better evaluation value by the two metrics has a higher interpretability for the agent's behavior. Comparing the correct answer rates for video games and robot manipulation, the correct answer rate for robot manipulation is consistently lower for all teaching methods. The robot manipulation is a 3D environment that requires more depth information consideration than video games, and the action space of the agent is vast. Therefore, the correct answer rate for robot manipulation was lower than that for video games. Ours had the highest correct answer rate for all tasks at 66.02%, and the lowest rate of ‘‘unknown’’ at 2.5%. These results indicate that mask-attention (Ours) enhances users’ understanding of agent behavior and serves as a highly interpretable visual explanation method for the agent model. On the other hand, Greydanus et al.’s method had a lower correct answer rate than w/o attention and a higher percentage of ‘‘unknown’’ answers. The results indicate that the Greydanus et al.’s method is less interpretable than mask attention and provides incorrect information for the users.

Figure 11 shows the details of the answers to the prediction questions regarding the target grasping object. Here, the target grasping objects of the agent model in all teaching methods are ‘‘dumbbells’’. That is, the correct answer to this question is dumbbells. In w/o attention, 47.1% of participants answered dumbbells, but 29.4% of participants answered unknown. From this result, it can be concluded that the w/o attention teaching was not sufficient to determine which object was the object to be grasped, since only RGB images were used. In Greydanus et al., 41.2% of participants chose dumbbells, while responses for plastic wrap and tennis ball case ranged between $20 - 30\%$, showing scattered answers. In contrast to this result, Ours has a high correct answer rate as 94.7% of participants answered dumbbells. Therefore, the mask-attention in Ours demonstrates high interpretability in understanding the agent’s grasping object, attributable to the constraints imposed by the mask-attention loss.

From these results, we confirmed that the mask-attention module and mask-attention loss enhance the interpretability of mask-attentions, effectively indicating the behavior of the agent model.

V. CONCLUSION

In this paper, we focus on two perspectives: policy and state value, in order to clarify the basis for judgments about the agent model’s behavior in the framework of deep reinforcement learning. Then, we proposed Mask Attention A3C (Mask A3C), a visual explanation method for DRL agents based on the Actor-Critic method. Mask A3C implements an attention mechanism for policy and state value, which are output branches of the Actor-Critic method, and generates mask-attention that highlights significant regions associated with each branch’s outputs. Since policy represents the probability distribution of the action selection and state value represents the value of the current state, these are important elements in the decision-making of a DRL agent. Therefore, visualization of the mask-attention enables us to visually explain the reason for the agent’s decision from both policy and state value perspectives. At the same time, applying the Mask-attention Loss prevents the agent from gazing at areas that do not affect its decision-making during learning, thus improving the interpretability of the mask-attention. Experimental results with the video game and robot manipulation show that the mask-attention represents an important regions for interpreting the agent’s decision-making. To our knowledge, this is the first study to analyze DRL agents from two perspectives: policy and state value. In our study, we focused on policy and state value, but the simple structure of our mask-attention module allows for its application to other DRL methods, including value-based DRL algorithms in addition to Actor-Critic methods.

The threshold value for selecting the pseudo-maskatt in Mask-attention Loss and the introduction step need to be set to optimal values for the task. In the future, we plan to conduct detailed investigations into these hyperparameters. Also, a large number of studies on interpretability focusing on the decision-making of DRL agents have realized visual interpretations of DRL agents’ decisions. Thus, these studies face challenges that the interpretation of a DRL agent’s gazing map is received differently by different users. Therefore, we consider it necessary to develop a method to explain the agent’s decision-making in natural language.

REFERENCES

[1] K. Shao, Z. Tang, Y. Zhu, N. Li, and D. Zhao, ‘‘A survey of deep reinforcement learning in video games,’’ 2019, *arXiv:1912.10944*.

[2] X. Liang, X. Du, G. Wang, and Z. Han, ‘‘A deep reinforcement learning network for traffic light cycle control,’’ *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1243–1253, Feb. 2019.

[3] Z. Yao, X. Liang, G.-P. Jiang, and J. Yao, ‘‘Model-based reinforcement learning control of electrohydraulic position servo systems,’’ *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 3, pp. 1446–1455, Jun. 2022.

[4] A. Namdari, M. A. Samani, and T. S. Durrani, ‘‘Lithium-ion battery prognostics through reinforcement learning based on entropy measures,’’ *Algorithms*, vol. 15, no. 11, p. 393, Oct. 2022.

[5] D. Han, B. Mulyana, V. Stankovic, and S. Cheng, ‘‘A survey on deep reinforcement learning algorithms for robotic manipulation,’’ *Sensors*, vol. 23, no. 7, p. 3762, Apr. 2023.

[6] X. Chen, L. Yao, J. McAuley, G. Zhou, and X. Wang, ‘‘Deep reinforcement learning in recommender systems: A survey and new perspectives,’’ *Knowl.-Based Syst.*, vol. 264, Mar. 2023, Art. no. 110335.

[7] I. Sorokin, A. Seleznev, M. Pavlov, A. Fedorov, and A. Ignateva, ‘‘Deep attention recurrent Q-network,’’ presented at the Adv. Neural Inf. Process. Syst. (NeurIPS) Workshops, 2015.

[8] T. Zahavy, N. Ben-Zrihem, and S. Mannor, ‘‘Graying the black box: Understanding DQNs,’’ in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1899–1908.

[9] R. Zhang, Z. Liu, L. Zhang, J. A. Whritner, K. S. Müller, M. M. Hayhoe, and D. H. Ballard, ‘‘AGIL: Learning attention from human for visuomotor tasks,’’ in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 663–679.

[10] S. Greydanus, A. Koul, J. Dodge, and A. Fern, ‘‘Visualizing and understanding Atari agents,’’ in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 1792–1801.

[11] L. Weitkamp, E. van der Pol, and Z. Akata, ‘‘Visual rationalizations in deep reinforcement learning for Atari games,’’ in *Proc. Benel. Conf. Artif. Intell. (BNAIC)*, 2018, pp. 151–165.

[12] A. Mott, D. Zoran, M. Chrzanowski, D. Wierstra, and D. J. Rezende, ‘‘Towards interpretable reinforcement learning using attention augmented agents,’’ in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019, pp. 12350–12359.

[13] A. Manchin, E. Abbasnejad, and A. van den Hengel, ‘‘Reinforcement learning with attention that works: A self-supervised approach,’’ in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2019, pp. 223–230.

[14] W. Shi, G. Huang, S. Song, Z. Wang, T. Lin, and C. Wu, ‘‘Self-supervised discovering of interpretable features for reinforcement learning,’’ *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2712–2724, May 2022.

[15] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, ‘‘Asynchronous methods for deep reinforcement learning,’’ in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

[16] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, ‘‘Reinforcement learning with unsupervised auxiliary tasks,’’ in *Proc. Int. Conf. Learn. Represent. (ICLR)*. OpenReview, 2017.

[17] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. 32nd Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 37, Lille, France, 2015, pp. 1889–1897.

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[19] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[20] J. Duan, Y. Guan, S. E. Li, Y. Ren, Q. Sun, and B. Cheng, "Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6584–6598, Nov. 2022.

[21] H. Itaya, T. Hirakawa, T. Yamashita, H. Fujiyoshi, and K. Sugiura, "Visual explanation using attention mechanism in actor-critic-based deep reinforcement learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–10.

[22] C. Rupprecht, C. Ibrahim, and C. J. Pal, "Finding and visualizing weaknesses of deep reinforcement learning agents," in *Proc. Int. Conf. Learn. Represent. (ICLR)*. OpenReview, 2020.

[23] H. Liu, M. Zhuge, B. Li, Y. Wang, F. Faccio, B. Ghanem, and J. Schmidhuber, "Learning to identify critical states for reinforcement learning from videos," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 1955–1965.

[24] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 3215–3222.

[25] A. P. Badia, B. Piot, S. Kapturowski, P. Sprechmann, A. Vitvitskyi, D. Guo, and C. Blundell, "Agent57: Outperforming the Atari human benchmark," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 507–517.

[26] S. Kapturowski, V. Campos, R. Jiang, N. Rakić ević, H. van Hasselt, C. Blundell, and A. P. Badia, "Human-level Atari 200x faster," 2022, *arXiv:2209.07550*.

[27] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," in *Proc. AAAI Fall Symp. Ser.*, 2015, pp. 29–37.

[28] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*. OpenReview, 2016.

[29] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Proc. Conf. Neural Inf. Process. Syst.*, Dec. 2000, pp. 1008–1014.

[30] A. Nair, P. Srinivasan, S. Blackwell, C. Alcicek, R. Fearon, A. De Maria, V. Panneershelvam, M. Suleyman, C. Beattie, S. Petersen, S. Legg, V. Mnih, K. Kavukcuoglu, and D. Silver, "Massively parallel methods for deep reinforcement learning," presented at the Int. Conf. Mach. Learn. Deep Learn. Workshop, 2015.

[31] S. Kapturowski, G. Ostrovski, J. Quan, R. Munos, and W. Dabney, "Recurrent experience replay in distributed reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*. OpenReview, 2019.

[32] T. L. Paine, C. Gulcehre, B. Shahriari, M. Denil, M. Hoffman, H. Soyer, R. Tanburn, S. Kapturowski, N. Rabinowitz, D. Williams, G. Barth-Maron, Z. Wang, and N. de Freitas, "Making efficient use of demonstrations to solve hard exploration problems," in *Proc. Int. Conf. Learn. Represent. (ICLR)*. OpenReview, 2020.

[33] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[34] W. Samek, G. Montavon, A. Binder, S. Lapuschkin, and K.-R. Müller, "Interpreting the predictions of complex ML models by layer-wise relevance propagation," 2016, *arXiv:1611.08191*.

[35] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

[36] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 839–847.

[37] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, and J. Jiao, "Weakly supervised instance segmentation using class peak response," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3791–3800.

[38] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[39] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," in *Proc. Int. Conf. Learn. Represent. (ICLR)*. OpenReview, 2017.

[40] V. Petsiuk, A. Das, and K. Saenko, "RISE: Randomized input sampling for explanation of black-box models," in *Proc. Brit. Mach. Vis. Conf.*, 2018, p. 151.

[41] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4190–4197.

[42] L. Li, B. Wang, M. Verma, Y. Nakashima, R. Kawasaki, and H. Nagahara, "SCOUTER: Slot attention-based classifier for explainable image recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 1026–1035.

[43] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network: Learning of attention mechanism for visual explanation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10697–10706.

[44] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015, pp. 802–810.

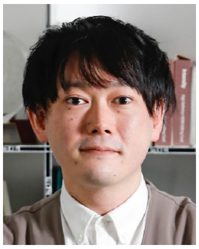
[45] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI gym," 2016, *arXiv:1606.01540*.

[46] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance GPU-based physics simulation for robot learning," in presented at the 35th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track, 2021.

[47] R. Araki, T. Yamashita, and H. Fujiyoshi, "ARC2017 RGB-D dataset for object detection and segmentation," in *Proc. Late Breaking Results Poster Int. Conf. Robot. Autom. (ICRA)*, 2018.

[48] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2013, vol. 30, no. 1, pp. 1–3.

HIDENORI ITAYA (Graduate Student Member, IEEE) received the B.E. and M.S. degrees in computer science from Chubu University, Japan, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree. He has been with DENSO WAVE Inc., since 2021.



TSUBASA HIRAKAWA (Member, IEEE) received the Ph.D. degree in computer science from Hiroshima University, Japan, in 2017. From 2017 to 2019, he was a Researcher Fellow with Chubu University, Japan. He has been a specially appointed Associate Professor with the Chubu Institute for Advanced Studies, Chubu University, since 2019. He has been a Lecturer with the Department of Computer Science, Chubu University, since 2021. He held a Fellowship with Japan Society for the Promotion of Science, from 2014 to 2017. He was a Visiting Researcher with ESIEE Paris, France, from 2014 to 2015.

TAKAYOSHI YAMASHITA (Member, IEEE) received the Ph.D. degree in computer science from Chubu University, Japan, in 2011. He was with OMRON Corporation, from 2002 to 2014. He was a Lecturer with the Department of Computer Science, Chubu University, from 2014 to 2017, where he was an Associate Professor, from 2017 to 2021, and has been a Professor, since 2021. His current research interests include object detection, object tracking, human activity understanding, pattern recognition, and machine learning. He is a member of IEICE and IPSJ.

HIRONOBU FUJIYOSHI (Member, IEEE) received the Ph.D. degree in electrical engineering from Chubu University, Japan, in 1997. From 1997 to 2000, he was a Postdoctoral Fellow with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, working on the DARPA video surveillance and monitoring (VSAM) effort and the humanoid vision project for the HONDA humanoid robot. From 2005 to 2006, he was a Visiting Researcher with the Robotics Institute, Carnegie Mellon University. He is currently a Professor with the Department of Robotics, Chubu University. His research interests include computer vision, video understanding, and pattern recognition. He is a member of IEICE and IPSJ.

KOMEI SUGIURA (Member, IEEE) received the B.E. degree in electrical and electronic engineering and the M.S. and Ph.D. degrees in informatics from Kyoto University, in 2002, 2004, and 2007, respectively. From 2006 to 2008, he was a Research Fellow with Japan Society for the Promotion of Science. From 2006 to 2009, he was with ATR Spoken Language Communication Research Laboratories. He is currently a Professor with Keio University. From 2008 to 2020, he was a Senior Researcher with the National Institute of Information and Communications Technology, Japan, before joining Keio University, Japan, in 2020. His research interests include multimodal language understanding, service robots, machine learning, spoken dialogue systems, cloud robotics, imitation learning, and recommender systems.