JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 37, 517-533 (2021)
DOI: 10.6688/JISE.202105_37(3).0002

Residual Network for Deep Reinforcement Learning with Attention Mechanism

HANHUA ZHU[1] AND TOMOYUKI KANEKO[2]
[1]*Graduate School of Interdisciplinary Information Studies*
[2]*Interfaculty in Information Studies*
University of Tokyo
Tokyo, 113-0033 Japan
E-mail: zhu-hanhua@g.ecc.u-tokyo.ac.jp; kaneko@acm.org

Making decisions based on a good representation of the environment is advantageous in deep reinforcement learning (DRL). In this work, we propose a new network structure for DRL, Deep Residual Attention Reinforcement Learning (DRARL), by incorporating an attention-based structure into the network structure of Importance Weighted Actor-Learner Architecture (IMPALA). DRARL helps the model learn a better representation by helping the model focus on the crucial features. The effectiveness of DRARL was empirically evaluated in a subset of Atari games, with popular RL algorithms, IMPALA, PPO, and A2C. The experiments show that DRARL works robustly with the three algorithms and improves sample efficiency in seven out of ten games. Furthermore, the visualization of important features empirically shows that the DRARL helps the model concentrate on the crucial features and therefore improves the performance and sample efficiency.

Keywords: deep reinforcement learning, representation learning, attention mechanism, Atari games, visualization of reinforcement learning

1. INTRODUCTION

Reinforcement learning (RL) algorithms have recently been firmly established as feasible approaches in several complex tasks such as Go and Atari games. Deep reinforcement learning (DRL), which combines RL with deep neural networks, sheds light on the viability of training directly from image input which is a popular data source in real-world tasks such as autonomous driving [1,2]. This combination increases the successful applications of RL techniques but also brings challenges. As the number of dimensions in the input increases (*e.g.*, by using larger image size), the agent usually needs more experiences and the learning becomes more difficult. To remedy the problem, we need to improve sample efficiency, by obtaining such a good representation of an input that can identify important features in the input.

Visual attention mechanisms inspired by human perception have recently achieved great success in caption generation [3], image classification [4], and machine translation [5] by focusing on a subset of the input or computing the relevance between different parts of inputs. In image classification tasks, visual attention mechanisms compute

Received August 9, 2020; revised October 16, 2020; accepted November 16, 2020.
Communicated by Chia-Hui Chang.

weights of pixel features and help the model distinguish between the important features and the irrelevant features when dealing with high-dimensional images which induced us to investigate the possible benefits from combining attention mechanisms with DRL algorithms.

In this work, we extend our preliminary research [6] and propose a general network structure for DRL called Deep Residual Attention Reinforcement Learning (DRARL) which can be easily applied to multiple state-of-art DRL algorithms. Our contributions can be summarized as follows:

1. We incorporate the residual attention mechanism [4] into the network structure of a famous deep reinforcement learning model called IMPALA [2]. Unlike other existing models that apply attention mechanisms to computationally expensive recurrent networks, our model computes weights on input features in the single feed-forward process and is constructed by stacking multiple modules. DRARL shows 10.53% mean and 6.35% median improvements in the performance and demonstrates higher sample efficiency compared to the original IMPALA in seven of ten Atari games with the cost of a 4% increase in parameters.

2. In order to evaluate the general ability of DRARL, we apply it to the other two famous DRL algorithms, Advantage Actor-critic (A2C) and Proximal Policy Optimization (PPO) [7]. The results show that PPO and A2C enjoy the advantages of the additional attention mechanism. (7.91% mean improvements in PPO and slight mean improvements in A2C)

3. Interpretability is important for making reliable AI agents in general. To grasp the change inside the neural networks, we visualized how our attention structure works by visualizing the attention of the trained agents, which specifically, highlighting the important regions where the agents base its decision on the game screen. The results of visualization demonstrate that our network structure helps the model focus on the crucial area, which explains the improvement in the final performance and sample efficiency to some extent.

2. RELATED WORK

When playing games, human players tend to pay more attention to the crucial objects in the game screen such as the ball in a catching game or the treasure chest in a dungeon game as they are closely connected to the goals of the game. Under the guidance of inherent attention, we make decisions based more on these crucial parts than other parts such as background which are less relevant to the goals, and consequently get a higher score. Y. C. Leong *et al*. [8] conducted experiments on this utilization of the attention and showed that the attention biases both value computation during choice and value update during human learning which are also important in reinforcement learning algorithms.

Attention mechanisms have been incorporated into the reinforcement learning algorithms to select relevant information for each agent in a multi-agent environment [9], to reason about the relations between entities in a scene [10] and to reduce the computational operations by focusing on relatively small informative regions of the input image [11, 12]. L. Yuezhang *et al*. [13] combined the optical flow-based attention mechanism with Advantage Actor-Critic (A2C) but no significant improvement was observed in their experiments. Deep Attention Recurrent Q-Network (DARQN) [12] is an extension of Deep Q-Network [1] by adding soft and hard attention mechanisms. Study [14] presents a soft attention model to force the model to focus on crucial information. The purpose of these two works is similar to ours and all of them employ soft attention mechanisms but the structures of attention mechanisms are totally different. In DARQN, the attention mechanisms work sequentially after the convolutional neural networks (CNNs) and in research [14], the architecture of the model has been totally replaced with the multi-head attention structure, a famous attention mechanism introduced in [5]. However, in our attention mechanism, both paths of CNNs and attention run in parallel, which is easy to implement. Furthermore, both of two previous works depend on recurrent neural networks, on the contrary, we construct our model with feedforward networks, and consequently make the learning in a much simpler way.

3. METHOD

3.1 Deep Residual Attention Reinforcement Learning

We incorporate the residual attention mechanism to help our model find the crucial features among the state input. The proposed model is constructed by stacking multiple attention modules and each module has two branches: the mask branch and the trunk branch, following study [4]. An attention module is illustrated in Fig. 1. As shown in Fig. 1, the trunk branch $T(x)$ processes input features with a base network structure while the mask branch $M(x)$ adds soft weights on input features with a bottom-up top-down structure which mimics both the bottom-up fast feedforward process and the top-down attention feedback. The output of attention module is computed by:

$$H_{i,c}(x) = (1 + M_{i,c}(x)) * T_{i,c}(x), \tag{1}$$

where i ranges over all input features and c is the index of the channel in CNNs. If the mask branch $M(x)$ yields all zeros, the output $H(x)$ become $T(x)$, the output of the base network. Similar to the ideas in residual learning [15], this equation ensures that the performance will be no worse than the model without attention. The mask branch not only works as a feature selector in forward process, but also produces gradient filter during back propagation which makes model robust to the noise.

In this work, we use residual network [15] as a basic unit in both mask and trunk branch. A residual block consists of two convolution layers and each of them is followed by a ReLU unit. The trunk branch has two residual blocks while the mask branch only has one. The network structure of the mask branch used in our model follows the design in study [4] and is simplified as unlike the large size of images processed in image classification tasks, the scale of features processed in current reinforcement learning tasks is relatively small. In the mask branch, a max-pooling is performed firstly to increase the receptive field. After reaching a suitable resolution, a residual block takes the responsibility of computing soft weights. Then, a bilinear interpolation is conducted to keep the output size the same to the input features. After two consecutive 1×1 convolution layers, a sigmoid layer finally normalizes the output range to $[0, 1]$.

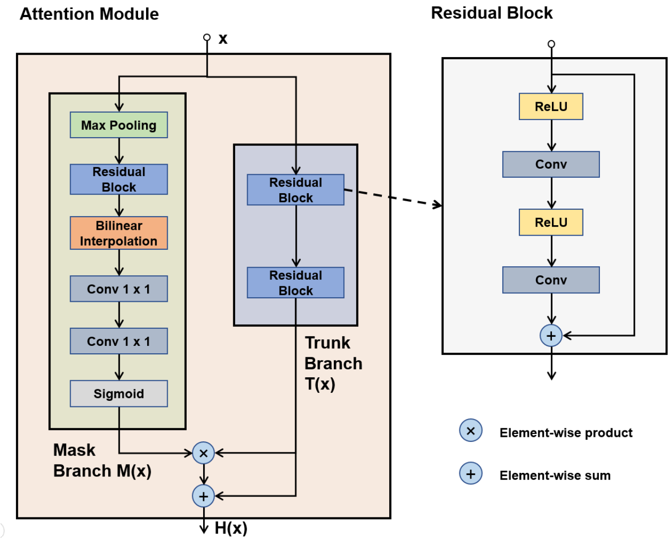


Fig. 1. The structure of the attention module.

3.2 Deep Reinforcement Learning

Among a family of state-of-the-art reinforcement learning algorithms, we chose three algorithms, IMPALA, A2C and PPO and apply our new network structure to them.

3.2.1 IMPALA

IMPALA is a state-of-the-art method, while it is sometimes referred to as the relatively large network structure among DRL algorithms. Instead of communicating the gradients of parameters between workers and a central parameter server, the actors in IMPALA communicate trajectories which consist of states, actions and rewards with a centralized learner. With these trajectories, the learner is able to access the information necessary for the training and therefore the parameter updating and the trajectory generation can be conducted in parallel. For the purpose of reducing the harmful discrepancy between the latest policy updated by the learner and the out-of-date policy used in trajectory generation, IMPALA uses a V-trace off-policy actor-critic algorithm.

The goal of IMPALA is to learn a policy π and a value function V^π. As the learning is off-policy, the algorithm needs to learn the V^π of policy π which is usually called target policy by using trajectories generated by a diffenent policy μ called behaviour policy. When given a trajectory $(s_i, a_i, r_i)_{i=t}^{i=t+n}$ which consists of sequences of state s, the reward r and the action a selected by policy π starting from time t to time $t+n$, the n-steps V-trace target for $V(s_t)$ at state s_t is defined as:

$$v_t = V(s_t) + \delta_t V + \gamma c_t (v_{t+1} - V(s_{t+1})), \tag{2}$$

where $\delta_t V = \rho_t(r_t + \gamma V(s_{t+1}) - V(s_t))$ represents temporal difference for V and and γ is the discount factor. Truncated importance sampling weights $\rho_t = \min(\overline{\rho}, \dfrac{\pi(a_t|s_t)}{\mu(a_t|s_t)})$,

$c_t = \min(\overline{c}, \dfrac{\pi(a_t|s_t)}{\mu(a_t|s_t)})$. ρ_t defines the fixed point of the update and $\overline{\rho}$ directly influences the value function the algorithm converges to. In details, the algorithm will converge to the value function of the target policy if $\overline{\rho}$ is infinite while close to the value function of the behavior policy when $\overline{\rho}$ is close to zero. c_t works as a trace cutting and measures how much a temporal difference at time $t+1$ will influence the update at a previous time t and $\overline{c}$ influences the speed of convergence.

At training time t, the parameters of value function θ are updated by gradient descent in the direction of

$$(v_t - V_\theta(s_t))\nabla_\theta V_\theta(s_t), \tag{3}$$

which adjusts the parameters θ to reduce the difference between the output $V_\theta(s_t)$ and the V-trace target v_t. The policy parameters ω are updated by policy gradient in the direction of

$$\rho_t \nabla_\omega \log \pi_\omega(a_t|s_t)(r_t + \gamma v_{t+1} - V_\theta(s_s)), \tag{4}$$

which adjusts the parameters ω to increase the log-probability of chosen action which lead to a higher state-action value and decrease the action which have a lower state-action value. $V_\theta(s_s)$ is used as a baseline to reduce the variance of the policy gradient estimate. Furthermore, an additional entropy bonus is added to prevent premature convergence:

$$-\nabla_\omega \sum_a \pi_\omega(a|s_t)\log \pi_\omega(a|s_t). \tag{5}$$

The overall update is conducted by summing these three gradient with appropriate coefficients.

3.2.2 A2C

A2C is the synchronous implementation of Asynchronous advantage actor-critic (A3C) [16]. Instead of asynchronous updating, A2C performs updates after each actor finishes its segment of experience. Same to the A3C, A2C updates the parameters with the gradient given by:

$$\nabla_{\theta'} \mathrm{log}\pi(\mathrm{a_t}|\mathrm{s_t};\theta')\mathrm{A}(\mathrm{s_t},\mathrm{a_t};\theta_\mathrm{v}) + \beta\nabla_{\theta'}\mathrm{H}(\pi(\mathrm{s_t};\theta')), \tag{6}$$

where θ' represents parameters updated to learn the target policy while θ_v is the parameters of value function for generating trajectories information used for learning. H is an additional entropy regularization term to ensure sufficient exploration and β is the hyperparameter controlling the strength of the entropy regularization. $A(s_t, a_t; \theta, \theta_v)$ is an advantage estimation defined as :

$$A(s_t, a_t; \theta_v) = r_t + \gamma V(s_{t+1}; \theta_v) - V(s_t; \theta_v). \tag{7}$$

3.2.3 PPO

PPO attains the data efficiency and reliable performance of trust region policy optimization (TRPO) [17] with a simpler implementation by maximizing a novel objective function. TRPO maximizes an objective function defined as:

$$\underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t] \tag{8}$$

$$\text{subject to} \quad \hat{\mathbb{E}}_\text{t}[\mathbf{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid \text{s}_\text{t}), \pi_\theta(\cdot \mid \text{s}_\text{t})]] \ \leq \ \delta, \tag{9}$$

where θ and θ_{odd} are the policy parameters to be updated and before the update. $\hat{\mathbb{E}}_t$ represents the empirical average over a finite batch of samples and δ is the parameter represents the bond on KL divergence. $\hat{A}_t$ is a truncated version of generalized advantage estimation at time step t computed with the length-T trajectory. $\hat{A}_t$ is defined as:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1}, \tag{10}$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ and λ is the discount factor.

In order to simplify the optimization process and avoid the excessively large policy update in the maximization of Eq. (8), PPO uses following clipped objective function to penalize changes to the policy:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(\text{r}_\text{t}(\theta)\hat{\text{A}}_\text{t}, \text{clip}(\text{r}_\text{t}(\theta), 1-\varepsilon, 1+\varepsilon)\hat{\text{A}}_\text{t})], \tag{11}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$, and ε is a hyperparameter controlling the clip operation.

Considering that parameters will be shared between the policy and value function if a neural network structure is used, combining policy objective function with a value function error term which is defined as $L_t^{VF} = (V_\theta(s_t) - V_t^{targ})^2$ where $V_t^{targ} = \hat{A}_t + V_{\theta_{old}}(s_t)$ [18] is necessary. Similar to the A2C, an entropy bonus $S[\pi_\theta](s_t)$ is also added to the objective function to ensure exploration. The final objective function is defined as:

$$L^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)], \tag{12}$$

where c_1, c_2 are coefficients.

3.3 Visualization of Attention

In order to explain how the attention mechanism influences the decision, we imply Gradient-weighted Class Activation Mapping (Grad-CAM) [19] which is widely used to produce visual explanations for the decisions made by CNN-based models. As the fully-connected layers lose the spatial information, Grad-CAM uses the gradient information from the last convolutional layer to compute the importance of each neuron for a decision. Weitkamp *et al*. [20] adapted this method to reinforcement learning models. Let y^a be the value of selected action a before the softmax layer and the kth feature map of convolutional layer A as A^k. The neuron importance weight α_k^a is defined as:

$$\alpha_k^a = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^a}{\partial A_{ij}^k}, \tag{13}$$

where i and j are the coordinates of feature maps, $\frac{1}{Z}\sum_i\sum_j$ is a global average pooling operation, and α_k^a represents a partial linearization of the network downstream from A

which computes the importance weights of each neuron of feature map k for a selected action a.

Then a weighted combination of forward activation maps is performed and for filtering the features that have a negative influence on the action of interest, a ReLU is applied to the linear combination of maps. The final localization map $L^a_{\text{Grad-CAM}}$ is defined as:

$$L^a_{\text{Grad-CAM}} = ReLU(\sum_k \alpha^a_k A^k). \tag{14}$$

We finally extrapolate the heat-map to the size of input state by conducting a bilinear interpolation because $L^a_{Grad-CAM}$ is a coarse heat-map of the same size as the convolutional feature map which is smaller than the input.

4. EXPERIMENTS

To evaluate the effectiveness of DRARL, we conducted the following experiments:

1. The experiments in one of the most well-used benchmarks for DRL, Atari game. We extended the experiments of IMPALA from a base set of five Atari games conducted in our preliminary work [6] to an extended set of 10 Atari games. In order to evaluate the general ability of DRARL, we also conducted experiments with two famous DRL algorithms, A2C and PPO. Because of the limitation of computational resources, we only had time to obtain results of the base set of Atari games in the experiments of A2C and PPO. But we think these five games could produce convincing evidence to prove the general ability of DRARL.

2. Ablation study on different baseline network structures. To identify whether performance increases are introduced by the attention mechanisms or by the increase in the number of learnable parameters, we designed two naive network structures as new baseline and conducted experiments in an Atari game where DRARL showed great improvements.

3. Visual Explanation. We analyzed the influence of the attention mechanism on the learned parameters by visualizing the important regions in the game screen with Grad-CAM and counting the number of parameters in different gradient intervals. Statistical analysis and visualization results show that DRARL helps models focus on the important features and thus learn a better representation.

4.1 Details of Training

For all the experiments, we evaluated models trained with two network structures: our DRARL and the original network structure of IMPALA in the Atari environment produced by OpenAI Gym [21]. The two network structures are shown in the Fig. 2. The original network structure of IMPALA has $1,089,232+257\times$ action numbers parameters while DRARL has $1,135,632+257\times$ action numbers parameters which is only 4% more than the IMPALA's.

For data processing and hyperparameters, we followed the study [2] where all the image data in the training was converted from RGB color space to Gray color space

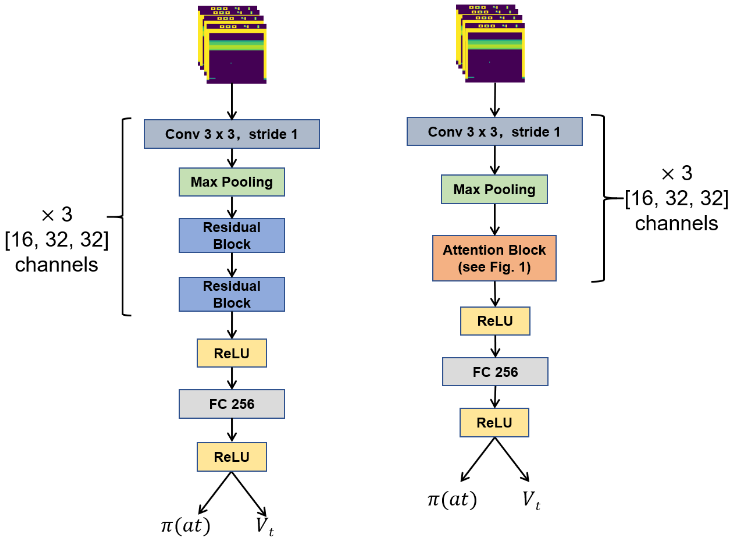


Fig. 2. Model architecture; (Left) IMPALA, which is same to the network structure proposed in study [2]; (Right) DRARL, using one attention block (explained in Section 3.1) instead of two residual blocks.

and resized into 84 × 84 at first. Then the last four frames (after frame skipping) were stacked together as the observation of current state. For models trained by IMPALA, the hyperparameters used in the training were totally the same as the setting in research [2]. We trained our models in the architecture of 32 workers and 1 centralized learner. For experiments of A2C and PPO, we used the implementation of A2C and PPO2 in OpenAI baselines [18] and followed the setting of initial parameters. All the experiments were conducted on a single machine with two NVIDIA 1080Ti GPUs and one 16-core 32-threads AMD CPU.

The results shown in this paper were the average test scores of three agents each of which is trained independently with same algorithm and network structure with 200 million environment steps. Results shown as percentages are computed with scores normalized by baseline's (IMPALA, A2C, and PPO) scores, which demonstrate the difference between our proposed one and the baseline more directly than human-normalized scores. Agents only had a single life during training while games over when agents lost the standard number of lives in the test environment. Each score listed in the Tables and Figures is the average score over 200 plays in the test environment. For convenience, we abbreviate the models trained with the baseline IMPALA's network structure as IMPALA in the Tables and Figures.

4.2 Performance in Atari Games

4.2.1 IMPALA

Table 1 summarizes the results of 10 Atari games, five of which were shown in our preliminary work [6]. The fourth column of the table shows the rate of increase of the

Table 1. The final scores of DRARL and original IMPALA in 10 Atari games. DRARL shows 10.53% **mean and** 6.35% **median higher performance over IMPALA (based on IMPALA-normalized results).**

Game	IMPALA	(std)	DRARL	(std)	Rate of increase	IMPALA paper
Alien	1442.83	(218.77)	**1570.05**	(182.47)	8.82%	15962.1
Amidar	**757.45**	(56.17)	749.50	(119.77)	−1.05%	1554.79
BeamRider	**7645.31**	(127.84)	7001.07	(64.89)	−8.43%	29608.05
Berzerk	650.58	(4.28)	**659.98**	(26.74)	1.44%	1852.70
Breakout	473.00	(13.88)	**497.29**	(18.80)	5.14%	787.34
Centipede	6716.95	(1097.22)	**6899.73**	(763.32)	2.72%	11049.75
Krull	5966.99	(171.38)	**8053.78**	(250.01)	34.97%	8147.70
KungFuMaster	25232.50	(1712.01)	**27141.00**	(1549.76)	7.56%	43375.50
RoadRunner	48939.33	(4560.99)	**55960.83**	(2151.47)	14.35%	57121.00
Seaquest	1631.57	(243.86)	**2280.00**	(237.82)	39.74%	1753.20

Table 2. The final scores of A2C with IMPALA network structure and DRARL in 5 Atari games. A2C+DRARL shows 0.45% **mean and** 2.55% **median higher performance over A2C (based on A2C-normalized results).**

Game	A2C	(std)	A2C+DRARL	(std)	Rate of increase
Amidar	988.99	(269.63)	**1014.23**	(136.04)	2.55%
Breakout	677.77	(17.13)	**685.49**	(8.44)	1.14%
Centipede	6034.03	(265.04)	**6564.29**	(192.46)	8.79%
Krull	8772.12	(1720.52)	**9621.70**	(554.61)	9.68%
Seaquest	**2198.60**	(431.93)	1760.77	(3.94)	−19.91%

performance of proposed DRARL compared to baseline IMPALA. The fifth column of the table shows the results published in study [2]. The results of our IMPALA shown in the second column are different from those in the study [2]. We investigate that the differences are introduced by some parameters such as the number of workers that are not written in the paper [2]. From Table I, we can find that DRARL outperforms the original IMPALA in eight of ten games. There is only one game BeamRider where the performance of proposed DRARL is worse (−8.43%) than IMPALA. We find that in BeamRider, the enemies move randomly, which means the next state is stochastic given only 4 past observed frames. In this stochastic environment, focusing on the current features may have negative influence on the planning of the future which is more important to get a higher score.

4.2.2 A2C

As shown in the Table 2, DRARL performs better than the network structure of IMPALA in 4 of 5 games when trained with A2C. However, the additional attention mechanism has a negative effect (−19.91%) on the performance of A2C in Seaquest where models benefit a lot (+39.74%) from attention when trained with IMPALA. We also find that the results of A2C+DRARL in Seaquest have an unusually low standard deviation

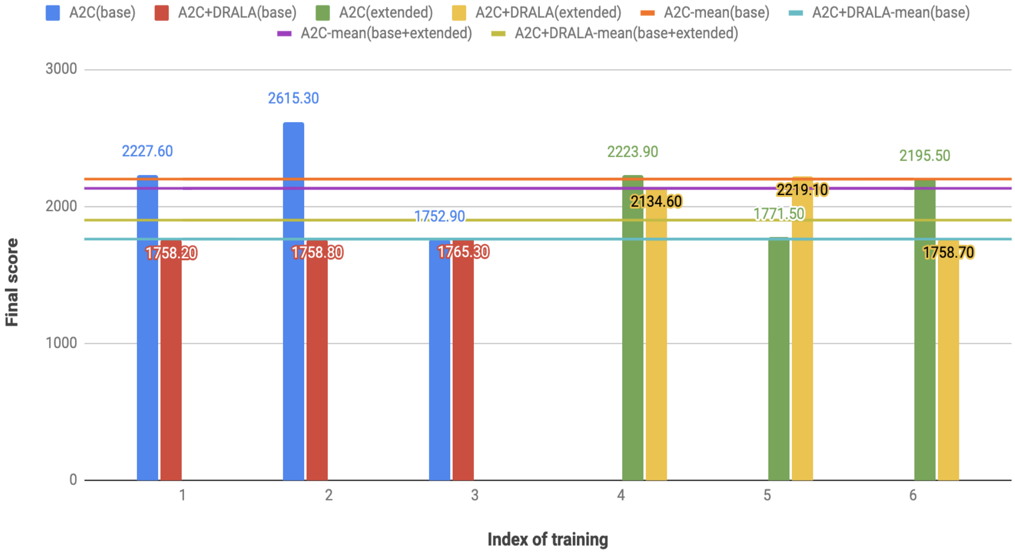


Fig. 3. Results of Sequest where both DRARL and IMPARA network structure are trained with A2C. Training index 1, 2, 3 are the results used to compute the mean value in Table 2 (marked as base in the figure) while 4, 5, 6 are the results from extended experiments in Seaquest(marked as extended in the figure). The mean scores of the six results are 2131.12 (IMPALA) and 1899.12 (DRARL). DRARL has a 10.89% decline compared to the IMPALA.

compared to the other results of Seaquest. We investigate the reason and plot the learning curves in Fig 4, from where we can find that there seems to be a key point causing reward jumps in Seaquest and when the models catch the point, they can converge to a higher score (about 2200), otherwise they will stay the same without any further improvement until the end (about 1700).

To investigate if DRARL really can not catch the key point during the training by A2C in Seaquest, we increased the number of training times from 3 to 6 and plot the results in Fig. 3. Fig. 3 shows that DRARL did catch the key point but in relatively small percentages (2 of 6 training) compared to the models trained with network structure of IMPALA (4 of 6 training). We also find that the rate of increase increases from -19.91% to -10.89% and the mean improvement increases from 0.45% to 2.26%, which could be further improved with more number of training times.

4.2.3 PPO

Similar to the results of A2C, DRARL outperforms the network structure of IMPALA in the first four games but has a 11.52% decrease in Seaquest, as shown in Table 3. We find that the key point in Sequest discussed above also exists during the training of PPO in Seaquest and DRARL catches 1 of 3 training while models trained with the network structure of IMPALA catches 3 of 3 training. We think that the the performance of two network structures will become similar if we increase the number of training times.

Table 3. The final scores of PPO with IMPALA network structure and DRARL in 5 Atari games. PPO+DRARL shows 7.91% **mean and** 8.43% **median higher performance over PPO (based on PPO-normalized results).**

Game	PPO	(std)	PPO+DRARL	(std)	Rate of increase	PPO paper
Amidar	1933.69	(757.45)	**2511.96**	(710.03)	29.91%	674.6
Breakout	560.55	(92.61)	**607.82**	(38.80)	8.43%	274.8
Centipede	12074.21	(404.99)	**13301.78**	(1397.78)	10.17%	4386.4
Krull	9567.93	(248.94)	**9811.74**	(77.47)	2.55%	7942.3
Seaquest	**2801.73**	(20.21)	2479.03	(291.73)	−11.52%	1204.5

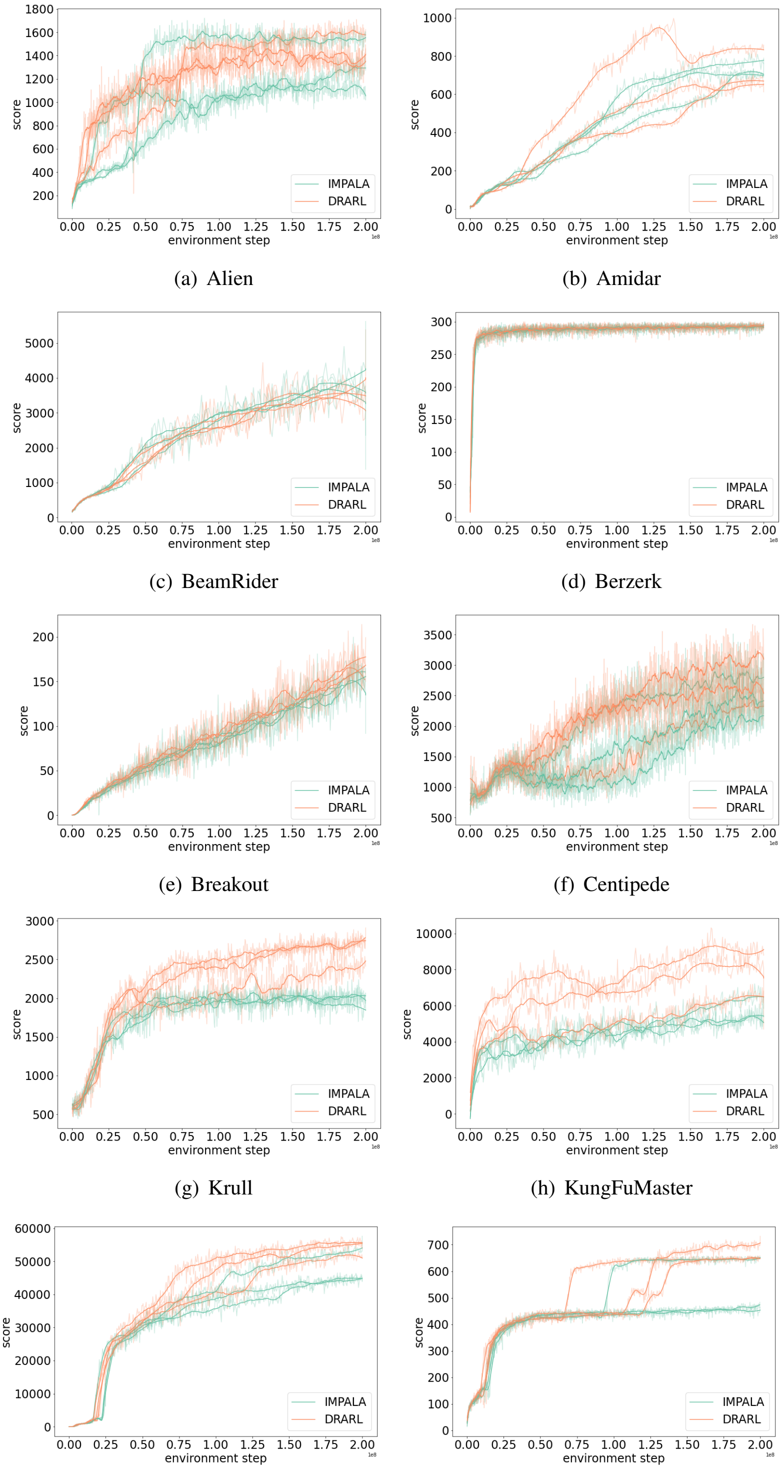
4.2.4 Sample efficiency

We plot learning curves of models trained by IMPALA with DRARL and original network structure in 10 Atari games in Fig. 4. We find that the DRARL contributes to a higher sample efficiency represented by reaching a higher performance in an earlier stage of training in Seaquest, Krull, RoadRunner, and KungFuMaster, which are the games enjoy the benefits from new network structures most. If we analyze the three curves from each network structure together, we can find the mean of three curves of DRARL is above the IMPALA's in Alien and Centipede. Even for the Amidar, where DRARL does not works well, one of three training curves is above all of the three curves of IMPALA. These results give us evidence that DRARL sometimes leads to a higher sampling efficiency in Alien, Centipede, and Amidar. Possibly because of the stochasticity in BeamRider, the additional attention mechanism slows down the training and results in a lower sample efficiency. For the remaining games, Berzerk and Breakout, DRARL shows almost the same sample efficiency as the IMPALA.

4.3 Ablation Study on Network Structure

In order to figure out which part of DRARL most contributed to the improvement in performance, we created two simplified network structures, Simple DRARL and Large IMPALA, for comparison. As shown in the Fig. 5, Simple DRARL has the same number of parameters with proposed DRARL while the number of Large IMPALA's parameters is a little bit smaller than DRARL (the difference comes from two 1×1 convolutional layers in mask branch, which is small enough to be ignored compared to the number of total parameters). Simple DRARL is designed to investigate the performance of the bottom-up top-down structure and Large IMPALA is set to demonstrate the effect of the increment in the number of parameters. We choose PPO as training algorithms and Amidar where our DRARL shows great improvement (+29.91%) with PPO as the experiment environment to evaluate the proposed method and baselines.

As shown in Table 4, although the Large IMPALA outperforms the IMPALA which demonstrates the increment in the number of parameters has a positive effect on performance, higher scores of Simple DRARL and DRARL shows that more improvement is brought by the additional attention mechanism. The higher score of DRARL compared to Simple DRARL also reveals the effectiveness of the bottom-up top-down structure in the mask branch.



(a) Alien (b) Amidar

(c) BeamRider (d) Berzerk

(e) Breakout (f) Centipede

(g) Krull (h) KungFuMaster

(i) RoadRunner (j) Seaquest

Fig. 4. Learning curves of models trained by IMPALA with or without DRARL in Atari games.

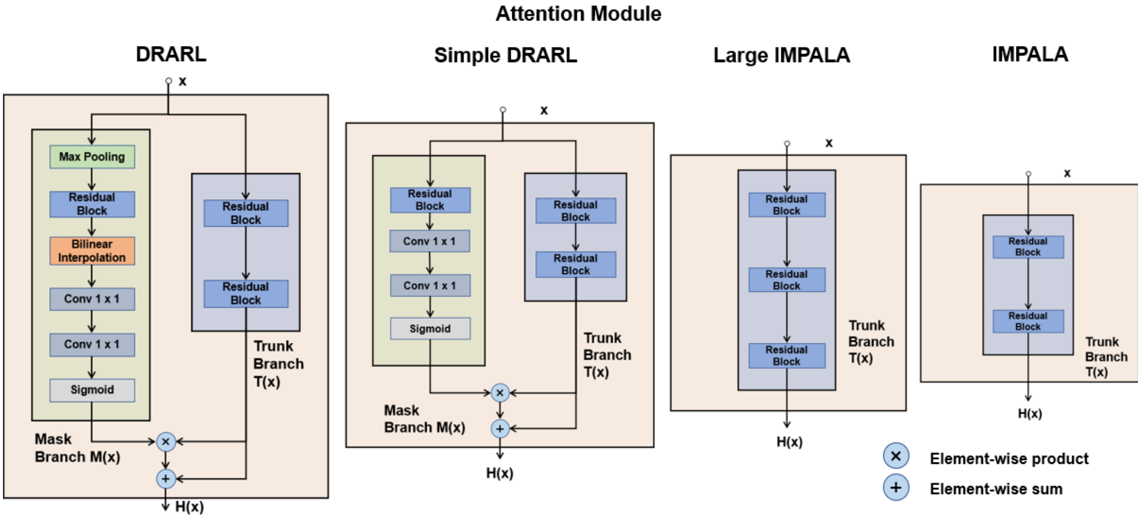


Fig. 5. The structure of a single network module (instead of each attention module in Fig 1) of DRARL, IMPALA and two new alternatives.

Table 4. The final scores of PPO with new baselines in Amidar.

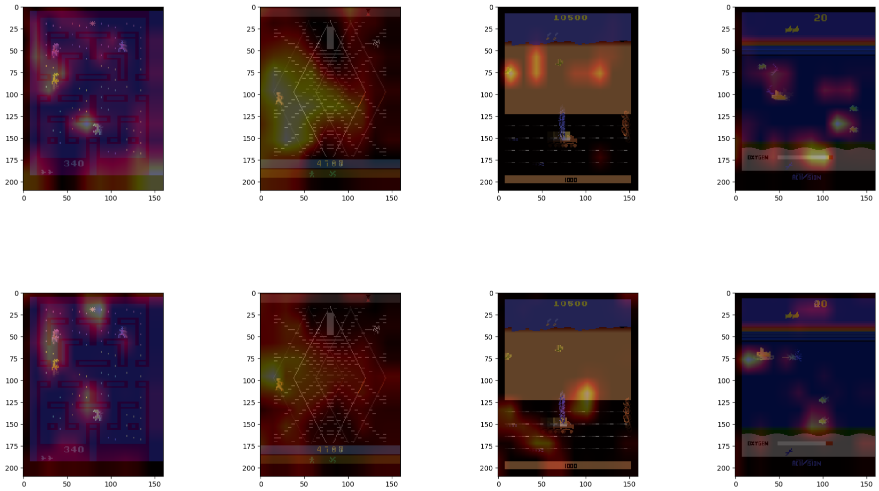
Game	IMPALA(std)	Large IMPALA	Simple DRARL	DRARL(std)
Amidar	1933.69 (757.45)	2100.71 (398.44)	2314.52 (108.71)	**2511.96** (710.03)

4.4 Visual Explanation

4.4.1 Visualization by Grad-CAM

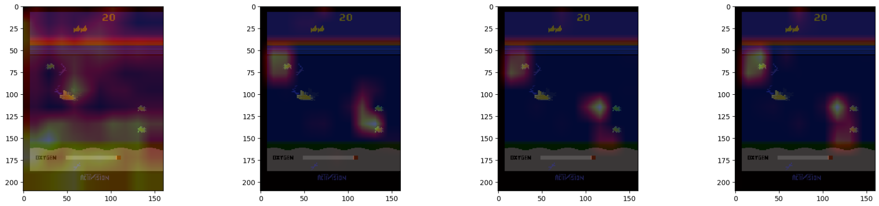
To explain the reason why the additional residual attention mechanism influences the training in a positive way, we visualized the important regions where the models base their decision on the game screen by Grad-CAM. The important regions can be considered as the attention of the models thus the regions should cover the crucial features if the trained models are able to focus on the crucial features. The important regions of the stacked four frames (former three frames are shown as afterimage) among the observation are shown in Fig. 6. As shown in Fig. 6, the important regions for the models trained by IMPALA are distributed in a wide range including irrelevant area while the models trained by DRARL relatively focus on the crucial features (the enemy and player in Alien, RoadRunner and Seaquest, the player in Krull). These results give empirical evidences that the residual attention mechanism helps IMPALA concentrate on the features which are closely connected to the reward and learn a better representation of the environment.

We also visualized the important regions for different training stages of IMPALA with DRARL to demonstrate the trend of focused features during trining. We visualized four stages, 0%(after initialization), 36% (after 72 million training steps), 72% (after 144 million training steps), and 100% (after 200 million training steps). From Fig. 7 we can find that after a short-term study, the model trained with DRARL is able to recognize



(a) Alien (b) Krull (c) RoadRunner (d) Seaquest

Fig. 6. The important regions in four Atari games trained by IMPALA with two DRARL and original network structure. **Top:** IMPALA **Bottom:** DRARL.



(a) DRARL-0% (b) DRARL-36% (c) DRARL-72% (d) DRARL

Fig. 7. The important regions for different training stages of IMPALA with DRARL in Seaquest.

the crucial features which are shown as the important region cover the enemies. With training steps increase, the region becomes smaller and moves slightly from the former location. The results of visualization correspond the learning curve in Fig 4 where a great improvement is demonstrated at the early stage of training while the curve tends to be flat after 144 million training steps.

It should be noted that the attention visualized by Grad-CAM sometimes deviates from the expected location and not always interpretable. We observed some dependency on games, where Grad-CAM works in (a) Alian and (b) Krull, sometimes failed in (d) Sequest shown in the bottom right region, and more frequently in (c) RoadRunner shown in the top yellow background, which is attribute to the limitation of adjusting the visualization method developed for computer vision to RL. The obscure attention pictures are mainly caused by the bilinear interpolation operation which is needed to resize the heat map to match the original game size.

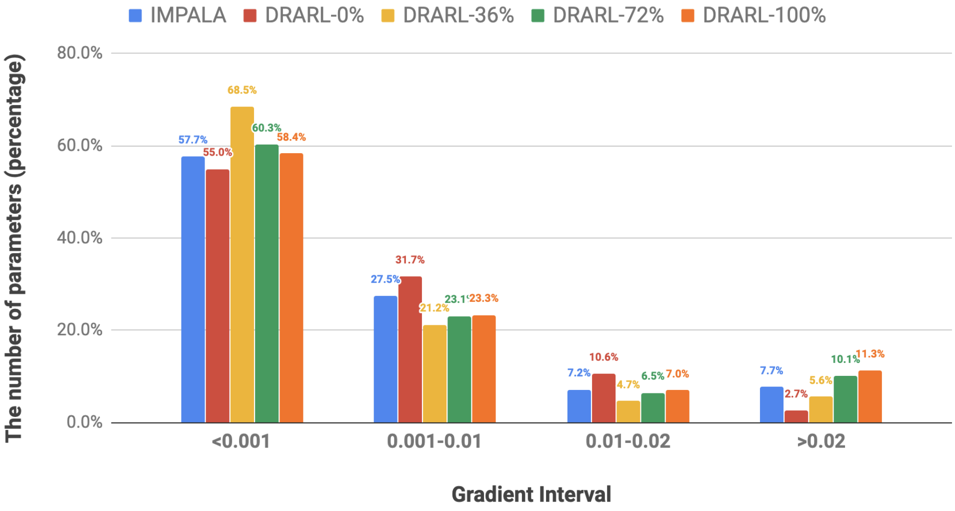


Fig. 8. Frequency of the absolute value of gradients on each input pixels with respect to the selected action. Gradients are accumulated for each interval among 100 state-action pairs in Seaquest.

4.4.2 Visualization of gradients

In order to see how agents focus on important area in the input, we made additional analysis statistically, while we already understood typical cases by Grad-CAM in the previous experiment. We collected gradient data of models trained by IMPALA with original IMPALA networks and DRARL on consecutive states in a trajectory. We recorded the absolute value of the gradients for each pixel with respect to the policy output for the selected action, and counted their occurrence for each interval. The results are the sum of numbers on 100 images input and shown as the percentage of total in Fig. 8. We can find the number in the middle gradient interval decrease (27.5% decrease to 23.3% in 0.001-0.01) and the the number in the larger values interval increase (7.7% increase to 11.3% in $>$ 0.02) compared to the baseline IMPALA. From the results of different training stages of IMPALA with DRARL, we can also find that with training steps increase, the number in the larger values interval (> 0.02) increase while the number in the lower values interval (< 0.001) decrease. These results show agents focus more on a subset of features along with training by DRARL, and indicate the attention mechanism helps trained agents focus on the important features.

5. DISCUSSION

In this work, we have investigated the effectiveness of combining attention mechanisms with DRL algorithms and propose DRARL which incorporates the residual attention mechanism into the network structure of IMPALA. With the attention mechanism, DRARL shows 10.53% mean and 6.35% median improvements in the performance and demonstrates higher sample efficiency compared to the original IMPALA in seven of ten Atari games. The experiments of A2C and PPO demonstrate that our network structure is also effective (7.91% mean improvements in PPO and slight mean improvements in A2C) when trained with other DRL algorithms. We also produce explanations about the improvement by visualizing the important features and the results show that the residual attention mechanism helps model concentrate on the features which are closely connected to the reward and learn a better representation of the environment, which produces an explanation of improvements in the final performance and sampling efficiency.

REFERENCES

1. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu *et al.*, "Human-level control through deep reinforcement learning," *Nature*, Vol. 518, 2015, p. 529.
2. L. Espeholt, H. Soyer, R. Munos, K. Simonyan *et al.*, "Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures," *arXiv preprint*, 2018, arXiv:1802.01561.
3. K. Xu, J. Ba, R. Kiros *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proceedings of International Conference on Machine Learning*, 2015, pp. 2048-2057.
4. F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6450-6458.
5. A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998-6008.
6. H. Zhu and T. Kaneko, "Deep residual attention reinforcement learning," in *Proceedings of IEEE International Conference on Technologies and Applications of Artificial Intelligence*, 2019, pp. 1-6.
7. J. Schulman, F. Wolski, P. Dhariwal *et al.*, "Proximal policy optimization algorithms," *arXiv preprint*, 2017, arXiv:1707.06347.
8. Y. C. Leong, A. Radulescu, R. Daniel, V. DeWoskin, and Y. Niv, "Dynamic interaction between reinforcement learning and attention in multidimensional environments," *Neuron*, Vol. 93, 2017, pp. 451-463.
9. S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97, 2019, pp. 2961-2970.
10. V. Zambaldi, D. Raposo, A. Santoro *et al.*, "Relational deep reinforcement learning," *arXiv preprint*, 2018, arXiv:1806.01830.
11. V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems*, 2014, pp. 2204-2212.
12. I. Sorokin, A. Seleznev, M. Pavlov, A. Fedorov, and A. Ignateva, "Deep attention recurrent q-network," *arXiv preprint*, 2015, arXiv:1512.01693.
13. L. Yuezhang, R. Zhang, and D. H. Ballard, "An initial attempt of combining visual selective attention with deep reinforcement learning," *arXiv preprint*, 2018, arXiv:1811.04407.
14. A. Mott, D. Zoran, M. Chrzanowski *et al.*, "Towards interpretable reinforcement learning using attention augmented agents," in *Advances in Neural Information Processing Systems*, 2019, pp. 12350-12359.

15. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.
16. V. Mnih, A. P. Badia, M. Mirza *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proceedings of International Conference on Machine Learning*, 2016, pp. 1928-1937.
17. J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of International Conference on Machine Learning*, 2015, pp. 1889-1897.
18. P. Dhariwal, C. Hesse, O. Klimov *et al.*, "Openai baselines," https://github.com/openai/baselines, 2017.
19. R. R. Selvaraju, M. Cogswell, A. Das *et al.*, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of IEEE International Conference on Computer Vision*, 2017, pp. 618-626.
20. L. Weitkamp, E. van der Pol, and Z. Akata, "Visual rationalizations in deep reinforcement learning for atari games," *arXiv preprint*, 2019, arXiv:1902.00566.
21. G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman *et al.*, "OpenAI gym," *arXiv preprint*, 2016, arXiv:1606.01540.

Hanhua Zhu received his Bachelor of Engineering in 2017 from Nanjing University of Posts and Telecommunications in China and his M.S. of Applied Computer Science from the University of Tokyo in Japan in 2020. Hanhua Zhu is now a Ph.D. student at the Graduate School of Interdisciplinary Information Studies, the University of Tokyo. His research interests include representation learning and reinforcement learning.

Tomoyuki Kaneko received his Bachelor of Liberal Arts in 1997 from the University of Tokyo in Japan and his M.S. (Multidisciplinary Sciences) and Ph.D. degrees from the same university in 1999 and 2002. He is now an Associate Professor at the Department of Interfaculty Initiative in Information Studies, the University of Tokyo. His research interests include artificial intelligence and machine learning in games.