META ATTENTION FOR OFF-POLICY ACTOR-CRITIC

Anonymous authors
Paper under double-blind review

ABSTRACT

Off-Policy Actor-Critic methods can effectively exploit past experiences and thus they have achieved great success in various reinforcement learning tasks. In many image-based and multi-source tasks, attention mechanism has been employed in Actor-Critic methods to improve their sampling efficiency. In this paper, we propose a meta attention method for state-based reinforcement learning tasks, which combines attention mechanism and meta-learning based on the Off-Policy Actor-Critic framework. Unlike previous attention-based work, our meta attention method introduces attention in the actor and the critic of the typical Actor-Critic framework rather than in multiple pixels of an image or multiple information sources. In contrast to existing meta-learning methods, the proposed meta-attention approach is able to function in both the gradient-based training phase and the agent's decision-making process. The experimental results demonstrate the superiority of our meta-attention method in various continuous control tasks, which are based on the Off-Policy Actor-Critic methods including DDPG, TD3, and SAC.

1 INTRODUCTION

Reinforcement Learning (RL) algorithms based on the Actor-Critic framework have achieved considerable success in many areas such as games, robot control, and planning. Compared with on-policy methods, off-policy methods possess more efficient sampling since they do not require new samples to be collected for each gradient step and make better use of experience (Haarnoja et al., 2018a). However, even for off-policy methods, traditional reinforcement learning algorithms still have extremely low sample efficiency (Yu, 2018). Recently, meta-learning (Hospedales et al., 2020) has become topical as a paradigm to accelerate RL by learning an inductive bias from past experience. By learning aspects of the learning strategy, such as fast adaptation strategies (Finn et al., 2017) (Rakelly et al., 2019), losses (Zhou et al., 2020) (Bechtle et al., 2020), optimization strategies (Duan et al., 2016), exploration strategies (Gupta et al., 2018), hyperparameters (Bechtle et al., 2020), and intrinsic rewards (Zheng et al., 2018), meta-learning has significantly improved sample efficiency over standard RL algorithms.

Improving sample efficiency through attention mechanism has also been proved to be very effective in image-based reinforcement learning(Barati & Chen, 2019; Chen et al., 2019). The application of attention mechanism in multi-agent system(Parnika et al., 2021; Iqbal & Sha, 2019) and multi-object task(Team et al., 2021) also shows its powerful capabilities in information processing. However, in the existing works, attention mechanisms often have clear application scenarios, such as image-based control or multiple sources of information (multi-agent or multi-target). The effective combination of attention mechanisms with current algorithms in a state-based single-agent environment is still a problem to be investigated

In this paper, we propose a meta attention method based on the attention mechanism. In the human decision-making process, people often modify their concepts based on the feedback and results to obtain a better decision. Inspired by this decision-making process, we use meta attention to adapt the features generated by the policy network based on the evaluation of value network. Our work differs from current attention-based work in that our meta attention approach works only within the Actor-Critic framework and does not depend on a specific scenario. We formalize the meta-learning process as a bi-level optimization problem. Our approach can be flexibly combined with various algorithms by using meta attention as a meta-learner and optimizing meta attention in the outer layer. Unlike the existing meta-learning methods, our meta attention approach can improve the performance of agent through gradients in the training stage and obtain better actions by adjusting the features in the execution stage.

We evaluated the proposed meta attention method in a series of continuous control tasks in Gym and Roboschool, including three 3D robot control tasks, two 2D control tasks, and one classic control task based on DDPG(Lillicrap et al., 2016), TD3(Fujimoto et al., 2018) and SAC(Haarnoja et al., 2018b). Besides, we also discussed the changes and impact caused by modifying the actor features through meta attention. Experimental results how that our meta-attention approach is not only effective in accelerating the learning progress of the agent in the training phase, but also improves the actions in the execution phase, further enhancing the performance of the agent.

2 Related Work

Attention Mechanism Attention is a behavioral and cognitive process of selectively attending to a discrete aspect of information, whether subjective or objective, while ignoring other perceptible information(de Santana Correia & Colombini, 2021). Typically, attention mechanism is mainly applied in the fields of computational vision and natural language processing. Accordingly, although the implementation methods are different, the application of attention mechanism in reinforcement learning is mainly focused on video games(Wu et al., 2021; Chen et al., 2019; Barati & Chen, 2019; Mott et al., 2019)(Manchin et al., 2019). Other works such as Peng et al. (2020) proposed a dynamic attention model with a dynamic encoder-decoder architecture, which dynamically explores node features to efficiently exploit hidden structural information at different construction steps. Li et al. (2021) applied the attention mechanism to generate feature vectors and input them into value and policy head during the feature extraction phase of PPO and PPG.Jiang & Lu (2018); Iqbal & Sha (2019); Mao et al. (2019) employed a multi-head attention mechanism to make one agent selectively pay attention to information from other agents. Team et al. (2021) used the attention mechanism to match multiple goals with the hidden state of the current state to obtain the goal-attention hidden state under different goals. This approach allows the agent to predict the expected return obtained by attending to a goal at the end of a scene.

Meta Reinforcement Learning Meta-learning is most often understood as learning to learn, which refers to learning from historical information or multiple learning episodes to improve learning algorithms. Since works which fed historical trajectories into Recurrent Neural Network to flesh out task-level information(Wang et al., 2016; Duan et al., 2016), various meta-learning methods have been proposed to strengthen agent performance. Houthooft et al. (2018); Kirsch et al. (2020); Zhou et al. (2020) used meta-learning methods to learn a loss function rather than artificial design to improve performance of agent in single or multiple tasks.Gupta et al. (2018); Stadie et al. (2018); Xu et al. (2018a) employed meta-learning methods to learn exploration instead of traditional exploration methods.Finn et al. (2017) meta learned a good model initialization of the model that can be quickly adapted to different tasks. Xu et al. (2018b) improved the performance of the agent by meta-learning discount factors.Rakelly et al. (2019)Fakoor et al. (2020) treats meta information as an unobservable state of Partially Observable Markov Decision Process, further improves the agent's performance in multi task learning. Although the attention mechanism may not have an explicit meta-learning object, it can also be considered as the king of the meta-learning method.

Bi-level Optimization Generally, meta-learning can be formalized as a bi-level optimization(BLO) problem. However, the solution of BLO problems is often challenging. Franceschi et al. (2018) proposed a framework for approximating the solution of BLO problems using the gradient method. Since then, many works based on the gradient methods to optimize meta leaner have successfully proved the feasibility of gradient method, such asLi et al. (2019); Finn & Levine (2018); Lian et al. (2020); Flennerhag et al. (2020). For the RL problem, Zhou et al. (2020) optimize the meta critic, the upper-level of a BLO, as an intrinsic motivation by gradient method. Kirsch et al. (2020) enables a population of agents to use and improve a single parameterized objective function through gradient learning on different tasks. Rajeswaran et al. (2019) proposed an implicit MAML algorithm by drawing upon implicit differentiation that effectively decouples the meta-gradient computation from the selection of an inner loop optimizer. Liu et al. (2019) proposed a surrogate objective function TMAML, which incorporates control variables into gradient estimation through automatic differentiation and improves the quality of the gradient estimation by reducing the variance without introducing bias.

3 METHODOLOGY

3.1 OFF-POLICY ACTOR-CRITIC

In general, the reinforcement learning task can be considered as finding the optimal policies in Markov Decision Processes (MDPs). The MDP is defined by a tuple $(\mathbb{S}, \mathbb{A}, \mathbb{P}, \mathbb{R})$, where $\mathbb{S}$ is a set of states, $\mathbb{A}$ is a set of actions, $\mathbb{P}$ is a set of probabilities to switch from a state s to s' for a given set of action a, and $R : \mathbb{S} \times \mathbb{A} \rightarrow \mathbb{R}$ is a scalar reward function. In the Actor-Critic framework, the policy network(Actor) $\pi_\phi(s)$ and the value network(Critic) $Q_\theta(s, a)$ are parameterized by a neural network respectively. At each time t, the agent receives an observation s_t and takes a action a_t based on its policy $\pi : \mathbb{S} \rightarrow \mathbb{A}$, then receives a reward r_t and a new state s_{t+1}. The tuple (s_t, a_t, r_t, s_{t+1}) describes a state transition and will be stored in a reply buffer $\mathcal{D}$ for off-policy learning. The objective of RL is to find the optimal policy π_ϕ to maximizes the expected cumulative return J:

$$J(\phi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R\left(s_t, a_t\right) \mid a_t \sim \pi_\phi\left(\cdot \mid s_t\right)\right] \tag{1}$$

Where γ is the discount factor, and $J(\phi)$ also can be written as the expected value for the Q-function:

$$J(\phi) = \mathbb{E}_{s\sim p_\pi} Q_\theta(s, a)|_{a_t \sim \pi_\phi(\cdot|s_t)}, \tag{2}$$

Where Q-function is the expected discounted sum of rewards following visitation at state s and execution of action a, and p_π is the state distribution induced by policy π. For off-policy Actor-Critic architectures such as DDPG, TD3 and SAC, the loss for the actor provided by the critic may be different, but the Q-function will be learned by minimizing the loss of the below equation as same:

$$\begin{aligned} L_\theta^{Critic} &= \mathbb{E}_{s_t \sim p_\pi, a_t \sim \pi_\phi}\left[\left(Q_\theta\left(s_t, a_t\right) - y_t\right)^2\right] \\ y_t &= r_t + \gamma Q_{\theta'}\left(s_{t+1}, \pi_{\phi'}\left(s_{t+1}\right)\right), \end{aligned} \tag{3}$$

where ϕ' and θ' represent the target network for the critic and the actor respectively. The actor loss usually differs in details according to different algorithms, however , they all follow a form of the following formula:

$$L_\phi^{Actor} = -J(\phi) = -\left.\mathbb{E}_{s\sim p_\pi} Q_\theta(s, a)\right|_{a=\pi_\phi(s)} \tag{4}$$

3.2 META ATTENTION METHOD

The attention mechanism has been widely used in image recognition and natural language processing since it can model the human pattern recognition process. It allocates attention to the important part of information while automatically ignores low-value features. According to (Vaswani et al., 2017), attention function can be described as mapping a query and a set of key-value pairs to an output, where the output is computed as a weighted sum of the values, with the weight assigned to each value being computed by a compatibility function of the query with the corresponding key:

$$\text{Attention(Query, Key, Value)} = \sum_i \text{Similarity } \left(Query_i, Key_i\right) * \text{ Value }_i$$

In order to introduce the attention mechanism into the Actor-Critic framework, we first split the actor and critic network into two parts . We take the last layer of actor network $\hat{\pi}$ as an action net to produce actions and the rest as actor feature net $\bar{\pi}(s)$ to extract features, with the entire policy network being denoted as $\pi_\phi(s) = \hat{\pi}(\bar{\pi}(s))$. Similarly, the value network can also be divided into a critic net $\hat{Q}$ and a critic feature net $\bar{Q}(s)$, and the whole value network can be expressed as $Q_\theta(s) = \hat{Q}(\bar{Q}(s, a))$. We use the feature net as encoder to obtain Query, Key and Value by the following formula:

$$\text{Querry} = \bar{\pi}(s) \quad \text{Key} = \bar{Q}(s, \pi_\phi(s)) \quad \text{Value} = \bar{\pi}(s)$$

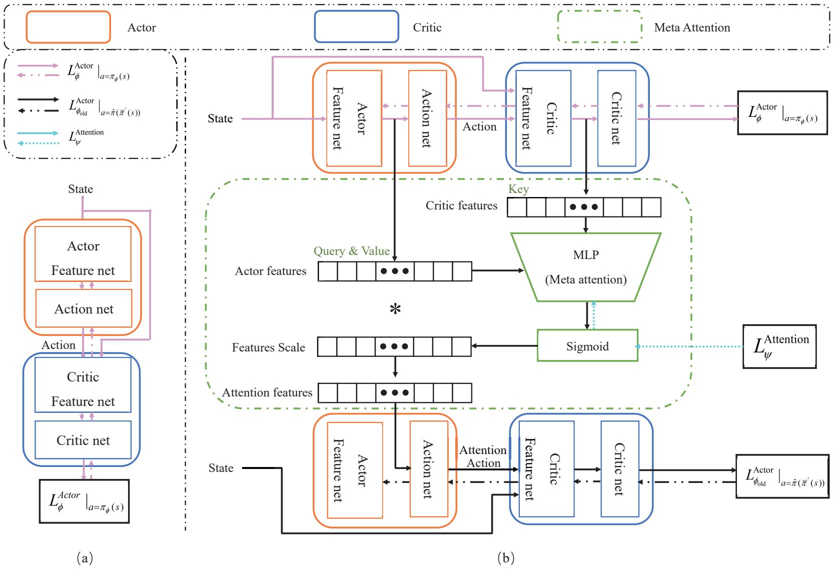


Figure 1: (a) Traditional Actor-Critic framework. (b) The meta attention Actor-Critic framework. The part circled by the orange box represents the actor, the part circled by the blue box represents the critic, and the part circled by a green dashed line represents the attention method. There is two orange(blue) boxes in (b), but they refer to the same actor(critic). Lines with different colors represent the update process dominated by different losses.

We input Query(actor features) and Key(critic features) into meta attention network $f_\psi(Query, Key)$ (a three-layers MLP parameterized by ψ) to calculate the similarity of each feature dimension. To enhance or reduce the features in specific corresponding dimensions, we multiply the output after the sigmoid function by 2 to obtain the feature scale. In this paper, we use $\circ$ to denote the Hadamard product, and by calculating the Hadamard product $\circ$ of Value(actor features) and scale, we obtain the attention features $\bar{\pi}'(s)$ after modifying some dimensions:

$$\text{Attention Features} = \bar{\pi}'_\psi(s) = f_\psi(\bar{\pi}(s), \bar{Q}(s, \pi_\phi(s))) \circ 2 \circ \bar{\pi}(s) \tag{5}$$

This calculation process corresponds to the part in the green dotted line box in Figure 1 (b).

There are also two critical problems in the optimization process. 1) how to affect the agent's decision-making and training process through attention features; and 2) how to optimize the meta attention network to achieve the correct matching Query(actor features) and Key(critic features) to generate the proper feature scales. To address these issues, we formalize the entire optimization process as a bi-level optimization problem, referring to the meta attention method as the outer level, and to the task performed by the agent as the inner level:

$$\psi^* = \arg\min_\psi L_\psi^{\text{Attention}}(\mathcal{D}; \phi^*; \psi) \tag{6}$$

$$\text{s.t. } \phi^* = \arg\min_\phi [L_\phi^{Actor}(\mathcal{D}; \phi | a \sim \pi_\phi(s)) + L_\phi^{Actor}\big(\mathcal{D}; \phi; \psi | a \sim \hat{\pi}(\bar{\pi}'_\psi(s))\big], \tag{7}$$

where $L_\psi^{\text{Attention}}$ is the meta optimization objective, and L_ϕ^{Actor} is the actor loss in the form of Eq.(4).

For the first problem, we first implement a traditional back-propagate dominated by the actor loss $L_\phi^{\text{Actor}}|_{a=\pi_\phi(s)}$ on training data d_{trn}:

$$\phi_{old} = \phi - \eta \frac{\partial L_\phi^{\text{Actor}}(d_{trn} | a \sim \pi_\phi(s))}{\partial \phi}, \tag{8}$$

Where η is the learning rate, this process corresponds to the upper part of Figure 1 (b) and the first step of meta-training in Algorithm1. Then, we generate a new attention action $a =$

$\hat{\pi}(\bar{\pi}'_{\psi}(s))$ from the attention features through action net and feed into the critic to get the loss $L_{\phi}^{\text{Actor}}(\mathcal{D};\phi;\psi)\,|_{a=\hat{\pi}(\bar{\pi}'_{\psi}(s))}$ for back-propagation:

$$\phi_{new} = \phi_{old} - \eta \frac{\partial L_{\phi_{old}}^{\text{Actor}}\left(d_{trn};\psi|a \sim \hat{\pi}(\bar{\pi}'_{\psi}(s))\right)}{\partial \phi} \tag{9}$$

This allows the agent to obtain better actions by simply modifying the features compared to the original action without increasing the batch size, strengthening the agent's learning of good actions and reducing the probability of producing poor actions after the gradient step. This process corresponds to the lower part of (b) in Figure 1 and the second step of meta-training in Algorithm1.

For the second problem, we have two basic assumptions: 1) a good meta attention network will inevitably generate feature scales that can generate actions with higher value since it better associates the relationship between actor features and critic features; 2) in the process of back-propagation, high-value actions enhance actors more than low-value actions because good actions will strengthen the actor's tendency to make such choices, while bad actions cause the actor to try other actions. Furthermore, since the meta attention network is involved in the back propagation process, we use the utility of meta attention in this process as the attention loss $L_{\psi}^{\text{Attention}}$ and the attention loss on validation data d_{val} is defined as:

$$L_{\psi}^{\text{Attention}} = \tanh\left(L_{\phi_{new}}^{\text{Actor}}\left(d_{val}|a \sim \pi_{\phi_{new}}(s)\right) - L_{\phi_{old}}^{\text{Actor}}\left(d_{val}|a \sim \pi_{\phi_{old}}(s)\right)\right) \tag{10}$$

Under this definition, when performing gradient descent updates, it is ensured that the meta attention is always updated along the direction that improves the performance of the agent. This process corresponds to the blue line part of (b) in Figure 1 and the meta-test in Algorithm1.

3.3 Beyond the Gradient

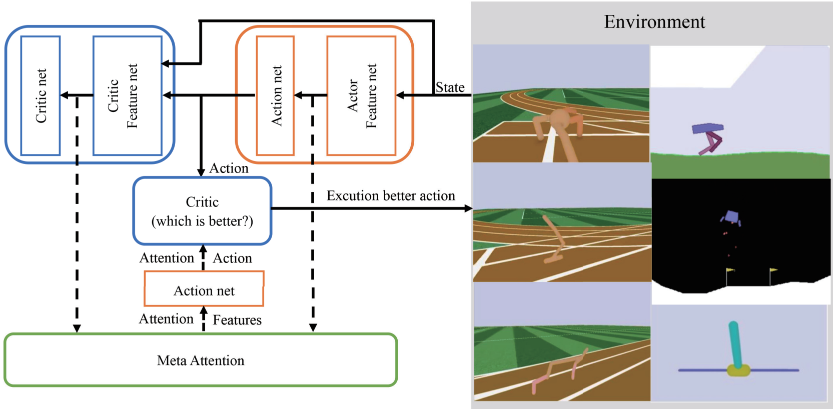


Figure 2: When the agent interacts with the environment, in addition to the action produced by the traditional Actor-critic framework, meta attention also attempts to make further adjustments to the initially produced results in the hope of achieving better results.

Unlike previous gradient-based meta-learning approaches, our proposed meta attention method can affect the learning process of the agent through the gradient and participate in each of their decisions. In the traditional Actor-Critic framework, the critic estimates each action's value based on the current state, which provides a criterion for evaluating the action produced by an actor. The Critic and the trained meta attention make it possible for meta attention methods to participate in decision-making.

When people are given the rules for scoring in human decision-making, they often modify their previous ideas to obtain better results. Similarly, we combine the attention method with the critic to introduce this decision-making method into the AC framework. For the deterministic policy, when an agent is given a state from the environment, we can get another attention action by adjusting the actor features in addition to the actions directly produced by the traditional Actor-Critic framework. Then the action with the higher Q value given by the critic is executed. This decision process is shown in Figure 2 and corresponds to the environment step in Algorithm 1.

This method of action selection is like an actor dancing once more after a performance, getting another score by adapting his or her actions based on the feedback given by the critics, and choosing which set of actions to be performed formally by comparing the two scores.

For stochastic policies such as SAC, their actions under a continuous control task are typically sampled from a Gaussian distribution. Although it is meaningless to compare sampled actions, we can still compare and modify the actor features in terms of the mean value of Gaussian distribution. Another attentional action is then sampled from the Gaussian distribution and the parameters (mean and variance) generated by the attentional features are modified.

Although the Q-value given by the critic may be inaccurate in the early stage of training, and off-policy algorithms typically have serious over estimation problem, these issues are beyond the scope of this paper.

Algorithm 1 Meta attention for off-policy Actor-Critic

```
Initialized parameters $\theta, \phi, \psi$, $D = \emptyset$ and learning rate $\eta, \lambda$
  for each iteration do
    for each environment step do
      if the actor dances twice then
        $a_t \sim \pi_\phi(s_t)$ or $a_t \sim \hat{\pi}(\bar{\pi}'_\psi(s))$          % Select the action with higher Q-value
      else
        $a_t \sim \pi_\phi(s_t)$                                   % Traditional action generation process
      end if
      $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t), r_t$                     % Observe reward $r_t$ and new state $s_{t+1}$
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r_t, s_{t+1})\}$         % Store the transition in the replay buffer
    end for
    for each gradient step do
      Sample mini-batch $d_{trn}$ from $\mathcal{D}$
      $L_\theta^{Critic} \leftarrow$ Eq.(3)
      $\theta \leftarrow \theta - \eta \nabla_\theta L_\theta^{Critic}$                     % Critic update
      meta-training:
      $\phi_{old} \leftarrow$ Eq.(8)                                 % Vanilla Actor update
      $\phi_{new} \leftarrow$ Eq.(9)                                 % Actor update based on attention action
      meta-test:
      $L_\psi^{\text{Attention}} \leftarrow$ Eq.(10)                          % Attention loss
      meta-optimisation:
      $\phi \leftarrow \phi_{new}$                                      % Update the actor parameters
      $\psi \leftarrow \psi - \lambda \nabla_\psi L_\psi^{\text{Attention}}$                 % Update the meta attention parameters
    end for
  end for
```

4 Experiment

4.1 Setup

To evaluate our meta attention framework, we selected four 3D continuous robot control tasks in Roboschool and two 2D continuous robot control tasks in Gym without modifying the original environments or rewards. Although MuJoCo is widely used to test various reinforcement learning algorithms, Roboschool provides a more realistic and difficult robot control task than MuJoCo. Compared with MuJoCo, Roboschool reduces the alive bonus and increases drive costs to encourage movement in a low-energy manner. Roboschool also made other adjustments, such as encouraging the Ant robot to have two or more legs on the ground to ensure that the robot moves in a more natural posture.

We applied our meta attention framework to DDPG, TD3 and SAC as the basic algorithms. Algorithms that use meta attention methods only in the training stage will be annotated as **_MATT** (such as DDPG_MATT), while algorithm that uses the meta attention method in both the training stage and agent decision-making process will be annotated as **_MATT_DT**(such as DDPG_MATT_DT).

We also compare our **_MATT** and **_MATT_DT** with another SOTA work (Zhou et al., 2020) which proposed Meta Critic (**_MC**) to enforce the basic algorithm.

All our codes are built on the code provided by Zhou et al. (2020). We ran tasks for 1 million or 5 hundred thousands game steps depending on the environment, and we evaluate our policy over 10 episodes without exploration every 1000 game steps. All results for each task are averaged over 10 random seeds (trials) and network initializations. 95% confidence intervals are shown as shaded regions over time steps. Following Fujimoto et al. (2018), curves are smoothed evenly for clarity (window_size=10).

Experiments in all environments had the same network structure and hyperparameters, with a learning rate of 0.003 for DDPG and TD3 but 0.001 for SAC. All experiments are performed on 4 servers with 4 NVIDIA-TITAN-V GPU each and 3 servers with 6 NVIDIA-TITAN-V GPUs. Each server is equipped by 2 Intel(R) Xeon(R) Gold 5218 CPUs.

4.2 Evaluation of Meta Attention Method

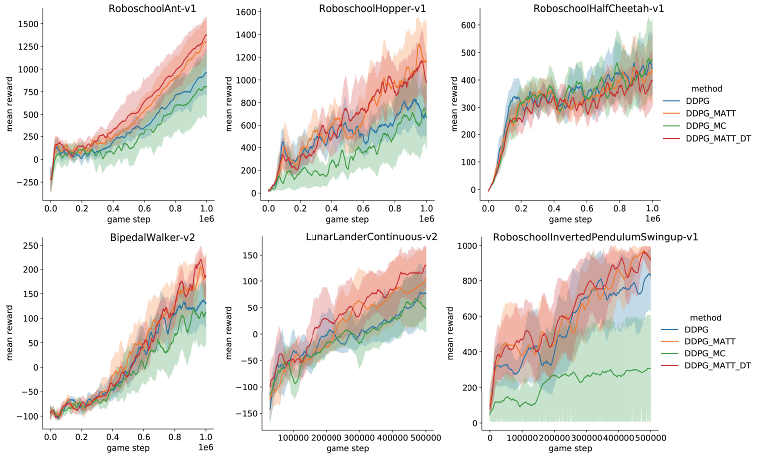


Figure 3: The learning curves of DDPG and enhancement algorithms. The shaded region represents the 95% confidence interval.

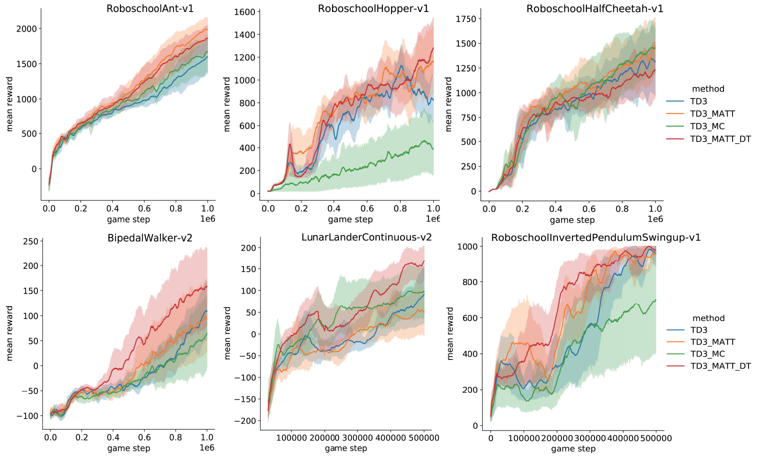


Figure 4: The learning curve of TD3 and enhancement algorithms. The shaded region represents the 95% confidence interval.

Deterministic policy Figure 3 and Figure 4 demonstrate the comparison between the vanilla algorithms and their enhancive versions based on DDPG and TD3. From the figures, we observes that **_MATT** and **_MATT_DT** achieve significant performance improvements in all but the *RoboschoolHalfCheetah-v1* environment. **DDPG_MATT** perform almost as well as **DDPG_MATT_DT** under the environments *BipedalWalker-v2*, *LunarLanderContinuous-v2* and *RoboschoolInvertedPendulumSwingup-v1*. But in these enviroment, **TD3_MATT_DT** perform significantly better than **TD3_MATT**, and we speculate that this phenomenon may be related to more accurate Q-values.

Similar to Meta Critic, **_MATT** and **_MC** only work in the training stage and influence the actor through gradients, but **DDPG_MC** performs even worse than the original algorithm in four environments, while **DDPG_MATT** shows performance improvements in all environments except *RoboschoolHalfCheetah-v1*. **TD3_MATT** also shows high performance in more environment than **TD3_MC**. From the data provided in Zhou et al. (2020), we can also find a similar situation: the performance of **_MC** decreases in the first 1 million steps in deterministic policy based algorithms. This also shows that our meta attention method completes the meta-learning process faster than Meta Critic.

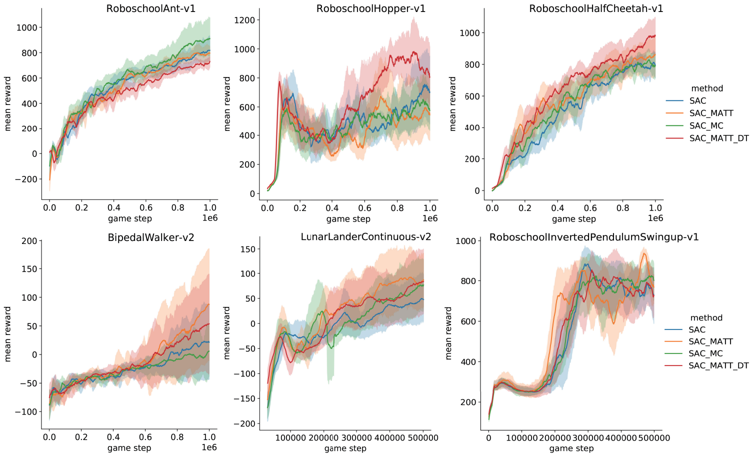


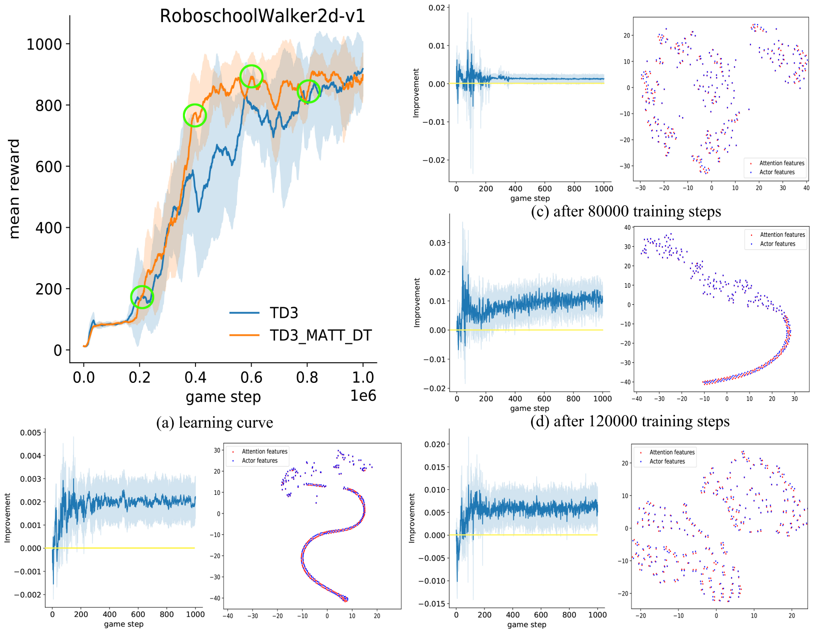
Figure 5: The learning curve of SAC and enhancement algorithms. The shaded region represents the 95% confidence interval.

Stochastic policy Figure 5 compares the SAC and its enhancive versions. Although **SAC_MATT** has only achieved performance improvement in environments *BipedalWalker-v2* and *LunarLanderContinuous-v2*, **SAC_MATT_DT** still achieves performance improvements in all environments other than *RoboschoolAnt-v1* and *RoboschoolInvertedPendulumSwingup-v1*. This suggests that our method is also effective in stochastic policy, but the improvement in stochastic policy is less pronounced than in Deterministic policy. We speculate that, compared with deterministic strategies, sampled actions in random strategies bring more uncertainty to the learning of meta attention. The performance of **SAC_MATT** and **SAC_MC** in the four environments is similar, which is the same as or slightly higher than the original algorithm. The performance of the first 1 million steps also matches the data provided in the Zhou et al. (2020).

4.3 Further Analysis

In order to analyze the effect of meta attention on the features, we ran 5 random seeds(trails) in *RoboschoolWalker2d-v1* environment. In each trial, we ran a total of 1 million game steps and evaluate our policy over 10 episodes without exploring every 1000 game steps. Figure 6 (a) shows an overview of this experiment, and the shaded region represents the 95% confidence interval.

We recorded the Q-values of attention and actor features during the evaluation stage to verify the impact of meta attention on performance. In the evaluation phase, we took 1 million consecutive Q-values of the actions generated by attention features and actor features. These 1 million samples correspond to 1 million time steps in the evaluation stage or 10 evaluations. We used the error between Q-values of the attention action and vanilla action as performance improvements brought



(a) learning curve

(b) after 40000 training steps

(c) after 80000 training steps

(d) after 120000 training steps

(e) after 160000 training steps

Figure 6: (a) Learning curves, the shaded part represents the 95% confidence interval; (b-e) The results obtained by sampling at different stages (green circle in (a)) of training. The left part represent the difference in Q-values of actions generated by attention features and actor features, respectively The right part represents t-SNE visualization of attention features and actor features, where red dots represent attention features and blue dots represent actor features.

by meta attention. We sampled at four different time points in the training phase, and these results are represented in Figure 6 (b-e).

To analyze the diversity of attention features and actor features, we sampled 250 attention features and actor features on continuous 250 time steps during evaluation on one trail. We used t-SNE to visualize these total 500 features. Similarly, we sampled at five different time points in the training stages, and these results are shown in the right part of Figure 6 (b-e).

Figure 6 (b-e) shows the results of sampling at approximately 40000, 80000, 120000, 160000 training steps on one trail of 1 million time steps or 200000 training steps. In terms of improvement(left part of Figure 6 (b-e)), the actions obtained after the modification of our meta attention usually achieved higher Q-values at different stages of training. In terms of the distribution (right part of Figure 6 (b-e)), attention features have changed in most cases compared with actor features, as shown by the apparent separation of red points and blue points. This suggests that our meta-attention approach is effective in modifying features and improving performance in the current state throughout the training phase.

5 Conclusion

In this paper, we present the meta attention method, a derivative-based meta learner for off-policy Actor-Critic reinforcement learning methods. By modifying the actor features, we speed up the training process and enable agents to make better decisions when interacting with the environment. The experimental results demonstrate the effectiveness of our method in both stochastic policy and deterministic policy. Moreover, our method can be flexibly integrated into a variety of contemporary off-policy Actor-Critic methods to boost performance. In future work, we will use the multi-head attention to improve the meta learner and decouple the meta attention method from the value function to reduce the impact of the over estimation problem.

REFERENCES

Elaheh Barati and Xuewen Chen. An actor-critic-attention mechanism for deep reinforcement learning in multi-view environments. In Sarit Kraus (ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 2002–2008. ijcai.org, 2019. doi: 10.24963/ijcai.2019/277. URL https://doi.org/10.24963/ijcai.2019/277.

Sarah Bechtle, Artem Molchanov, Yevgen Chebotar, Edward Grefenstette, Ludovic Righetti, Gaurav S. Sukhatme, and Franziska Meier. Meta learning via learned loss. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, pp. 4161–4168. IEEE, 2020. doi: 10.1109/ICPR48806.2021.9412010. URL https://doi.org/10.1109/ICPR48806.2021.9412010.

Yilun Chen, Chiyu Dong, Praveen Palanisamy, Priyantha Mudalige, Katharina Muelling, and John M. Dolan. Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019, Macau, SAR, China, November 3-8, 2019*, pp. 3697–3703. IEEE, 2019. doi: 10.1109/IROS40897.2019.8968565. URL https://doi.org/10.1109/IROS40897.2019.8968565.

Alana de Santana Correia and Esther Luna Colombini. Attention, please! A survey of neural attention models in deep learning. *CoRR*, abs/2103.16775, 2021. URL https://arxiv.org/abs/2103.16775.

Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl^2: Fast reinforcement learning via slow reinforcement learning. *CoRR*, abs/1611.02779, 2016. URL http://arxiv.org/abs/1611.02779.

Rasool Fakoor, Pratik Chaudhari, Stefano Soatto, and Alexander J. Smola. Meta-q-learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SJeD3CEFPH.

Chelsea Finn and Sergey Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=HyjC5yWCW.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 2017. URL http://proceedings.mlr.press/v70/finn17a.html.

Sebastian Flennerhag, Andrei A. Rusu, Razvan Pascanu, Francesco Visin, Hujun Yin, and Raia Hadsell. Meta-learning with warped gradient descent. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rkeiQlBFPB.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1563–1572. PMLR, 2018. URL http://proceedings.mlr.press/v80/franceschi18a.html.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1582–1591. PMLR, 2018. URL http://proceedings.mlr.press/v80/fujimoto18a.html.

Abhishek Gupta, Russell Mendonca, Yuxuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 5307–5316, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/4de754248c196c85ee4fbdcee89179bd-Abstract.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018a.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *CoRR*, abs/1812.05905, 2018b. URL http://arxiv.org/abs/1812.05905.

Timothy M. Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J. Storkey. Meta-learning in neural networks: A survey. *CoRR*, abs/2004.05439, 2020. URL https://arxiv.org/abs/2004.05439.

Rein Houthooft, Yuhua Chen, Phillip Isola, Bradly C. Stadie, Filip Wolski, Jonathan Ho, and Pieter Abbeel. Evolved policy gradients. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 5405–5414, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/7876acb66640bad41f1e1371ef30c180-Abstract.html.

Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2961–2970. PMLR, 2019. URL http://proceedings.mlr.press/v97/iqbal19a.html.

Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 7265–7275, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/6a8018b3a00b69c008601b8becae392b-Abstract.html.

Louis Kirsch, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Improving generalization in meta reinforcement learning using learned objectives. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=S1evHerYPr.

Yiying Li, Yongxin Yang, Wei Zhou, and Timothy M. Hospedales. Feature-critic networks for heterogeneous domain generalization. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3915–3924. PMLR, 2019. URL http://proceedings.mlr.press/v97/li19l.html.

Yunfei Li, Tao Kong, Lei Li, Yifeng Li, and Yi Wu. Learning to design and construct bridge without blueprint. *CoRR*, abs/2108.02439, 2021. URL https://arxiv.org/abs/2108.02439.

Dongze Lian, Yin Zheng, Yintao Xu, Yanxiong Lu, Leyu Lin, Peilin Zhao, Junzhou Huang, and Shenghua Gao. Towards fast adaptation of neural architectures with meta learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=r1eowANFvr.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL `http://arxiv.org/abs/1509.02971`.

Hao Liu, Richard Socher, and Caiming Xiong. Taming MAML: efficient unbiased meta-reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4061–4071. PMLR, 2019. URL `http://proceedings.mlr.press/v97/liu19g.html`.

Anthony Manchin, Ehsan Abbasnejad, and Anton van den Hengel. Reinforcement learning with attention that works: A self-supervised approach. In Tom Gedeon, Kok Wai Wong, and Minho Lee (eds.), *Neural Information Processing - 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12-15, 2019, Proceedings, Part V*, volume 1143 of *Communications in Computer and Information Science*, pp. 223–230. Springer, 2019. doi: 10.1007/978-3-030-36802-9_25. URL `https://doi.org/10.1007/978-3-030-36802-9_25`.

Hangyu Mao, Zhengchao Zhang, Zhen Xiao, and Zhibo Gong. Modelling the dynamic joint policy of teammates with attention multi-agent DDPG. In Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor (eds.), *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pp. 1108–1116. International Foundation for Autonomous Agents and Multiagent Systems, 2019. URL `http://dl.acm.org/citation.cfm?id=3331810`.

Alexander Mott, Daniel Zoran, Mike Chrzanowski, Daan Wierstra, and Danilo Jimenez Rezende. Towards interpretable reinforcement learning using attention augmented agents. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 12329–12338, 2019. URL `https://proceedings.neurips.cc/paper/2019/hash/e9510081ac30ffa83f10b68cde1cac07-Abstract.html`.

P. Parnika, Raghuram Bharadwaj Diddigi, Sai Koti Reddy Danda, and Shalabh Bhatnagar. Attention actor-critic algorithm for multi-agent constrained co-operative reinforcement learning. In Frank Dignum, Alessio Lomuscio, Ulle Endriss, and Ann Nowé (eds.), *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, pp. 1616–1618. ACM, 2021. URL `https://dl.acm.org/doi/10.5555/3463952.3464178`.

Bo Peng, Jiahai Wang, and Zizhen Zhang. A deep reinforcement learning algorithm using dynamic attention model for vehicle routing problems. *CoRR*, abs/2002.03282, 2020. URL `https://arxiv.org/abs/2002.03282`.

Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. Meta-learning with implicit gradients. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 113–124, 2019. URL `https://proceedings.neurips.cc/paper/2019/hash/072b030ba126b2f4b2374f342be9ed44-Abstract.html`.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5331–5340. PMLR, 2019. URL `http://proceedings.mlr.press/v97/rakelly19a.html`.

Bradly C. Stadie, Ge Yang, Rein Houthooft, Xi Chen, Yan Duan, Yuhuai Wu, Pieter Abbeel, and Ilya Sutskever. Some considerations on learning to explore via meta-reinforcement learning. *CoRR*, abs/1803.01118, 2018. URL http://arxiv.org/abs/1803.01118.

Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michaël Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Roberta Raileanu, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. Open-ended learning leads to generally capable agents. *CoRR*, abs/2107.12808, 2021. URL https://arxiv.org/abs/2107.12808.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Jane X. Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Rémi Munos, Charles Blundell, Dharshan Kumaran, and Matthew Botvinick. Learning to reinforcement learn. *CoRR*, abs/1611.05763, 2016. URL http://arxiv.org/abs/1611.05763.

Haiping Wu, Khimya Khetarpal, and Doina Precup. Self-supervised attention-aware reinforcement learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 10311–10319. AAAI Press, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17235.

Tianbing Xu, Qiang Liu, Liang Zhao, and Jian Peng. Learning to explore with meta-policy gradient. *CoRR*, abs/1803.05044, 2018a. URL http://arxiv.org/abs/1803.05044.

Zhongwen Xu, Hado van Hasselt, and David Silver. Meta-gradient reinforcement learning. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 2402–2413, 2018b. URL https://proceedings.neurips.cc/paper/2018/hash/2715518c875999308842e3455eda2fe3-Abstract.html.

Yang Yu. Towards sample efficient reinforcement learning. In Jérôme Lang (ed.), *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pp. 5739–5743. ijcai.org, 2018. doi: 10.24963/ijcai.2018/820. URL https://doi.org/10.24963/ijcai.2018/820.

Zeyu Zheng, Junhyuk Oh, and Satinder Singh. On learning intrinsic rewards for policy gradient methods. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 4649–4659, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/51de85ddd068f0bc787691d356176df9-Abstract.html.

Wei Zhou, Yiying Li, Yongxin Yang, Huaimin Wang, and Timothy M. Hospedales. Online meta-critic learning for off-policy actor-critic methods. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/cceff8faa855336ad53b3325914caea2-Abstract.html.