

ATTENTION-BASED CURIOSITY-DRIVEN EXPLORATION IN DEEP REINFORCEMENT LEARNING

Patrik Reizinger and Márton Szemenyei*

Budapest University of Technology and Economics
Department of Control Engineering and Information Technology
Budapest, Hungary
rpatrik96@sch.bme.hu

ABSTRACT

Reinforcement Learning enables to train an agent via interaction with the environment. However, in the majority of real-world scenarios, the extrinsic feedback is sparse or not sufficient, thus intrinsic reward formulations are needed to successfully train the agent. This work investigates and extends the paradigm of curiosity-driven exploration. Our aim is to develop means for the better incorporation of state- and/or action-dependent information into existing intrinsic reward formulations. First, a probabilistic approach is taken to exploit the advantages of the attention mechanism, which is successfully applied in other domains of Deep Learning. Combining them, we propose new methods, such as Attention-aided Advantage Actor-Critic, an extension of the Actor-Critic framework. Second, another curiosity-based approach - Intrinsic Curiosity Module - is extended. The proposed model utilizes attention to emphasize features for the dynamic models within Intrinsic Curiosity Module, moreover, we also modify the loss function, resulting in a new curiosity formulation, which we call rational curiosity (RCM). The corresponding implementation can be found at <https://github.com/rpatrik96/AttA2C/>.

Index Terms— Reinforcement Learning, curiosity, exploration, attention

1. INTRODUCTION

Learning through reinforcement is a powerful concept, as it - similarly to unsupervised learning - does not need labeled samples, which are both time- and money-consuming to collect. Moreover, Reinforcement Learning (RL) is rather similar to the concept of human learning, since it defines the learning process as an interaction between an agent and its environment. Interestingly, in some cases, it was proven that an RL agent optimizes the same objective function as primates do, as noted in [1].

Speaking of optimization, an optimum can only be referred to w.r.t. some given criteria, thus RL has to define its own criterion, as it is the case for Artificial Intelligence (AI) generally. In optimization, we generally face the so-called bias-variance problem, which has a new designation in the RL domain: it is called the exploration-exploitation dilemma. Here, exploitation refers to preferring prior knowledge, i.e. bias. In some sense, exploitation can be thought of as a risk-minimizing way of actions. Nonetheless, this policy may not always be the best option. Clearly, some sort of compromise should be obtained between exploitation and exploration - which stands for a variance-preferring policy.

RL agents are generally trained based on the feedback, i.e. using rewards collected from the environment. However, the environment is not guaranteed to have an inner dynamics that rewards the agent in a way that corresponds with its goals. For example, consider an agent seeking a time-optimal solution and a non-negative reward. In this scenario, the agent will be reinforced to interact on an infinite time-horizon, thus, contradicting its goal. As the example shows, only having an extrinsic incentive may be impractical. Furthermore, rewards are in most real-life scenarios sparse, so it is more difficult for the agent to assign credit to useful actions and omit disadvantageous ones in the future.

To mitigate the above-mentioned conflict, several methods were developed to supplement the objective function of an RL agent to improve its performance. The contribution of this paper concerns the extension of existing methods for intrinsic reward formulation for sake of better incorporation of state-/action-dependent information. Before reviewing our work in detail, let us summarize the main contributions in this field. In model-based RL, the World model architecture [2] is one answer for challenges mentioned above. On the other hand, curiosity-based methods modify the loss function (or even the network architecture) by adding terms to incentivize exploration. In this topic, the Intrinsic Curiosity Module (ICM) module of [3] is a major contribution, which will be used extensively in this work to build upon. ICM introduces inner dynamics (forward and inverse) to quantify the prediction er-

*Supported by the ÚNKP-19-2 New National Excellence Program of the Ministry for Innovation and Technology

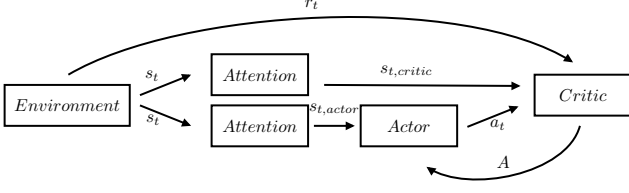


Fig. 1: The AttA2C architecture

ror of the next state and action, which is utilized as an intrinsic reward - a more extensive study can be found in [4]. Another promising work in this field is the notion of disagreement, which is an ensemble-based curiosity formulation [5].

In this paper, we explore the capabilities of applying attention [6] in order to incentivize exploration and improve generalization. We test our proposed methods using Atari games from OpenAI Gym [7]. Our main contributions are as follows:

- AttA2C, an attention-aided A2C variant,
- feature- and action-selective extension of the ICM [3],
- a rational curiosity formulation.

2. ATTENTION-BASED CURIOSITY

As the review of the literature shows, curiosity-based methods provide powerful extensions for the agents. In other (non-RL) domains[8], the same holds for the attention mechanism [6], which defines a tunable probability distribution. In this section, both curiosity-driven exploration and attention are combined, resulting in modifications of the Advantage Actor-Critic (A2C) [9] and the ICM [3]. Furthermore, we develop a new loss function formulation as well.

2.1. AttA2C

RL agents often include some variant of Actor-Critic network, which has two components: the actor predicts action probabilities, while the critic provides an estimate about how good the state is. Assuming an Actor-Critic network, we supplemented the architecture based on the Generative Adversarial Network (GAN)[10]–Actor-Critic correspondence [11], which is summarized in Table 1. Highlighting the only difference (typeset in bold in the table), i.e. the fact that while an Actor-Critic network feeds the same input/features to its both heads, the GAN does quite the opposite.

Taking this similarity a step further, we conjecture that separating the feature space into two parts can be advantageous, as different features may be useful for the actor and for the critic. The proposed architecture aims to overcome this disadvantage by utilizing attention. As shown in Figure 1, the input from the environment is fed through separate attention layers, thus both heads can put emphasis onto the most

GAN	Actor-Critic
Generator	Actor
Discriminator	Critic
Label	Reward
Input	State
Latent space	State

Table 1: The connection between GAN and Actor-Critic (AC) concepts [11]

important parts of the latent process, which is formulated as follows:

$$a_t = Actor(Attn_{\pi}(\phi(s_t))) \quad (1)$$

$$A_{\pi}(s_t, a_t) = Critic(Attn_A(\phi(s_t))), \quad (2)$$

where *Actor* (which is basically the policy π) and *Critic* denote the corresponding networks, ϕ is the feature transform of the input (if present), while *Attn* is a shorthand for the probability distribution of the attention mechanism. Subscripts denote that two separate attention layers are used to predict the policy π and the advantage function A .

2.2. Feature- and action selective ICM

Being able to separate the most useful features cannot only be advantageous for the Actor-Critic network, but also for the curiosity formulation. Thus, we introduce attention mechanism to ICM to make it action- and feature-selective. To do that, attention is applied onto the concatenation of the inputs of the forward and inverse models (denoted by the *fwd* and *inv* subscripts, respectively):

$$\hat{\phi}_{t+1} = Forward(Attn_{fwd}([\phi(s_t), a_t])) \quad (3)$$

$$\hat{a}_t = Inverse(Attn_{inv}([\phi(s_t), a_t])), \quad (4)$$

where $\hat{\cdot}$ stands for predicted values. The equations above describe the single attention case, but we also experimented with double attention, in which we swapped the order of concatenation and attention. In this case, the attention-weighted features and actions are concatenated. The reason for this second formulation was to separate the weighting between the two domains. The main advantage of the latter formulation could be that using double attention could ensure that both in the feature and action spaces there will be a subspace which is emphasized. This is not the case when using single attention,

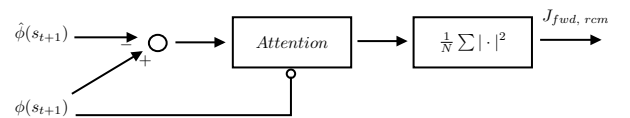


Fig. 2: Forward loss formulation of the RCM agent

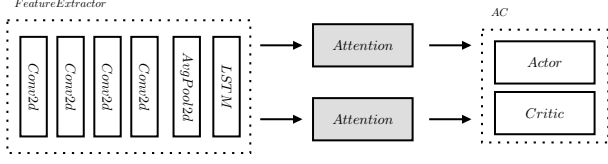


Fig. 3: The A2C/Attention-aided Advantage Actor-Critic (AttA2C) network (gray indicates optional layer)

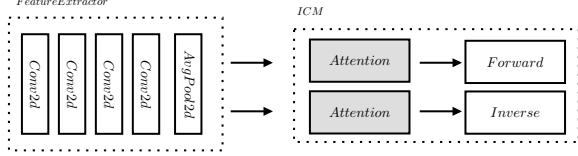


Fig. 4: The ICM/Rational Curiosity Module (RCM) network (gray indicates optional layer)

which can be problematic if the distributions are not in the same order of magnitude, implying that one domain is more important due to using different value scales.

2.3. Rational curiosity

The curiosity-driven exploration strategies of [3] and [5] represent a rather powerful approach to train agents in sparse reward settings. Nonetheless, these models are based on the assumption that enforcing curiosity-driven exploration is a good choice for every state and action, even though this is not true in every scenario. E.g. consider the noisy TV experiment of [4], where the agent has control over generating new (and thus, unexpected) instances of random noise, resulting in high values of the ICM/disagreement losses.

To classify the states based on the usefulness of curiosity, the state space $\mathcal{S}$ can be divided into two subsets: $\mathcal{S}^+$ and $\mathcal{S}^-$, where the former denotes the subset of states with useful curiosity, while the latter the harmful curiosity settings. Curiosity is termed as harmful if being driven by this curiosity does not ensure the fulfillment of the original objective, not even in the long run. I.e. sacrificing short-term rewards to develop general skills is useful, but overfitting random noise is not.

Thus, the rational objective - Rational Curiosity Module (RCM) - of curiosity-driven exploration would be to minimize the probability of being in states in $\mathcal{S}^-$, which corresponds to:

$$\min p(s_{t+1} \in \mathcal{S}^- | s_t, a_t), \quad (5)$$

i.e. it is intended to select actions which do not lead the agent into $\mathcal{S}^-$, independent from its actual state.

To achieve that, an attention network [6] is used on the forward loss function of the ICM module [3]. The inverse loss is not used, as for each action prediction, there is only one true action, where weighting has no effect. The new forward

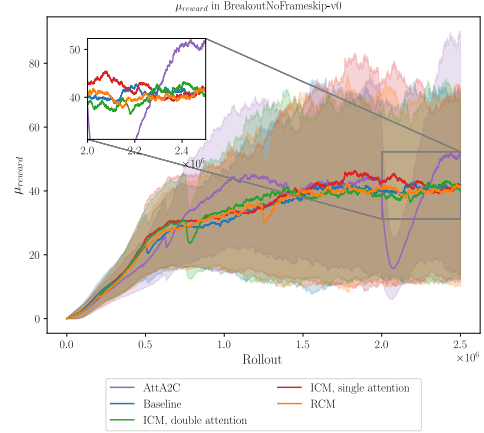


Fig. 5: Mean reward in the Breakout environment (v0)

loss term is illustrated in Figure 2, where the round-headed arrow denotes that $\phi(s_{t+1})$ is used as the control term in the attention layer. I.e. it determines the weighting of the Mean Squared Error (MSE) (shown as two steps, i.e. subtraction and squared mean) through a probability distribution. Thus, the loss function is modified in the following manner:

$$J_{rcm} = J_{A2C} + \beta \text{Attn}(J_{fwd}, \phi(s_{t+1})) + (1 - \beta) J_{inv}, \quad (6)$$

where RCM is the term for the new model, J_{A2C} is the objective function for the A2C network, J_{fwd} for the forward and J_{inv} for the inverse dynamics, weighted with a scalar β . This formula is motivated by the fact that $\phi(s_{t+1})$ encodes fully whether whether the state of the agent is in $\mathcal{S}^+$ or in $\mathcal{S}^-$. We hypothesize that this formulation can help to utilize curiosity only in situations where the agent can benefit from that on the long run, but omits curiosity otherwise.

2.4. Implementation

The proposed methods are implemented in PyTorch [12], the agents are based on the implementation of [3] (shown in Figure 3 and Figure 4). The extension is indicated with the gray boxes, as those layers are part of some of the proposed methods. Five configurations were implemented: AttA2C, single- and double-attention ICM and RCM, and the traditional ICM agent is used as a baseline [3].

For test purposes, the Atari environments of the OpenAI Gym [7] are used via the Stable-baselines [13] package. Three environments were chosen, i.e. Breakout, Pong and Seaquest (as in [5]), which provide single- (Pong, Seaquest) and multiplayer (Pong) tasks with one (Breakout, Pong) and multiple (Seaquest) objectives. For each of the three, both the deterministic (v4) and the stochastic (v0, with nonzero action repeat probability) variants were evaluated. All agents were trained on 4 environments in a parallel manner for 2,500,000 rollouts with 5 steps each, using 4 stacked frames.

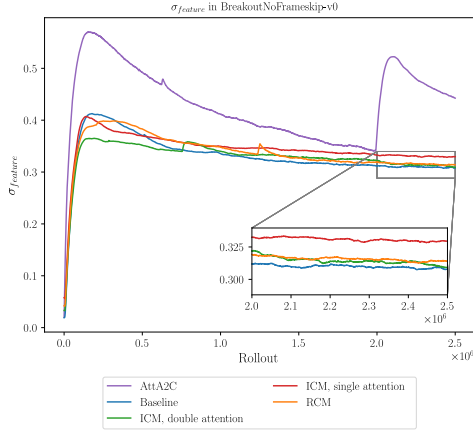


Fig. 6: Feature standard deviation in the Breakout environment (v0)

Agent	Mean normalized reward	Std. dev.
Baseline	92.33	4.98
AttA2C	88.01	13.36
$ICM_{1 \times Att}$	94.88	4.46
$ICM_{2 \times Att}$	92.17	5.68
RCM	95.31	6.48

Table 2: Comparison of the agents’ normalized mean performance (higher is better)

3. RESULTS

We monitored two metrics to compare performance and generalization: mean reward for the former, and mean standard deviation of the features for the latter. Due to lack of space, only the results for the more difficult, stochastic (v0) environments are shown here (the others are available on GitHub).

For Breakout (Figure 5, with a 1σ confidence interval), the selective versions of ICM had a consistently good performance. The AttA2C agent trained faster at the beginning, and after experiencing a jump in variance, it outperformed the other agents. The standard deviation of the feature space shown in Figure 6, the significantly higher values in case of AttA2C hold basically for every environment, but with smaller ”jumps”). In our experiments, agents (not concerning AttA2C for this statement) with higher standard deviation performed generally better. In case of Pong (Figure 7), the single-attention ICM performed as the best, followed by the RCM agent. In this case, AttA2C trained slower, but managed to achieve comparable rewards - the reason for this could be the smaller gradients due to attention between feature space and the actor/critic. The RCM agent was the best for the most complex environment, Seaquest, as Figure 8 shows. In this case, the selective ICM agents were overtaken by the original ICM, while AttA2C experienced much slower training.

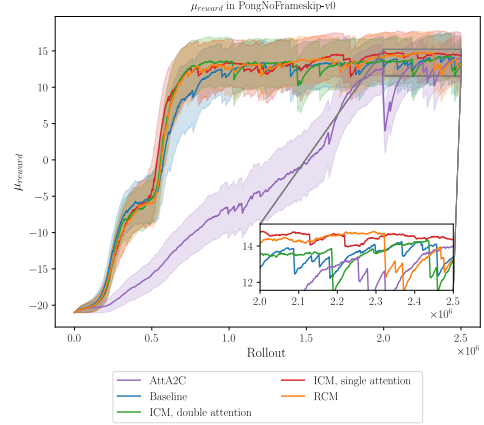


Fig. 7: Mean reward in the Pong environment (v0)

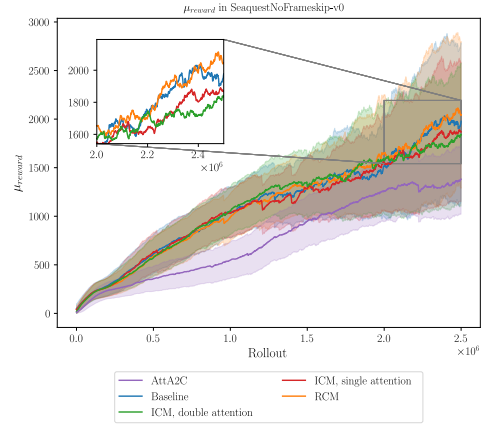


Fig. 8: Mean reward in the Seaquest environment (v0)

To be able to provide a concise but quantitative summary of the agents’ performance in all environments (including v0 and v4), we normalized the highest rewards of each agent, with 100 denoting the best performance. This way, we were able to compare the mean performances, summarized in Table 2 (including a 1σ confidence interval). As it shows, the RCM agent performed the best, followed by the single attention ICM. Note that AttA2C had both the lowest mean and the highest variance, mainly due to the good performance in Breakout, but a moderate one in the other scenarios.

4. DISCUSSION

This work investigated the paradigm of curiosity-driven exploration in RL, which was extended by the attention mechanism. We proposed three different methods to incorporate attention for utilizing curiosity in a selective manner. The new models were tested in the OpenAI Gym environment and have shown consistent improvement to the baseline models used for comparison.

5. REFERENCES

- [1] Richard S. Sutton and Andrew G. Barto, *Reinforcement learning: An Introduction*, 2017.
- [2] David Ha and Jürgen Schmidhuber, “World models,” *arXiv preprint arXiv:1803.10122*, 2018.
- [3] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell, “Curiosity-driven exploration by self-supervised prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 16–17.
- [4] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros, “Large-scale study of curiosity-driven learning,” *arXiv preprint arXiv:1808.04355*, 2018.
- [5] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta, “Self-supervised exploration via disagreement,” *arXiv preprint arXiv:1906.04161*, 2019.
- [6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” 2017.
- [7] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba, “OpenAI Gym,” 2016.
- [8] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He, “AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1316–1324.
- [9] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in *International Conference on Machine Learning*, 2016, pp. 1928–1937.
- [10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.
- [11] David Pfau, “Connecting GANs, Actor-Critic methods and multilevel optimization,” 2017.
- [12] Adam Paszke, Gregory Chanan, Zeming Lin, Sam Gross, Edward Yang, Luca Antiga, and Zachary Devito, “Automatic differentiation in PyTorch,” *Advances in Neural Information Processing Systems 30*, , no. Nips, pp. 1–4, 2017.
- [13] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu, “Stable baselines,” <https://github.com/hill-a/stable-baselines>, 2018.