MetaIRN: Metagradient Based Intrinsic Rewards Via Attention Network

Philipp Link
p.link@stud.uni-hannover.de

Julius Heidmann
julius.heidmann@stud.uni-hannover.de

Luan Liebig-Schultz
luan.liebig-schultz@stud.uni-hannover.de

Abstract

In this project, we train a PPO agent via meta-gradients, leveraging intrinsic rewards generated by an attention network to tackle Temporal Credit Assignment (TCA)-only tasks where memory is explicitly decoupled. Our objective is to quantify the contribution of attention-based intrinsic rewards that are optimized exclusively for maximizing extrinsic returns. In particular, we investigate whether this approach accelerates PPO training and focuses credit assignment on early key actions in fully observable settings. Our study reveals that the proposed intrinsic-reward PPO agent with an attention-based meta-gradient mechanism was unable to effectively capture the relevant temporal dependencies, leaving the impact of our approach on the TCA problem inconclusive.

1 Introduction

Reinforcement Learning agents often face substantial challenges when operating in environments with sparse reward structures. Since meaningful feedback in such environments is provided only rarely, it becomes strictly difficult for agents to determine which actions where truly impactful upon later success. This often results in a very slow learning process or failure to learn at all. To address this issue, promising approaches Jiang et al. [2021] Zheng et al. [2018] associate actions with temporal distant rewards by assigning credit to them depending on their contribution to later results. In literature, this concept is often referred to as TCA Ni et al. [2023] and describes measuring the influence that an action has on a particular outcome.

In this project, we investigate to what extent a self-attention mechanism can address the described TCA problem. We therefore employ a meta-gradient approach in which a one-head-attention network in the outer loop predicts intrinsic rewards (r_{in}), while the inner loop trains a PPO agent based on these. Our main objective is to leverage the r_{in} in combination with the extrinsic rewards (r_{ex}) to accelerate training for the PPO agent. Whereby, the meta-gradient is trained on maximizing the expected extrinsic reward only. Furthermore, we explicitly decouple TCA from memory effects in a controlled environment to measure the attention network's direct impact on an isolated TCA task solely. Our code can be accessed by our Github repository[1].

2 Related Work

The following section briefly outlines papers that form the basis of our approach.

[1] https://github.com/XfensorX/IntrinsicAttention/tree/main

2.1 SAIR & LIRPG

Jiang et al. [2021] propose a self-attention-based temporal intrinsic-reward model (SAIR) for training policies in sparse-reward, partially observable environments. More specifically, the proposed SAIR model employs a meta-gradient which generates r_{in} through the self-attention network in the outer loop and combines those with the given r_{ex} to update the policy and guide parameter updates in the inner loop.

More precise, within the meta-gradient inner loop, the PPO policy parameters θ are updated θ' by computing the gradient of the objective function J w.r.t. θ and given the trajectory $\mathcal{D}$ using the sum of r_{ex} and r_{in}: $\theta' \leftarrow \theta + \alpha \, \nabla_\theta J_{\text{ex+in}}\big(\theta \mid \mathcal{D}\big)$.

Whereas in the outer loop, the attention network parameters η are updated η' by computing the gradient of J using only r_{ex} w.r.t. the updated PPO policy parameters θ' from the inner loop and η: $\eta' \leftarrow \eta + \alpha \, \nabla_\eta J_{\text{ex}}$, with $\nabla_\eta J_{\text{ex}} = \big(\nabla_{\theta'} J_{\text{ex}}(\theta' \mid \mathcal{D})\big)\big(\nabla_\eta \theta'\big)$. In this way, r_{in} shapes and facilitates the policy learning process, while the main objective remains to maximize the accumulated expected extrinsic return.

Therefore, the attention networks purpose is to capture temporal features encoded in the trajectories $\mathcal{D}$ consisting of states, actions, and r_{ex} collected by the current policy θ, so that it can learn temporal dependencies between the current action and the future performance to promote and facilitate the training process of the policy.

Furthermore, a very similar approach is described in the LIRPG framework Zheng et al. [2018]. It learns a parametric intrinsic-reward function $r_{\text{in},\eta}(s_t, a_t)$ through a non-recurrent NN (MLP, CNN) only being able to depict the current state and action, parameterized by η, in the outer loop. The parameters η are updated via a meta-gradient to maximize only the extrinsic return J_{ex}, while the policy θ in the inner loop is trained on a weighted combination of r_{ex} and $r_{\text{in},\eta}(s_t, a_t)$: $J_{\text{ex+in}} = \mathbb{E}_\theta \left[\sum_{t=0}^{T} \gamma^t \left(r_{\text{ex},t} + \lambda \, r_{\text{in},\eta}(s_t, a_t)\right)\right]$, whereas both the policy and intrinsic-reward networks are updated through the mentioned objective functions, similar to SAIR. Unlike LIRPG's non-recurrent intrinsic-reward network, which lacks temporal context, SAIR's self-attention module can extract temporal dependencies across the entire trajectory, while both employ importance sampling to compute the meta-gradient from the same trajectory.

2.2 Decoupling TCA from Memory

Furthermore, Ni et al. [2023] propose an approach to disentangle TCA from memory. Here, memory refers to an agent's ability to recall a distant past event at a later time step, whereas TCA denotes the ability to determine when an action that deserved credit occurred. In particularly, TCA is hereby measured using a forward view by defining a credit assignment length as the minimal number of future steps until an optimal action produces a measurable advantage, rather than the traditional backward view. The memory length is defined as the minimal number of steps between the observation of an informative cue and the point where this information must be used. Therefore, several environments are introduced to measure a Transformer's performance on isolated memory or TCA tasks. In particular, we highlight the Umbrella Length in chapter 4, as we deploy this in our project.

We further enhance the approach to use environments which strictly isolate TCA and memory from each other, to better measure the TCA capability of our intrinsic-reward PPO agent.

3 Approach

Building upon the two previously discussed approaches in chapter 2, our goal is to investigate whether a PPO agent trained via a meta-gradient, similar to the depicted SAIR Jiang et al. [2021] implementation in table 1, can improve TCA compared to a vanilla PPO agent without intrinsic rewards. We further aim to maximize the expected extrinsic reward by employing a single-head attention layer followed by a linear layer as our intrinsic-reward network, and combine extrinsic and intrinsic rewards to accelerate the PPO agent's learning.

To get an isolated measurement of the TCA ability of the agents, we strictly eliminate memory effects by implementing the PPO actor and critic networks as simple feed-forward neural networks, explicitly avoiding memory-based-architectures such as LSTMs, GRUs, or Transformers, since these could allow the agent to store past information and thus introduce a memory component. This ensures that any performance gains are solely due to an improvement of TCA and not memory.

To ensure that only TCA abilities aus measured, we employ the Umbrella Length environment (see chapter 4), where TCA behavior is isolated from memory effects. Hereby, no memory is required, since the optimal action is observable at every step. The challenge lies purely in assigning credit over a long temporal gap, as only the first action affects the final outcome, while intermediate rewards are noise and not meaningful for the final reward. In this setting, the intrinsic-reward network is permitted to access the entire trajectory within an episode (forward view) to determine reward assignments but cannot use information across episodes.

Algorithm 1 SAIR Algorithm Jiang et al. [2021]

```
Require: learning rate $\alpha$ and intrinsic-reward proportion $\lambda$
    Initialize: Initialize the policy network parameters $\theta$ and the intrinsic-reward network parameters $\eta$ with random values
    REPEAT
    The agent uses the policy $\pi_\theta$ to interact with the environment, and obtains the trajectory $\mathcal{D}$ and the extrinsic reward $r^{\text{ex}}$
    Compute the intrinsic reward $r^{\text{in}}$ for every sampled state $s_i$ using $\mathcal{D}$ and the self-attention intrinsic-reward network with the parameters $\eta$
    Approximate the gradient $\nabla_\theta J_{\text{ex+in}}\big(\theta \mid \mathcal{D}\big)$
    Update: $\theta' \leftarrow \theta + \alpha\, \nabla_\theta J_{\text{ex+in}}\big(\theta \mid \mathcal{D}\big)$
    Approximate the gradient $\nabla_\eta J_{\text{ex}} = \big(\nabla_{\theta'} J_{\text{ex}}(\theta' \mid \mathcal{D})\big)\big(\nabla_\eta \theta'\big)$
    Update: $\eta' \leftarrow \eta + \alpha\, \nabla_\eta J_{\text{ex}}$
    $\theta \leftarrow \theta'$
    UNTIL done
```

During training, the intrinsic-reward attention network processes complete trajectories (including extrinsic rewards) from each episode, similar to SAIR, and predicts intrinsic rewards for every timestep. Its purpose is to identify the key action, which determines the final outcome and assigning greater importance to it by redistributing credit in form of intrinsic rewards from the delayed extrinsic reward to the earlier relevant timestep.

An architecture overview of our intrinsic-reward PPO agent is illustrated in figure 1.

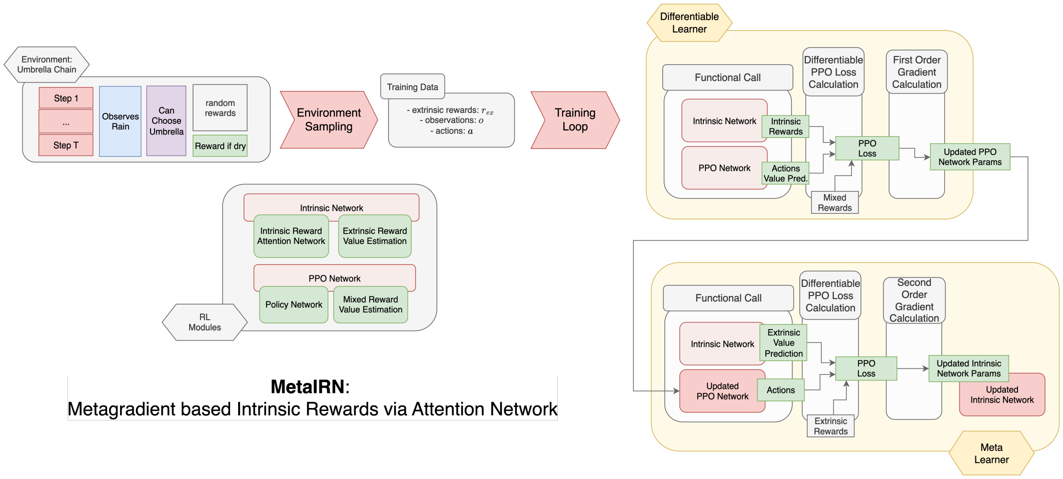


Figure 1: Architecture Overview

For a more detailed implementation overview, we refer to our repository [2].

[2] https://github.com/XfensorX/IntrinsicAttention/tree/main

Our central research question is whether this combination of meta-gradient optimization and attention-based generated intrinsic rewards can enhance the agent's ability to assign credit in long-horizon, TCA-only scenarios and thus accelerate PPO training.

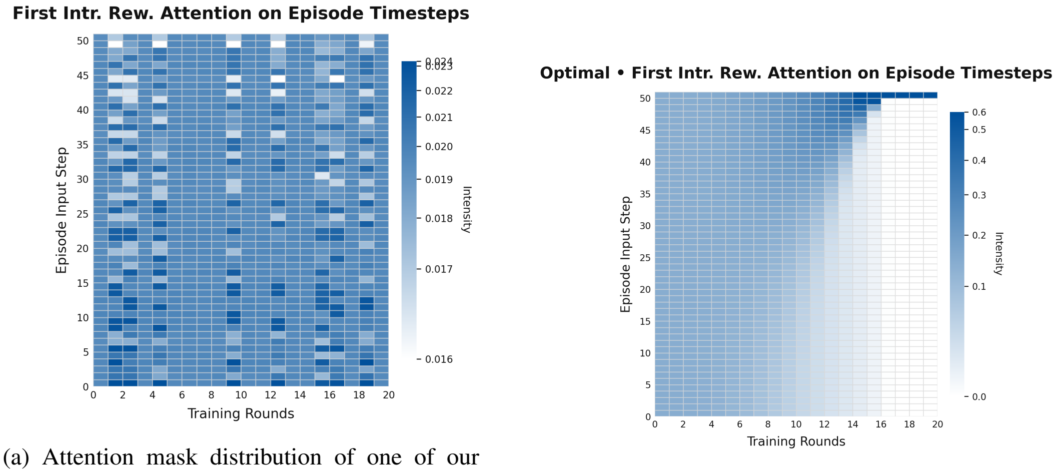
4 Environment - Umbrella Length

The Umbrella environment Osband et al. [2020] is a fully observable MDP designed to test pure TCA isolated from memory effects. Here, only the first action of an episode affects the final outcome and is fixed throughout the whole episode whereby all intermediate actions are ignored. The only informative reward occurs at the terminal step and depends solely on whether the initial action $a_1 \in \{\text{pick, don't pick}\}$ matches the final observation `need_umbrella`. Intermediate rewards are pure noise and provide no learning signal or matching relation about the correct action choice. Hence, the *credit-assignment length* equals the episode length, while the *memory requirement* is zero, because the task is fully observable and so `need_umbrella` is visible at every step. We've slightly modified the environment, so that the intermediate rewards, which serve as distracting noise, are set to $R_i \in \{-0.1, 0.1\}$. The final reward is set to $R_f \in \{-10, 10\}$ depending on whether the first action of the agent matches to `need_umbrella`. In an optimal case our attention-based intrinsic-reward PPO model should learn, that only the first action determines the final outcome and should therefore learn to place a sharp credit spike to the first step and zero elsewhere, reflecting the fact that only the initial decision determines the final return.

5 Experiments

For our training experiments, we conducted five runs for both the vanilla PPO and our intrinsic-reward PPO agent in the umbrella environment, using umbrella chain lengths of [2,5,10,20,50] and ten distinct random seeds. To ensure well-tuned configurations, we performed hyperparameter optimization with SMAC3 Lindauer et al. [2022], employing the BOHB algorithm, whereby the negative mean episode return served as the cost function. We applied a multi-fidelity approach on environment steps, going up from 30k to 100k and evaluated 100 configurations using an umbrella chain length of 50.

We plot our results for the vanilla and our intrinsic-reward PPO agent in comparison to the optimal performance for how the attention for the intrinsic reward intrinsic reward should redistribute over time for the first action, as explained in 4.



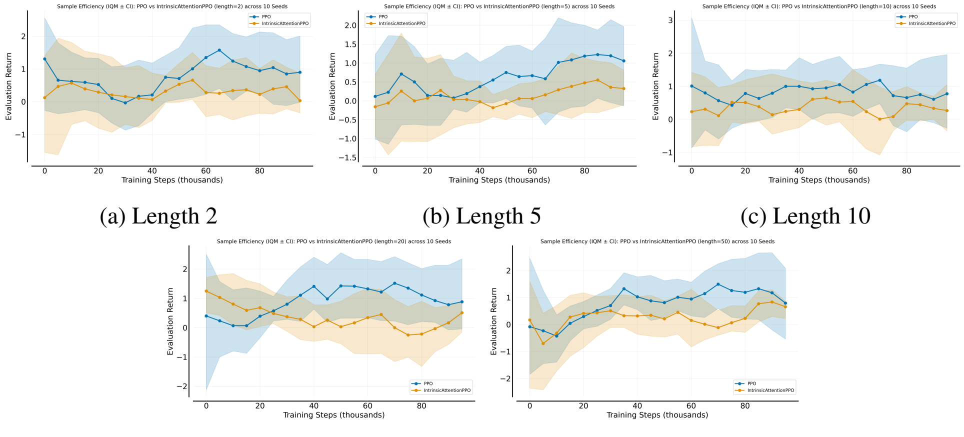
(a) Attention mask distribution of one of our intrinsic-reward PPO networks

(b) Optimal attention mask distribution

Figure 2: Comparison of first reward and optimal attention

Therefore, figure 2 shows the comparison between the attention on the environments steps over time for creating the intrinsic reward of our intrinsic-reward PPO model after a whole optimization 2a and a general optimal attention distribution 2b. Where one can see, in the optimal attention distribution plot, how the focus clearly shifts towards the agent's last environment step (top row) over time. Indicating an understanding of the temporal dependency linking the final reward to the agent's first action. Whereas for the intrinsic-reward PPO model attention mask no dependencies are captured.

Furthermore, the following plots show the sample-efficiency curves for both the intrinsic-reward PPO and the vanilla PPO agent, illustrating the mean evaluation return over the training steps for varying umbrella lengths [2,5,10,20,50] across 10 seeds.



(a) Length 2

(b) Length 5

(c) Length 10

(d) Length 20

(e) Length 50

Figure 3: Sample efficiency plots for different umbrella lengths plotted across 10 seeds (IQM+-CI).

As can be seen in figure 3, the vanilla PPO agent is able to slightly improve its performance over time, particularly in environments with shorter umbrella lengths. This can be explained due to the reduced temporal distance the agent needs to account for, which it fails to capture effectively for longer umbrella lengths. However, this effect can't be validated, since the agent isn't improving significantly at all.

In comparison, our intrinsic-reward PPO agent fails to meaningfully capture the temporal dependencies and therefore does not achieve a significant performance improvement. Several factors could explain this outcome. One reason could be that the meta-gradient implementation may be fragile or error-prone, preventing effective learning. The intrinsic network may be too small, with too few attention layers to extract useful temporal features, as we only use one attention and linear layer for our intrinsic-reward network. The HPO search space might be too constrained, limiting the discovery of well-performing configurations. Or the extrinsic value head, which has a strong influence on training, may require further research to ensure it enhances rather than disrupts performance.

Consequently, no definitive statement can be made as to why the performance does not improve and these results should therefore be interpreted with caution.

As further depicted in figure 4 one can see the IQM Evaluation Return of the fully trained vanilla PPO and the intrinsic-reward PPO over different umbrella lengths. The figure shows that neither agent is able to solve the environment at all. Surprisingly, this is the case even for small environment lengths. For our intrinsic-reward PPO model, this could be due to the reasons mentioned earlier. However, we expected better performance from the native PPO agent.

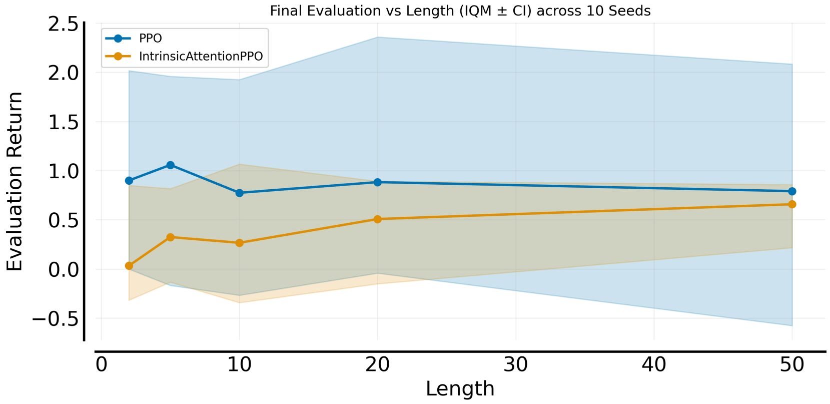


Figure 4: Performance over the umbrella length after training per agent

6 Discussion

Our study investigated whether adding an attention-based intrinsic-reward network to a meta-gradient learning system can improve the performance of a PPO agent by using intrinsic rewards to address the issue of TCA. Thereby, we took inspiration from existing approaches on this topic which we combined with a special TCA-only environment. We employed an implementation of a meta-gradient optimization process to facilitate training for PPO agents. Furthermore, we compared our intrinsic-reward PPO agent to it's theoretical optimal performance and to vanilla PPO by running multiple trainings on different seeds and varying umbrella lengths for both agents, testing the agent's TCA abilities. To good configurations and a fair comparison, we employed a BOHB based HPO via SMAC3 for each run.

Deriving from our experiments, we conclude that our intrinsic-reward PPO agent was unable to capture the relevant temporal dependencies. This is likely due to points mentioned above. The vanilla PPO agent however is able to improve it's performance very slightly over time but not in a very significant manner.

In relation to our research question, it is not possible to draw reliable conclusions about the impact of our intrinsic-reward attention network on the TCA problem, given the uncertainty surrounding the reasons why our agent failed to learn effectively. Nonetheless, refining our promising implementation leaves room for future insights and could yield potential improvements in this area. Despite it's current limitations, we provide a certain approach and environment that can serve as a foundation for measuring and exploring TCA using attention mechanisms. With additional interest and resources, our approach can be extended to gain meaningful insights in the field on how intrinsic rewards and attention can be leveraged to address the TCA problem.

References

Zhuo Jiang, Daiying Tian, Qingkai Yang, and Zhihong Peng. Self-attention based temporal intrinsic reward for reinforcement learning. In *2021 China Automation Congress (CAC)*, pages 2022–2026, 2021. doi: 10.1109/CAC53003.2021.9727314.

Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. Smac3: A versatile bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23(54):1–9, 2022. URL `http://jmlr.org/papers/v23/21-0888.html`.

Tianwei Ni, Michel Ma, Benjamin Eysenbach, and Pierre-Luc Bacon. When do transformers shine in rl? decoupling memory from credit assignment. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL `http://papers.nips.cc/paper_files/paper/2023/hash/9dc5accb1e4f4a9798eae145f2e4869b-Abstract-Conference.html`.

Ian Osband, Yotam Doron, Matteo Hessel, John Aslanides, Eren Sezener, Andre Saraiva, Katrina McKinney, Tor Lattimore, Csaba Szepesvári, Satinder Singh, Benjamin Van Roy, Richard S. Sutton, David Silver, and Hado van Hasselt. Behaviour suite for reinforcement 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=rygf-kSYwH`.

Zeyu Zheng, Junhyuk Oh, and Satinder Singh. On learning intrinsic rewards for policy gradient methods. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 4649–4659, Red Hook, NY, USA, 2018. Curran Associates Inc.