Agent-Time Attention for Sparse Rewards Multi-Agent Reinforcement Learning

Jennifer She
Stanford University
Stanford, CA, USA
jenshe@alumni.stanford.edu

Jayesh K. Gupta
Microsoft
Bellevue, WA, USA
jayesh.gupta@microsoft.com

Mykel J. Kochenderfer
Stanford University
Stanford, CA, USA
mykel@stanford.edu

ABSTRACT

Sparse and delayed rewards pose a challenge to single agent reinforcement learning. This challenge is amplified in multi-agent reinforcement learning (MARL) where credit assignment of these rewards needs to happen not only across time, but also across agents. We propose Agent-Time Attention, a neural network model with auxiliary losses for redistributing sparse and delayed rewards in collaborative MARL. We provide a simple example to demonstrate how providing agents with their own local redistributed rewards over shared global redistributed rewards leads to better policies.

KEYWORDS

Multi-Agent Reinforcement Learning, Sparse Rewards

ACM Reference Format:
Jennifer She, Jayesh K. Gupta, and Mykel J. Kochenderfer. 2022. Agent-Time Attention for Sparse Rewards Multi-Agent Reinforcement Learning. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

Sparse and delayed rewards are difficult for reinforcement learning (RL) because the number of possible trajectories grows exponentially with time horizon and this makes attributing rewards to intermediate observations and actions exponentially more difficult [3, 9]. One approach to improve learning is to supply additional rewards through reward shaping in order to transform sparse delayed rewards problems into dense ones [13]. However, reward shaping is often difficult because it requires environment-specific knowledge. For single agent problems, frameworks like RUDDER and SECRET [1, 5] allow constructing neural network models for reward redistribution; learning how sparse delayed rewards can be transformed into dense rewards for effective policy optimization.

Unfortunately, in the multi-agent reinforcement learning (MARL) setting, sparse and delayed rewards have not been explored extensively [8]. Existing cooperative MARL methods instead are focused on the problem of deducing an agent's contribution to the overall team's success, assuming access to dense global team rewards. These methods can take various forms, be it *implicit* like [11, 12, 14, 15, 17, 21] where the global state-action value is decomposed as aggregation of each agent's state-action value while assigning the shared global rewards to each agent based on their actions, or *explicit* such as COMA [6] and LIIR [4] based on computing difference rewards [20] against particular reward baselines. Implicit methods often encounter limitations in expressiveness with no clear strategy for continuous action domains, while explicit methods face limitations on reasoning about individual effect of individual agent actions on the shared global rewards.

In this work we instead focus on multi-agent domains with delayed or sparse global rewards. Therefore any MARL framework will require reasoning about both the contribution's from different agents as well as team's actions in the past i.e. solve the *credit assignment* problem along both the agent and time axes. We focus on an improving explicit method to ensure applicability on continuous action problems. To this end, we propose Agent-Time Attention (ATA), a neural network model trained on auxiliary losses for redistributing global, sparse and delayed team rewards across both time and agents into dense, local agent rewards. This model can be applied on top of different single agent RL methods such as Q-learning and policy gradient methods without additional modifications under the CTDE paradigm. We perform a pedagogical experiment on a multi-agent one-dimensional coin environment to emphasize the importance of holistically reasoning about credit assignment along both agent and time axes at the same time.

We found that simply extending standard policy gradient methods with ATA, not just outperforms baselines that just do credit assignment on either the agent or time axis, but also their straightforward combinations where they still reason about credit-assignment along these two axes separately.

2 MARL REWARD REDISTRIBUTION

RUDDER [1] described reward redistribution as a two-step process: 1) learning a function g that predicts the expected return for a given observation-action sequence. 2) *contribution analysis* that determines how much an observation-action pair contributes to the final prediction.

In the multi-agent case too, the basic principles behind reward redistribution remains the same. However, there is an additional demand to redistribute rewards not just along time but also across agents. Inspired by success of self-attention at *implicit* credit assignment between agents in LICA [22] and transformers at deep learning problems [7, 10, 18] we propose a transformer based architecture for multi-agent return prediction which we term as Agent-Time Attention (ATA). The network architecture is described in Figure 1. Observation-action pairs at each timestep t for each agent i result into the return predictions of $\hat{R}_i^t$ during execution.

The data for training this model is collected during policy optimization. The reader might observe that in cooperative MARL we only have global team rewards from the environment. To be able to train this architecture in a similar manner as RUDDER, we aggregate the features after the transformer layer, $\hat{Z}_1^t, \ldots \hat{Z}_n^t$, into

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

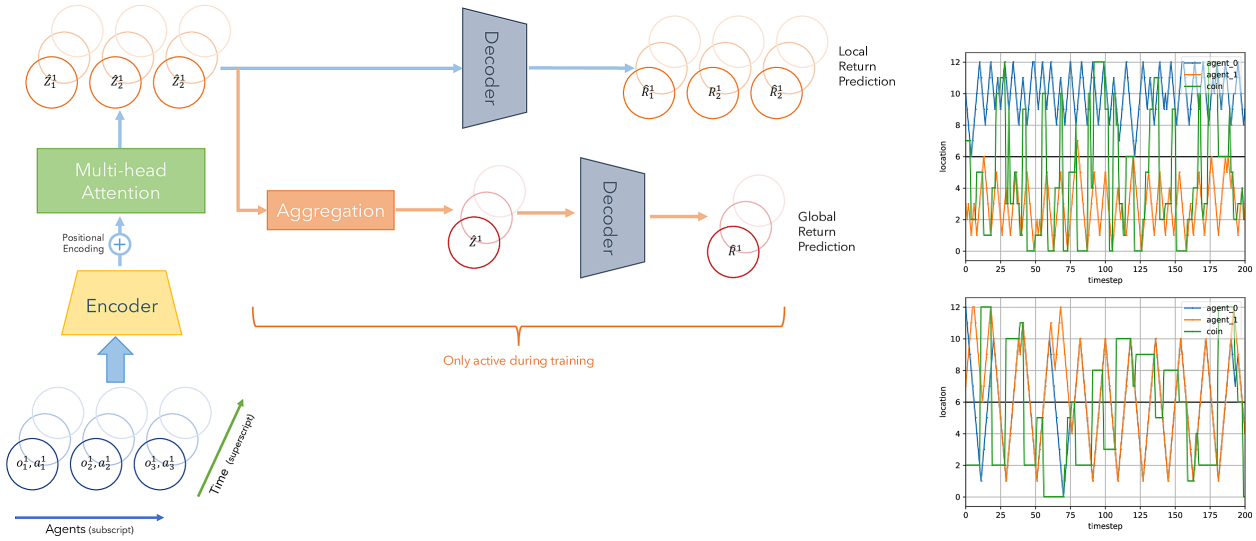

Figure 1: Left: Network architecture of Agent-Time Attention (ATA) for Reward Redistribution. The blue arrows are active during both training and execution while orange arrows are active only during training. ATA first encodes individual agent observation-actions to a latent space. Together with positional encoding ATA applies a single layer of multi-head attention [19] to another latent space. These are directly decoded to predict individual agent returns at particular time ($\hat{R}_i^t$ for i-th agent at t timestep). We aggregate features in latent space across agents and decode them to predict global team returns at particular time ($\hat{R}^t$). Right: sample episodes during training, of agents and coin positions over time. Top is only reward redistribution along time axis. Bottom is ATA.

$\hat{Z}^t$ before decoding with the same linear layer as used for local return predictions during execution. The training loss is:

$$\ell = (1-\lambda)\left\|\hat{R}^t - R^t\right\|^2 + \lambda\left\|\hat{R}_i^t - R_i^t\right\|^2 \quad (1)$$

where R^t and R_i^t are Monte Carlo returns estimated from the actual rollouts collected during policy optimization.

As recommended by RUDDER, we use *differences in return predictions* for contribution analysis. The agent i's reward at timestep t is therefore $\tilde{r}_i^t = \hat{R}_i^{t+1} - \hat{R}_i^t$. In our initial tests, we too found that this performed better than other contribution analysis methods like integrated gradients [16] or layer-wise relevance propagation [2]. For example, integrated gradients are more computationally expensive and result in many possible redistributed rewards when the reward redistribution model overfits. The attention-weighed method from SECRET is also too constrained in that it enforces $\tilde{r}_t = \alpha R_t$, where $\alpha \geq 0$.

Our MARL reward redistribution model has the flexibility to provide agents with individual rewards in place of global rewards without requiring a MARL-specific policy optimization method. From past literature, global rewards tend to result in lazier agent behaviors and purely individual rewards tend to result in more selfish, potentially greedy agent behaviors. A balance can be achieved under different scenarios by tuning λ in Equation (1).

3 1D COIN ENVIRONMENT

To understand why RUDDER like reward redistribution of global team rewards $\tilde{r}^t$ is not enough for MARL, and why we need to think about redistribution across both time and agents, we construct a simple one dimensional coin collection environment.

The environment consists of a line of length 13 with two agents and one coin generated at random positions on the line. An agent has partial observations and can move left or right. If an agent reaches the coin, the global team reward increments by p_1, and if both agents reaches the coin at the same time, the global reward increments by p_2. Once either agents reaches the coin, the coin is randomly placed at a different position. The episode length is 200. The global reward is provided at the end of each episode.

We compare our ATA model to a baseline RUDDER model that takes as input concatenation of the observations and actions of all agents into a single input $(o_{0t}, a_{0t}, o_{1t}, ...)$ and then predicts the global team reward redistribution only. We use independent policy gradient (IPG) to train the agent policies.

Figure 1 shows example episodes mid-training using these reward redistribution methods for $p_1 = 0.25$ and $p_2 = 1$. The RUDDER baseline encourages a lazy behavior of agents collecting coins in their own half of the space, while our reward redistribution model that does redistribution along both axes can encourage behavior where agents tend to move together, across the entire space. This is preferred specifically in the case where $p_2 > p_1$ because getting the coin together leads to higher returns. Although both methods converge to similar solutions, ATA is much faster.

4 CONCLUSION

We proposed Agent-Time Attention (ATA) for redistributing sparse global team rewards into dense agent-specific rewards that can be trained along with existing RL methods such as policy gradient. We identified a simple case demonstrating failure of just redistributing rewards over time for the multi-agent case.

REFERENCES

[1] Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. 2019. RUDDER: Return Decomposition for Delayed Rewards. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE* 10, 7 (2015), e0130140.

[3] Christoph Dann and Emma Brunskill. 2015. Sample Complexity of Episodic Fixed-Horizon Reinforcement Learning. *Advances in Neural Information Processing Systems (NIPS)* (2015).

[4] Yali Du, Lei Han, Meng Fang, Ji Liu, Tianhong Dai, and Dacheng Tao. 2019. LIIR: Learning Individual Intrinsic Reward in Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[5] Johan Ferret, Raphael Marinier, Matthieu Geist, and Olivier Pietquin. 2020. Self-Attentional Credit Assignment for Transfer in Reinforcement Learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*. https://doi.org/10.24963/ijcai.2020/368

[6] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[7] Quentin Fournier, Gaétan Marceau Caron, and Daniel Aloise. 2021. A Practical Survey on Faster and Lighter Transformers. *arXiv preprint arXiv:2103.14636* (2021).

[8] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor. 2019. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems* 33, 6 (2019), 750–797.

[9] Nan Jiang and Alekh Agarwal. 2018. Open problem: The dependence of sample complexity lower bounds on planning horizon. In *Conference on Learning Theory*.

[10] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2021. Transformers in Vision: A Survey. *ACM Comput. Surv.* (dec 2021). https://doi.org/10.1145/3505244

[11] Sheng Li, Jayesh K Gupta, Peter Morales, Ross Allen, and Mykel J Kochenderfer. 2021. Deep Implicit Coordination Graphs for Multi-agent Reinforcement Learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

[12] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. 2019. MAVEN: Multi-Agent Variational Exploration. *Advances in Neural Information Processing Systems (NeurIPS)* (2019).

[13] Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning (ICML)*.

[14] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning (ICML)*.

[15] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*.

[16] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*.

[17] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

[18] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732* (2020).

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[20] David H Wolpert and Kagan Tumer. 2002. Optimal payoff functions for members of collectives. In *Modeling Complexity in Economic and Social Systems*. World Scientific, 355–369.

[21] Tianjun Zhang, Huazhe Xu, Xiaolong Wang, Yi Wu, Kurt Keutzer, Joseph E Gonzalez, and Yuandong Tian. 2020. Multi-Agent Collaboration via Reward Attribution Decomposition. *arXiv preprint arXiv:2010.08531* (2020).

[22] Meng Zhou, Ziyu Liu, Pengwei Sui, Yixuan Li, and Yuk Ying Chung. 2020. In *Advances in Neural Information Processing Systems (NeurIPS)*.