Using Assistance Rewards Without Introducing Bias: Overcoming Sparse Rewards in Multi-Agent Reinforcement Learning

Extended Abstract

Yue Yang
Monash University
Melbourne, Australia
yue.yang1@monash.edu

Bernd Meyer
Monash University
Melbourne, Australia
bernd.meyer@monash.edu

Frits de Nijs*
Monash University
Melbourne, Australia
frits.nijs@monash.edu

ABSTRACT

Reinforcement learning agents may fail to learn good policies when their reward function is too sparse. Auxiliary reward shaping functions can help guide exploration towards the true rewards, but risk producing sub-optimal policies as agents now target a modified objective function. Our paper addresses this challenge by introducing a general framework for incorporating auxiliary reward functions without introducing a bias in the true objective. Agents train an ensemble of reward-function-specific policies, sharing experiences collected with one policy to all other policies in the ensemble. A top-level control policy then learns to choose the best policy to maximize the true objective. We show that this scheme does not affect the convergence properties of the underlying reinforcement learning algorithm, while avoiding potential biasing of the agent's objective. We also adapted our proposed algorithm using off-policy PPO with MA-Trace correction for state value estimation. To our knowledge, this is the first work to adapt off-policy PPO in a multi-agent setting. We also demonstrate that our approach operates effectively with various assistance reward designs, removing the need for detailed reward function crafting or fine-tuning.

KEYWORDS

Multi-agent RL; Sparse Environment; Assistance Reward

ACM Reference Format:
Yue Yang, Bernd Meyer, and Frits de Nijs. 2025. Using Assistance Rewards Without Introducing Bias: Overcoming Sparse Rewards in Multi-Agent Reinforcement Learning: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

1 INTRODUCTION

In reinforcement learning, learning the true objective in a sparse reward multi-agent setting is challenging, often requiring understanding and coordination among agents' behaviors and actions [4, 7, 15, 17]. To address this, recent studies have introduced *auxiliary reward functions*, additional objectives that provide extra information to incentivize effective exploration and facilitate the learning process [7, 8, 15, 18].

* Corresponding author.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

However, introducing auxiliary rewards can also introduce biases to the system. This paper introduces a novel strategy, Multi-agent Task Ensemble (MATE), for utilizing auxiliary information in reinforcement learning without the issues previously mentioned. MATE constructs an ensemble of auxiliary policies, each optimizing a distinct auxiliary reward function independently. The algorithm leverages these dense auxiliary rewards to guide agents in taking semantically meaningful actions, thus enriching their experience and enhancing performance with off-policy observations. A high-level controller policy, trained self-supervised on the true objective, selects the optimal auxiliary policy at each stage. Agents receive both assistance and true rewards, transitioning to new states. The assistance policy enhances learning by offering valuable experiences; if ineffective, the higher-level controller, trained on the true rewards, steps in to minimize negative impacts. We demonstrate that our Multi-agent Task Ensemble (MATE) framework converges to the optimal policy when using an off-policy algorithm like tabular Q-learning. For on-policy methods such as PPO, we adapt them to off-policy settings via MA-Trace correction. Empirically, MATE surpasses baseline methods in environments with sparse rewards and biased assistance reward designs. MATE efficiently utilizes any form of assistance reward, leveraging their benefits without requiring extensive customization or tuning of the reward function.

2 RELATED WORK

Two main methods for using auxiliary functions are: 1) combining true and auxiliary objectives into a new reward function [2, 11, 12], and 2) adjusting the policy gradient based on the similarity between assistance and true policies [9, 19, 22]. Both approaches modify the original learning objectives, potentially leading to unintended behaviors. Additionally, integrating auxiliary objectives requires careful tuning of parameters and coefficients, presenting significant challenges [3, 5]. Transfer Learning trains agents on a source task with individual rewards, usually in a simpler environment for efficient learning, then fine-tunes the learned policies on a target task with sparser team rewards [20, 24, 25]. However, this approach can suffer from negative transfer if the source task is too dissimilar, impacting performance negatively on the target task. Imitation learning focuses on replicating expert actions to achieve similar outcomes but often lacks exploration and creativity, especially if expert behaviors do not match the desired objectives [1, 6, 14].

3 METHODOLOGY

Assistance Reward Environment: An assistance-reward Dec-POMDP augments the usual Dec-POMDP tuple with m additional

side reward signals $R^a(s, \mathbf{a}) \rightarrow \mathbb{R}$. The resulting tuple is $(N, S, \{A_i\}, P, \mathcal{R}, \{\Omega_i\}, O, \gamma)$, where $\mathcal{R} = R^o, \{R^a\}$. Here, we annotate the true or 'real' (sparse) reward function with R^o, to distinguish it from the assistance rewards R^a for $a \in \{1 \dots m\}$. In this formulation, N is the finite set of agents; S is the finite set of states; A_i is the set of possible actions for agent i; P is the state transition probability function; O is the set of possible observations; $\{\Omega_i\}$ is the set of observation functions for each agent, where $\Omega_i : S \rightarrow O$ determines the observations received by agent i based on the current state; $\gamma \in [0, 1]$ is the discount factor. It is assumed that the assistance reward signals R^a are denser than the original reward function, although they may induce policies that are sub-optimal for the original reward function. In other words, we do not require that a policy which is optimal for R^a would also be (near) optimal for R^o, nor do we place any restrictions on the magnitude of R^a compared to R^o.

Assistance Task Ensemble: We propose to address the challenge of biased assistance reward by training an Assistance Task Ensemble. In this paper, we focus on collaborative multi-agent problems solved via centralized training and decentralized execution.

Each agent policy consists of $m+2$ sub-policies; a real-reward policy $\pi_i^o(a_i \mid \Omega_i(s_t))$ targeting R^o, a set of m executor policies $\pi_i^m(a_i \mid \Omega_i(s_t))$ each targeting their respective assistance reward R^m, and a controller policy $\pi^{\text{CTR}}(\pi_i^e \mid \Omega_i(s_t))$ distributing over the set of executor policies $E_i = \{\pi_i^o\} \cup \{\pi_i^1, \dots, \pi_i^m\}$ in order to maximise the real reward R^o. Together, an agent's ensemble policies form a control policy $\pi_i^{\text{MATE}}(a_i \mid \Omega_i(s_t))$ for the original problem, defined by:

$$\pi_i^{\text{MATE}}(o_i) = \left(a_i \sim \pi_i^e(\Omega_i(s_t)) \mid \pi_i^e \sim \pi_i^{\text{CTR}}(\Omega_i(s_t))\right), \quad (1)$$

Intuitively, the action selection process of an agent works as follows: each time step t, the agent observes the current observation o_t from the environment. Based on this observation, the agent's controller policy selects the appropriate executor policy from the candidate policies $\pi_e \in \{\pi_i^1, \dots, \pi_i^m, \pi_i^o\}$, which in turn specifies the probability of selecting concrete action $a_{t,i} \in A_i$.

Let $\boldsymbol{\pi}^{\text{MATE}}$ denote the joint policy for all N agents, and define the objective function as

$$J(\boldsymbol{\pi}^{\text{MATE}}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R^o\left(s_t, \{\pi_i^{\text{MATE}}(\Omega_i(s_t))\}_{i=1}^N\right)\right], \quad (2)$$

where $R^o(s_t, \pi_i^{\text{MATE}}(\Omega_i(s_t)))$ is the true objective reward at time t, $\gamma \in [0, 1]$ is the discount factor, and $\{\pi_i^{\text{MATE}}(\Omega_i(s_t))\}_{i=1}^N$ is the joint action at time t. The learning problem is then to find the optimal joint policy $\boldsymbol{\pi}^{\text{MATE}^*} = \arg\max_{\boldsymbol{\pi}^{\text{MATE}}} J(\boldsymbol{\pi}^{\text{MATE}})$.

For the executor sub-policies $\boldsymbol{\pi}^j$, which represent the joint policy across all agents associated with reward r_j, we train each policy concurrently to maximize its performance with respect to its corresponding reward function. The objective function is:

$$J(\boldsymbol{\pi}^j) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R^j(s_t, \{\pi_i^{\text{MATE}}(\Omega_i(s_t))\}_{i=1}^N)\right]. \quad (3)$$

For MATE integrated with the on-policy Multi-Agent Proximal Policy Optimization [21, MAPPO], which features centralized training and decentralized execution, we adjust it to include off-policy observations from assistance trajectories. This adaptation employs MA-Trace off-policy corrections [23], akin to how V-Trace is used for off-policy adaptations in PPO [10].

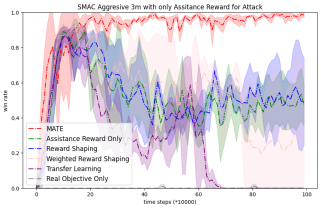

Figure 1: Aggre 3m (QMIX)

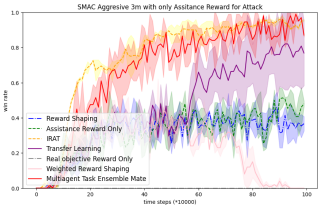

Figure 2: Aggre 3m (MAPPO)

4 EXPERIMENTS

Environment Setup and Benchmarks: We evaluate MATE's ability to integrate assistance rewards on a modified sparse version of the Multi-Agent StarCraft environment [16, SMAC]. We created a version of the '3 marines' map where the true reward is 1 for win and 0 otherwise. To assist agents in learning the true reward, we add an assistance reward based only on damage dealt. This assistance reward is sub-optimal since it places no value on self-preservation. In SMAC, the opposing team uses scripted strategies to attack the agent's units, resulting in a likelihood of winning akin to flipping a coin under this assistance reward function.

We implemented MATE on top of two base RL algorithms: 1) QMIX [13], and 2) MAPPO [21] with off-policy corrections. We compare against baseline strategies Assistance Reward Only (AR) and Objective Reward Only (OR) which learn from only one of the reward functions, basic Reward Shaping (RS) and Weighted Reward Shaping (WRS) that integrate both rewards into one scalar reward function, Transfer Learning (TL) that switches from assistance to real reward function mid-way training, and state-of-the-art Individual Reward Assisted Team Policy Learning [19, IRAT] that uses dual policies per agent with discrepancy constraints (on MAPPO).

Results: Results using QMIX (Fig. 1) and MAPPO (Fig. 2) demonstrate that only the MATE algorithm and IRAT are able to consistently achieve close to 100% win rate in our sparse SMAC environment. As expected, strategies using the assistance reward directly (AR, RS, and WRS) are not able to learn a successful policy, showing the predicted coin-flip win rate effect. Transfer learning, which shifts focus from initial objectives to actual goals later in training, can destabilize learning performance, especially with value-based approaches. Using only the real reward (win rate) is too sparse to learn a good policy. Both MATE QMIX and MATE MAPPO could learn a good strategy and can attain optimal outcomes for the environment with a biased assistance reward.

5 DISCUSSION AND CONCLUSION

We propose a method to handle sparse reward signals with any design of assistance reward without introducing bias in the final policy. By integrating an upper-level controller and implementing a clear separation between the assistance and the real objective policy in training, our approach achieves superior performance when compared to state-of-the-art algorithms across a variety of scenarios. Due to its flexibility and simplicity, our method adapts well to various problem domains, particularly in environments where the impact of assistance objectives on learning goals is unclear.

REFERENCES

[1] Felipe Leno Da Silva and Anna Helena Reali Costa. 2019. A survey on transfer learning for multiagent reinforcement learning systems. *Journal of Artificial Intelligence Research* 64 (2019), 645–703.

[2] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual Multi-Agent Policy Gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32(1). AAAI Press, Washington, DC, USA, 2974–2982. https://doi.org/10.1609/aaai.v32i1.11794

[3] Ze Gong and Yu Zhang. 2020. What Is It You Really Want of Me? Generalized Reward Learning with Biased Beliefs about Domain Dynamics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34(03). AAAI Press, Washington, DC, USA, 2485–2492. https://doi.org/10.1609/aaai.v34i03.5630

[4] Joshua Hare. 2019. Dealing with Sparse Rewards in Reinforcement Learning. arXiv:1910.09281 [cs.LG] https://arxiv.org/abs/1910.09281

[5] Yujing Hu, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, and Changjie Fan. 2020. Learning to utilize shaping rewards: A new approach of reward shaping. *Advances in Neural Information Processing Systems* 33 (2020), 15931–15941.

[6] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. 2017. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)* 50, 2 (2017), 1–35.

[7] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. 2016. Reinforcement Learning with Unsupervised Auxiliary Tasks. arXiv:1611.05397 [cs.LG]

[8] Siyuan Li, Rui Wang, Minxue Tang, and Chongjie Zhang. 2019. Hierarchical reinforcement learning with advantage-based auxiliary rewards. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, Article 126, 11 pages.

[9] Somdeb Majumdar, Shauharda Khadka, Santiago Miret, Stephen Mcaleer, and Kagan Tumer. 2020. Evolutionary Reinforcement Learning for Sample-Efficient Multiagent Coordination. In *Proceedings of the 37th International Conference on Machine Learning*, Hal Daumé III and Aarti Singh (Eds.), Vol. 119. PMLR, Online, 6651–6660.

[10] Wenjia Meng, Qian Zheng, Gang Pan, and Yilong Yin. 2023. Off-Policy Proximal Policy Optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37(8). AAAI Press, Washington, DC, USA, 9162–9170. https://doi.org/10.1609/aaai.v37i8.26099

[11] Andrew Y. Ng and Stuart J. Russell. 2000. Algorithms for Inverse Reinforcement Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 663–670.

[12] Aida Rahmattalabi, Jen Jen Chung, Mitchell Colby, and Kagan Tumer. 2016. D++: Structural credit assignment in tightly coupled multiagent domains. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4424–4429.

[13] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2020. Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *Journal of Machine Learning Research* 21, 178 (2020), 1–51. http://jmlr.org/papers/v21/20-081.html

[14] Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. 2021. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems* 34 (2021), 11702–11716.

[15] Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom van de Wiele, Vlad Mnih, Nicolas Heess, and Jost Tobias Springenberg. 2018. Learning by Playing Solving Sparse Reward Tasks from Scratch. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 4344–4353.

[16] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. 2019. The StarCraft Multi-Agent Challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (Montreal QC, Canada) *(AAMAS '19)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2186–2188.

[17] Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. 2018. Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards. arXiv:1707.08817 [cs.AI]

[18] Vivek Veeriah, Matteo Hessel, Zhongwen Xu, Janarthanan Rajendran, Richard L Lewis, Junhyuk Oh, Hado P van Hasselt, David Silver, and Satinder Singh. 2019. Discovery of Useful Questions as Auxiliary Tasks. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc.

[19] Li Wang, Yupeng Zhang, Yujing Hu, Weixun Wang, Chongjie Zhang, Yang Gao, Jianye Hao, Tangjie Lv, and Changjie Fan. 2022. Individual Reward Assisted Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 23417–23432.

[20] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 1–40.

[21] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 24611–24624.

[22] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient Surgery for Multi-Task Learning. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 5824–5836.

[23] Michał Zawalski, Błażej Osiński, Henryk Michalewski, and Piotr Miłoś. 2024. Off-Policy Correction For Multi-Agent Reinforcement Learning. arXiv:2111.11229 [cs.LG]

[24] Zhuangdi Zhu, Kaixiang Lin, Anil K. Jain, and Jiayu Zhou. 2023. Transfer Learning in Deep Reinforcement Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 11 (2023), 13344–13362. https://doi.org/10.1109/TPAMI.2023.3292075

[25] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.