Off-Policy Correction For Multi-Agent Reinforcement Learning

Extended Abstract

Michał Zawalski
University of Warsaw
Warsaw, Poland
m.zawalski@uw.edu.pl

Błażej Osiński
University of Warsaw
Warsaw, Poland
b.osinski@mimuw.edu.pl

Henryk Michalewski
Google Research
henrykm@google.com

Piotr Miłoś
Polish Academy of Sciences
Warsaw, Poland
pmilos@impan.pl

ABSTRACT

Multi-agent reinforcement learning (MARL) provides a framework for problems involving multiple interacting agents. Despite similarity to the single-agent case, multi-agent problems are often harder to train and analyze theoretically. In this work, we propose MA-Trace, a new on-policy actor-critic algorithm, which extends V-Trace to the MARL setting. The key advantage of our algorithm is its high scalability in a multi-worker setting. To this end, MA-Trace utilizes importance sampling as an off-policy correction method, which allows distributing the computations with negligible impact on the quality of training. Furthermore, our algorithm is theoretically grounded – we provide a fixed-point theorem that guarantees convergence. We evaluate the algorithm extensively on the StarCraft Multi-Agent Challenge, a standard benchmark for multi-agent algorithms. MA-Trace achieves high performance on all its tasks and exceeds state-of-the-art results on some of them.

KEYWORDS

Reinforcement Learning; V-Trace; Importance Sampling; Scalability

ACM Reference Format:
Michał Zawalski, Błażej Osiński, Henryk Michalewski, and Piotr Miłoś. 2022. Off-Policy Correction For Multi-Agent Reinforcement Learning: Extended Abstract. 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

In this work, we propose MA-Trace, a new on-policy actor-critic algorithm, which adheres to the centralized training and decentralized execution paradigm [3–5]. The key component of MA-Trace is the usage of importance sampling. This mechanism, based on V-Trace [2], provides off-policy correction for training data. As we demonstrate empirically, it allows distributing the computations efficiently in a multi-worker setup. Another advantage of MA-Trace is the fact that it is theoretically grounded. We provide a fixed-point theorem that guarantees convergence.

The on-policy algorithms directly optimize the objective; thus, they tend to be more stable and robust to hyperparameter choices than off-policy methods [1, 9, 10]. However, it is often impractical to use an on-policy algorithm in the distributed setting. When data collection is performed by many workers, the communication latency, asynchronicity, and other factors make the behavioral policies lag behind the target one. This results in a shift of the collected data towards off-policy distribution, which hurts the training. V-Trace reduces this shift by utilizing importance weights, thus permitting highly scalable training. Similarly, MA-Trace can be distributed to many workers to significantly reduce the wall-time of training with no negative impact on the results.

We evaluate MA-Trace on the StarCraft Multi-Agent Challenge [7]. Our approach achieves competitive performance on all tasks and exceeds the state-of-the-art results on some of them.

For an extended version of this paper, we refer to [12]. The videos and further analysis can be found on the project webpage https://sites.google.com/view/ma-trace. The source code of our experiments is available at https://github.com/awarelab/seed_rl.

Our main contributions are the following:

(1) We introduce MA-Trace – a simple, scalable, and effective multi-agent reinforcement learning algorithm with theoretical guarantees.
(2) We confirm that the training using MA-Trace can be easily distributed to multiple workers with a nearly perfect speed-up and no negative impact on the quality.
(3) We provide extensive experimental validation of the MA-Trace algorithm on the StarCraft Multi-Agent Challenge.

2 MA-TRACE ALGORITHM

MA-Trace follows the paradigm of centralized training, decentralized execution. Each actor-agent chooses its action taking into account its local observation. On the other hand, the critic network, which provides estimates of the value function, operates only during training, so it does not need to obey decentralization requirements. We study two versions of the algorithm, differing in the input to the critic: MA-Trace (state) uses the environment states, while MA-Trace (obs) employs the joint observation of all agents.

The value function V^π corresponding to policy π can be obtained by repeated application of the Bellman operator. This requires on-policy data. The central innovation of V-Trace in the single-agent setting and MA-Trace in the multi-agent setting is to allow for slightly off-policy data by utilizing importance sampling. To this

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.

end, we use the V-Trace-inspired policy evaluation operator $\mathcal{R}$, defined as

$$\mathcal{R}V(s) := V(s) + \mathbb{E}_\mu\left[\sum_{t=0}^{+\infty}\gamma^t(c_0\cdots c_{t-1})\rho_t(r_t+\gamma V(s_{t+1})-V(s_t))|s_0=s\right], \quad (1)$$

where $c_t = c(s_t,a_t), \rho_t := \rho(s_t,a_t)$ are importance sampling corrections given by

$$c_t := \min\left(\overline{c}, \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}\right), \quad \rho_t := \min\left(\overline{\rho}, \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}\right), \quad (2)$$

where μ is a policy that collected the data and $\overline{c}, \overline{\rho}$ are hyperparameters. The operator $\mathcal{R}$ leads to a n-step Monte-Carlo target

$$v_t := V(s_t) + \sum_{u=t}^{t+n+1}\gamma^{u-t}\left(\prod_{i=t}^{u-1}c_i\right)\rho_u\big(r_u+\gamma V(s_{u+1})-V(s_u)\big). \quad (3)$$

For updating the networks parameters ω_i of the i-th actor we use policy gradient updates. We also need importance sampling to correct for using the off-policy behavioral policy μ:

$$\hat{g}_{i,t} = \rho_t(\nabla_{\omega_i}\log(\pi_{\omega_i}(a_t|s_t)))(r_t+\gamma V(s_{t+1})-V(s_t)). \quad (4)$$

In case of MA-Trace (obs) we need to use respective local observations instead of states when estimating $\hat{g}_{i,t}$.

The operator $\mathcal{R}$ enjoys the fixed point property. We present a proof of the following Theorem in [12].

Theorem 1. *Let c_t, ρ_t be importance sampling weights (2) and $0 \le \overline{c} \le \overline{\rho}$. Assume also that $\mathbb{E}_\mu \rho_0 \ge \beta \in (0,1]$. Then the operator $\mathcal{R}$ is a C_∞ contraction with a unique fixed point $V^{\tilde{\pi}}$ which is a value function of a policy $\tilde{\pi}$ given by*

$$\tilde{\pi}(a|s) := \frac{\min(\overline{\rho}\mu(a|s), \pi(a|s))}{\sum_{b\in\mathcal{A}}\min(\overline{\rho}\mu(b|s), \pi(b|s))}.$$

The contraction constant is smaller than $1-(1-\gamma)\beta < 1$.

3 EXPERIMENTS

3.1 Environment

We evaluate MA-Trace on the StarCraft Multi-Agent Challenge (SMAC) [8], which is based on a popular real-time strategy game StarCraft II. It provides 14 micromanagement tasks of varying difficulty and structure. The aim is to win a battle against a built-in AI engine by using a team of agents. Each unit has a limited sight range, which makes the environment partially observable. To facilitate the training, SMAC provides dense rewards. The team receives points for inflicting damage and defeating units. This scheme sometimes might reinforce undesired behaviors, which is the case e.g. in the *3s_vs_5z* task (see Figure 1 and discussion of the environment on the project webpage).

3.2 Main result

MA-Trace (obs) reaches competitive results and in some cases exceeds the state-of-the-art. We compare with a selection of the state-of-the-art algorithms on SMAC following [11] and [6]. More detailed comparison can be found in [12].

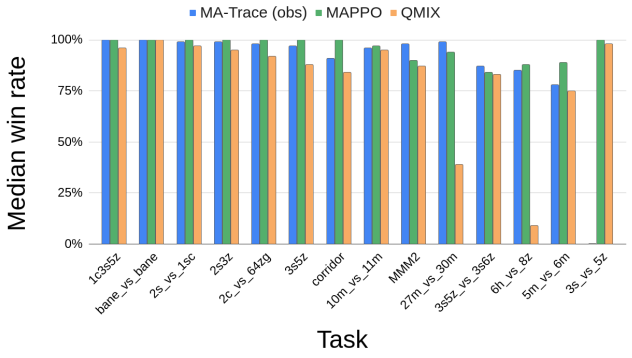

Figure 1: Median win rate of MA-Trace compared with state-of-the-arts algorithms on SMAC. In *3s_vs_5z*, our agent learns to keep the enemies alive and let them regenerate, to inflict more damage. This way the agent maximizes the rewards but scores no win.

3.3 Discussion and further experiments

Advantage of using importance sampling. Using the importance weights is the key algorithmic innovation of MA-Trace (and V-Trace), responsible for the strong performance we report. Indeed, already for 30 actor workers, using the weights is essential. Otherwise, the algorithm is unstable and suffers from poor asymptotic performance: on 7 out of 14 environments its final score is not better than 50% of MA-Trace.

Training scaling. The importance sampling enables V-Trace to be truly scalable in multi-node setups. MA-Trace enjoys the same property. Importantly, we do not observe any significant degradation in the training performance when trained in the multi-node setup.

Input for the critic network. We found that MA-Trace (state) performs slightly better than MA-Trace (obs) in many tasks, though the differences are small. However, in two harder tasks (*6h_vs_8z* and *corridor*), MA-Trace (state) learns much slower and often fails, unlike the (obs) variant. This is perhaps surprising, as the full state contains additional information (e.g., about invisible opponents).

Sharing actors networks. We follow a common approach of sharing the policy network between agents. MA-Trace with separate policies failed to learn on the 5 hardest tasks. In some works that use a shared network, e.g. [6], the observations are enriched with the agent ID to preserve individuality. This might be beneficial if agents should be assigned different roles within the team. However, we find these benefits rather minor and opt for the input provided by the environment (i.e., without ID).

4 CONCLUSIONS

In our work, we introduced MA-Trace, a new multi-agent reinforcement learning algorithm with theoretical guarantees. Despite its on-policy nature, the computations can be distributed to many workers with nearly perfect speed-up and no significant impact on the training quality. This was achieved by using importance sampling weights.

We believe that such a scheme could be used to strengthen other on-policy algorithms, which are typically considered stable and robust though less time-efficient in practice.

REFERENCES

[1] Josh Achiam. 2018. Spinning Up in Deep RL. https://spinningup.openai.com.

[2] Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. 2018. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research)*, Jennifer G. Dy and Andreas Krause (Eds.), Vol. 80. PMLR, 1406–1415. http://proceedings.mlr.press/v80/espeholt18a.html

[3] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual Multi-Agent Policy Gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 2974–2982. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17193

[4] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 6379–6390. https://proceedings.neurips.cc/paper/2017/hash/68a9750337a418a86fe06c1991a1d64c-Abstract.html

[5] Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research)*, Jennifer G. Dy and Andreas Krause (Eds.), Vol. 80. PMLR, 4292–4301. http://proceedings.mlr.press/v80/rashid18a.html

[6] Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. 2020. Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *CoRR* abs/2003.08839 (2020). arXiv:2003.08839 https://arxiv.org/abs/2003.08839

[7] Mikayel Samvelyan, Tabish Rashid, Christian Schröder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob N. Foerster, and Shimon Whiteson. 2019. The StarCraft Multi-Agent Challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor (Eds.). International Foundation for Autonomous Agents and Multiagent Systems, 2186–2188. http://dl.acm.org/citation.cfm?id=3332052

[8] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philiph H. S. Torr, Jakob Foerster, and Shimon Whiteson. 2019. The StarCraft Multi-Agent Challenge. *CoRR* abs/1902.04043 (2019).

[9] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[10] John N. Tsitsiklis and Benjamin Van Roy. 1997. An analysis of temporal-difference learning with function approximation. *IEEE Trans. Autom. Control.* 42, 5 (1997), 674–690. https://doi.org/10.1109/9.580874

[11] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre M. Bayen, and Yi Wu. 2021. The Surprising Effectiveness of MAPPO in Cooperative, Multi-Agent Games. *CoRR* abs/2103.01955 (2021). arXiv:2103.01955 https://arxiv.org/abs/2103.01955

[12] Michał Zawalski, Błażej Osiński, Henryk Michalewski, and Piotr Miłoś. 2021. Off-Policy Correction For Multi-Agent Reinforcement Learning. arXiv:cs.LG/2111.11229