# Model-Based Actor-Critic for Multi-Objective Reinforcement Learning with Dynamic Utility Functions

Extended Abstract

Johan Källström
Linköping University
Linköping, Sweden
johan.kallstrom@liu.se

Fredrik Heintz
Linköping University
Linköping, Sweden
fredrik.heintz@liu.se

## ABSTRACT

Many real-world problems require a trade-off between multiple conflicting objectives. Decision-makers' preferences over solutions to such problems are determined by their utility functions, which convert multi-objective values to scalars. In some settings, utility functions change over time, and the goal is to find methods that can efficiently adapt an agent's policy to changes in utility. Previous work on learning with dynamic utility functions has focused on model-free methods, which often suffer from poor sample efficiency. In this work, we instead propose a model-based actor-critic, which explores with diverse utility functions through imagined rollouts within a learned world model between interactions with the real environment. An experimental evaluation on Minecart, a well-known benchmark for multi-objective reinforcement learning, shows that by learning a model of the environment the quality of the agent's policy is improved compared to model-free algorithms.

## KEYWORDS

Multiple Objectives; Reinforcement Learning; Model-Based Learning

**ACM Reference Format:**
Johan Källström and Fredrik Heintz. 2023. Model-Based Actor-Critic for Multi-Objective Reinforcement Learning with Dynamic Utility Functions: Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

## 1 INTRODUCTION

Multi-objective reinforcement learning (MORL) provides methods that allow agents to learn optimal policies in multi-objective Markov decision processes (MOMDPs) [4, 10]. A MOMDP uses vector reward signals [12], with each element representing one of the objectives, in contrast to the scalar rewards used in single-objective reinforcement learning [11], resulting in vector returns. To evaluate the outcomes of different solutions in relation to each other, a utility function is used to convert the vector return to a scalar, representing a specific trade-off among the objectives.

In some scenarios, the utility function is not fixed over time. In MORL research this is referred to as the *Dynamic Utility Scenario* [4]. For instance, the utility of a mining company's distribution of its equipment over its mines will change as the prices of different ores change. To handle such changes in utility efficiently, it is desirable to reuse information from learning with previously encountered utility functions, instead of restarting learning from scratch. Recent work proposes to use a single neural network to represent multiple policies, by conditioning the network on preference weights, and enforces diversity in the contents of the replay buffer in terms of the *crowding distance* of the returns of stored trajectories [1, 2, 7, 14]. Performance is improved compared to training several separate networks. Preference weights represent a linear utility function, where each weight specifies the corresponding objective's importance in relation to the other objectives. Conditioned networks have proven useful in the dynamic utility scenario and other settings [5, 8, 9, 15].

Previous work in MORL for learning with dynamic utility functions has focused on model-free learning, which often suffers from poor sample efficiency. In this work, we instead propose a model-based actor-critic, based on DreamerV2 [3]. An experimental evaluation shows that the model-based agent (MO-Dreamer) significantly outperforms the model-free state-of-the-art in an environment with frequent utility changes. To the best of our knowledge this is the first study of model-based multi-objective reinforcement learning in the dynamic utility scenario.

## 2 METHOD

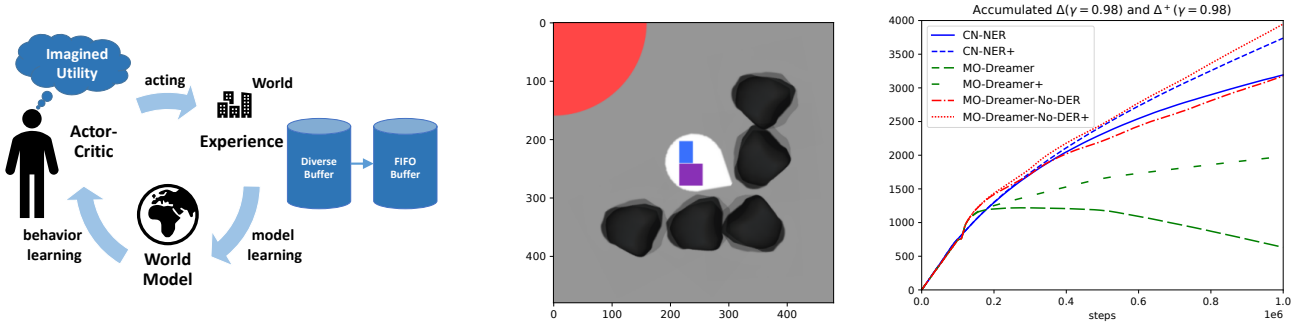
An overview of MO-Dreamer is shown in Figure 1. MO-Dreamer uses the same recurrent state space model, image predictor, and discount predictor as DreamerV2, but predicts vector rewards to enable modelling of MOMDPs: $\hat{\boldsymbol{r}}_t \sim p_\phi(\hat{\boldsymbol{r}}_t | h_t, z_t)$, where $h_t$ and $z_t$ are given by the recurrent and representation models with parameters $\phi$. We make the assumption that the elements of the multi-objective reward are statistically independent, and represent them as individual univariate Gaussians with unit variance in the world model. The reward predictor's contribution to the world model loss is then $-\sum_{i=1}^{n} \ln p_\phi(r_{i,t} | h_t, z_t)$ for a MOMDP with $n$ objectives.

The world model is trained with data collected from the agent's past experiences with the real environment. In the dynamic utility scenario, it is important to quickly learn environment features suitable for multiple utility functions. To ensure diversity in the data used in the early stages of learning, we use two connected replay buffers and enforce diversity in the first buffer based on crowding distance as in [1]. When the main buffer is full, the trajectory that contributes the least to diversity is moved to the secondary buffer, which uses a first-in-first-out (FIFO) principle. We then sample trajectories from either buffer with a probability proportional to the number of environment steps contained in each of them. This means that the early stages of learning will prioritise sampling from the diverse buffer, while later stages of learning will sample from either buffer with equal probability.

*Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.* 

**Table 1: Average episodic Δ and $\Delta^+$ over ten iterations**

| Algorithm | Δ overall | Δ last 250k steps | $\Delta^+$ overall | $\Delta^+$ last 250k steps |
|---|---|---|---|---|
| CN-NER | 0.0791 ± 0.0041 | 0.0292 ± 0.0024 | 0.0926 ± 0.0038 | 0.0476 ± 0.0023 |
| MO-Dreamer | **0.0112** ± 0.0159 | **−0.0185** ± 0.0018 | **0.0348** ± 0.0124 | **0.0096** ± 0.0007 |
| MO-Dreamer-No-DER | 0.0583 ± 0.0415 | 0.0309 ± 0.0424 | 0.0724 ± 0.0329 | 0.0484 ± 0.0330 |




**(a) Model-based multi-objective actor-critic** **(b) Minecart environment** **(c) Average cumulative Δ and $\Delta^+$**

**Figure 1: Learning method (a), evaluation environment (b), and performance over ten iterations (c)**

We use an actor-critic setup to learn the behaviour of the agent, where the critic learns a multi-objective value function that guides the updates of the actor's policy. To enable single-network representations of the action distributions as well as the state value functions of multiple policies, we condition both actor and critic on the current utility weights: $\hat{a}_t \sim p_\psi(\hat{a}_t|\hat{z}_t, \boldsymbol{w})$ and $\boldsymbol{v}_\xi(\hat{z}_t, \boldsymbol{w}) \approx E_{p_\phi p_\psi}\left[\sum_{\tau \geq t} \hat{\gamma}^{\tau-t} \hat{\boldsymbol{r}}_\tau | \boldsymbol{w}\right]$, where $\psi$ and $\xi$ are the parameters of the actor and critic. When the critic gets a certain weight as input in combination with the current model state, it will output the corresponding vector of objective values. When the actor gets a weight and model state pair as input, it will select the best policy for optimising the corresponding utility. The actor-critic is trained through imagination rollouts within the world model, which makes it possible to revisit states and utility weights previously encountered to improve the policy for the corresponding situation.

## 3 EXPERIMENTAL EVALUATION

We perform experiments on the Minecart benchmark [1], illustrated in Figure 1, with the default configuration for the mines, $\gamma = 0.98$, and train for 1M steps. We use frequent utility changes, where changes occur in each episode. As in previous work, we use a linear utility function represented by preference weights. For linear utility functions a convex coverage set (CCS) contains all optimal policies [10, 13]. As evaluation metric we use regret against an approximated CCS, calculated using the heuristic proposed in [1]: $\Delta(\boldsymbol{g}, \boldsymbol{w}) = \boldsymbol{V}^*_{\boldsymbol{w}} \cdot \boldsymbol{w} - \boldsymbol{g} \cdot \boldsymbol{w}$, where $\boldsymbol{V}^*{}_{\boldsymbol{w}}$ is the optimal value in the CCS for the current weight vector $\boldsymbol{w}$, and $\boldsymbol{g}$ is the discounted return. Since we are using an estimate for the optimal utility, there is a chance that the regret estimate could become negative, if the agents learn policies that outperform the heuristic. To investigate if, and to what extent, this happens we also calculate results where negative regrets are clipped and denote those metrics by $\Delta^+ = max(\Delta, 0)$.

We use utility conditioned DQN [6] with diverse experience replay (DER) buffer [1] combined with near on-policy experience replay (NER) [14] as baseline (CN-NER), since prior work has shown that it has state-of-the-art performance in environments with frequent utility changes. Table 1 and Figure 1 show that MO-Dreamer significantly outperforms the model-free baseline in terms of average episodic regret and cumulative regret. We can also see that MO-Dreamer improves on the approximate CCS after convergence, as indicated by the negative value of the average episodic Δ and the sloping curve of the average cumulative Δ. An ablation study shows that MO-Dreamer without diversified dataset and sampling performs significantly worse than the full agent. In addition to reducing regret, MO-Dreamer completes on average 57183.2 ± 760.4 episodes over the allocated 1M environment steps, while the baseline CN-NER completes only 40408.3 ± 893.7 episodes. This illustrates the high quality of the learned world model, resulting in a robust policy that can complete many successful episodes in few steps.

## 4 CONCLUSION

In this work, we proposed MO-Dreamer, a model-based agent for learning in environments with dynamic utility functions. MO-Dreamer uses imagination rollouts with a diverse set of utility functions to explore which policy to follow to optimise the return for a given set of objective preferences. An experimental evaluation showed that MO-Dreamer significantly outperforms the model-free state-of-the-art algorithm for MORL on the Minecart benchmark with frequently changing preference weights. In future work we intend to study how learned world models can be used for transfer learning in multi-objective decision-making problems. For instance, we would like to study how the world model learned when acting with a linear utility function can be used to transfer to non-linear utility functions.

## ACKNOWLEDGMENTS

This work was supported by the Swedish Governmental Agency for Innovation Systems (grant NFFP7/2017-04885), and the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by the supercomputing resource Berzelius provided by National Supercomputer Centre at Linköping University and the Knut and Alice Wallenberg foundation.

## REFERENCES

[1] Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. 2019. Dynamic weights in multi-objective deep reinforcement learning. In *International Conference on Machine Learning*. PMLR, 11–20.

[2] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[3] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. 2020. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193* (2020).

[4] Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems* 36, 1 (2022), 1–59.

[5] Johan Källström and Fredrik Heintz. 2019. Tunable dynamics in agent-based simulation using multi-objective reinforcement learning. In *Adaptive and Learning Agents Workshop (ALA-19) at AAMAS, Montreal, Canada, May 13-14, 2019*. 1–7.

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[7] Xiaodong Nian, Athirai A Irissappane, and Diederik Roijers. 2020. DCRAC: Deep conditioned recurrent actor-critic for multi-objective partially observable environments. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 931–938.

[8] David O'Callaghan and Patrick Mannion. 2021. Tunable behaviours in sequential social dilemmas using multi-objective reinforcement learning. In *Proceedings of the 20th international conference on autonomous agents and multiagent systems*. 1610–1612.

[9] Mathieu Reymond, Eugenio Bargiacchi, and Ann Nowé. 2022. Pareto Conditioned Networks. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. 1110–1118.

[10] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research* 48 (2013), 67–113.

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

[12] Peter Vamplew, Benjamin J Smith, Johan Källström, Gabriel Ramos, Roxana Rădulescu, Diederik M Roijers, Conor F Hayes, Fredrik Heintz, Patrick Mannion, Pieter JK Libin, Richard Dazeley, and Cameron Foale. 2022. Scalar reward is not enough: A response to Silver, Singh, Precup and Sutton (2021). *Autonomous Agents and Multi-Agent Systems* 36, 2 (2022), 1–19.

[13] Peter Vamplew, John Yearwood, Richard Dazeley, and Adam Berry. 2008. On the limitations of scalarisation for multi-objective reinforcement learning of pareto fronts. In *Australasian joint conference on artificial intelligence*. Springer, 372–378.

[14] Shang Wang, Mathieu Reymond, Athirai A Irissappane, and Diederik M Roijers. 2022. Near On-Policy Experience Sampling in Multi-Objective Reinforcement Learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. 1756–1758.

[15] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. 2019. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 14636–14647.