Off-the-Grid MARL: Datasets with Baselines for Offline Multi-Agent Reinforcement Learning

Claude Formanek[1,2], Asad Jewa[1], Jonathan Shock[2], Arnu Pretorius[1]

[1] InstaDeep, [2] University of Cape Town

TLDR: Offline MARL is a nascent research field that has, to date, been hampered by the lack of standardised datasets to measure progress. To address this, we provide a diverse collection of high-quality multi-agent datasets with baselines.

⇩ Download

Being able to harness the power of large datasets for developing autonomous systems could unlock enormous value for real-world applications. Many important industrial systems are multi-agent in nature and, in industry, large amounts of logged data from distributed system processes are stored. Offline MARL provides a promising paradigm for building effective decentralised online controllers from such datasets.

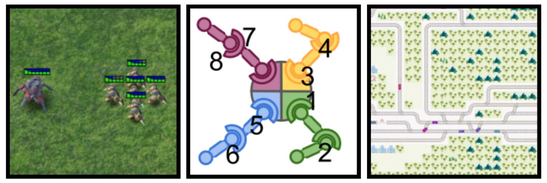
However, offline MARL is still in its infancy, and, therefore, lacks standardised benchmarks, baselines and evaluation protocols typically found in more mature subfields of RL. This deficiency makes it difficult for the community to sensibly measure progress. In this work, we aim to fill this gap by releasing off-the-grid MARL (OG-MARL): a framework for generating offline MARL datasets and algorithms. We release an initial set of datasets and baselines for cooperative offline MARL Our datasets provide settings that are characteristic of real-world systems, including complex dynamics, non-stationarity, partial observability, suboptimality and sparse rewards, and are generated from popular online MARL benchmarks.

Framework for Offline MARL

```python
from og_marl.environments.smac import SMAC
from og_marl.offline_tools import MAOfflineEnvLogger
from og_marl.systems.executor_base import RandomExecutor

# Instantiate environment
env = SMAC("3m")

# Executor that chooses random actions for each agent
executor = RandomExecutor()

# Wrap env in offline logger
wrapped_env = MAOfflineEnvLogger(
    environment=env
)

# Reset environment
timestep = wrapped_env.reset()

# Get some random actions from executor
all_agent_actions = executor.select_actions()

# Step the environment
next_timestep = wrapped_env.step(
    actions=all_agent_actions
)
```

- Utilities for generating and analysing offline MARL datasets.
- Standardised re-implementation of popular offline MARL algorithms.
- Framework for developing novel offline MARL algorithms.
- Website to store and distribute datasets.

Diverse Multi-Agent Environments

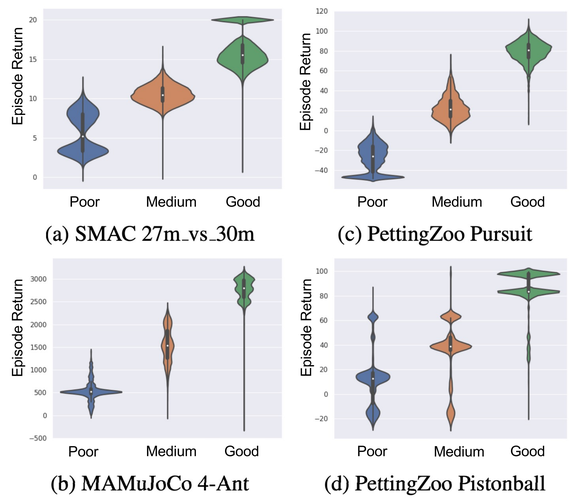


(a) SMAC (b) MAMuJoCo (c) Flatland

(d) Pistonball (e) Coop Pong (f) Pursuit

We provide datasets for several popular MARL benchmark environments including the SMAC (Samvelyan et al., 2019), Multi-Agent MuJoCo (Peng et al., 2021), Flatland (Mohanty et al., 2020) and environments from PettingZoo (Terry et al., 2021). Together these environments cover a broad range of task characteristics including: i) discrete and continuous action spaces, ii) vector and pixel-based observations, iii) dense and sparse rewards, iv) a varying number of agents (from 2 to 27 agents), and finally v) heterogeneous and homogeneous agents.

Dataset Quality Classes

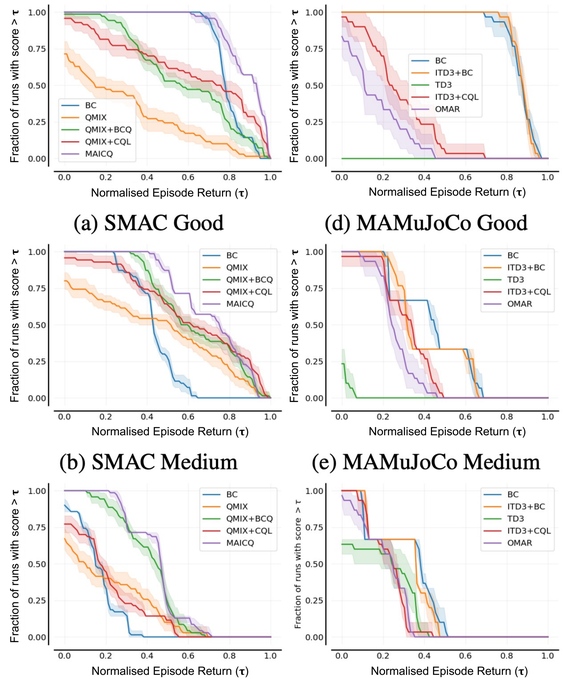


(a) SMAC 27m_vs_30m (c) PettingZoo Pursuit

(b) MAMuJoCo 4-Ant (d) PettingZoo Pistonball

To generate the transitions in the datasets, we recorded environment interactions with online MARL algorithms . For each scenario, we provide three types of datasets. The dataset types are characterised by the quality of the joint policy that generated the trajectories in the dataset. We use violin plots to visualised the distribution of episode returns in the datasets.

Comprehensive Baseline Results



(a) SMAC Good (d) MAMuJoCo Good

(b) SMAC Medium (e) MAMuJoCo Medium

(c) SMAC Poor (f) MAMuJoCo Poor

We provide baselines for all of our datasets. Above we show the performance profiles [Agarwal et al., 2021] for popular offline MARL algorithms [Pan et al.,(2022), Yang et al., (2021)]. The performance profiles aggregate across all SMAC and MA MuJoCo scenarios respectively. Roughly speaking, if a curve is above another it means that algorithms performs better than the other, with respect to the episode return at the end of offline training.

References

- Agarwal, Rishabh, et al., "Deep reinforcement learning at the edge of the statistical precipice", *Advances in Neural Information Processing Systems* (2021).
- Samvelyan, Rashid, et al., "The StarCraft multi-agent challenge", *Autonomous Agents and Multi-Agent Systems (2019)*
- Nygren*, et al., "*Flatland-RL : Multiagent reinforcement learning on trains*", ArXiv (2020)*
- Terry, Black, et al., "PettingZoo: Gym for multi-agent reinforcement learning", Advances in Neural Information Processing Systems (2021)
- Pan, Huang, et al., "Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification", *International Conference on Machine Learning (2022)*
- Yang, Ma, et al. "Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning", *Advances in Neural Information Processing Systems* (2021)

DEEP LEARNING INDABA

AAMAS 2023 LONDON