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

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

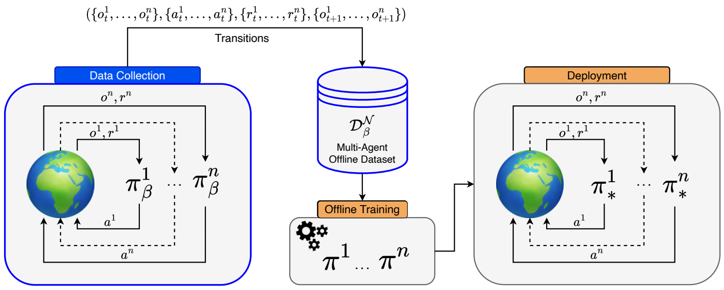
Being able to harness the power of large datasets for developing cooperative multi-agent controllers promises to unlock enormous value for real-world applications. Many important industrial systems are multi-agent in nature and are difficult to model using bespoke simulators. However, in industry, distributed processes can often be recorded during operation, and large quantities of demonstrative data stored. Offline multi-agent reinforcement learning (MARL) provides a promising paradigm for building effective decentralised controllers from such datasets. However, offline MARL is still in its infancy and therefore lacks standardised benchmark datasets and baselines typically found in more mature subfields of reinforcement learning (RL). These deficiencies make 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 growing repository of high-quality datasets with baselines for cooperative offline MARL research. Our datasets provide settings that are characteristic of real-world systems, including complex environment dynamics, heterogeneous agents, non-stationarity, many agents, partial observability, suboptimality, sparse rewards and demonstrated coordination. For each setting, we provide a range of different dataset types (e.g. `Good`, `Medium`, `Poor`, and `Replay`) and profile the composition of experiences for each dataset. We hope that OG-MARL will serve the community as a reliable source of datasets and help drive progress, while also providing an accessible entry point for researchers new to the field.

1 Introduction

RL algorithms typically require extensive online interactions with an environment to be able to learn robust policies (Yu, 2018). This limits the extent to which previously-recorded experience may be leveraged for RL applications, forcing practitioners to instead rely heavily on optimised environment simulators that are able to run quickly and in parallel on modern compute hardware.

In a simulation, it is not atypical to be able to generate years of operating behaviour of a specific system (Berner et al., 2019; Vinyals et al., 2019). However, achieving this level of online data generation throughput in real-world systems, where a realistic simulator is not readily available, can be challenging or near impossible. More recently, the field of offline RL has offered a solution to this challenge by bridging the gap between RL and supervised learning. In offline RL, the aim is to develop algorithms that are able to leverage large existing datasets of sequential decision-making to learn optimal control strategies that can be deployed online (Levine et al., 2020). Many researchers

Submitted to the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks. Do not distribute.




```python
from og_marl import SMAC
from og_marl import QMIX
from og_marl import OfflineLogger

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

# Wrap env in offline logger
env = OfflineLogger(env)

# Make multi-agent system
system = QMIX(env)

# Collect data
system.run_online()

# Load dataset
dataset = env.get_dataset("Good")

# Train offline
system.run_offline(dataset)
```

Figure 1: **Top**: an illustration of offline MARL. Behaviour policies collect experiences and store them in an offline dataset. New policies are trained from the offline data without any online environment interactions. At the end of training, the policies are deployed in the environment. **Right**: a code snippet demonstrating how to record new datasets, as well as load existing ones, using OG-MARL.

believe that offline RL could help unlock the full potential of RL when applied to the real world, where success has been limited (Dulac-Arnold et al., 2021).

Although the field of offline RL has experienced a surge in research interest in recent years (Prudencio et al., 2023), the focus on offline approaches specific to the multi-agent setting has remained relatively neglected, despite the fact that many real-world problems are naturally formulated as multi-agent systems (e.g. traffic management (Zhang et al., 2019), a fleet of ride-sharing vehicles (Sykora et al., 2020), a network of trains (Mohanty et al., 2020) or electricity grid management (Khattar and Jin, 2022)). Moreover, systems that require multiple agents (programmed and/or human) to execute coordinated strategies to perform optimally, arguably have a higher barrier to entry when it comes to creating bespoke simulators to model their online operating behaviour.

Offline RL research in the single agent setting has benefited greatly from publicly available datasets and benchmarks such as D4RL (Fu et al., 2020) and RL Unplugged (Gulcehre et al., 2020). Without such offerings in the multi-agent setting to help standardise research efforts and evaluation, it remains challenging to gauge the state of the field and reproduce results from previous work. Ultimately, to develop new ideas that drive the field forward, standardised sets of tasks and baselines are required.

In this paper, we present OG-MARL, a rich set of datasets specifically curated for cooperative offline MARL. We generated diverse datasets on a range of popular cooperative MARL environments. For each environment, we provide different types of behaviour resulting in *Good*, *Medium* and *Poor* datasets as well as *Replay* datasets (a mixture of the previous three). We developed and applied a quality assurance methodology to validate our datasets to ensure that they contain a diverse spread of experiences. Together with our datasets, we provide initial baseline results using state-of-the-art offline MARL algorithms.

The OG-MARL code and datasets are publicly available through our website.[1] Additionally, we invite the community to contribute their own datasets to the growing repository on OG-MARL and use our website as a platform for storing and distributing datasets for the benefit of the research community. We hope the lessons contained in our methodology for generating and validating datasets help future researchers to produce high-quality offline MARL datasets and help drive progress.

2 Related Work

Datasets. In the single-agent RL setting, D4RL (Fu et al., 2020) and RL Unplugged (Gulcehre et al., 2020) have been important contributions, providing a comprehensive set of offline datasets for benchmarking offline RL algorithms. While not originally included, D4RL was later extended by Lu et al. (2022) to incorporate datasets with pixel-based observations, which they highlight as a notable

[1] https://sites.google.com/view/og-marl

deficiency of other datasets. The ease of access to high-quality datasets provided by D4RL and RL Unplugged has enabled the field of offline RL to make rapid progress over the past years (Kostrikov et al., 2021; Ghasemipour et al., 2022; Nakamoto et al., 2023). However, these repositories lack datasets for MARL, which we believe, alongside additional technical difficulties such as large joint action spaces (Yang et al., 2021), has resulted in slower progress in the field.

Offline Multi-Agent Reinforcement Learning. To date, there has been a limited number of papers published on cooperative offline MARL, resulting in benchmarks, datasets and algorithms that do not adhere to any unified standard, making comparisons between works difficult. In brief, Zhang et al. (2021) carried out an in-depth theoretical analysis of finite-sample offline MARL. Jiang and Lu (2021) proposed a decentralised multi-agent version of the popular offline RL algorithm BCQ (Fujimoto et al., 2019) and evaluated it on their own datasets of a multi-agent version of MuJoCo (MAMuJoCo) (Peng et al., 2021). Yang et al. (2021) highlighted how extrapolation error accumulates rapidly in the number of agents and propose a new method they call *Implicit Constraint Q-Learning* (ICQ) to address this. The authors evaluate their method on their own datasets collected using the popular *StarCraft Mulit-Agent Challenge* (SMAC) (Samvelyan et al., 2019). Pan et al. (2022) showed that *Conservative Q-Learning* (CQL) (Kumar et al., 2020), a very successful offline RL method, does not transfer well to the multi-agent setting since the multi-agent policy gradients are prone to uncoordinated local optima. To overcome this, the authors proposed a zeroth-order optimization method to better optimize the conservative value functions, and evaluate their method on their own datasets of a handful of SMAC scenarios, the two agent HalfCheetah scenario from MAMuJoCo and some simple Multi Particle Environments (MPE) (Lowe et al., 2017). Meng et al. (2021) propose a *multi-agent decision transformer* (MADT) architecture, which builds on the *decision transformer* (DT) (Chen et al., 2021), and demonstrated how it can be used for offline pre-training and online fine-tuning in MARL by evaluating their method on their own SMAC datasets. Barde et al. (2023) explored a model-based approach for offline MARL and evaluated their method on MAMuJoCo.

Datasets and baselines for Offline MARL. In all of the aforementioned works, the authors generate their own datasets for their experiments and provide only a limited amount of information about the composition of their datasets (e.g. spread of episode returns and/or visualisations of the behaviour policy). Furthermore, each paper proposes a novel algorithm and typically compares their proposal to a set of baselines specifically implemented for their work. The lack of commonly shared benchmark datasets and baselines among previous papers has made it difficult to compare the relative strengths and weaknesses of these contributions and is one of the key motivations for our work.

Finally, we note works that have already made use of the pre-release version of OG-MARL. Formanek et al. (2023) investigated selective "reincarnation" in the multi-agent setting and Zhu et al. (2023) explored using diffusion models to learn policies in offline MARL. Both these works made use of OG-MARL datasets for their experiments, which allows for easier reproducibility and more sound comparison with future work using OG-MARL.

3 Preliminaries

Multi-Agent Reinforcement Learning. There are three main formulations of MARL tasks: competitive, cooperative and mixed. The focus of this work is on the cooperative setting. Cooperative MARL can be formulated as a *decentralised partially observable Markov decision process* (Dec-POMDP) (Bernstein et al., 2002). A Dec-POMDP consists of a tuple $\mathcal{M} = (\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}, \{\mathcal{O}^i\}, P, E, \rho_0, r, \gamma)$, where $\mathcal{N} \equiv \{1, \dots, n\}$ is the set of n agents in the system and $s \in \mathcal{S}$ describes the full state of the system. The initial state distribution is given by ρ_0. Each agent $i \in \mathcal{N}$ receives only partial information from the environment in the form of a local observation o^i_t, given according to an emission function $E(o_t|s_t, i)$. At each timestep t, each agent chooses an action $a^i_t \in \mathcal{A}^i$ to form a joint action $\mathbf{a}_t \in \mathcal{A} \equiv \prod_i^N \mathcal{A}^i$. Due to partial observability, each agent typically maintains an observation history $o^i_{0:t} = (o^i_0, \dots, o^i_t)$, or implicit memory, on which it conditions its policy $\mu^i(a^i_t|o^i_{0:t})$, when choosing an action. The environment then transitions to a new state in response to the joint action selected in the current state, according to the state transition function $P(s_{t+1}|s_t, \mathbf{a}_t)$ and provides a shared scalar reward to each agent according to a reward function $r(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. We define an agent's return as its discounted cumulative rewards over the T episode timesteps, $G = \sum_{t=0}^{T} \gamma^t r_t$, where $\gamma \in (0, 1]$ is the discount factor. The goal of MARL in a Dec-POMDP is to find a joint policy $(\pi^1, \ldots, \pi^n) \equiv \pi$ such that the return of each agent i, following π^i, is maximised with respect to the other agents' policies, $\pi^{-i} \equiv (\pi \backslash \pi^i)$. That is, we aim to find π such that $\forall i : \pi^i \in \arg\max_{\hat{\pi}^i} \mathbb{E}\left[G \mid \hat{\pi}^i, \pi^{-i}\right]$

Offline Reinforcement Learning. An offline RL algorithm is trained on a static, previously collected dataset $\mathcal{D}_\beta$ of transitions (o_t, a_t, r_t, o_{t+1}) from some (potentially unknown) behaviour policy π_β, without any further online interactions. There are several well-known challenges in the offline RL setting which have been explored, predominantly in the single-agent literature. The primary issues are related to different manifestations of data distribution mismatch between the offline data and the induced online data. Levine et al. (2020) provide a detailed survey of the problems and solutions in offline RL.

Offline Multi-Agent Reinforcement Learning. In the multi-agent setting, offline MARL algorithms are designed to learn an optimal *joint* policy $(\pi^1, \ldots, \pi^n) \equiv \pi$, from a static dataset $\mathcal{D}_\beta^{\mathcal{N}}$ of previously collected multi-agent transitions $(\{o_t^1, \ldots, o_t^n\}, \{a_t^1, \ldots, a_t^n\}, \{r_t^1, \ldots, r_t^n\}, \{o_{t+1}^1, \ldots, o_{t+1}^n\})$, generated by a set of interacting behaviour policies $(\pi_\beta^1, \ldots,\ \pi_\beta^n) \equiv \pi_\beta$.

4 Task Properties

In order to design an offline MARL benchmark which is maximally useful to the community, we carefully considered the properties that the environments and datasets in our benchmark should satisfy. A major drawback in most prior work has been the limited diversity in the tasks that the algorithms were evaluated on. Meng et al. (2021) for example only evaluated their algorithm on SMAC datasets and Jiang and Lu (2021) only evaluated on MAMuJoCo datasets. This makes it difficult to draw strong conclusions about the generalisability of offline MARL algorithms. Moreover, these environments fail to test the algorithms along dimensions which may be important for real-world applications. In this section, we outline the properties we believe are important for evaluating offline MARL algorithms.

Centralised and Independent Training. The environments supported in OG-MARL are designed to test algorithms that use decentralised execution, i.e. at execution time, agents need to choose actions based on their local observation histories only. However, during training, centralisation (i.e. sharing information between agents) is permissible, although not required. *Centralised training with decentralised execution* (CTDE) (Kraemer and Banerjee, 2016) is one of the most popular MARL paradigms and is well-suited for many real-world applications. Being able to test both centralised and independent training algorithms is important because it has been shown that neither paradigm is consistently better than the other (Lyu et al., 2021). As such, both types of algorithms can be evaluated using OG-MARL datasets and we also provide baselines for both centralised and independent training.

Different types of Behaviour Policies. We generated datasets with several different types of behaviour policies including policies trained using online MARL with fully independent learners (e.g. independent DQN and independent TD3), as well as CTDE algorithms (e.g. QMIX and MATD3). Furthermore, some datasets generated with CTDE algorithms used a state-based critic while others used a joint-observation critic. It was important for us to consider both of these critic setups as they are known to result in qualitatively different policies (Lyu et al., 2022). More specific details of which algorithms were used to generate which datasets can be found in Table B.1 in the appendix.

Partial Information. It is common for agents to receive only local information about their environment, especially in real-world systems that rely on decentralised components. Thus, some of the environments in OG-MARL test an algorithm's ability to leverage agents' *memory* in order to choose optimal actions based only on partial information from local observations. This is in contrast to settings such as MAMuJoCo where prior methods (Jiang and Lu, 2021; Pan et al., 2022) achieved reasonable results without instilling agents with any form of memory.

Different Observation Modalities. In the real world, agent observations come in many different forms. For example, observations may be in the form of a feature vector or a matrix representing a pixel-based visual observation. Lu et al. (2022) highlighted that prior single-agent offline RL datasets failed to test algorithms on high-dimensional pixel-based observations. OG-MARL tests algorithms on a diverse set of observation modalities, including feature vectors and pixel matrices of different sizes.

Continuous and Discrete Action Spaces. The actions an agent is expected to take can be either discrete or continuous across a diverse range of applications. Moreover, continuous action spaces can often be more challenging for offline MARL algorithms as the larger action spaces make them more prone to extrapolation errors, due to out-of-distribution actions . OG-MARL supports a range of environments with both discrete and continuous actions.

Homogeneous and Heterogeneous Agents. Real-world systems can either be homogeneous or heterogeneous in terms of the types of agents that comprise the system. In a homogeneous system, it may be significantly simpler to train a single policy and copy it to all agents in the system. On the other hand, in a heterogenous system, where agents may have significantly different roles and responsibilities, this approach is unlikely to succeed. OG-MARL provides datasets from environments that represent both homogeneous and heterogeneous systems.

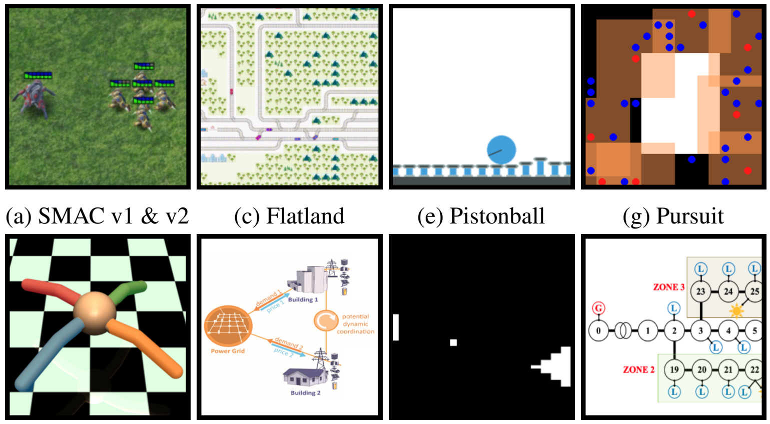
Number of Agents. Practical MARL systems may have a large number of agents. Most prior works to date have evaluated their algorithms on environments with typically fewer than 8 agents (Pan et al., 2022; Yang et al., 2021; Jiang and Lu, 2021). In OG-MARL, we provide datasets with between 2 and 27 agents, to better evaluate *large-scale* offline MARL (see Table B.1).

Sparse Rewards. Sparse rewards are challenging in the single-agent setting, but in the multi-agent setting, it can be even more challenging due to the multi-agent credit assignment problem (Zhou et al., 2020). Prior works focused exclusively on dense reward settings (Pan et al., 2022; Yang et al., 2021). To overcome this, OG-MARL also provides datasets with sparse rewards.

Team and Individual Rewards. Some environments have team rewards while others can have an additional local reward component. Team rewards exacerbate the multi-agent credit assignment problem, and having a local reward component can help mitigate this. However, local rewards may result in sub-optimality, where agents behave too greedily with respect to their local reward and as a result jeopardize achieving the overall team objective. OG-MARL includes tasks to test algorithms along both of these dimensions.

Procedurally Generated and Stochastic Environments. Some popular MARL benchmark environments are known to be highly deterministic (Ellis et al., 2022). This limits the extent to which the generalisation capabilities of algorithms can be evaluated. Procedurally generated environments have proved to be a useful tool for evaluating generalisation in single-agent RL (Cobbe et al., 2020). In order to better evaluate generalisation in offline MARL, OG-MARL includes stochastic tasks that make use of procedural generation.

Realistic Multi-Agent Domains. Almost all prior offline MARL works have evaluated their algorithms exclusively on game-like environments such as StarCraft (Yang et al., 2021) and particle simulators (Pan et al., 2022). Although a large subset of open research questions may still be readily investigated in such simulated environments, we argue that in order for offline MARL to become more practically relevant, benchmarks in the research community should begin to closer reflect real-world problems of interest. Therefore, in addition to common game-like benchmark environments, OG-MARL also supports environments which simulate more real-world like problems including energy management and control (Vazquez-Canteli et al., 2020; Wang et al., 2021). While there remains a large gap between these environments and truly real-world settings, it is a step in the right direction to keep pushing the field forward and enable useful contributions in the development of new algorithms and improving our understanding of key difficulties and failure modes.



(a) SMAC v1 & v2 (c) Flatland (e) Pistonball (g) Pursuit

(b) MAMuJoCo (d) CityLearn (f) Co-op Pong (h) Voltage Control

Figure 2: MARL environments for which we provide datasets in OG-MARL.

5 Environments

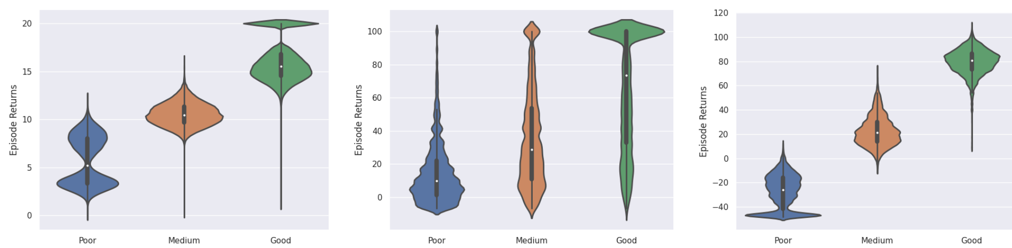
SMAC v1 *(hetero- and homogeneous agents, local observations)*. SMAC is the most popular cooperative offline MARL environment used in the literature(Gorsane et al., 2022). SMAC focuses on the micromanagement challenge in StarCraft 2 where each unit is controlled by an independent agent that must learn to cooperate and coordinate based on local (partial) observations. SMAC played an important role in moving the MARL research community beyond grid-world problems and has also been very popular in the offline MARL literature (Yang et al., 2021; Meng et al., 2021; Pan et al., 2022). Thus, it was important for OG-MARL to support a range of SMAC scenarios.

SMAC v2 *(procedural generation, local observations)*. Recently some deficiencies in SMAC have been brought to light. Most importantly, SMAC is highly deterministic, and agents can therefore learn to *memorise* the best policy by conditioning on the environment timestep only. To address this, SMACv2 (Ellis et al., 2022) was recently released and includes non-deterministic scenarios, thus providing a more challenging benchmark for MARL algorithms. In OG-MARL, we publicly release the first set of SMACv2 datasets.

MAMuJoCo *(hetero- and homogeneous agents, continuous actions)*. The MuJoCo environment (Todorov et al., 2012) has been an important benchmark that helped drive research in continuous control. More recently, MuJoCo has been adapted for the multi-agent setting by introducing independent agents that control different subsets of the whole MuJoCo robot (MAMuJoCo) (Peng et al., 2021). MAMuJoCo is an important benchmark because there are a limited number of continuous action space environments available to the MARL research community. MAMuJoCo has also been widely adopted in the offline MARL literature (Jiang and Lu, 2021; Pan et al., 2022). Thus, in OG-MARL we provide the largest openly available collection of offline datasets on scenarios in MAMuJoCo (Pan et al. (2022), for example, only provided a single dataset on 2-Agent HalfCheetah).

PettingZoo *(pixel observations, discrete and continuous actions)*. OpenAI's Gym (Brockman et al., 2016) has been widely used as a benchmark for single agent RL. PettingZoo is a gym-like environment-suite for MARL (Terry et al., 2021) and provides a diverse collection of environments. In OG-MARL, we provide a general-purpose environment wrapper which can be used to generate new datasets for any PettingZoo environment. Additionally, we provide initial datasets on three PettingZoo environments including *PistonBall*, *Co-op Pong* and *Pursuit* (Gupta et al., 2017). We chose these environments because they have visual (pixel-based) observations of varying sizes; an important dimension along which prior works have failed to evaluate their algorithms.

Flatland *(real-world problem, procedural generation, sparse local rewards)*. The train scheduling problem is a real-world challenge with significant practical relevance. Flatland (Mohanty et al., 2020) is a simplified 2D simulation of the train scheduling problem that is an appealing benchmark for



(a) SMAC 27m_vs_30m (b) PettingZoo Co-op Pong (c) PettingZoo Pursuit

Figure 3: Violin plots of the probability distribution of episode returns for selected datasets in OG-MARL. In blue the `Poor` datasets, in orange the `Medium` datasets and in green the `Good` datasets. Wider sections of the violin plot represent a higher probability of sampling a trajectory with a given episode return, while the thinner sections correspond to a lower probability. The violin plots also include the median, interquartile range and min/max episode return for the datasets.

cooperative MARL for several reasons. Firstly, it randomly generates a new train track layout and timetable at the start of each episode, thus testing the generalisation capabilities of MARL algorithms to a greater degree than many other environments. Secondly, Flatland has a very sparse and noisy reward signal, as agents only receive a reward on the final timestep of the episode. Finally, agents have access to a local reward component. These properties make the Flatland environment a novel, challenging and realistic benchmark for offline MARL.

Voltage Control and CityLearn *(real-world problem, continuous actions)*. Energy management (Yu et al., 2021) is another appealing real-world application for MARL, especially given the large potential efficiency gains and corresponding positive effects on climate change that could be had (Rolnick et al., 2022). As such, we provide datasets for two challenging MARL environments related to energy management. Firstly, we provide datasets for the *Active Voltage Control on Power Distribution Networks* environment (Wang et al., 2021). Secondly, we provide datasets for the CityLearn environment (Vazquez-Canteli et al., 2020) where the goal is to develop agents for distributed energy resource management and demand response between a network of buildings with batteries and photovoltaics.

6 Datasets

To generate the transitions in the datasets, we recorded environment interactions of partially trained online algorithms, as has been common in prior works for both single-agent (Gulcehre et al., 2020) and multi-agent settings (Yang et al., 2021; Pan et al., 2022). For discrete action environments, we used QMIX (Rashid et al., 2018) and independent DQN and for continuous action environments, we used independent TD3 (Fujimoto et al., 2018) and MATD3 (Lowe et al., 2017; Ackermann et al., 2019). Additional details about how each dataset was generated are included in Appendix C.

Diverse Data Distributions. It is well known from the single-agent offline RL literature that the quality of experience in offline datasets can play a large role in the final performance of offline RL algorithms (Fu et al., 2020). In OG-MARL, we include a range of dataset distributions including `Good`, `Medium`, `Poor` and `Replay` datasets in order to benchmark offline MARL algorithms on a range of different dataset qualities. The dataset types are characterised by the quality of the joint policy that generated the trajectories in the dataset, which is the same approach taken in previous works (Meng et al., 2021; Yang et al., 2021; Pan et al., 2022). To ensure that all of our datasets have sufficient coverage of the state and action spaces, while also containing minimal repetition i.e. not being too narrowly focused around a single strategy, we used 3 independently trained joint policies to generate each dataset, and additionally added a small amount of exploration noise to the policies. The boundaries for the different categories were assigned independently for each environment and were related to the maximum attainable return in the environment. Additional details about how the different datasets were curated can be found in Appendix C.

Table 1: Results on the *Pursuit* and *Co-op Pong* datasets. The mean episode return with one standard deviation across all seeds is given. In each row the best mean episode return is in bold.

Scenario	Dataset	BC	QMIX	QMIX+BCQ	QMIX+CQL	MAICQ
Co-op Pong	Good	31.2±3.5	0.6±3.5	1.9±1.1	**90.0**±**4.7**	75.4±3.9
	Medium	21.6±4.8	10.6±17.6	20.3±12.2	64.9±15.0	**84.6**±**0.9**
	Poor	1.0±0.9	14.4±16.0	30.2±20.7	52.7±8.5	**74.8**±**7.8**
Pursuit	Good	78.3±1.8	6.7±19.0	66.9±14.0	54.4±6.3	**92.7**±**3.7**
	Medium	15.0±1.6	-24.4±20.2	16.6±10.7	20.6±10.3	**35.3**±**3.0**
	Poor	-18.5±1.6	-43.7±5.6	**-0.7**±**4.0**	-19.6±3.3	-4.1±0.7

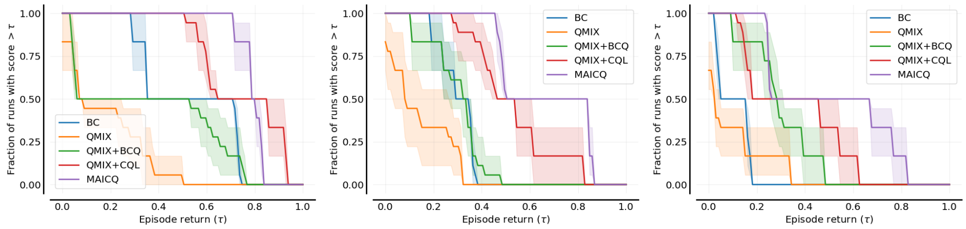
Statistical characterisation of datasets. It is common in both the single-agent and multi-agent offline RL literature for researchers to curate offline datasets by unrolling episodes using an RL policy that was trained to a desired *mean* episode return. However, authors seldom report the distribution of episode returns induced by the policy. Reporting only the mean episode return of the behaviour policy can be misleading (Agarwal et al., 2021). To address this, we provide violin plots to visualise the distribution of expected episode returns. A violin plot is a powerful tool for visualising numerical distributions as they visualise the density of the distribution as well as several summary statistics such as the minimum, maximum and interquartile range of the data. These properties make the violin plot very useful for understanding the distribution of episode returns in the offline datasets, assisting with interpreting offline MARL results. Figure 3 provides a sample of the violin plots for different scenarios (the remainder of the plots can be found in the appendix). In each figure, the difference in shape and position of the three violins (blue, orange and green) illustrates the difference in the datasets with respect to the expected episode return. Additionally, we provide a table with the mean and standard deviation of the episode returns for each of the datasets in Table C.1, similar to Meng et al. (2021).

7 Baselines

In this section, we present the initial baselines that we provide with OG-MARL. This serves two purposes: *i)* to validate the quality of our datasets and *ii)* to enable the community to use these initial results for development and performance comparisons in future work. In the main text, we present results on two PettingZoo environments (*Pursuit* and *Co-op Pong*), since these environments and their corresponding datasets are a novel benchmark for offline MARL. Furthermore, it is the first set of environments with pixel-based observations to be used to evaluate offline MARL algorithms. We include all additional baseline results in Appendix D (Table D.4 and Table D.5).

Baseline Algorithms. State-of-the-art algorithms were implemented from seminal offline MARL work. For discrete action environments we implemented *Behaviour Cloning* (BC), QMIX (Rashid et al., 2018), QMIX with *Batch Constrained Q-Learning* (Fujimoto et al., 2019) (QMIX+BCQ), QMIX with *Conservative Q-Learning* (Kumar et al., 2020) (QMIX+CQL) and MAICQ (Yang et al., 2021). For continuous action environments, Behaviour Cloning (BC), Independent TD3 (ITD3), ITD3 with *Behaviour Cloning* regularisation (Fujimoto and Gu, 2021) (ITD3+BC), ITD3 with *Conservative Q-Learning* (ITD3+CQL) and OMAR (Pan et al., 2022) were implemented. Appendix D provides additional implementation details on the baseline algorithms.

Experimental Setup. On *Pursuit* and *Co-op Pong*, all of the algorithms were trained offline for 50000 training steps with a fixed batch size of 32. At the end of training, we evaluated the performance of the algorithms by unrolling the final joint policy in the environment for 100 episodes and recording the mean episode return over the episodes. We repeated this procedure for 10 independent seeds as per the recommendation by Gorsane et al. (2022). We kept the online evaluation budget (Kurenkov and Kolesnikov, 2022) fixed for all algorithms by only tuning hyper-parameters on *Co-op Pong* and keeping them fixed for *Pursuit*. Controlling for the online evaluation budget is important when comparing offline algorithms because online evaluation may be expensive, slow or dangerous in



(a) Good (b) Medium (c) Poor

Figure 4: Performance profiles (Agarwal et al., 2021) aggregated across all seeds on *Pursuit* and *Co-op Pong*. Shaded regions show pointwise 95% confidence bands based on percentile bootstrap with stratified sampling.

real-world problems, making online hyper-parameter fine-tuning infeasible. See Appendix D for a further discussion on hyper-parameter tuning in OG-MARL.

Results. In Table 1 we provide the unnormalised mean episode returns for each of the discrete action algorithms on the different datasets for *Pursuit* and *Co-op Pong*.

Aggregated Results. In addition to the tabulated results we also provide *aggregated* results as per the recommendation by Gorsane et al. (2022). In Figure 4 we plot the performance profiles (Agarwal et al., 2021) of the discrete action algorithms by aggregating across all seeds and the two environments, *Pursuit* and *Co-op Pong*. To facilitate aggregation across environments, where the possible episode returns can be very different, we adopt the normalisation procedure from Fu et al. (2020). On the `Good` datasets, we found that MAICQ and QMIX+CQL both outperformed behaviour cloning (BC). QMIX+BCQ did not outperform BC and vanilla QMIX performed very poorly. On the `Medium` datasets, MAICQ and QMIX+CQL once again performed the best, significantly outperforming BC. QMIX+BCQ marginally outperformed BC and vanilla QMIX failed. Finally, on the `Poor` datasets, MAICQ, QMIX+CQL and QMIX+BCQ all outperformed BC but MAICQ was the best by some margin. These results on PettingZoo environments, with pixel observations, further substantiate that MAICQ is the current state-of-the-art offline MARL algorithm in discrete action settings.

8 Discussion

Limitations and future work. The primary limitation of this work is that it focuses on the cooperative setting. Additionally, the datasets used in OG-MARL were exclusively generated by online MARL policies. Future work could explore the inclusion of datasets from alternate sources, such as hand-designed or human controllers, which may exhibit distinct properties (Fu et al., 2020). Moreover, an exciting research direction considers the offline RL problem as a sequence modeling task (Chen et al., 2021; Meng et al., 2021), and we aim to incorporate such models as additional baselines in OG-MARL in future iterations.

Potential Negative Societal Impacts. While the potential positive impacts of efficient decentralized controllers powered by offline MARL are promising, it is essential to acknowledge and address the potential negative societal impacts (Whittlestone et al., 2021). Deploying a model trained using offline MARL in real-world applications requires careful consideration of safety measures (Gu et al., 2022; Xu et al., 2022). Practitioners should exercise caution to ensure the safe and responsible implementation of such models.

Conclusion. In this work, we highlighted the importance of offline MARL as a research direction for applying RL to real-world problems. We specifically focused on the lack of a standard set of benchmark datasets, which is a significant obstacle to progress. To address this issue, we presented a set of relevant and diverse datasets for offline MARL. We profiled our datasets by visualising the distribution of episode returns in violin plots and tabulated mean and standard deviations. We validated our datasets by providing a set of initial baseline results with state-of-the-art offline MARL algorithms. Finally, we open-sourced all of our software tooling for generating new datasets and provided a website with our code, as well as for hosting and sharing the datasets. It is our hope that the research community will adopt and contribute towards OG-MARL as a framework for offline MARL research and that it helps to drive progress in this nascent field.

References

J. Ackermann, V. Gabler, T. Osa, and M. Sugiyama. Reducing overestimation bias in multi-agent domains using double centralized critics. *ArXiv Preprint*, 2019. 7

R. Agarwal, D. Schuurmans, and M. Norouzi. An optimistic perspective on offline reinforcement learning. *ArXiv Preprint*, 2019. 23

R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 2021. 8, 9, 26, 27

P. Barde, J. Foerster, D. Nowrouzezahrai, and A. Zhang. A model-based solution to the offline multi-agent reinforcement learning coordination problem. *ArXiv Preprint*, 2023. 3

C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. de Oliveira Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang. Dota 2 with large scale deep reinforcement learning. *ArXiv Preprint*, 2019. 1

D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 2002. 3

G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *ArXiv Preprint*, 2016. 6

L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. Decision transformer: reinforcement learning via sequence modeling. *Advances in Neural Information Processing Systems*, 2021. 3, 9

K. Cobbe, C. Hesse, J. Hilton, and J. Schulman. Leveraging procedural generation to benchmark reinforcement learning. *International Conference on Machine Learning*, 2020. 5

G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal, and T. Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Springer Machine Learning*, 2021. 2

B. Ellis, S. Moalla, M. Samvelyan, M. Sun, A. Mahajan, J. N. Foerster, and S. Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *ArXiv Preprint*, 2022. 5, 6

C. Formanek, C. R. Tilbury, J. P. Shock, K. ab Tessera, and A. Pretorius. Reduce, reuse, recycle: Selective reincarnation in multi-agent reinforcement learning. *Workshop on Reincarnating Reinforcement Learning at ICLR*, 2023. 3, 17

J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4rl: Datasets for deep data-driven reinforcement learning. *ArXiv Preprint*, 2020. 2, 7, 9

S. Fujimoto and S. S. Gu. A minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 2021. 8, 23, 24

S. Fujimoto, H. Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. *International Conference on Machine Learning*, 2018. 7

S. Fujimoto, D. Meger, and D. Precup. Off-policy deep reinforcement learning without exploration. *International Conference on Machine Learning*, 2019. 3, 8, 23, 24

T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. I. au2, and K. Crawford. Datasheets for datasets. *ArXiv Preprint*, 2021. 16

K. Ghasemipour, S. S. Gu, and O. Nachum. Why so pessimistic? estimating uncertainties for offline rl through ensembles, and why their independence matters. *Advances in Neural Information Processing Systems*, 2022. 3

R. Gorsane, O. Mahjoub, R. J. de Kock, R. Dubb, S. Singh, and A. Pretorius. Towards a standardised performance evaluation protocol for cooperative MARL. *Advances in Neural Information Processing Systems*, 2022. 6, 8, 9

S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, Y. Yang, and A. Knoll. A review of safe reinforcement learning: Methods, theory and applications. *ArXiv Preprint*, 2022. 9

C. Gulcehre, Z. Wang, A. Novikov, T. Paine, S. Gómez, K. Zolna, R. Agarwal, J. S. Merel, D. J. Mankowitz, C. Paduraru, et al. Rl unplugged: A suite of benchmarks for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 2020. 2, 7

J. K. Gupta, M. Egorov, and M. Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. *International Conference on Autonomous Agents and Multiagent Systems*, 2017. 6

J. Hu, S. Jiang, S. A. Harding, H. Wu, and S.-w. Liao. Rethinking the implementation tricks and monotonicity constraint in cooperative multi-agent reinforcement learning, 2021. 23

J. Jiang and Z. Lu. Offline decentralized multi-agent reinforcement learning. *ArXiv Preprint*, 2021. 3, 4, 5, 6

V. Khattar and M. Jin. Winning the citylearn challenge: Adaptive optimization with evolutionary search under trajectory-based guidance. *ArXiv Preprint*, 2022. 2

I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning. *Deep RL Workshop at NeurIPS*, 2021. 3

L. Kraemer and B. Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Elsevier Neurocomputing*, 2016. 4

A. Kumar, J. Fu, G. Tucker, and S. Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Neural Information Processing Systems*, 2019. 23

A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 2020. 3, 8, 23, 24

V. Kurenkov and S. Kolesnikov. Showing your offline reinforcement learning work: Online evaluation budget matters. *International Conference on Machine Learning*, 2022. 8, 24

S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *ArXiv Preprint*, 2020. 1, 4

R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017. 3, 7

C. Lu, P. J. Ball, T. G. Rudner, J. Parker-Holder, M. A. Osborne, and Y. W. Teh. Challenges and opportunities in offline reinforcement learning from visual observations. *Decision Awareness in Reinforcement Learning Workshop at ICML*, 2022. 2, 5

X. Lyu, Y. Xiao, B. Daley, and C. Amato. Contrasting centralized and decentralized critics in multi-agent reinforcement learning. *International Conference on Autonomous Agents and Multi-Agent Systems*, 2021. 4

X. Lyu, A. Baisero, Y. Xiao, and C. Amato. A deeper understanding of state-based critics in multi-agent reinforcement learning. *ArXiv Preprint.*, 2022. 4

L. Meng, M. Wen, Y. Yang, C. Le, X. Li, W. Zhang, Y. Wen, H. Zhang, J. Wang, and B. Xu. Offline pre-trained multi-agent decision transformer: One big sequence model conquers all starcraftii tasks. *ArXiv Preprint*, 2021. 3, 4, 6, 7, 8, 9

S. Mohanty, E. Nygren, F. Laurent, M. Schneider, C. Scheller, N. Bhattacharya, J. Watson, A. Egli, C. Eichenberger, C. Baumberger, G. Vienken, I. Sturm, G. Sartoretti, and G. Spigler. Flatland-rl : Multi-agent reinforcement learning on trains. *ArXiv Preprint*, 2020. 2, 6

M. Nakamoto, Y. Zhai, A. Singh, Y. Ma, C. Finn, A. Kumar, and S. Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *Workshop on Reincarnating Reinforcement Learning at ICLR*, 2023. 3

L. Pan, L. Huang, T. Ma, and H. Xu. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification. *International Conference on Machine Learning*, 2022. 3, 4, 5, 6, 7, 8, 24

B. Peng, T. Rashid, C. Schroeder de Witt, P.-A. Kamienny, P. Torr, W. Böhmer, and S. Whiteson. Facmac: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems*, 2021. 3, 6

R. F. Prudencio, M. R. O. A. Maximo, and E. L. Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 2

T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. *International Conference on Machine Learning*, 2018. 7, 8, 23

D. Rolnick, P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, A. S. Ross, N. Milojevic-Dupont, N. Jaques, A. Waldman-Brown, A. S. Luccioni, T. Maharaj, E. D. Sherwin, S. K. Mukkavilli, K. P. Kording, C. P. Gomes, A. Y. Ng, D. Hassabis, J. C. Platt, F. Creutzig, J. Chayes, and Y. Bengio. Tackling climate change with machine learning. *ACM Computing Surveys*, 2022. 7

M. Samvelyan, T. Rashid, C. Schroeder de Witt, G. Farquhar, N. Nardelli, T. G. Rudner, C.-M. Hung, P. H. Torr, J. Foerster, and S. Whiteson. The starcraft multi-agent challenge. *International Conference on Autonomous Agents and MultiAgent Systems*, 2019. 3, 16

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2018. 23

Q. Sykora, M. Ren, and R. Urtasun. Multi-agent routing value iteration network. *International Conference on Machine Learning*, 2020. 2

J. Terry, B. Black, N. Grammel, M. Jayakumar, A. Hari, R. Sullivan, L. S. Santos, C. Dieffendahl, C. Horsch, R. Perez-Vicente, et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 2021. 6

E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012. 6

J. R. Vazquez-Canteli, S. Dey, G. Henze, and Z. Nagy. Citylearn: Standardizing research in multi-agent reinforcement learning for demand response and urban energy management. *ArXiv Preprint*, 2020. 5, 7

O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 2019. 1

J. Wang, W. Xu, Y. Gu, W. Song, and T. C. Green. Multi-agent reinforcement learning for active voltage control on power distribution networks. *Advances in Neural Information Processing Systems*, 2021. 5, 7

J. Whittlestone, K. Arulkumaran, and M. Crosby. The societal implications of deep reinforcement learning. *Journal of Artificial Intelligence Research*, 2021. 9

H. Xu, X. Zhan, and X. Zhu. Constraints penalized q-learning for safe offline reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 9

Y. Yang, X. Ma, L. Chenghao, Z. Zheng, Q. Zhang, G. Huang, J. Yang, and Q. Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 2021. 3, 5, 6, 7, 8, 24

L. Yu, S. Qin, M. Zhang, C. Shen, T. Jiang, and X. Guan. A review of deep reinforcement learning for smart building energy management. *IEEE Internet of Things Journal*, 2021. 7

Y. Yu. Towards sample efficient reinforcement learning. *International Joint Conference on Artificial Intelligence*, 2018. 1

H. Zhang, S. Feng, C. Liu, Y. Ding, Y. Zhu, Z. Zhou, W. Zhang, Y. Yu, H. Jin, and Z. Li. CityFlow: A multi-agent reinforcement learning environment for large scale city traffic scenario. *ACM International World Wide Web Conference*, 2019. 2

K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar. Finite-sample analysis for decentralized batch multiagent reinforcement learning with networked agents. *IEEE Transactions on Automatic Control*, 2021. 3

M. Zhou, Z. Liu, P. Sui, Y. Li, and Y. Y. Chung. Learning implicit credit assignment for cooperative multi-agent reinforcement learning. *Arxiv Preprint*, 2020. 5

Z. Zhu, M. Liu, L. Mao, B. Kang, M. Xu, Y. Yu, S. Ermon, and W. Zhang. Madiff: Offline multi-agent learning with diffusion models. *Arxiv Preprint*, 2023. 3, 17

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See section 8.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See section 8.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Our datasets and code are open-sourced.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All of the training details are in section 7 and the hyperparameter details are in Appendix D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figure 4.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix D.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [Yes] See our datasheet in Appendix A and licence in Appendix E.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See our datasheet in Appendix A.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See our datasheet in Appendix A.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See our datasheet in Appendix A.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]