MULTI-TASK SEQUENCE MODELS GENERALISE IN OFFLINE MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent sequence model architectures have demonstrated great promise in offline multi-agent reinforcement learning (MARL). However, even for this expressive model class, generalising to tasks *unseen* in the training data remains a core challenge. A sensible response to this challenge is to simply scale the amount of offline data available for training. Yet, in this work, we find that task diversity has a stronger influence on generalisation than sheer dataset size. To obtain our findings, we study offline MARL sequence models trained on single-task datasets, clearly demonstrating their limited ability to zero-shot transfer to held-out test tasks. Leveraging this insight, we train and test multi-task versions of offline sequence modeling architectures. We identify three key design choices for successful offline multi-task training: (i) task-balanced mini-batches, (ii) treating value estimation as classification and (iii) agent masking to handle variable team sizes. Using multi-task datasets from three challenging cooperative environments (`Connector`, `RWARE`, and `LBF`), we investigate generalisation to unseen tasks and the scaling behaviour of our multi-task offline algorithms. **We show that our multi-task sequence models generalise better across all environments compared to single-task models, and achieve a mean improvement of 219% on held-out test tasks.** Moreover, our offline MARL sequence models consistently outperform behaviour cloning (a surprisingly strong baseline). Our results clearly show that scaling task diversity by increasing the number of tasks used during training leads to improved generalisation gains over simply scaling the dataset size at a fixed level of task diversity.

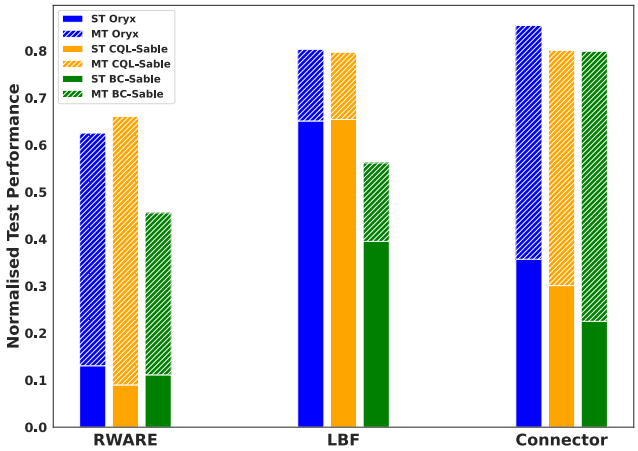


Figure 1: *Test task performance difference between single-task and multi-task sequence models.* Three multi-agent sequence models—CQL-Sable, BC-Sable and Oryx (Formanek et al., 2025)—were trained using either a single task (ST) or a set of multiple training tasks (MT). Average zero-shot performance was measured across a held-out set of test tasks. The upper bar represents the performance gap between ST and MT sequence models on unseen test tasks. **Averaged across all three algorithms, we observe a test performance increase of over 442% on `RWARE`, 29% on `LBF`, and 187% on `Connector`.**

1 INTRODUCTION

Building agents that generalise to tasks beyond those present in their training data is a central challenge in reinforcement learning (RL), and a prerequisite for deploying agents in the real world (Kirk et al., 2023). In many domains, collecting fresh data online by interacting with a live system is costly or risky, so practitioners turn to offline RL from logged trajectories (Levine et al., 2020). While single-agent work has studied the train–test generalisation gap (Mediratta et al., 2024), the multi-agent case remains under-explored. Despite recent progress in offline MARL (Yang et al., 2021b; Shao et al., 2023; Meng et al., 2023; Li et al., 2025; Formanek et al., 2025), prior work have largely been restricted to training and evaluating on the same task, without examining generalisation to unseen tasks.

In this work, we study the generalisation of single-task models, and then introduce a challenging multi-task benchmark for offline MARL, which builds on widely adopted MARL environments `LBF`, `RWARE` (Papoudakis et al., 2021), and `Connector` (Bonnet et al., 2024). Using this benchmark, we evaluate three state-of-the-art offline multi-agent sequence models, namely Oryx (Formanek et al., 2025), as well as two offline versions of Sable (Mahjoub et al., 2025) (CQL-Sable and BC-Sable). Across all three environments, we show that these models exhibit poor generalisation when trained only on a dataset from a single task. However, when trained *simultaneously* on a dataset consisting of a diverse set of multiple tasks, their ability to zero-shot transfer to unseen tasks significantly improves. Furthermore, we verify that similar results cannot be obtained by simply increasing the size of the dataset for a fixed number of tasks, but rather that the key driver is increasing dataset diversity by adding more tasks, which consistently leads to improved test performance. Finally, we find that for a fixed data budget, increasing the model's capacity has a positive impact on generalisation for challenging tasks.

We identify three key design choices for multi-agent sequence models to be successfully trained across multiple tasks simultaneously: (i) task balanced batching, which makes the model unbiased over a mixture of tasks, (ii) value learning via classification (Farebrother et al., 2024) which improves the models ability to handle tasks with varying reward scales (Kumar et al., 2022a), and (iii) masking and shuffling active agents in the sequence, which allows the models to dynamically handle varying numbers of agents across tasks.

Our findings show that offline MARL sequence models trained on diverse multi-task datasets show promising signs of generalisation to unseen tasks, as compared to single-task alternatives. In contrast to the findings of Mediratta et al. (2024), we observe that our offline MARL methods do outperform behaviour cloning, a consistent and surprisingly strong baseline to beat. Finally, our work discovers the first promising signs of performance scaling (Hilton et al., 2023) with increases in model capacity for offline MARL on difficult unseen tasks.

In summary, our main contributions are as follows:

- We develop a challenging multi-task offline MARL benchmark, which includes 30 large training sets and 22 test sets across `LBF`, `Connector`, and `RWARE`.
- We present two novel MARL sequence models (BC-Sable and CQL-Sable) and three design choices that enable these models — and Oryx (Formanek et al., 2025) — to be trained on multi-task datasets.
- We show that the zero-shot generalisation capacity of all three multi-agent sequence models scales significantly (219% on average) as the number of tasks in the training data increases.
- We study the effect of dataset and model size on generalisation, clearly establishing that sheer dataset size in not the main driver of test performance, and that for difficult tasks, model scaling positively affects generalisation.
- **All of our (anonymized) code is available for download[1]. We will make all of our code and datasets publicly available upon publication.**

[1] `https://sites.google.com/view/multi-task-marl`

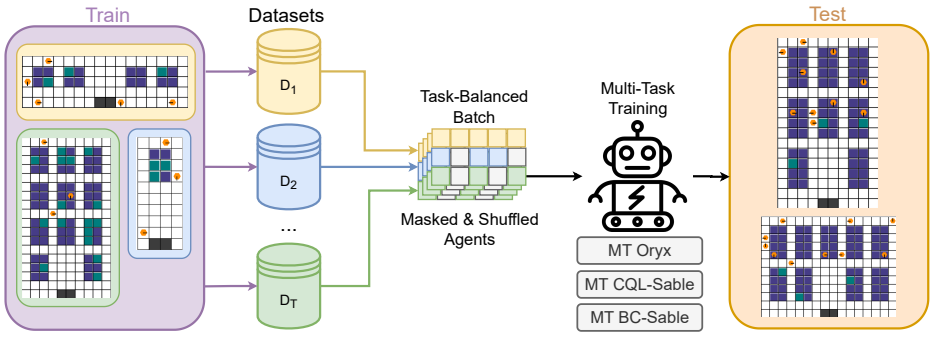


Figure 2: *Our offline multi-task multi-agent training and testing setup.* In this setup, there is a set of training tasks, each with a static dataset of pre-collected trajectories that together form a diverse multi-task dataset. This dataset is then used for training, without any additional online interactions with either the training tasks or the testing tasks. At evaluation time, the trained model is evaluated on each of the held-out test tasks, and the average test performance is calculated.

2 Multi-Task Sequence Modelling for Offline MARL

2.1 Preliminaries

Problem formulation. We formalise a cooperative MARL *task* as a Dec-POMDP (Kaelbling et al., 1998), defined by the tuple $\mathcal{M}_\dagger = \langle \mathcal{N}, \mathcal{S}, \boldsymbol{\mathcal{A}}, P, R, \{\Omega^i\}_{i\in\mathcal{N}}, \{E_i\}_{i\in\mathcal{N}}, \gamma \rangle$, where $\dagger$ denotes the particular task selected from an environment. For example, in a simulated robotic warehouse environment, a task corresponds to a specific warehouse layout and the number of robotic workers collecting and depositing requested shelf items. At each timestep t within a task, the environment is in state $s_t \in \mathcal{S}$. Each agent $i \in \mathcal{N}$ selects an action $a_t^i \in \mathcal{A}^i$ based on its local action-observation history $\tau_t^i = (o_0^i, a_0^i, \ldots, o_t^i)$. The agents' actions form a joint action $\boldsymbol{a}_t \in \boldsymbol{\mathcal{A}} = \prod_{i\in\mathcal{N}} \mathcal{A}^i$, which, when executed, yields a shared reward $r_t = R(s_t, \boldsymbol{a}_t)$, transitions the environment to $s_{t+1} \sim P(\cdot|s_t, \boldsymbol{a}_t)$, and provides each agent i with a new observation $o_{t+1}^i \sim E_i(\cdot|s_{t+1}, \boldsymbol{a}_t)$. The agent then updates its history as $\tau_{t+1}^i = (\tau_t^i, a_t^i, o_{t+1}^i)$. The task-specific objective is to learn a joint policy $\pi(\boldsymbol{a}|\boldsymbol{\tau})$ that maximises the expected discounted return over a horizon of timesteps H: $J_\dagger(\boldsymbol{\pi}) = \mathbb{E}_\pi \left[\sum_{t=0}^{H} \gamma^t r_t \right]$.

To create our train-test evaluation setup, we consider offline datasets $\mathcal{D}_{\text{train}} = \{\mathcal{D}_\dagger : \dagger \in \mathcal{T}_{\text{train}}\}$ collected from a set of training tasks $\mathcal{T}_{\text{train}}$. Our objective is to learn a single joint policy $\boldsymbol{\pi}_{\text{train}}$, using only the fixed multi-task training data (i.e. without any additional online interaction), to maximise the expected zero-shot performance on a set of *unseen* test tasks $\mathcal{T}_{\text{test}}$, given as

$$J(\boldsymbol{\pi}) = \mathbb{E}_{\dagger\sim\mathcal{T}_{\text{test}}}[J_\dagger(\boldsymbol{\pi})|\boldsymbol{\pi} = \boldsymbol{\pi}_{\text{train}}] .$$

By optimising the above objective, we are minimising the generalisation gap between training and test tasks. A simplified visual representation of the problem setting is depicted in Figure 2.

Multi-Agent Sequence Models. Centralised control, where a single policy outputs the joint action, is theoretically optimal but scales poorly due to an exponential growth of the action space (de Kock et al., 2025). However, autoregressive factorisation is an efficient way to parametrise the joint policy, by expressing the joint distribution over n agents as a product of conditional distributions:

$$\pi(\boldsymbol{a}|\boldsymbol{\tau}) = \prod_{k=1}^{n} \pi^{i_k}\left(a^{i_k} \mid \boldsymbol{\tau}, a^{i_1}, \ldots, a^{i_{k-1}}\right) .$$

Here i_k denotes an agent index from an ordered set $\{i_1, \ldots, i_n\} \in S_n$, where S_n is the set of permutations of $\{1, ..., n\}$. This factorisation decomposes joint decision-making into a sequence of conditional actions, enabling scalable coordination, efficient parallel training and, in certain cases, providing desirable convergence properties (Zhong et al., 2024b). Sequence models provide a natural parameterisation of such policies, closely mirroring the autoregressive next token prediction process in text and image generation, and have been demonstrated to work well on a large range of MARL settings (Wen et al., 2022; Mahjoub et al., 2025; Daniel et al., 2024; Formanek et al., 2025).

2.2 Multi-Task Sequence Models for Offline MARL

Building on existing multi-agent sequence models for offline MARL (Formanek et al., 2025), we propose a few simple yet essential modifications that enable training on multiple tasks with varying numbers of agents simultaneously, while allowing seamless zero-shot transfer. By design, our multi-task sequence models do not receive explicit task IDs or have task specific output heads, since this would limit their zero-shot transferability to new tasks. Instead, our models have to infer task information from observations, agent counts, and environment dynamics.

Dynamic agent padding, shuffling and masking. In order to dynamically handle variable numbers of agents across tasks, we zero-pad the inputs for absent agents and mask their contributions in the loss. Moreover, we randomise the ordering of both active and inactive agents at each training update, which encourages the model to share representations and transfer knowledge across agents.

Multi-task training loss. Given a set of training tasks $\mathcal{T}_{\text{train}} = \{\dagger_1, \ldots, \dagger_M\}$, with offline buffers $\{\mathcal{D}_\dagger\}_{\tau \in \mathcal{T}_{\text{train}}}$, we train a multi-task sequence model by minimizing the average per-task loss

$$\min_\theta \quad \frac{1}{M} \sum_{\dagger \in \mathcal{T}_{\text{train}}} \Big[\mathcal{L}(\theta; \mathcal{D}_\dagger) \Big]. \tag{1}$$

The loss $\mathcal{L}$ changes depending on the algorithm used, which in our case includes autoregressive versions of behaviour cloning (BC) (Pomerleau, 1988; Bain & Sammut, 1995), Conservative Q-learning (CQL) (Kumar et al., 2020) and Implicit Constraint Q-learning (ICQ) (Yang et al., 2021b; Formanek et al., 2025).

Task-balanced batching. For each training update, we build a single unified mini-batch by evenly sampling across different tasks. Given a batch size B, we compute $q = \lfloor B \,/\, |\mathcal{T}_{\text{train}}| \rfloor$ and $r = B - q|\mathcal{T}_{\text{train}}|$. Each task $\dagger \in \mathcal{T}_{\text{train}}$, contributes q samples; the remaining r samples are assigned by round-robin across tasks up to the value r. This yields stochastic gradients that are unbiased over a uniform mixture of tasks (each task equally weighted), rather than a size-weighted mixture. The resulting task-balanced batching also mitigates "head-task" dominance seen with dataset-proportional sampling, a known issue in domain generalisation from long-tailed datasets (Cui et al., 2019).

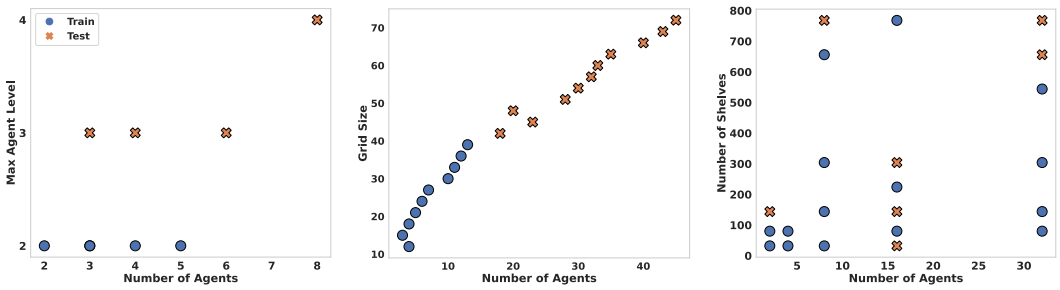
Value function learning via classification. To mitigate gradient interference from varying reward scales across tasks, we replace scalar TD regression with a classification objective. Specifically, we use HL-Gauss (Imani & White, 2018; Farebrother et al., 2024), which projects each scalar TD target onto a discrete support by smoothing with a Gaussian distribution, and trains the value function with categorical cross-entropy over the resulting histogram. This choice, consistent with prior multi-task training architectures (Kumar et al., 2022a), improves stability and reduces loss-scale sensitivity compared to mean squared error.

3 Experiments

3.1 Experimental design

Tasks. We considered three challenging MARL environments, `LBF`, `RWARE` (Papoudakis et al., 2021) and `Connector` (Bonnet et al., 2024). These are all widely used MARL benchmarks, with `RWARE` also proposed as a suitable multi-task benchmark in previous work (Schäfer, 2022) and `Connector` being of particular interest due to its agent scaling properties Formanek et al. (2025). For each environment, we selected several different level configurations to serve as distinct tasks. These tasks were then partitioned into train and test sets (see Appendix A), taking care to ensure that the test tasks were different in meaningful ways to the training tasks, as shown in Figure Figure 3.

Datasets. For each task, we construct an offline dataset $\mathcal{D}_\dagger$ by recording a set of rollouts at fixed intervals from an online training run of Sable (Mahjoub et al., 2025), a state-of-the-art MARL sequence model. This yields a mixed dataset with the same number of rollouts per task but not necessarily the same number of transitions, since episode lengths differ across tasks, hence the necessity for task-balanced batching. Observations and actions are standardised per environment. For sequence modeling, we sample fixed-length trajectory chunks (context length reported with other hyperparameters in Appendix C). Rewards are left unclipped during training and for comparability across tasks, we report normalised returns, where each task's episode return is normalised by the final episode return achieved by the online system on that task.



(a) LBF (b) Connector (c) RWARE

Figure 3: *Distributional shift between train and test tasks.* Each point represents a task with the number of agents in each task plotted against a specific task property: in LBF, the maximum agent level, in Connector the grid size, and in RWARE the number of shelves. While these dimensions are important to distinguish tasks, it should be noted there are additional parameters which change across tasks, not shown here (e.g. the layout of shelves in RWARE tasks).

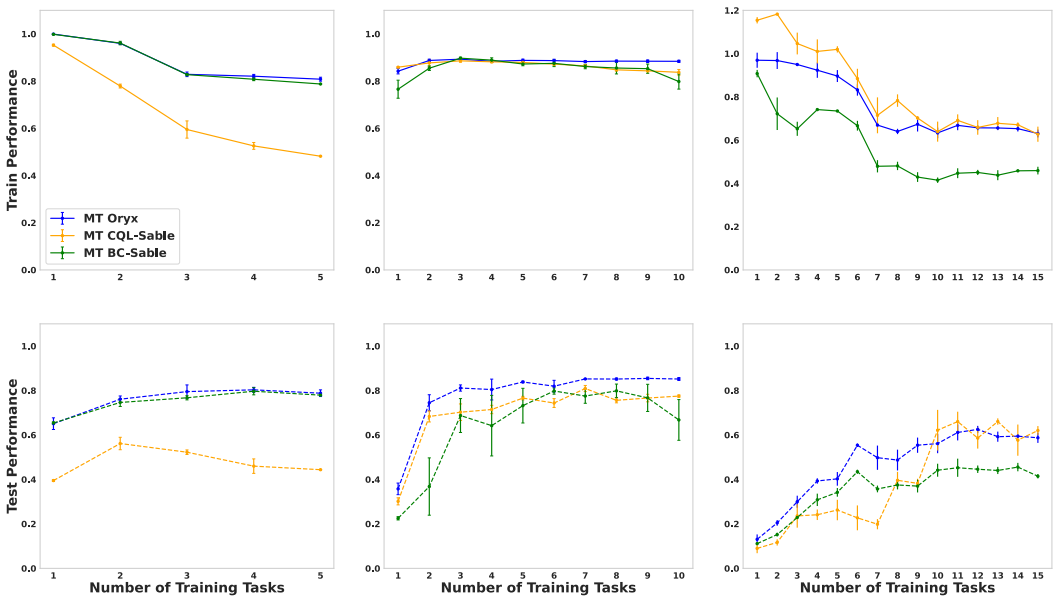
Algorithms. The main algorithm we consider is an adapted version of Oryx (Formanek et al., 2025), which we modify for multi-task training. As described in section 2, this includes (i) dynamic padding, masking and agent shuffling, (ii) task-balanced batching, and (iii) value learning using HL-Gauss (Farebrother et al., 2024). We refer to this version of Oryx as MT Oryx. In addition, we develop two new strong baselines. The first is MT BC-Sable, which is an offline variant of Sable that uses simple behaviour cloning to train an autoregressive policy, along with dynamic padding and masking of agents, and task-balanced batching. The second is MT CQL-Sable, another offline variant of Sable that uses an autoregressive version of the CQL loss (Kumar et al., 2020), along with all three MT enhancements as in MT Oryx. The Sable network backbone is consistent across all three algorithms. Therefore, the only significant difference between MT Oryx and the other two baselines is the loss function $\mathcal{L}$ used. We chose CQL because of its proven generalisation and scaling capabilities in the single-agent setting (Kumar et al., 2022a; Chebotar et al., 2023), and BC for its competitive generalisation performance as demonstrated in prior work (Mediratta et al., 2024). Hyperparameter details for all three algorithms are listed in Appendix C.

Evaluation protocol. In our experiments, we are interested in the expected zero-shot performance of the trained model on the held-out test tasks. To measure this, we compute the absolute episode return (Gorsane et al., 2022), by running the best checkpoint achieved during training for 320 independent evaluation episodes and averaging the episode returns for each task in the test set. To compare across tasks and environments with potentially different reward scales, we normalise the absolute episode return by dividing it by the maximum expected episode return achieved on the respective task by the online Sable algorithm. Each run configuration was repeated across three random seeds, with the mean and standard deviation being reported in each case.

3.2 Multi-Task Training Improves Generalisation

Experiment. We vary the number of tasks in the training set, while keeping the test set fixed. We then train our multi-task sequence models on different subsets of the training datasets and measure the performance on the test tasks. For LBF, we consider a total of 5 training tasks, for Connector 10 and for RWARE 15, incrementing training by a single task from 1 to the maximum for each environment. We plot the performance across training task counts when evaluated on the same training tasks as well as the held-out test tasks in Figure 4.

Discussion. We observe that performance on the training tasks remains high across all environments, even as the number of tasks increases. This indicates that the model can successfully learn across multiple tasks simultaneously. However, in RWARE we note a progressive decline in training performance as the number of training tasks grows. We attribute this to the higher complexity of RWARE tasks and the need to scale model capacity with task diversity to maintain performance. Interestingly, even as train task performance degrades, test task performance improves nearly monotonically as the number of training tasks increases, highlighting the importance of diverse multi-task data for generalisation. On LBF, we observe that MT CQL-Sable's performance decreases. We



(a) LBF (b) Connector (c) RWARE

Figure 4: *The effect of increasing task diversity on performance.* **Top:** training tasks. **Bottom:** held-out test tasks. When we train our sequence models using only a single task, we observe strong performance on that single training task (see first point on each plot in the top row). However, the performance on the held-out test tasks is much lower, i.e. the generalisation gap is large. **As we increase the number of tasks in the training set, we observe a steady increase in the test task performance across all three environments.**

hypothesise that this is due to the high proportion of expert trajectories in the LBF dataset, as the data collection policy quickly converges to the optimal behaviour. Prior work has shown that CQL is particularly sensitive to overly narrow or high-quality datasets, and benefits from mixed quality datasets (Schweighofer et al., 2022). To further examine this, we include an ablation on trajectories' quality in Appendix B.

Across all algorithms and environments, performance tends to plateau after a certain number of training tasks. We attribute this saturation to the limits of the current model capacity, pointing to the necessity of scaling up the model size to obtain maximum performance on highly diverse multi-task datasets (see subsection 3.4). To summarise the overall effect of multi-task training with a fixed model size, we measure and report the maximum performance gain on test tasks in Figure 1. Averaged across all three algorithms, test performance improves by over 442% on RWARE, 29% on LBF, and 187% on Connector. These results validate the effectiveness of multi-task training as a means of unlocking substantial performance gains on unseen test tasks.

3.3 Multi-Task Offline MARL can Generalise Better than Behaviour Cloning

The findings from Mediratta et al. (2024) paint a bleak outlook for the generalisation capabilities of Offline RL algorithms compared to simple behaviour cloning. To establish if we observe a similar trend, we aggregate the normalised episode returns across all test tasks from LBF, RWARE and Connector, when trained using the full training set, to compare our three algorithms. In Table 1, we show the mean and standard error for each algorithm.

We want to know which offline training objective performed the best in terms of generalisation to the test tasks. We considered three objectives: behaviour cloning, conservative Q-learning, and the autoregressive ICQ loss from Formanek et al. (2025). We find that indeed BC outperforms CQL. However, interestingly, the autoregressive ICQ loss in Oryx significantly outperforms both BC and CQL, a promising result supporting the ability of offline MARL to generalise to unseen tasks.

Table 1: *Comparison of test task performance of all three models.*The mean and standard error of the performance across all test tasks on RWARE, LBF and Connector for each of the multi-task algorithms (largest mean highlighted with bold). In the final column the combined mean across all tasks from the three environments is computed. **In contrast to the findings by Mediratta et al. (2024), we find that on each environment the best performing algorithm is an Offline RL method (MT CQL-Sable or MT Oryx), rather than the BC model. When aggregated across all the test tasks combined, MT Oryx performs the best.**

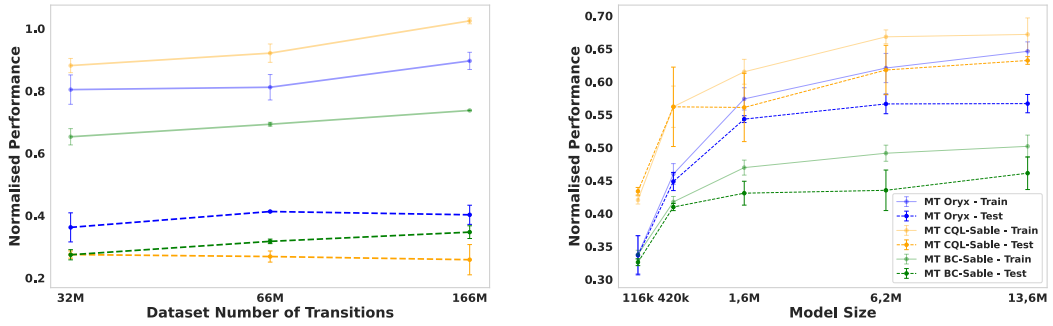
	Algorithm	RWARE	LBF	Connector	Combined
●	MT Oryx	0.587 ± 0.054	$\mathbf{0.803 \pm 0.026}$	$\mathbf{0.852 \pm 0.002}$	$\mathbf{0.759 \pm 0.023}$
●	MT CQL-Sable	$\mathbf{0.620 \pm 0.066}$	0.562 ± 0.029	0.668 ± 0.018	0.633 ± 0.024
●	MT BC-Sable	0.415 ± 0.050	0.797 ± 0.030	0.775 ± 0.004	0.664 ± 0.027

3.4 Can we further improve generalisation by increasing the size of the datasets and models?

A natural question that arises is what is the optimal dataset size and model size for generalisation. Can we improve the generalisation capabilities by simply increasing the size of the dataset for a given set of training tasks? Similarly, can we improve generalisation by increasing the size of the model? To test this we design two experiments.

Experiment (a). To determine whether increasing the size of the datasets (in terms of number of transitions rather than number of tasks helps performance) we conducted a sweep over dataset sizes for several multi-task datasets on RWARE. The results of the sweep are presented in Figure 5a. Similar to the results by Mediratta et al. (2024), we find that there is little evidence that scaling up the number of transitions helps generalisation nearly as much as adding more tasks.

Experiment (b). To study the effect of model size, we train various models with different numbers of parameters, ranging from 116k to 13M, using the RWARE dataset. For simplicity, we mainly vary the embedding dimension of the model's encoder-decoder network from 64 (116k parameters) to 768 (13M parameters). We report the average episode return, normalised by the online performance, on both the training and test tasks in Figure 5b.

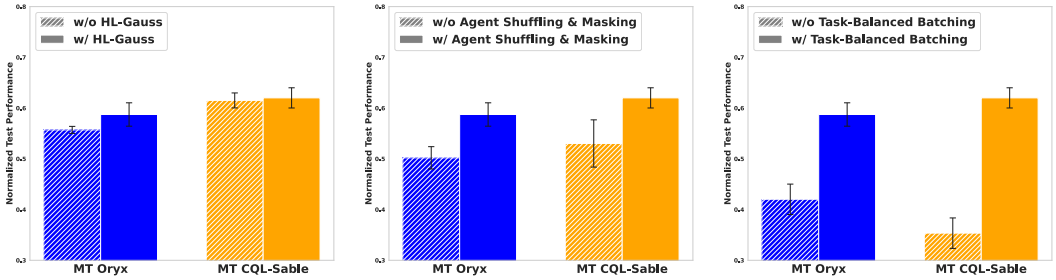


(a) Dataset size scaling.

(b) Model size scaling.

Figure 5: *The impact of scaling up dataset* ***(left)*** *and model size* ***(right).*** When we fix the number of RWARE tasks in the dataset to 5 but grow the number of transitions in the dataset, we observe an increase in train performance, while the test performance plateaus. On the other hand, when we train each of our MT sequence models on the full 15 task RWARE dataset, **we observe a clear scaling trend with respect to the model size in terms of both train and test performance.**

Discussion. The results in Figure 5a indicate that simply increasing the number of transitions in the training dataset improves train task performance but does not lead to better generalisation on held-out test tasks, highlighting the importance of task diversity in multi-task datasets, since from Figure 4c we can conclude that adding additional tasks has a greater benefit.. In contrast, scaling model capacity (Figure 5b)—from an embedding dimension of 64 (116k parameters) to 512 (6.2M parameters)—consistently improved both training and test performance. This finding is particularly



(a) HL-Gauss (b) Agent Masking & Shuffling (c) Task-Balanced Batching

Figure 6: *Ablation studies*. **Left:** Using HL-Gauss improves test performance for MT Oryx by ≈8%, while the effect on MT CQL-Sable is marginal. **Middle:** Disabling agent masking and shuffling reduces test performance by ≈16% on average for both algorithms. **Right:** Removing task-balanced batching has the highest impact with ≈ 37% drop in test performance on average for both MT Oryx and MT CQL-Sable.

encouraging: it suggests that large, diverse multi-task datasets may be the missing ingredient needed to make ever-larger and more general offline MARL models viable. Notably, this result contrasts with the single-task setting reported by (Formanek et al., 2025), where the optimal embedding dimension was just 64, underscoring the unique potential of multi-task data for enabling scale.

3.5 Ablation Studies

HL-Gauss. To test the effect of using HL-Gauss (Farebrother et al., 2024) for multi-task learning, we conduct an ablation on the full set of `RWARE` training tasks where we run MT Oryx and MT CQL-Sable with and without HL-Gauss for value function learning (e.g. standard TD mean-squared-error). We compare the algorithms on multi-task `RWARE` since the task-to-task variance in episode returns is significant and therefore more challenging to accurately learn a multi-task value function. As shown in Figure 6a, using HL-Gauss leads to slightly better performance (≈ 8% improvement) on test tasks for MT Oryx, while the effect on MT CQL-Sable is marginal.

Agent shuffling and masking. To test the impact of *not* masking and shuffling agents we conduct a similar ablation to above on `RWARE`. We observe decrease in performance of ≈ 16% on average for both algorithms on the test tasks, when we do not mask and shuffle agents (see Figure 6b).

Task-balanced batching. Finally, we conducted an ablation on how we sample data from the multi-task dataset. In the first case we use our proposed task-balanced batching method, which includes a fair mix of samples from each task in every batch. In the alternative approach we choose a random task at each update step and sample a full batch from the chose single task. The results in Figure 6c shows a 37% decrease in test performance on average for both MT Oryx and MT CQL-Sable without task-balanced batching.

4 Related Work

Offline MARL. Most prior work in offline MARL uses single-task training and evaluation, while focusing on finding solutions to key challenges particular to offline multi-agent learning. Seminal early papers include Jiang & Lu (2021) and Yang et al. (2021a), who introduced multi-agent methods for constrained Q-value estimation. Since then, numerous additional works have aimed to tackle challenges such as extrapolation error (Shao et al., 2023; Eldeeb et al., 2024), coordination (Barde et al., 2024; Tilbury et al., 2024; Zhou et al., 2025), offline training stability (Pan et al., 2022; Wang et al., 2023; Matsunaga et al., 2023; Wu et al., 2023a; Bui et al., 2025; Liu et al., 2024b; Li et al., 2025), opponent modeling (Jing et al., 2024), offline-to-online transfer (Zhong et al., 2024a; Formanek et al., 2023) and theoretical understanding (Cui & Du, 2022b;a; Zhong et al., 2022; Zhang et al., 2023b; Xiong et al., 2023; Wu et al., 2023a).

Sequence Models for RL. Formulating RL as a sequence modelling problem has gained significant attention. Chen et al. (2021) introduced the Decision Transformer (DT), later extended in various

ways (Zheng et al., 2022; Yamagata et al., 2023; Wu et al., 2023b). Lee et al. (2022) trained a multi-task DT that learned across tasks and could be quickly fine-tuned. Meng et al. (2023) introduced MADT, an extension of the DT to the multi-agent setting. The Multi-Agent Transformer (MAT) (Wen et al., 2022) addressed the online setting with auto-regressive action selection, and Mahjoub et al. (2025) improved on MAT with Sable, which replaces the Transformer with a Retentive Network (Sun et al., 2023) and adds temporal memory, achieving state-of-the-art results. Building on this line, Formanek et al. (2025) proposed Oryx, an offline MARL sequence model derived from an autoregressive version of Implicit Constraint Q-Learning (ICQ) (Yang et al., 2021b) and offline-specific modifications to Sable, also achieving state-of-the-art performance.

Multi-Task RL. Multi-task training has most prominently been investigated in single-agent continuous-control and robotics problems with a focus on representation and transfer learning (Xu et al., 2020; Kalashnikov et al., 2021; Kumar et al., 2022b; Cheng et al., 2022). Although shown to be useful in most cases, Yu et al. (2021) find that naively adding more multi-task data to an offline RL training dataset can sometimes lead to a decrease in performance on downstream tasks, particularly when the distributional shift between tasks is large. In terms of generalisation, Kumar et al. (2022a) and He et al. (2023) highlight the potential for high-capacity models trained on large and diverse multi-task datasets to produce agents that can generalise more broadly when fine-tuned on previously unseen tasks. Most closely related to our work is that of Mediratta et al. (2024), who evaluate the zero-shot generalisation capabilities of several offline single-agent RL methods by training them on a set of training tasks and testing them on a set of holdout tasks. They find that current offline RL methods do not generalise well and are typically outperformed by simple behaviour cloning.

Multi-Task MARL. Multi-task MARL faces both architectural and evaluation challenges when agents must generalise beyond single-task training, motivating formal definitions and benchmarks for task generalisation(Schäfer, 2022). Rosen et al. (2024) give a formal, goal-oriented theory that proves how a learned world value function can enable provably optimal zero-shot task generalisation in goal-based multi-agent settings. MaskMA (Liu et al., 2024a) introduces a mask-based framework that adapts to varying agent- and action-spaces and shows strong zero-shot transfer on unseen SMAC (Samvelyan et al., 2019) maps. Unlike our approach, their work builds on MADT (Meng et al., 2023), while we focus on sequence model architectures related to Oryx (Formanek et al., 2025), which have been shown to outperform MADT. The offline coordination-skill discovery method ODIS (Zhang et al., 2023a) extracts task-invariant coordination primitives from multi-task trajectories and shows that this can be used to deploy coordination policies to unseen SMAC tasks without additional online interaction. Related work, HiSSD (Liu et al., 2025) proposes a hierarchical separation between common cooperative (temporal) skills and task-specific controllers. None of the above studies investigates the effect of task diversity on test performance, instead keeping the number of training tasks fixed.

5 Conclusion

In this work, we studied generalisation in offline MARL and showed that task diversity is a key driver of improved test performance. We introduced a simple yet effective recipe for building multi-task sequence models, which consistently narrows the train–test gap and achieves significant performance gains on unseen test tasks. Our findings suggest that future progress in offline MARL should prioritise (i) constructing large and diverse, multi-task datasets, and (ii) carefully tuning their models' capacity for the given data budget to maximise zero-shot generalisation. We release code, datasets, task splits, and training scripts to encourage reproducibility and to establish stronger benchmarks for evaluating generalisation in offline MARL.

Limitations and future work. Our work is limited to centralised sequence model architectures, and although these represent a powerful and performant model class, promising future work could include extending our analysis to decentralised and CTDE algorithms. Additional areas of interest include studying the limits of transfer across environments (not only tasks), and investigating accelerating fine-tuning in safety-critical and data-scarce real-world domains.

REFERENCES

Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine intelligence 15*, pp. 103–129, 1995.

Paul Barde, Jakob Foerster, Derek Nowrouzezahrai, and Amy Zhang. A model-based solution to the offline multi-agent reinforcement learning coordination problem. In *International Conference on Autonomous Agents and Multiagent Systems*, 2024.

Clément Bonnet, Daniel Luo, Donal John Byrne, Shikha Surana, Sasha Abramowitz, Paul Duckworth, Vincent Coyette, Laurence Illing Midgley, Elshadai Tegegn, Tristan Kalloniatis, et al. Jumanji: a diverse suite of scalable reinforcement learning environments in jax. In *The Twelfth International Conference on Learning Representations*, 2024.

The Viet Bui, Thanh Hong Nguyen, and Tien Mai. Comadice: Offline cooperative multi-agent reinforcement learning with stationary distribution shift regularization. In *International Conference on Learning Representations*, 2025.

Yevgen Chebotar, Quan Vuong, Karol Hausman, Fei Xia, Yao Lu, Alex Irpan, Aviral Kumar, Tianhe Yu, Alexander Herzog, Karl Pertsch, et al. Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions. In *Conference on Robot Learning*, pp. 3909–3928. PMLR, 2023.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Yuan Cheng, Songtao Feng, Jing Yang, Hong Zhang, and Yingbin Liang. Provable benefit of multitask representation learning in reinforcement learning. *Advances in Neural Information Processing Systems*, 35:31741–31754, 2022.

Qiwen Cui and Simon S Du. When are offline two-player zero-sum markov games solvable? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35. Curran Associates, Inc., 2022a.

Qiwen Cui and Simon S Du. Provably efficient offline multi-agent reinforcement learning via strategy-wise bonus. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35. Curran Associates, Inc., 2022b.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

Jemma Daniel, Ruan de Kock, Louay Ben Nessir, Sasha Abramowitz, Omayma Mahjoub, Wiem Khlifi, Claude Formanek, and Arnu Pretorius. Multi-agent reinforcement learning with selective state-space models. *arXiv preprint arXiv:2410.19382*, 2024.

Ruan de Kock, Arnu Pretorius, and Jonathan Shock. Is an exponentially growing action space really that bad? validating a core assumption for using multi-agent rl. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 2490–2492, 2025.

Eslam Eldeeb, Houssem Sifaou, Osvaldo Simeone, Mohammad Shehab, and Hirley Alves. Conservative and risk-aware offline multi-agent reinforcement learning. *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2024. ISSN 2372-2045. doi: 10.1109/tccn.2024.3499357. URL http://dx.doi.org/10.1109/TCCN.2024.3499357.

Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taïga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, et al. Stop regressing: training value functions via classification for scalable deep rl. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 13049–13071, 2024.

Claude Formanek, Omayma Mahjoub, Louay Ben Nessir, Sasha Abramowitz, Ruan de Kock, Wiem Khlifi, Simon Du Toit, Felix Chalumeau, Daniel Rajaonarivonivelomanantsoa, Arnol Fokam, et al. Oryx: a performant and scalable algorithm for many-agent coordination in offline marl. *Advances in neural information processing systems*, 2025.

Juan Claude Formanek, Callum Rhys Tilbury, Jonathan Phillip Shock, Kale ab Tessera, and Arnu Pretorius. Reduce, reuse, recycle: Selective reincarnation in multi-agent reinforcement learning. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023. URL https://openreview.net/forum?id=_Nz9lt2qQfV.

Rihab Gorsane, Omayma Mahjoub, Ruan John de Kock, Roland Dubb, Siddarth Singh, and Arnu Pretorius. Towards a standardised performance evaluation protocol for cooperative marl. *Advances in Neural Information Processing Systems*, 35:5510–5521, 2022.

Haoran He, Chenjia Bai, Kang Xu, Zhuoran Yang, Weinan Zhang, Dong Wang, Bin Zhao, and Xuelong Li. Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning. *Advances in neural information processing systems*, 36:64896–64917, 2023.

Jacob Hilton, Jie Tang, and John Schulman. Scaling laws for single-agent reinforcement learning, 2023. URL https://arxiv.org/abs/2301.13442.

Ehsan Imani and Martha White. Improving regression performance with distributional losses. In *International conference on machine learning*, pp. 2157–2166. PMLR, 2018.

Jiechuan Jiang and Zongqing Lu. Offline decentralized multi-agent reinforcement learning, 2021. URL https://arxiv.org/abs/2108.01832.

Yuheng Jing, Kai Li, Bingyun Liu, Yifan Zang, Haobo Fu, QIANG FU, Junliang Xing, and Jian Cheng. Towards offline opponent modeling with in-context learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 1998.

Dmitry Kalashnikov, Jake Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Scaling up multi-task robotic reinforcement learning. In *5th Annual Conference on Robot Learning*, 2021.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76:201–264, 2023.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Aviral Kumar, Rishabh Agarwal, Xinyang Geng, George Tucker, and Sergey Levine. Offline q-learning on diverse multi-task data both scales and generalizes. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022a.

Aviral Kumar, Anikait Singh, Frederik Ebert, Mitsuhiko Nakamoto, Yanlai Yang, Chelsea Finn, and Sergey Levine. Pre-training for robots: Offline rl enables learning new tasks from a handful of trials. *arXiv preprint arXiv:2210.05178*, 2022b.

Kuang-Huei Lee, Ofir Nachum, Mengjiao Sherry Yang, Lisa Lee, Daniel Freeman, Sergio Guadarrama, Ian Fischer, Winnie Xu, Eric Jang, Henryk Michalewski, et al. Multi-game decision transformers. *Advances in neural information processing systems*, 35:27921–27936, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Chao Li, Ziwei Deng, Chenxing Lin, Wenqi Chen, Yongquan Fu, Weiquan Liu, Chenglu Wen, Cheng Wang, and Siqi Shen. Dof: A diffusion factorization framework for offline multi-agent reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jie Liu, Yinmin Zhang, Chuming Li, Zhiyuan You, Zhanhui Zhou, Chao Yang, Yaodong Yang, Yu Liu, and Wanli Ouyang. Maskma: Towards zero-shot multi-agent decision making with mask-based collaborative learning. *Transactions on Machine Learning Research*, 2024a.

Sicong Liu, Yang Shu, Chenjuan Guo, and Bin Yang. Learning generalizable skills from offline multi-task data for multi-agent cooperation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zongkai Liu, Qian Lin, Chao Yu, Xiawei Wu, Yile Liang, Donghui Li, and Xuetao Ding. Offline multi-agent reinforcement learning via in-sample sequential policy optimization, 2024b. URL https://arxiv.org/abs/2412.07639.

Omayma Mahjoub, Sasha Abramowitz, Ruan John de Kock, Wiem Khlifi, Simon Verster Du Toit, Jemma Daniel, Louay Ben Nessir, Louise Beyers, Juan Claude Formanek, Liam Clark, et al. Sable: a performant, efficient and scalable sequence model for marl. In *Forty-second International Conference on Machine Learning*, 2025.

Daiki E. Matsunaga, Jongmin Lee, Jaeseok Yoon, Stefanos Leonardos, Pieter Abbeel, and Kee-Eung Kim. Alberdice: Addressing out-of-distribution joint actions in offline multi-agent rl via alternating stationary distribution correction estimation, 2023. URL https://arxiv.org/abs/2311.02194.

Ishita Mediratta, Qingfei You, Minqi Jiang, and Roberta Raileanu. The generalization gap in offline reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Linghui Meng, Muning Wen, Chenyang Le, Xiyun Li, Dengpeng Xing, Weinan Zhang, Ying Wen, Haifeng Zhang, Jun Wang, Yaodong Yang, et al. Offline pre-trained multi-agent decision transformer. *Machine Intelligence Research*, 20(2):233–248, 2023.

Ling Pan, Longbo Huang, Tengyu Ma, and Huazhe Xu. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification, 2022. URL https://arxiv.org/abs/2111.11188.

Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V Albrecht. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Simon Rosen, Abdel Mfougouon Njupoun, Geraud Nangue Tasse, Steven James, and Benjamin Rosman. Optimal task generalisation in multi-agent reinforcement learning. In *Coordination and Cooperation for Multi-Agent Reinforcement Learning Methods Workshop*, 2024.

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. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

Lukas Schäfer. Task generalisation in multi-agent reinforcement learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, pp. 1863–1865. International Foundation for Autonomous Agents and Multiagent Systems, 2022.

Kajetan Schweighofer, Marius-constantin Dinu, Andreas Radler, Markus Hofmarcher, Vihang Prakash Patil, Angela Bitto-Nemling, Hamid Eghbal-Zadeh, and Sepp Hochreiter. A dataset perspective on offline reinforcement learning. In *Conference on Lifelong Learning Agents*, pp. 470–517. PMLR, 2022.

Jianzhun Shao, Yun Qu, Chen Chen, Hongchang Zhang, and Xiangyang Ji. Counterfactual conservative q learning for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:77290–77312, 2023.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

Callum Rhys Tilbury, Juan Claude Formanek, Louise Beyers, Jonathan Phillip Shock, and Arnu Pretorius. Coordination failure in cooperative offline MARL. In *ICML 2024 Workshop: Aligning Reinforcement Learning Experimentalists and Theorists*, 2024.

Xiangsen Wang, Haoran Xu, Yinan Zheng, and Xianyuan Zhan. Offline multi-agent reinforcement learning with implicit global-to-local value regularization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 52413–52429. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a46c84276e3a4249ab7dbf3e069baf7f-Paper-Conference.pdf.

Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems*, 35:16509–16521, 2022.

Chengjie Wu, Pingzhong Tang, Jun Yang, Yujing Hu, Tangjie Lv, Changjie Fan, and Chongjie Zhang. Conservative offline policy adaptation in multi-agent games. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=C8pvL8Qbfa.

Yueh-Hua Wu, Xiaolong Wang, and Masashi Hamaya. Elastic decision transformer. *Advances in neural information processing systems*, 36:18532–18550, 2023b.

Wei Xiong, Han Zhong, Chengshuai Shi, Cong Shen, Liwei Wang, and Tong Zhang. Nearly minimax optimal offline reinforcement learning with linear function approximation: Single-agent mdp and markov game, 2023. URL https://arxiv.org/abs/2205.15512.

Zhiyuan Xu, Kun Wu, Zhengping Che, Jian Tang, and Jieping Ye. Knowledge transfer in multi-task deep reinforcement learning for continuous control. *Advances in Neural Information Processing Systems*, 33:15146–15155, 2020.

Taku Yamagata, Ahmed Khalil, and Raul Santos-Rodriguez. Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl. In *International Conference on Machine Learning*, pp. 38989–39007. PMLR, 2023.

Yiqin Yang, Xiaoteng Ma, Chenghao Li, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning, 2021a. URL https://arxiv.org/abs/2106.03400.

Yiqin Yang, Xiaoteng Ma, Chenghao Li, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:10299–10312, 2021b.

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Sergey Levine, and Chelsea Finn. Conservative data sharing for multi-task offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34:11501–11516, 2021.

Fuxiang Zhang, Chengxing Jia, Yi-Chen Li, Lei Yuan, Yang Yu, and Zongzhang Zhang. Discovering generalizable multi-agent coordination skills from multi-task offline data. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=53FyUAdP7d.

Yuheng Zhang, Yu Bai, and Nan Jiang. Offline learning in markov games with general function approximation, 2023b. URL https://arxiv.org/abs/2302.02571.

Qinqing Zheng, Amy Zhang, and Aditya Grover. Online decision transformer. In *international conference on machine learning*, pp. 27042–27059. PMLR, 2022.

Hai Zhong, Xun Wang, Zhuoran Li, and Longbo Huang. Offline-to-online multi-agent reinforcement learning with offline value function memory and sequential exploration, 2024a. URL https://arxiv.org/abs/2410.19450.

Han Zhong, Wei Xiong, Jiyuan Tan, Liwei Wang, Tong Zhang, Zhaoran Wang, and Zhuoran Yang. Pessimistic minimax value iteration: Provably efficient equilibrium learning from offline datasets, 2022. URL https://arxiv.org/abs/2202.07511.

Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32): 1–67, 2024b.

Yihe Zhou, Yuxuan Zheng, Yue Hu, Kaixuan Chen, Tongya Zheng, Jie Song, Mingli Song, and Shunyu Liu. Cooperative policy agreement: Learning diverse policy for offline marl. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(21):23018–23026, Apr. 2025. doi: 10.1609/aaai.v39i21.34465. URL https://ojs.aaai.org/index.php/AAAI/article/view/34465.