MangoBench: A Benchmark for Multi-Agent Goal-Conditioned Offline Reinforcement Learning

Yi Wang[1] Ningze Zhong[1] Zhiheng Fu[2] Longguang Wang[1] Ye Zhang[1*] Yulan Guo[1*]
[1]Sun Yat-sen University [2]The Hong Kong Polytechnic University

Abstract

Offline Multi-Agent Reinforcement Learning (MARL) is critical for coordinating multiple agents in costly and unsafe environments, yet existing methods struggle with high sensitivity to reward functions and weak generalization to new goals, limiting its practical impact. Inspired by single-agent Offline Goal-Conditioned RL (OGCRL), we propose the first goal-conditioned offline MARL framework, extending OGCRL to multi-agent settings under both fully decentralized and centralized training with decentralized execution (CTDE) paradigms. To systematically evaluate this setting, we introduce MangoBench, the first fully cooperative multi-goal benchmark for MARL, covering 3 environments, 4 agent types, and 47 tasks, designed to assess joint-control locomotion, synchronous and asynchronous bimanual manipulation, and robustness to high-dimensional inputs. Extensive experiments demonstrate that our baselines achieve strong multi-goal generalization under sparse rewards, yet no method dominates all tasks, revealing both the intrinsic complexity and the unexplored potential of goal-conditioned offline MARL.

1. Introduction

Multi-Agent Reinforcement Learning (MARL) investigates how multiple agents learn to act and coordinate within a shared environment through interaction, cooperation, or competition [29, 39]. It has broad practical relevance, offering a principled framework for addressing complex multi-agent problems in real-world scenarios such as autonomous driving, collaborative robotics, and smart grids [6, 20, 33, 38].

Offline MARL, a branch of MARL, learns policies purely from pre-collected datasets, effectively avoiding costly, unsafe, and impractical exploration in physical environments [21, 28, 30, 34]. Despite these advantages, applying offline MARL in real-world settings remains challenging due to reward sensitivity and limited generalization. Reinforcement learning aims to maximize the expected cumulative reward, even minor perturbations in the reward function can lead to drastic policy shifts. Moreover, the dependence on task-specific, handcrafted reward functions often limits the agent's ability to generalize to new goals or environments. These challenges hinder the practicality of offline MARL, motivating the need for more robust and generalizable learning frameworks.

In the single-agent domain, Offline Goal-Conditioned Reinforcement Learning (OGCRL) [22, 23] has demonstrated notable advantages over conventional paradigms [10, 19, 31]. By incorporating goal relabeling and random goal sampling, each trajectory in the offline dataset can be transformed into a learning instance that maps any state to any goal, thereby enriching the space of state-goal combinations and improving generalization. Additionally, in OGCRL, the reward is typically defined as 0 when the agent reaches the goal and -1 otherwise, which greatly simplifies reward design. In short, the OGCRL framework for multi-task offline RL eliminates manual reward and goal design. It only requires basic roll-outs, as the necessary rewards and goals are automatically generated. Considering these strengths of OGCRL and the aforementioned drawbacks of offline MARL, a natural question arises: *Can we extend offline MARL to a goal-conditioned setting to overcome its inherent limitations?*

Motivated by prior work in multi-agent learning [5, 17, 21, 29, 35], we establish the *first* **goal-conditioned offline MARL** framework by extending OGCRL algorithms to multi-agent scenarios under both fully decentralized and centralized training with decentralized execution (CTDE) paradigms, as shown in Fig. 1. To enable goal-conditioned learning under the fully decentralized setting, we introduce a structured factorization of the global goal into individualized goal representations, allowing each agent to make decisions solely based on local information. In the CTDE setting, we integrate individual goals into a unified global objective to enhance value learning, while conditioning each actor on its local goal for decentralized execution.

Another major challenge in extending OGCRL to offline

*Corresponding authors.

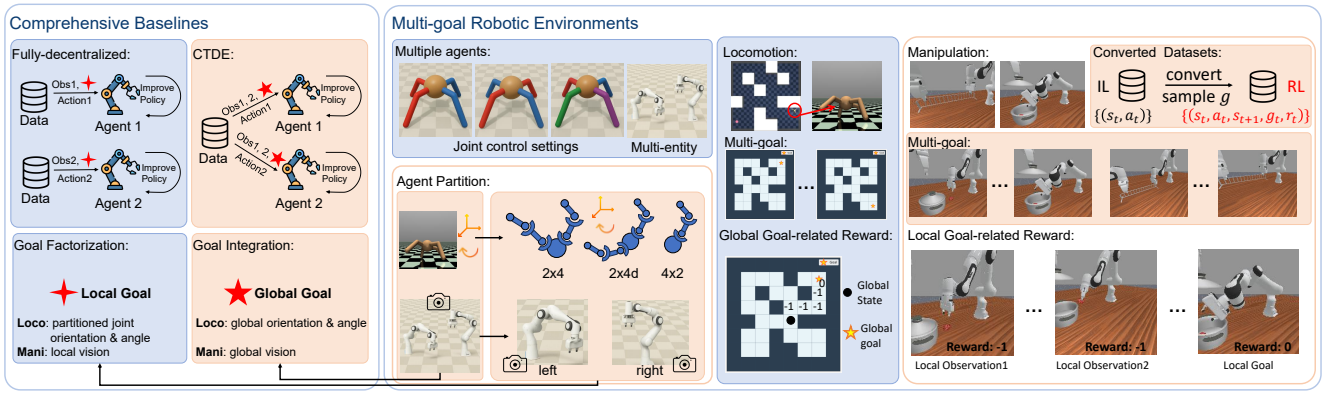


Figure 1. **Overview of MangoBench.** Through goal relabeling and structured robot factorization into local and global goals, we design goal-conditioned offline MARL baselines under both decentralized and CTDE paradigms, and introduce environments supporting joint-control locomotion and multi-entity manipulation with goal-related rewards, standardized RL datasets, and multi-goal evaluation.

MARL lies in the absence of suitable benchmarks. Existing MARL benchmarks [2, 3, 7, 12, 14, 17, 24, 25, 27] primarily evaluate online algorithms using dense, task-specific rewards, making them unsuitable for goal-conditioned offline scenarios that rely on sparse, goal-based rewards and require multi-goal generalization. Moreover, most current environments [1, 7, 12, 17, 24, 25] focus only on single-goal evaluations, leading to biased and incomplete assessments of goal-conditioned policies [23].

To address these gaps, we create **MangoBench**, the *first* fully cooperative **multi-goal benchmark** tailored for goal-conditioned offline MARL. MangoBench provides a comprehensive evaluation suite consisting of 3 environments, 4 agent types, 45 locomotion tasks, 2 manipulation tasks, and 6 baseline algorithms (see in Fig. 1). It supports diverse multi-agent configurations, including multi-entity and joint-control settings. For multi-entity tasks, the reward function adopts a localized formulation that encapsulates each entity's individual contribution, while for joint-control tasks, it assumes a global formulation that measures the collective efficacy of the unified control system. Moreover, we design progressively challenging multi-agent experiments to evaluate key capabilities such as joint-control locomotion, synchronous and asynchronous bimanual manipulation, and robustness to high-dimensional visual inputs.

Extensive experiments demonstrate that our baselines achieve strong performance on MangoBench even under sparse reward conditions, while exhibiting notable generalization across multiple tasks. Interestingly, no single method consistently outperforms others across all settings, highlighting both the complexity and the unexplored potential of this emerging research frontier.

Our contributions are summarized as follows:

- We propose the first systematic framework for goal-conditioned offline MARL through extending OGCRL under both fully decentralized and CTDE paradigms, improving its generalization ability.
- We create MangoBench, the first fully cooperative multi-goal benchmark, supporting diverse multi-agent tasks and evaluation protocols.
- Extensive experiments demonstrate the superior performance and generalization of our baselines while revealing the inherent complexity and open challenges in this new setting.

2. Related Works

Prior works [2, 3, 7, 12, 14, 17, 24, 25, 27] have benchmarked multi-agent learning performance by online reinforcement learning with dense rewards designed for specific tasks, which are not suitable for evaluating goal-conditioned offline MARL. Additionally, the tasks in [1, 7, 12, 17, 24, 25] do not use multiple goals for evaluation, even though we can collect offline datasets for these tasks, directly evaluating multi-task policies (goal-conditioned policies) only on the single, original goal when using these tasks will result in limited evaluation [23].

Inspired by [23, 24, 37], we introduce MangoBench, the first benchmark focusing on goal-conditioned offline MARL, which has the properties of comprehensive baselines under both fully decentralized and CTDE paradigms, multi-goal evaluation, diverse multi-agent configurations, standard RL datasets, challenges of tasks (e.g., joint-control

Table 1. Comparison of Multi-Agent Environments

Environment	Type	Multi-Goal	Stochasticity	Number of Tasks
VMAS [1]	Cooperative + Competitive	No	No	27
SMACv2 [7]	Cooperative	No	Yes	15
MPE [17]	Cooperative + Competitive	No	No	9
SISL [12]	Cooperative	No	No	3
Pommerman [25]	Cooperative + Competitive	No	No	3
MA-MUJOCO [24]	Cooperative	No	No	14
MangoBench (Ours)	Cooperative	Yes	Yes	47

Table 2. Properties of Reward Function in Multi-Agent Environments

Environment	Properties of Reward Function
VMAS [1]	Customized rewards based on scenarios (navigation, sampling, balance, etc.)
SMACv2 [7]	Rewards based on hit-point damage dealt and enemy units killed, with a bonus for winning the battle
MPE [17]	Rewards usually based on distance to landmark
SISL [12]	Dense cooperation rewards (walking distance, capture count, target collection, etc.)
Pommerman [25]	Immediate rewards such as victory or defeat, survival time, and mine detonation, given step by step
MA-MUJOCO [24]	Reuse dense velocity/displacement and other rewards of the original single agent as global team rewards
MangoBench (Ours)	Simple and generalizable rewards only depend on whether the state reaches goal

locomotion, synchronous and asynchronous bimanual manipulation) and simple rewards design. We claim that this setting is the best way for us to scale up the datasets collecting process in multi-agent learning, for it does not need any manually designed rewards or goals. Given any roll-outs (s, a, s'), we can immediately get the full data tuple (s, a, s', r, g) with our settings due to the self-generated rewards and goals. We summarize the properties of the previous environments in Table 1 and their reward function in Table 2 to highlight the superiority of our MangoBench in terms of task comprehensiveness, diversity, and the simplicity of rewards.

3. Goal-Conditioned Offline MARL

3.1. Problem Definition

We propose a new setting in offline MARL, termed **goal-conditioned offline MARL**, which aims to solve cooperative tasks **through generalizable policies without reward engineering**. We define this problem under a partially observable Markov game [15], defined as $\mathcal{M} = \big(\mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^{N}, P, \gamma\big)$ (a Markov process without rewards) and an unlabeled dataset $\mathcal{D}$. Multiple agents share a common environment and learn jointly without further online interaction. $\mathcal{N} = \{1, \ldots, N\}$ denotes the agent set, and $\mathcal{S}$ represents the joint state space of all agents. Each agent i receives a partial observation $\mathcal{O}_i$ correlated with the state $\mathcal{S} \rightarrow \mathcal{O}_i$ and takes actions from its action space $\mathcal{A}_i$. The joint action space is $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$. Each episode is conditioned on a goal $g \in \mathcal{S}$, which can be factorized into local goals g_i based on each agent's observation $\mathcal{O}_i$. Each agent follows a goal-conditioned policy $\pi_i(a_i \mid o_i, g_i)$ and interacts through the transition function $P : \mathcal{S} \times \mathcal{A}_1 \times ... \times \mathcal{A}_N \rightarrow \mathcal{S}$.

Learning Objective. For **multi-entity tasks**, the reward is defined locally:

$$r(o_i, g_i) = \begin{cases} r_1, & o_i \in \text{GoalStates}(g_i), \\ r_2, & \text{otherwise}, \end{cases} \tag{1}$$

where r_1 and r_2 are constants (e.g., $r_1 = 0, r_2 = -1$). For **joint-control tasks**, the reward depends on global goal attainment:

$$r(\boldsymbol{o}, \boldsymbol{g}) = \begin{cases} r_1, & \boldsymbol{o} \in \text{GoalStates}(\boldsymbol{g}), \\ r_2, & \text{otherwise}. \end{cases} \tag{2}$$

We assume access to a fixed offline dataset $\mathcal{D}$ containing K trajectories:

$$\mathcal{D} = \{\tau^{(k)} = (s_0, a_0, s_1, \ldots, s_T)\}_{k=1}^{K}, \tag{3}$$

collected by a behavior policy μ[1]. The objective is to learn policies $\pi_i(a_i \mid o_i, g_i)$ that maximize the expected discounted goal-conditioned return:

$$\max_{\pi_i} \mathbb{E}_{(o_i^t, a_i^t, o_i^{t+1}, g_i) \sim \mathcal{D}} \Big[\sum_{t=0}^{\infty} \gamma^t r(o_i^t, g_i) \Big], \tag{4}$$

where $\gamma \in [0, 1)$ denotes the discount factor. This formulation explicitly captures the cooperative multi-agent nature, the offline learning constraint, and the binary goal-conditioned reward structure.

3.2. Training Paradigm

Fully Decentralized Setting: In the fully decentralized setting, each agent i learns its own goal-conditioned policy $\pi_i(a_i \mid o_i, g_i)$ using only local observations o_i, local actions a_i, and local goals g_i, without relying on inter-agent communication during training and execution. The learning objective of each agent is formulated under a unified goal-conditioned framework:

$$\nabla_{\theta_i} J_i = \mathbb{E}_{(o_i, a_i, g_i) \sim \mathcal{D}} \Big[\nabla_{\theta_i} \log \pi_i(a_i \mid o_i, g_i)\, Q_i(o_i, a_i, g_i) \Big]. \tag{5}$$

Different baselines adopt variant formulations of this unified objective (see Appendix for details). Equation (5) illustrates that each agent optimizes its parameters to increase the likelihood of actions leading to higher expected cumulative rewards $Q_i(o_i, a_i, g_i)$ under its goal condition g_i. The fundamental mechanism allows each agent independently maximizes its expected goal-achievement return.

[1] We use s_t and a_t to denote the joint states and actions of all agents at time t, which can be partitioned for each agent as o_i^t and a_i^t, respectively. The same notation applies to s_{t+1} and a_{t+1}.

CTDE: In CTDE, each agent i executes its goal-conditioned policy $\pi_i(a_i \mid o_i, g_i)$ independently using only its local observation o_i and local goal g_i, but during training, it has access to additional global information such as the joint observations $\boldsymbol{o} = (o_1, \ldots, o_N)$, joint actions $\boldsymbol{a} = (a_1, \ldots, a_N)$, and shared goals $\boldsymbol{g} = (g_1, \ldots, g_N)$. The unified training objective is formulated as a centralized goal-conditioned framework:

$$\nabla_{\theta_i} J_i = \mathbb{E}_{(\boldsymbol{o},\boldsymbol{a},\boldsymbol{g})\sim\mathcal{D}}\Big[\nabla_{\theta_i} \log \pi_i(a_i \mid o_i, g_i)\, Q_i(\boldsymbol{o}, \boldsymbol{a}, \boldsymbol{g})\Big], \tag{6}$$

where $Q_i(\boldsymbol{o}, \boldsymbol{a}, \boldsymbol{g})$ is the centralized goal-conditioned action-value function that depends on the joint observation-action tuple and captures inter-agent interactions during training. Equation (6) indicates that each agent optimizes its policy parameters using gradients guided by a centralized critic, enabling coordinated learning while maintaining decentralized execution at test time.

3.3. Proposed Baseline Algorithms

Given the lack of existing algorithms tailored for goal-conditioned offline MARL, we introduce six baseline methods, four derived from goal-conditioned offline RL and two from offline MARL, to establish a foundation for future research and advancements in this direction.

- **GCMBC**: We propose goal-conditioned multi-agent behavior cloning (GCMBC), a fully decentralized extension of GCBC [11, 18], serving as a fundamental baseline representing the behavioral performance of the dataset.
- **ICRL**: Independent Contrastive Reinforcement Learning (ICRL), a fully decentralized multi-agent extension of CRL [8, 23], designed to evaluate the effectiveness of contrastive value learning in handling sparse rewards for multi-agent tasks.
- **IHIQL** and **HIQL-CTDE**: To build stronger benchmark baselines and evaluate different training paradigms, we extend the state-of-the-art HIQL [22, 23] into the goal-conditioned offline MARL setting, introducing IHIQL (fully decentralized) and HIQL-CTDE (centralized training with decentralized execution), both leveraging hierarchical policies to enhance robustness against sparse rewards and improve long-horizon reasoning.
- **GCOMIGA** and **GCOMAR**: To evaluate whether the current offline MARL can robustly handle the noise caused by sparse rewards to learn useful policy under the goal-conditioned environment, we designed GCOMIGA and GCOMAR by goal relabeling and randomly goal sampling, the goal-conditioned variants of OMIGA [34] and OMAR [21].

The details and formulas of our baseline algorithms are shown in the Appendix.

4. Proposed Benchmark: MangoBench

To better evaluate the goal-conditioned offline MARL baselines, we create MangoBench. In this section, we will introduce our multi-agent setting, converted RL datasets, multi-goal evaluation, rewards design, and different tasks for cooperative locomotion and manipulation of our benchmarks.

4.1. Cooperative Locomotion

Multi-Agent Setting: We introduce a structured factorization of the robot body into different joints to make multiple agents within a single robot locomote cooperatively. The locomotion tasks involve three distinct multi-agent settings, including 2 agents $\times$ 4 joints, 2 agents $\times$ 4 joints (diagonal) and 4 agents $\times$ 2 joints, as illustrated in Fig. 1.

- 2 agents $\times$ 4 joints: The ant has 4 legs, each with 2 joints, totaling 8 joints. In this setting, the 8 joints are split between two agents, each controlling 4 joints. The division typically follows a left-right or front-back grouping.
- 2 agents $\times$ 4 joints (D): This variant also assigns four joints to each of the two agents, but the assignment is done diagonally. For instance, one agent may control the front-left and rear-right legs, while the other controls the front-right and rear-left legs. This cross-pattern configuration introduces more complex coordination dynamics.
- 4 agents $\times$ 2 joints: Each leg (2 joints) is controlled by a separate agent. Thus, 4 agents are assigned one leg each, requiring a higher level of coordination among more agents, each with more localized control.

The global goal corresponds to the complete joint states of the ant, whereas the local goals are derived from the structured factorization of the robot's body, where each body part is assigned a goal based on its corresponding local joints for baseline training.

Multi-goal Evaluation: During the evaluation phase, each agent independently utilizes its own observation space to generate actions. These individual actions are then combined to form the final action set to interact with the environment. We utilize open-source datasets in OGBench [23]. For complete assessment, we evaluate performance based on five predefined goals as outlined in OGBench [23].

Goal-related Rewards Design: For joint-control tasks, we design a global goal-related reward to evaluate the collective efficacy of the unified control system. The reward is computed based on the global observation $\boldsymbol{o}$ and the global goal $\boldsymbol{g}$, reflecting the overall performance of the entire robot (e.g., the locomotion system's global orientation and position). The same scalar reward value is then broadcast to all agents, ensuring consistent optimization across all control modules. This design allows all agents in the unified control system to jointly pursue the same global objective while maintaining synchronized reward feedback.

AntMaze Task: In this task, multiple agents jointly con-

trol different body parts to navigate through the maze and reach the goal. Following OGBench [23], we construct four maze variants with increasing spatial complexity, including **medium**, **large**, **giant**, and **teleport**. The datasets also match the OGBench taxonomy: **Navigate** (hiqh-quality), **Stitch** (short goal-reaching trajectories), and **Explore** (low-quality but high-coverage). By varying maze scale and dataset type, we systematically assess how algorithms adapt to increasing environmental complexity, cooperative skill composition, and data diversity.

Ant-Soccer Task: In the Ant-Soccer environment, a simulated ant is controlled by two or four cooperative agents, each manipulating specific joints to push a football and reach the goal. The task emphasizes coordinated control for smooth locomotion and effective ball interaction. Following OGBench [23], we construct two maze variants with different complexity, including **Arena**, **Maze**. In Arena setting, agents operate in an open field without obstacles, learning stable walking gaits and precise force application to push the ball and reach the goal while maintaining balance and continuous contact. The Maze setting introduces walls and narrow corridors between the start and goal positions, where agents must combine locomotion, ball control, and navigation planning to maneuver through the maze without trapping the ball, making it significantly more challenging than the open-field setting.

4.2. Cooperative Manipulation

Multi-Agent Setting: To construct diverse multi-agent forms for a comprehensive evaluation of the baseline algorithms, we introduce multiple agent entities in the cooperative manipulation environment. For each robotic arm, its local visual observation is randomly sampled as the local goal, whereas the global goal is defined from the sampled global visual observation including both arms and the surrounding environment, serving as the goal input for training the goal-conditioned baselines.

Converted Standard RL datasets: We transform the open-source *RoboFactory* datasets [26] into a format suitable for goal-conditioned offline MARL. Specifically, we reorganize the data into a standard Markov Decision Process (MDP) structure, where each sample includes a randomly sampled goal and a corresponding goal-conditioned reward, as shown in Fig. 1. In contrast to the original datasets which is designed primarily for imitation learning frameworks such as Diffusion Policy (DP) [4], our reformulation enables direct training of goal-conditioned offline MARL algorithms.

Multi-goal Evaluation: To avoid biased evaluation, each manipulation task is associated with 5 clearly defined sequential multiple goals (seen in Fig. 1 and the Appendix) to facilitate consistent comparison across algorithms. To ensure robustness, all evaluation metrics are averaged over 100 random seeds.

Goal-related Rewards Design: For multi-entity tasks, the reward function adopts a localized formulation that captures each entity's individual contribution to the overall task performance. Each agent receives a goal-conditioned reward computed from its local observation o_i and local goal g_i, enabling each entity independently optimizes its goal-conditioned behavior while contributing to the cooperative task outcome.

2-Agent Tasks: We adopt several two-agent cooperative manipulation tasks from *RoboFactory* [26] for our experiments. The *lift-barrier* is a synchronous manipulation task that requires two robotic arms to simultaneously grasp both sides of a barrier and lift it to a target height, demanding precise temporal coordination between agents. In contrast, the *place-food* task involves asynchronous cooperation, where one arm must first open the pot lid before the other arm can place the food inside. Through these synchronous and asynchronous cooperation tasks, we aim to highlight how different goal-conditioned offline MARL algorithms handle synchronous versus sequential collaboration in multi-arm manipulation.

5. Results and Analysis

As presented in Fig. 2, we evaluated the baselines on the benchmark, full results on more challenging datasets are shown in the Appendix. We discuss the findings in Section 5.1 and 5.2. To better analyze the difference between the fully decentralized and CTDE settings, we further compare the results of IHIQL and HIQL-CTDE separately in Section 5.3. We also analyze the reasons for the poor performance of existing offline MARL in Section 5.4.

5.1. Results on Locomotion Tasks

Overall Performance. The agents were trained for 1 million gradient steps and the results were averaged over five seeds and five predefined goals. As shown in Fig. 2 and the full results in the Appendix, IHIQL emerges as the state-of-the-art algorithm for multi-agent offline goal-conditioned tasks, owing to its hierarchical policy that mitigates noise from sparse rewards. ICRL struggles with generalization and long-horizon reasoning, often failing in larger mazes (see the result of *antmaze-large-stitch* in Fig. 2). GCMBC performs suboptimally, as behavior cloning fails to utilize valuable information, especially from failure cases in low-quality datasets like *explore*. Meanwhile, GCOMIGA and GCOMAR fail in most tasks due to their inability to cope with the challenges introduced by sparse rewards, a phenomenon we analyze further in Section 5.4. Nonetheless, no current algorithm consistently excels across all tasks.

Comparison between Single-agent Setting. The comparison between our multi-agent baselines and their single-agent counterparts demonstrates both the effectiveness and

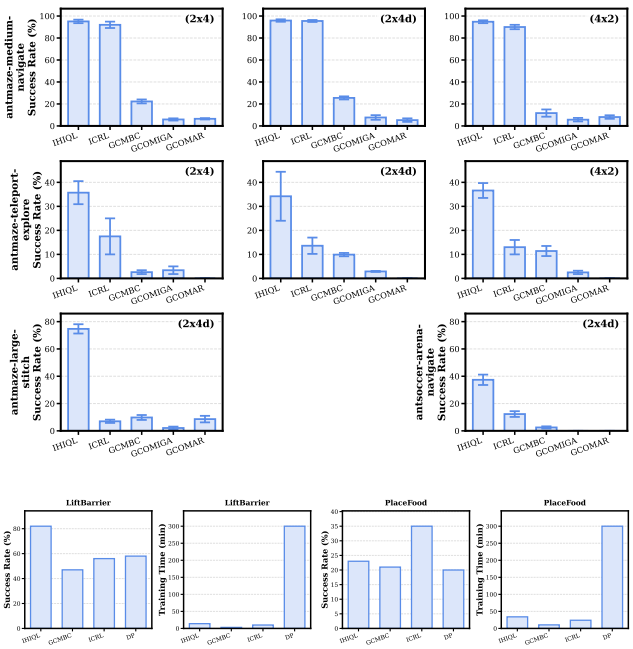


Figure 2. **Benchmark Results.** For locomotion tasks, we report each method's average success rate (%) across five test-time goals, averaged over 5 seeds with standard deviations shown as error bars. Due to the poor performance of GCOMIGA and GCOMAR, they are excluded from more challenging manipulation tasks. For manipulation tasks, we report the average success rate (%) over 100 seeds for three top-performing baselines and the imitation learning method DP [4], along with training time.

robustness of our proposed baselines. Across most tasks, the performance of our baselines remains comparable to, or even exceeds, that of the single-agent algorithms [23]. Notably, in tasks such as *antmaze-teleport-explore* and *antmaze-large-stitch* (see Fig. 2), the multi-agent variants achieve substantially higher success rates. Moreover, in all *teleport* mazes, the multi-agent IHIQL consistently outperforms single-agent HIQL [22], showing improved resilience to stochastic dynamics. This advantage stems from the decentralized structure, where each agent focuses on a localized subset of states and actions, effectively reducing the complexity of learning and making it easier to estimate stochastic transitions within its subspace [5]. These findings validate that our baselines can reliably handle coordination and uncertainty in complex multi-agent settings.

Challenges in Complex Coordination Tasks. Nevertheless, in more challenging environments such as the multi-agent *Ant-Soccer* tasks, performance declines markedly compared to single-agent baselines, with GCOMIGA and GCOMAR failing completely. Rendered videos reveal that agents in *Ant-Soccer* struggle to maintain stable locomotion while simultaneously manipulating

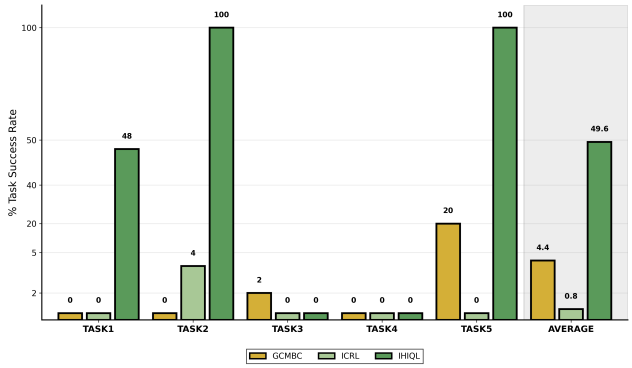


Figure 3. **Multiple Goals Evaluation.** This figure reports success rates of the three baseline algorithms on *antmaze-medium-explore* task under 5 goals, which indicates that only use single goal to evaluate goal-conditioned offline MARL algorithms can lead to inaccurate conclusions.

the ball-often rolling over, getting trapped in maze corners, or producing conflicting torques between limbs. This difficulty arises because the environmental object (the ball) introduces external dependencies not represented in the agents' intrinsic states, making coordinated control more complex. These results highlight the limitations of current methods and the necessity of our benchmark in exposing such coordination challenges. Future work could extend goal-conditioned offline MARL under the CTDE paradigm to improve global coordination, enabling agents to manage dynamic object interactions and remain robust under suboptimal or noisy datasets [23, 36].

Multiple Goals Evaluation. Fig. 3 shows multi-goal evaluation on the *antmaze-medium-explore* $2 \times 4d$ task, emphasizing the need for multi-goal evaluation over existing MARL benchmarks [1, 7, 12, 17, 24, 25], which use a single fixed goal and cannot assess generalization in goal-conditioned offline MARL. Fig. 3 highlights that evaluating with a single goal can lead to inaccurate assessments. If the goal coincides with task 4, all baselines drop to a 0 success rate, indicating failure to learn from low-quality data and contradicting their average performance. Conversely, when aligned with task 5, IHIQL *2x4d* achieves a perfect score of 1 despite an average of 49.6, clearly overestimating the algorithm's true capability.

5.2. Results on Manipulation Tasks

Overall Performance. To ensure fair comparison, the gradient steps for each task were aligned with the number of transitions in our converted RL datasets-15,000 iterations for *lift-barrier* and 38,800 for *place-food*. We evaluated three high-performing baselines against the strong imitation learning method DP [4] under the same data size to validate the effectiveness of our proposed goal-conditioned multi-agent baselines. All results were averaged over 100 seeds,

and DP results followed [26].

As shown in Fig. 2, in the *lift-barrier* task, **IHIQL** achieves the highest success rate, outperforming DP [4] by 41.4% while requiring only 5% of its training time. **GCMBC** and **ICRL** also reach competitive performance with comparable efficiency. These results highlight both the *effectiveness and efficiency* of our goal-conditioned MARL baselines and the potential of our new setting to serve as a strong foundation for multi-agent robotic learning, achieving or surpassing imitation learning methods with dramatically lower computational cost.

In the more challenging *place-food* task, **ICRL** achieves the best performance, exceeding DP [4] by 75% while training 93% faster. **IHIQL** and **GCMBC** also outperform DP [4], further confirming the robustness and superiority of our proposed baselines and underscoring the promise of goal-conditioned offline MARL as a practical and general framework for complex multi-agent robotic tasks.

Influences of Different Types of Observation. To investigate how different observation modalities affect our proposed baselines, we evaluate two types of inputs: *state* and *vision*. The *state* input contains the joint configurations of the robotic arms that directly determine their control actions, while the *vision* input consists of camera images that indirectly encode the robot's motion and environment. Interestingly, our baselines achieve higher performance under the *vision* setting while fail under *state* setting, which contrasts with the findings in imitation learning [26] where visual inputs has less contribution to policy learning. We attribute this to the fact that visual observations capture richer environmental context beyond the agent's own state, enabling RL agents to better understand their interactions with the environment and learn more effective goal-conditioned behaviors to perform beyond the datasets, whereas imitation learning merely replicates behaviors within the dataset. Additionally, to mitigate the reduction in training efficiency caused by high-dimensional visual inputs, we downsample the images to a resolution of 64×64. Remarkably, even with such low-resolution inputs, our baselines still achieve strong performance, as shown in Fig. 2.

Multiple Goals Evaluation. The results on manipulation tasks in Table 3 further validate the necessity of our proposed MangoBench benchmark. As shown in the results, success rates under multi-goal evaluation consistently surpass those under single-goal evaluation. These findings underscore the importance of evaluating multi-agent policies in a multi-goal context, which is precisely the gap that MangoBench fills for goal-conditioned offline MARL.

Table 3. Single-goal v.s. multi-goal evaluation on *lift-barrier* task.

evaluation	IHIQL	GCMBC	ICRL
single-goal	78%	22%	37%
multi-goal	82%	47%	56%

5.3. Analysis of Fully Decentralized vs. CTDE Settings.

To further justify the choice of adopting fully decentralized methods in most of our baseline algorithms, we conduct a comparative analysis between two representative baselines implemented under the fully decentralized and CTDE settings. As shown in Table 4, the performance of IHIQL significantly surpasses that of HIQL-CTDE across various tasks. Notably, the success rate of HIQL-CTDE slightly declines as training iterations increase. When the task horizon becomes longer and the state space grows larger, the performance of HIQL-CTDE deteriorates rapidly, as evidenced by its poor success rate on the *antmaze-giant-navigate* task. To accommodate the use of global states in the CTDE framework, HIQL-CTDE requires two separate goal-representation networks, one for the centralized value function and another for the decentralized heterogeneous actors. However, this architectural expansion introduces significant instability in learning and leads to performance degradation, particularly in large-scale or long-horizon tasks. We attribute this to the increased difficulty of jointly optimizing multiple networks as the complexity of the problem grows.

This explanation is further supported by comparing IGCIVL and GCIVL-CTDE. The IGCIVL algorithm, a simplified flat variant of IHIQL that employs a single actor while sharing the same value function design, exhibits comparable performance to its CTDE counterpart. Moreover, GCIVL-CTDE demonstrates more stable training behavior than HIQL-CTDE, reinforcing our claim that the degradation in HIQL-CTDE stems primarily from the hierarchical structural complexity rather than the CTDE paradigm itself. Since the CTDE setting brings only marginal or even negligible performance gains while increasing model complexity and training time, our baseline algorithms predominantly adopt the fully decentralized approach for both effectiveness and efficiency.

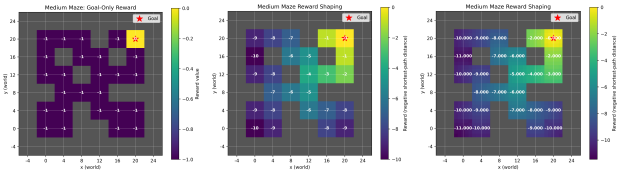
Table 4. **Fully decentralized v.s. centralized training decentralized execution.** Results on *AntMaze-navigate*. We report each method's average success rate (%) across the five test-time goals on each task. The results are averaged over 4 seeds, and we report standard deviations after ± sign.

Dataset	IHIQL	HIQL-CTDE	IGCIVL	GCIVL-CTDE
medium(2x4)	**95.1 ± 1.6**	74.0 ± 0.6	76.0 ± 3.4	75.0 ± 4.2
medium(2x4d)	**95.9 ± 1.1**	79.8 ± 4.2	68.2 ± 1.4	76.4 ± 1.6
medium(4x2)	**94.8 ± 1.3**	69.4 ± 4.8	64.2 ± 2.5	66.0 ± 2.4
large(2x4)	**85.4 ± 5.3**	44.0 ± 1.1	15.0 ± 1.4	20.5 ± 1.6
large(2x4d)	**92.2 ± 2.1**	51.2 ± 1.7	26.0 ± 0.6	21.2 ± 1.1
large(4x2)	**87.1 ± 2.2**	33.2 ± 2.8	9.1 ± 1.9	11.6 ± 2.9
giant(2x4)	**57.3 ± 2.1**	1.4 ± 0.8	0.0 ± 0.0	0.0 ± 0.0
giant(2x4d)	**50.3 ± 3.3**	2.2 ± 1.4	0.0 ± 0.0	0.4 ± 0.0
giant(4x2)	**35.5 ± 8.6**	1.6 ± 0.6	0.0 ± 0.0	0.0 ± 0.0
teleport(2x4)	**46.8 ± 1.7**	27.6 ± 2.3	37.8 ± 2.0	37.9 ± 0.2
teleport(2x4d)	**47.2 ± 2.6**	24.6 ± 0.3	38.0 ± 0.6	40.2 ± 0.8
teleport(4x2)	**48.7 ± 2.6**	22.4 ± 1.7	40.6 ± 0.9	38.6 ± 1.6

5.4. Analysis of Existing Offline MARL.

We argue that the poor performance of GCOMIGA and GCOMAR in Fig. 2 arises because existing offline MARL methods fail to overcome the challenge of sparse rewards (i.e., the noise in value function induced by sparsity in the reward signal [23]). To further analyze this issue, we conduct experiments on the *antmaze-medium-navigate* environment under three configurations: (1) GCOMIGA with goal-conditioned sparse reward, as seen in Eq. (2), (2) OMIGA [34] with sparse reward $\mathcal{R}$, (3) OMIGA [34] with shaped rewards $\mathcal{R}_1$ and $\mathcal{R}_2$. The visualizations of these different reward functions are shown in Fig. 4. Note that both the training and evaluation of OMIGA [34] are conducted under the standard offline MARL setting without goal conditioning. As shown in Fig. 5, the results clearly indicate that the poor performance of GCOMIGA on MangoBench is primarily due to the inability of OMIGA to learn effectively under sparse reward conditions. When rewards are shaped denser, OMIGA [34] demonstrates significantly better and more stable performance. We believe that OMAR [21] suffers from similar limitations.

Furthermore, recent studies in online MARL [16] have reported that agents often converge to local optima under sparse reward settings, leading to failure. In real-world robotic and multi-agent scenarios, sparse rewards are ubiquitous, as designing dense reward functions is often impractical. However, to date, no offline MARL algorithm has been explicitly designed to address the challenges posed by sparse rewards as some offline RL do [9, 13, 22, 32]. We hope that future research will focus on developing algorithms capable of robust learning in such sparse-reward environments.



(a) Sparse reward $\mathcal{R}$ (b) Shaped reward $\mathcal{R}_1$ (c) Shaped reward $\mathcal{R}_2$

Figure 4. **Visualization of different reward settings.** (a) Sparse reward $\mathcal{R}$, (b) shaped reward $\mathcal{R}_1$, and (c) shaped reward $\mathcal{R}_2$. The rewards become denser from left to right.

6. Future Research

Designing Improved Algorithms under the CTDE Setting. Our analysis reveals that the performance degradation of HIQL-CTDE primarily stems not from the CTDE paradigm itself, but from the increased architectural complexity introduced by the global goal representation network and the resulting misalignment between centralized and decentralized goal representations. The direct integration of a global goal encoder, while conceptually reasonable, creates additional learning challenges that hinder stable optimization, especially in large-scale or long-horizon tasks. These observations suggest that the CTDE framework still holds untapped potential for hierarchical goal-conditioned learning. We encourage future research to explore more efficient and coherent architectures for HIQL-CTDE, ones that preserve the coordination advantages of CTDE while mitigating its training instability. Such designs could potentially build upon the strong performance of IHIQL and further advance the understanding of goal representation learning in multi-agent settings.

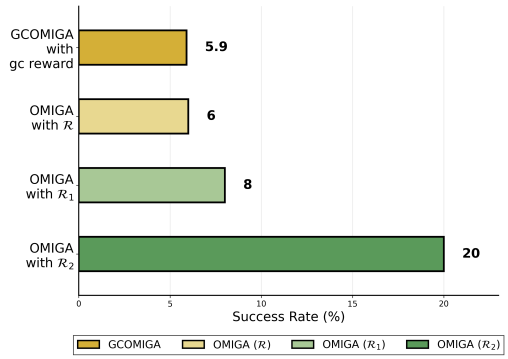


Figure 5. **Results of OMIGA under different rewards.**

Solving Sparse Reward Problem in Offline MARL. The analysis of OMIGA [34] and OMAR [21] reveals that existing offline MARL fail to overcome the challenge of sparse rewards. Since the dense rewards in real-world tasks (especially in multi-agent tasks) are difficult to design, we invite researchers to design more offline MARL or goal-conditioned offline MARL algorithms to address this issue.

7. Conclusion

In this paper, we present the first framework for goal-conditioned offline MARL, extending OGCRL to multi-agent settings under both fully decentralized and CTDE paradigms. By introducing structured goal factorization and unified goal conditioning, our approach enables robust coordination without handcrafted rewards. To support systematic evaluation, we propose MangoBench, the first multi-goal benchmark for offline MARL, encompassing diverse multi-agent setting, challenging cooperative tasks, standardized RL datasets and appropriate goal-related reward design. Experiments show that our baselines achieve strong performance even under sparse rewards, while revealing open challenges for future research. We believe that MangoBench fosters the development of general-purpose behaviors in goal-conditioned offline MARL.

8. Acknowledgement

This work was partially supported by the National Natural Science Foundation of China (No. 62372491), the Guangdong S&T Programme (No. 2025B0101130003), the Special Financial Grant from China Postdoctoral Science Foundation (No. 2025T180433), and the Science and Technology Planning Project of Key Laboratory of Advanced IntelliSense Technology, Guangdong Science and Technology Department (2023B1212060024). The experimental and computational work in this research run on the Huawei Cloud AI Compute Service. We appreciate the stable compute supply from this platform.

References

[1] Matteo Bettini, Ryan Kortvelesy, Jan Blumenkamp, and Amanda Prorok. Vmas: A vectorized multi-agent simulator for collective robot learning. In *International Symposium on Distributed Autonomous Robotic Systems*, pages 42–56. Springer, 2022. 2, 3, 6

[2] Matteo Bettini, Amanda Prorok, and Vincent Moens. Benchmarl: Benchmarking multi-agent reinforcement learning. *Journal of Machine Learning Research*, 25(217):1–10, 2024. 2

[3] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[4] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44 (10-11):1684–1704, 2025. 5, 6, 7

[5] Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020. 1, 6

[6] Tianchen Deng, Guole Shen, Chen Xun, Shenghai Yuan, Tongxin Jin, Hongming Shen, Yanbo Wang, Jingchuan Wang, Hesheng Wang, Danwei Wang, et al. Mne-slam: Multi-agent neural slam for mobile robots. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1485–1494, 2025. 1

[7] Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Foerster, and Shimon Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:37567–37593, 2023. 2, 3, 6

[8] Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2022. Curran Associates Inc. 4

[9] Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022. 8

[10] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020. 1

[11] Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Manon Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. In *International Conference on Learning Representations*, 2021. 4

[12] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers 16*, pages 66–83. Springer, 2017. 2, 3, 6

[13] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. 8

[14] Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, et al. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4501–4510, 2020. 2

[15] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994. 3

[16] Boyin Liu, Zhiqiang Pu, Yi Pan, Jianqiang Yi, Yanyan Liang, and Du Zhang. Lazy agents: A new perspective on solving sparse reward problem in multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 21937–21950. PMLR, 2023. 8

[17] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6382–6393, Red Hook, NY, USA, 2017. Curran Associates Inc. 1, 2, 3, 6

[18] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on Robot Learning*, pages 1113–1132. Pmlr, 2020. 4

[19] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[20] Takayuki Osa and Tatsuya Harada. Robustifying a policy in multi-agent rl with diverse cooperative behaviors and adversarial style sampling for assistive tasks. In *2024 IEEE International Conference on Robotics and Automation*, pages 15158–15164. IEEE, 2024. 1

[21] Ling Pan, Longbo Huang, Tengyu Ma, and Huazhe Xu. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification. In *International Conference on Machine Learning*, pages 17221–17237. PMLR, 2022. 1, 4, 8

[22] Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: Offline goal-conditioned rl with latent states as actions. *Advances in Neural Information Processing Systems*, 36:34866–34891, 2023. 1, 4, 6, 8

[23] Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2, 4, 5, 6, 8

[24] Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Facmac: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems*, 34:12208–12221, 2021. 2, 3, 6

[25] Peng Peng, Liang Pang, Yufeng Yuan, and Chao Gao. Continual match based training in pommerman: Technical report. *arXiv preprint arXiv:1812.07297*, 2018. 2, 3, 6

[26] Yiran Qin, Li Kang, Xiufeng Song, Zhenfei Yin, Xiaohong Liu, Xihui Liu, Ruimao Zhang, and Lei Bai. Robofactory: Exploring embodied agent collaboration with compositional constraints. *arXiv preprint arXiv:2503.16408*, 2025. 5, 7

[27] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019. 2

[28] 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. 1

[29] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth International Conference on Machine Learning*, pages 330–337, 1993. 1

[30] Wei-Cheng Tseng, Tsun-Hsuan Johnson Wang, Yen-Chen Lin, and Phillip Isola. Offline multi-agent reinforcement learning with knowledge distillation. *Advances in Neural Information Processing Systems*, 35:226–237, 2022. 1

[31] Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. Benchmarking model-based reinforcement learning. *arXiv preprint arXiv:1907.02057*, 2019. 1

[32] Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning*, pages 36411–36430. PMLR, 2023. 8

[33] Weizheng Wang, Le Mao, Ruiqi Wang, and Byung-Cheol Min. Multi-robot cooperative socially-aware navigation using multi-agent reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation*, pages 12353–12360. IEEE, 2024. 1

[34] Xiangsen Wang, Haoran Xu, Yinan Zheng, and Xianyuan Zhan. Offline multi-agent reinforcement learning with implicit global-to-local value regularization. *Advances in Neural Information Processing Systems*, 36:52413–52429, 2023. 1, 4, 8

[35] S Whiteson, M Samvelyan, T Rashid, CS De Witt, G Farquhar, N Nardelli, TGJ Rudner, CM Hung, PHS Torr, and J Foerster. The starcraft multi-agent challenge. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pages 2186–2188, 2019. 1

[36] Rui Yang, Han Zhong, Jiawei Xu, Amy Zhang, Chongjie Zhang, Lei Han, and Tong Zhang. Towards robust offline reinforcement learning under diverse data corruption. In *The Twelfth International Conference on Learning Representations*, 2024. 6

[37] Yifu Yuan, Hongyao Tang, Cong Wang, Yan Zheng, and Jianye Hao. Ed2: environment dynamics decomposition world models for continuous control. *Visual Intelligence*, 3: 23, 2025. 2

[38] Vladimir Yugay, Theo Gevers, and Martin R. Oswald. Magic-slam: Multi-agent gaussian globally consistent slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6741–6750, 2025. 1

[39] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384, 2021. 1