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

Supplementary Material

1. Baseline Algorithms

1.1. Goal-Conditioned Multi-Agent Behavior Cloning (GCMBC)

We propose goal-conditioned multi-agent behavior cloning (GCMBC), a fully decentralized extension of GCBC [3, 5] that serves as a fundamental baseline capturing the behavioral competence present in the dataset. Following the GCBC paradigm, GCMBC trains a goal-conditioned policy by sampling future states from the same trajectory as relabeled goals, enabling the policy to learn a general-purpose capability of reaching any achievable goal state from a given observation [15]. In the fully-decentralized setting with N agents, each agent receives a local observation $o_i \in \mathcal{S}$, is assigned a corresponding relabeled local goal g_i, and then executes an action a_i. Each agent maintains its own decentralized policy network π_i, and these policies are trained to predict actions conditioned on their respective observations and goals, according to the following objective:

$$J_{\pi_i}(\theta_i) = \mathbb{E}_{(o_i,a_i)\sim\mathcal{D},\, g_i\sim p^{\mathcal{D}}_{\text{traj}}(g_i|\tau)}\big[\log \pi_i(a_i \mid o_i, g_i)\big], \tag{1}$$

where $\mathcal{D}$ means the dataset, τ means the trajectory being sampled.

1.2. Independent Contrastive Reinforcement Learning (ICRL)

Independent Contrastive Reinforcement Learning (ICRL), a fully decentralized multi-agent extension of CRL [1, 9], which estimates the value function by contrastive learning. The objective of value function is as follows:

$$\begin{aligned} J(f_i) = \mathbb{E}_{(o_i,a_i)\sim\mathcal{D},\, g_i\sim p^{\mathcal{D}}_{\text{geom}}(g_i|o_i,a_i),\, g_i^-\sim p^{\mathcal{D}}_{\text{rand}}(g_i)} \\ \big[\log\sigma\big(f_i(o_i,a_i,g_i)\big) + \log\big(1-\sigma\big(f_i(o_i,a_i,g_i^-)\big)\big)\big], \end{aligned} \tag{2}$$

where $p^{\mathcal{D}}_{\text{geom}}$ means the truncated geometric future state distribution [9], $p^{\mathcal{D}}_{\text{rand}}$ means the uniform state distribution over the dataset $\mathcal{D}$, and σ denotes the sigmoid function. f_i is parameterized as follows:

$$f_i(o_i,a_i,g_i) = \frac{\phi(o_i,a_i)^\top \psi(g_i)}{\sqrt{d}}, \tag{3}$$

with $\phi : \mathcal{O}_i \times \mathcal{A}_i \to \mathcal{Z}_i = \mathbb{R}^d$ and $\psi : \mathcal{O}_i \to \mathcal{Z}_i = \mathbb{R}^d$.

This contrastive objective Eq. 2 encourages f_i to assign higher values to observation-goal pairs that are genuinely reachable, while pushing down values for randomly sampled, mismatched goals, allowing ICRL to learn a goal-reaching value function through contrastive discrimination.

We adopt advantage-weighted regression (AWR) [10, 11] to extract a policy from the learned value functions for manipulation tasks. Although prior work shows that behavior-constrained DDPG (DDPG+BC) [2] can generally outperform AWR in offline control [8], DDPG+BC is considerably more sensitive to the parameter α, often requiring careful tuning to obtain stable performance. In contrast, AWR provides a more robust, stable, and hyperparameter-insensitive policy extraction mechanism, making it particularly suitable for establishing a fundamental and reproducible baseline for the new environment. Note that future work may further enhance results by incorporating DDPG+BC or other stronger policy extraction techniques. The policy extraction function based on AWR is as follows:

$$\begin{aligned} J_{\text{AWR}}(\pi_i) = \mathbb{E}_{(o_i,a_i,o_i')\sim\mathcal{D},\, g_i\sim p^{\mathcal{D}}_{\text{mixed}}(g_i|o_i)} \\ \big[e^{\alpha(f_i(o_i,a_i,g_i) - f_i^V(o_i,g_i))}\log\pi(a_i \mid o_i, g_i)\big], \end{aligned} \tag{4}$$

where p_{mixed} means a mixture of these four goal-sampling distributions following [9], and f_i^V is an additional contrastive value function only based on local observation and goal:

$$\begin{aligned} J(f_i^V) = \mathbb{E}_{o_i\sim\mathcal{D},\, g_i\sim p^{\mathcal{D}}_{\text{geom}}(g_i|o_i),\, g_i^-\sim p^{\mathcal{D}}_{\text{rand}}(g_i)} \\ \big[\log\sigma\big(f_i^V(o_i,g_i)\big) + \log\big(1-\sigma\big(f_i^V(o_i,g_i^-)\big)\big)\big]. \end{aligned} \tag{5}$$

1.3. Independent Hierarchical Implicit Q-Learning (IHIQL)

To achieve better performance in multi-agent offline reinforcement learning, we extend the state-of-the-art HIQL [7, 9] into the goal-conditioned offline MARL setting, introducing IHIQL (fully decentralized) which leverages hierarchical policies to enhance robustness against sparse rewards and improve long-horizon reasoning. IHIQL learns the value function following Eq. 6 and extracts each agent's hierarchical policy from the same value function learned by Eq. 6.

$$\begin{aligned} L_V(\theta_i^V) = \mathbb{E}_{(o_i,o_i')\sim\mathcal{D},\, g_i\sim p^{\mathcal{D}}_{\text{mixed}}(g|\tau)}\Big[\mathcal{L}_\tau^2\big(r(o_i,g_i) \\ + \gamma V_{\bar\theta_i^V}(o_i',g_i) - V_{\theta_i^V}(o_i,g_i)\big)\Big], \end{aligned} \tag{6}$$

where $r(o_i,g_i) = \mathbf{1}_{\{g_i\}}(o_i) - 1$ denotes the goal-conditioned reward, which means if the agent reaches the

goal, its reward is 0, or its reward is -1. $\bar{\theta}_i^V$ denotes the parameters of the target V network. $\mathcal{L}_\tau^2$ is the expectile loss with a parameter $\tau \in [0.5, 1)$ [7, 9]. Note that we denote $V_i(o_i, g_i) = V_i(o_i, \phi_i(o_i, g_i))$, where $\phi_i : \mathcal{O}_i \times \mathcal{O}_i \to \mathcal{Z}_i$ serves as a goal representation function for each agent.

The high-level policy $\pi_i^{\theta_i^h}(z_i^{t+k} \mid o_i^t, g_i)$ produces optimal k-step subgoals o_i^{t+k} for each agent, and uses $z_i^{t+k} = \phi(o_i^t, o_i^{t+k})$ as compact representations of subgoals [7, 9]. The low-level policy $\pi_i^{\theta_i^\ell}(a_i^t \mid o_i^t, z_i^{t+k})$ produces the optimal actions a_i^t of each agent, taking the current observation o_i^t and the subgoal z_i^{t+k} as input. The objective of high-level policy is as follows:

$$J_{\pi_i^h}(\theta_i^h) = \mathbb{E}_{(o_i^t, o_i^{t+k}, g_i)}\big[\exp\Big(\beta \cdot \tilde{A}_i^h(o_i^t, o_i^{t+k}, g_i)\Big) \cdot \log \pi_i^{\theta_i^h}(z_i^{t+k} \mid o_i^t, g_i)\big], \tag{7}$$

The objective of low-level policy is as follows:

$$J_{\pi_i^\ell}(\theta_i^\ell) = \mathbb{E}_{(o_i^t, a_i^t, o_i^{t+1}, o_i^{t+k})}\big[\exp\Big(\beta \cdot \tilde{A}_i^\ell(o_i^t, a_i^t, o_i^{t+k})\Big) \cdot \log \pi_i^{\theta_i^\ell}(a_i^t \mid o_i^t, z_i^{t+k})\big], \tag{8}$$

where β is the hyperparameter, $\tilde{A}_i^h(o_i^t, o_i^{t+k}, g_i) = V_{\theta_i^V}(o_i^{t+k}, g_i) - V_{\theta_i^V}(o_i^t, g_i)$ and $\tilde{A}_i^l(o_i^t, o_i^{t+k}, g_i) = V_{\theta_i^V}(o_i^{t+1}, o_i^{t+k}) - V_{\theta_i^V}(o_i^t, o_i^{t+k})$ means the temporal difference advantage based on the value function. The high-level policy aims to produce a subgoal to reach the goal, and the low-level policy aims to produce an action to reach the subgoal. This hierarchical policies overcome the noise in value function result from the sparse reward.

1.4. HIQL under Centralized Training with Decentralized Execution Paradigm (HIQL-CTDE)

To further evaluate different training paradigms, we extend HIQL [7, 9] into the centralized training with decentralized execution (CTDE) paradigm, where training leverages centralized value learning while execution remains fully decentralized. HIQL-CTDE maintains the same hierarchical structure as the decentralized version, but the value function is now trained with global observations and global subgoals, allowing the critic to capture cross-agent dependencies and reduce non-stationarity during offline training. The centralized value function is defined as:

$$L_V(\theta_i^V) = \mathbb{E}_{(\mathbf{o}, \mathbf{o}') \sim \mathcal{D},\ \mathbf{g} \sim p_{\text{mixed}}^{\mathcal{D}}(\mathbf{g} \mid \tau)}\Big[\mathcal{L}_\tau^2\big(r(\mathbf{o}, \mathbf{g}) + \gamma V_{\bar{\theta}_i^V}(\mathbf{o}', \mathbf{g}) - V_{\theta_i^V}(\mathbf{o}, \mathbf{g})\big)\Big], \tag{9}$$

where $\mathbf{o} = (o_1, \ldots, o_N)$, $\mathbf{g} = (g_1, \ldots, g_N)$ are the global observations and goals of all N agents. The decentralized policy extraction functions are the same as Eq 7 and Eq. 8. Note that $V_i(\mathbf{o}, \mathbf{g}) = V_i(\mathbf{o}, \phi_i^{\text{global}}(\mathbf{o}, \mathbf{g}))$, where $\phi_i^{\text{global}} : \mathcal{S} \times \mathcal{S} \to \mathcal{Z}$. Since the centralized value function and the decentralized policy extraction function all need the goal representation network to encode the goal, we separately learn the global and local goal representation network, the local goal representation function is $\phi_i^{\text{local}}(o_i, g_i)$, where $\phi_i^{\text{local}} : \mathcal{O}_i \times \mathcal{O}_i \to \mathcal{Z}_i$.

1.5. Goal Conditioned Variant of OMIGA (GCOMIGA)

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 by goal relabeling and randomly goal sampling, the goal-conditioned variants of OMIGA [14]. GCOMIGA follows the CTDE paradigm and uses value decomposition methods [12, 13] to decompose the global value function into a linear combination of local value functions:

$$\begin{aligned} Q_{\text{tot}}(\mathbf{o}, \mathbf{a}, \mathbf{g}) &= \sum_i w_i(\mathbf{o}) Q_i(o_i, a_i, g_i) + b(\mathbf{o}), \\ V_{\text{tot}}(\mathbf{o}, \mathbf{g}) &= \sum_i w_i(\mathbf{o}) V_i(o_i, g_i) + b(\mathbf{o}), \\ w_i &\geq 0, \quad \forall i = 1, \ldots, n. \end{aligned} \tag{10}$$

where $\mathbf{o} = (o_1, \ldots, o_N)$, $\mathbf{a} = (a_1, \ldots, a_N)$, $\mathbf{g} = (g_1, \ldots, g_N)$ are the global observations, global actions and global goals of all N agents. GCOMIGA extracts the local goal conditioned policy by:

$$J_(\pi_i) = \mathbb{E}_{(o_i, a_i) \sim \mathcal{D},\ g_i \sim p_{\text{mixed}}^{\mathcal{D}}(g \mid \tau)}\big[\exp\Big(\frac{w_i(o)}{\alpha}(Q_i(o_i, a_i, g_i) - V_i(o_i, g_i))\Big) \cdot \log \pi_i(a_i | o_i, g_i)\big]. \tag{11}$$

1.6. Goal Conditioned Variant of OMAR (GCOMAR)

Through goal relabeling and randomly goal sampling, we further designed GCOMAR, the goal conditioned variant of OMAR [6] following fully decentralized paradigm. The decentralized critics are conditioned on the local goals g_i and trained by Conservative Q-Learning algorithm [4]. The goal conditioned policy objective is as follows:

$$J_(\pi_i) = \mathbb{E}_{(o_i, a_i) \sim \mathcal{D},\ g_i \sim p_{\text{mixed}}^{\mathcal{D}}(g | \tau)}\big[(1 - \beta) Q_i\big(o_i, g_i, \pi_i(o_i, g_i)\big) + \beta\big(\pi_i(o_i, g_i) - \hat{a}_i\big)^2\big], \tag{12}$$

where $\beta \in [0, 1]$ means the parameter and $\hat{a}_i$ denotes the action generated by the zeroth-order optimizer [6].

2. Environmental Details

2.1. Observation Space, Action Space

We provide an overview of the multi-agent local observation and action spaces for each task in Table 1, and present the detailed specifications of action space for each agent in Table 2 and Table 3. Note that the global observation space is the union of the local observation spaces of all agents of the same type. Therefore, for simplicity, we uniformly refer to the local observation space as the observation space in the following.

Table 1. Dimension of Multi-Agent Observation and Action Space.

Environment Type	Observation Dim.	Action Dim.
antmaze 2×4	21	4
antmaze $2 \times 4d$	21	4
antmaze 4×2	17	2
antsoccer 2×4	34	4
antsoccer $2 \times 4d$	34	4
antsoccer 4×2	30	2
bimanual manipulation	$64 \times 64 \times 3$	8

Table 2. Multi-Agent Action Space of Ant. Since action spaces of the corresponding agents in AntMaze and AntSoccer are the same, we denote them uniformly as **Ant**.

Agent Type	Agent ID	Action	Actuation Site
Ant 2×4	Agent 0	Torque	hip 1 (front left leg)
		Torque	ankle 1 (front left leg)
		Torque	hip 2 (front right leg)
		Torque	ankle 2 (front right leg)
	Agent 1	Torque	hip 4 (right back leg)
		Torque	ankle 4 (right back leg)
		Torque	hip 3 (back leg)
		Torque	ankle 3 (back leg)
Ant $2 \times 4d$	Agent 0	Torque	hip 1 (front left leg)
		Torque	ankle 1 (front left leg)
		Torque	hip 4 (right back leg)
		Torque	ankle 4 (right back leg)
	Agent 1	Torque	hip 2 (front right leg)
		Torque	ankle 2 (front right leg)
		Torque	hip 3 (back leg)
		Torque	ankle 3 (back leg)
Ant 4×2	Agent 0	Torque	hip 1 (front left leg)
		Torque	ankle 1 (front left leg)
	Agent 1	Torque	hip 2 (front right leg)
		Torque	ankle 2 (front right leg)
	Agent 2	Torque	hip 3 (back leg)
		Torque	ankle 3 (back leg)
	Agent 3	Torque	hip 4 (right back leg)
		Torque	ankle 4 (right back leg)

Table 3. Multi-Agent Action Space of Bimanual Manipulation

Agent ID	Dimension	Actuation Site
Agent 0	7	joint
	1	gripper
Agent 1	7	joint
	1	gripper

Given the high dimensionality of the observation space for each agent in the AntMaze and AntSoccer environments, we list the shared observation components among multiple agents in Table 4, and detail the specific components of each agent in AntMaze and AntSoccer in Table 5.

Table 4. Multi-Agent Shared Observation Components in AntMaze and AntSoccer.

Environment Type	Observation Type	Observation
AntMaze	qpos of root	x coordinate
		y coordinate
		z coordinate
		w orientation
		x orientation
		y orientation
		z orientation
	qvel of root	x coordinate velocity
		y coordinate velocity
		z coordinate velocity
		x coordinate angular
		y coordinate angular
		z coordinate angular
AntSoccer	qpos of root	x coordinate
		y coordinate
		z coordinate
		w orientation
		x orientation
		y orientation
		z orientation
	qpos of ball	x coordinate
		y coordinate
		z coordinate
		w orientation
		x orientation
		y orientation
		z orientation
	qvel of root	x coordinate velocity
		y coordinate velocity
		z coordinate velocity
		x coordinate angular
		y coordinate angular
		z coordinate angular
	qvel of ball	x coordinate velocity
		y coordinate velocity
		z coordinate velocity
		x coordinate angular
		y coordinate angular
		z coordinate angular

Table 5. Multi-Agent Specific Observation Components in AntMaze and AntSoccer. Since specific observation components of the corresponding agents in AntMaze and AntSoccer are the same, we denote them uniformly as **Ant**.

Agent Type	Agent ID	Observation	Observed Body Site
Ant 2×4	Agent 0	Angle	hip 1 (front left leg)
		Angle	ankle 1 (front left leg)
		Angle	hip 2 (front right leg)
		Angle	ankle 2 (front right leg)
		Angular Velocity	hip 1 (front left leg)
		Angular Velocity	ankle 1 (front left leg)
		Angular Velocity	hip 2 (front right leg)
		Angular Velocity	ankle 2 (front right leg)
	Agent 1	Angle	hip 3 (back leg)
		Angle	ankle 3 (back leg)
		Angle	hip 4 (right back leg)
		Angle	ankle 4 (right back leg)
		Angular Velocity	hip 3 (back leg)
		Angular Velocity	ankle 3 (back leg)
		Angular Velocity	hip 4 (right back leg)
		Angular Velocity	ankle 4 (right back leg)
Ant $2 \times 4d$	Agent 0	Angle	hip 1 (front left leg)
		Angle	ankle 1 (front left leg)
		Angle	hip 4 (right back leg)
		Angle	ankle 4 (right back leg)
		Angular Velocity	hip 1 (front left leg)
		Angular Velocity	ankle 1 (front left leg)
		Angular Velocity	hip 4 (right back leg)
		Angular Velocity	ankle 4 (right back leg)
	Agent 1	Angle	hip 2 (front right leg)
		Angle	ankle 2 (front right leg)
		Angle	hip 3 (back leg)
		Angle	ankle 3 (back leg)
		Angular Velocity	hip 2 (front right leg)
		Angular Velocity	ankle 2 (front right leg)
		Angular Velocity	hip 3 (back leg)
		Angular Velocity	ankle 3 (back leg)
Ant 4×2	Agent 0	Angle	hip 1 (front left leg)
		Angle	ankle 1 (front left leg)
		Angular Velocity	hip 1 (front left leg)
		Angular Velocity	ankle 1 (front left leg)
	Agent 1	Angle	hip 2 (front right leg)
		Angle	ankle 2 (front right leg)
		Angular Velocity	hip 2 (front right leg)
		Angular Velocity	ankle 2 (front right leg)
	Agent 2	Angle	hip 3 (back leg)
		Angle	ankle 3 (back leg)
		Angular Velocity	hip 3 (back leg)
		Angular Velocity	ankle 3 (back leg)
	Agent 3	Angle	hip 4 (right back leg)
		Angle	ankle 4 (right back leg)
		Angular Velocity	hip 4 (right back leg)
		Angular Velocity	ankle 4 (right back leg)

2.2. Predefined Goals

To avoid biased evaluation, each manipulation task is associated with 5 clearly defined sequential multiple goals to facilitate consistent comparison across algorithms. We demonstrate our five predefined evaluation goals for cooperative manipulation tasks. Specifically, the goals for the *lift-barrier* task are shown in Fig. 1, and those for the *place-food* task are presented in Fig. 2. For cooperative locomotion tasks, the predefined goals follow the same design as in [9].

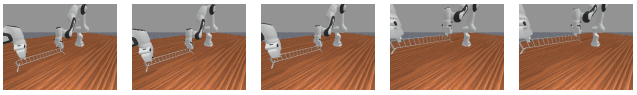

Figure 1. **Lift-Barrier Task.** Predefined evaluation goals (from left to right): descend, dock, grab, lift, and hold.

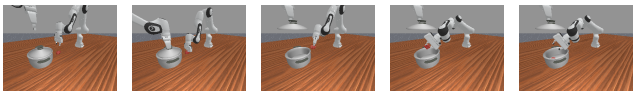

Figure 2. **Place-Food Task.** Predefined evaluation goals (from left to right): descend, grab, lift, move, and place.

3. Hyperparameters

We provide the complete hyperparameter settings for all baseline algorithms across tasks in Table 6. As AWR offers a more robust, stable, and hyperparameter-insensitive policy extraction mechanism, we adopt it uniformly to establish a strong, reproducible baseline for the manipulation environments.

4. Full Results

We report the full benchmark results on cooperative manipulation tasks and cooperative locomotion tasks in Table 7 and Table 8, respectively. The results suggest that IHIQL emerges as the state-of-the-art algorithm for most multi-agent offline goal-conditioned tasks. The results in Table 7 indicate that our goal-conditioned offline MARL baselines dramatically outperform the imitation learning method DP in both manipulation tasks while requiring only 5-7% of its training time, demonstrating effectiveness and efficiency of our proposed baseline algorithms. No single method achieves uniformly superior performance across all settings, underscoring the complexity of the problem and the significant untapped potential of this emerging research direction.

Table 6. Hyperparameters for baselines.

Hyperparameters	Value
Learning rate	0.0003
Optimizer	Adam
gradient steps	1000000 (locomotion)
gradient steps	15000 (lift-barrier)
gradient steps	38800 (place-food)
batch size	1024 (locomotion)
batch size	256 (manipulation)
discount factor γ	0.995 (antmaze-giant)
discount factor γ	0.99 (others)
target smoothing coefficient τ	0.005
hidden size	512
ICRL latent dimension	512
ICRL AWR α	3.0 (manipulation)
ICRL DDPG+BC α	0.1 (antmaze-navigate)
ICRL DDPG+BC α	0.1 (antmaze-stitch)
ICRL DDPG+BC α	0.003 (antmaze-explore)
ICRL DDPG+BC α	0.3 (antsoccer-navigate)
ICRL DDPG+BC α	0.3 (antsoccer-stitch)
IHIQL&HIQL-CTDE expectile κ	0.7
IHIQL&HIQL-CTDE subgoal k	25 (locomotion)
IHIQL&HIQL-CTDE subgoal k	10 (manipulation)
subgoal representation dimension	10
IHIQL&HIQL-CTDE AWR α	10.0 (antmaze-explore)
IHIQL&HIQL-CTDE AWR α	3.0 (others)
GCOMIGA mix network hidden size	128
GCOMIGA AWR α	10.0
GCOMAR actor rectification iters	2
GCOMAR actor rectification samples	20
GCOMAR actor rectification elites	5
GCOMAR actor coefficient	0.5 (antmaze-navigate)
GCOMAR actor coefficient	0.7 (antmaze-stitch)
GCOMAR actor coefficient	1.0 (antmaze-explore)
GCOMAR actor coefficient	1.0 (antsoccer)
GCOMAR cql α	5.0 (antmaze-navigate)
GCOMAR cql α	1.0 (antmaze-stitch)
GCOMAR cql α	1.0 (antmaze-explore)
GCOMAR cql α	5.0 (antsoccer)

Table 7. Comparison of training time and final success rate on the two cooperative manipulation tasks. All results are averaged over 100 random seeds.

Task	**Metric**	**IHIQL**	**GCMBC**	**ICRL**	**DP**
LiftBarrier	Success Rate (%)	**82**	47	56	58
	Training Time (min)	14	3	10	300
PlaceFood	Success Rate (%)	23	21	**35**	20
	Training Time (min)	34	10.4	24	300

Table 8. **Full benchmark table on locomotion tasks.** We report each method's average (binary) success rate (%) across the five test-time goals on each task. The results are averaged over 5 seeds , and we report the standard deviations after the ± sign. Numbers at or above 95% of the best value in the row are highlighted in bold.

Environment	Dataset Type	Dataset	IHIQL	ICRL	GCMBC	GCOMIGA	GCOMAR
antmaze	navigate	antmaze-medium-navigate-v0(2x4)	**95.1 ± 1.6**	**92.0 ± 2.9**	22.3 ± 1.8	5.9 ±1.0	6.5 ±0.6
		antmaze-medium-navigate-v0(2x4d)	**95.9 ± 1.1**	**95.5 ± 0.9**	25.5 ±1.4	7.7 ±2.1	5.3 ±1.6
		antmaze-medium-navigate-v0(4x2)	**94.8 ± 1.3**	90.0 ± 2.0	11.7 ± 3.3	5.7 ±1.6	8.1 ±1.6
		antmaze-large-navigate-v0(2x4)	**85.4 ± 5.3**	80.2 ± 2.4	15.0 ± 2.8	2.1 ±0.6	2.1 ±0.2
		antmaze-large-navigate-v0(2x4d)	**92.2 ± 1.9**	80.8 ± 3.3	21.2 ± 2.4	1.7 ±0.4	6.7 ±1.6
		antmaze-large-navigate-v0(4x2)	**87.1 ± 2.3**	72.0 ± 0.8	3.0 ± 1.3	1.3 ±0.2	2.3 ±0.9
		antmaze-giant-navigate-v0(2x4)	**57.3 ± 2.1**	19.8 ± 3.7	0.0 ± 0.0	0.0 ±0.0	0.0 ±0.0
		antmaze-giant-navigate-v0(2x4d)	**50.3 ± 2.9**	14.9 ± 2.3	0.1 ± 0.2	0.0 ±0.0	0.2 ±0.2
		antmaze-giant-navigate-v0(4x2)	**35.5 ± 8.3**	5.8 ± 1.6	0.0 ± 0.0	0.0 ±0.0	0.2 ±0.2
		antmaze-teleport-navigate-v0(2x4)	46.8 ± 1.7	**50.6 ± 2.1**	24.3 ± 2.9	10.0 ±2.1	5.9 ±1.2
		antmaze-teleport-navigate-v0(2x4d)	47.2 ± 2.4	**51.4 ± 1.4**	26.8 ± 4.3	10.8 ±3.0	5.0 ±1.3
		antmaze-teleport-navigate-v0(4x2)	**48.7 ± 2.4**	44.9 ± 1.7	15.1 ± 1.1	14.8 ±4.0	2.9 ±0.3
	stitch	antmaze-medium-stitch-v0(2x4)	**94.3 ± 1.4**	64.8 ± 3.7	43.7 ± 1.4	3.4 ±1.1	19.3 ±1.3
		antmaze-medium-stitch-v0(2x4d)	**93.9 ± 2.5**	62.9 ± 5.1	48.4 ± 1.2	4.4 ±1.3	10.3 ±3.5
		antmaze-medium-stitch-v0(4x2)	**93.3 ± 0.9**	60.7 ± 4.0	44.3 ± 1.4	7.1 ±3.4	18.5 ±2.4
		antmaze-large-stitch-v0(2x4)	**80.2 ± 2.1**	8.5 ± 1.8	0.6 ± 0.6	0.5 ±0.5	5.0 ±2.6
		antmaze-large-stitch-v0(2x4d)	**74.7 ± 3.4**	7.0 ± 1.2	9.8 ± 1.8	2.1 ±1.0	8.6 ±2.4
		antmaze-large-stitch-v0(4x2)	**64.5 ± 2.5**	5.8 ± 1.5	0.0 ± 0.0	0.7 ±0.5	1.9 ±1.1
		antmaze-giant-stitch-v0(2x4)	**5.1 ± 3.5**	0.0 ± 0.0	0.0 ± 0.0	0.0 ±0.0	0.0 ±0.0
		antmaze-giant-stitch-v0(2x4d)	**0.7 ± 0.7**	0.0 ± 0.0	0.0 ± 0.0	0.0 ±0.0	0.0 ±0.0
		antmaze-giant-stitch-v0(4x2)	**0.2 ± 0.3**	0.0 ± 0.0	0.0 ± 0.0	0.0 ±0.0	0.0 ±0.0
		antmaze-teleport-stitch-v0(2x4)	**37.9 ± 1.6**	33.4 ± 2.3	26.6 ± 3.8	10.7 ±2.2	5.7 ±2.9
		antmaze-teleport-stitch-v0(2x4d)	**39.2 ± 1.2**	29.8 ± 3.5	32.6 ± 1.6	3.3 ±0.4	2.9 ±1.6
		antmaze-teleport-stitch-v0(4x2)	**36.1 ± 2.7**	29.7 ± 2.7	22.5 ± 1.1	6.8 ±0.9	11.4 ±5.2
	explore	antmaze-medium-explore-v0(2x4)	**34.2 ± 6.6**	1.3 ± 0.6	1.3 ± 0.5	2.1 ±0.9	0.0 ±0.0
		antmaze-medium-explore-v0(2x4d)	**36.5 ± 9.6**	1.5 ± 1.1	3.0 ± 0.9	1.9 ±1.2	0.0 ±0.0
		antmaze-medium-explore-v0(4x2)	**24.8 ± 4.1**	0.1 ± 0.1	0.1 ± 0.2	1.8 ±0.8	0.0 ±0.0
		antmaze-large-explore-v0(2x4)	**5.4 ± 2.1**	0.0 ± 0.0	0.0 ± 0.0	0.6 ±0.2	0.0 ±0.0
		antmaze-large-explore-v0(2x4d)	**1.0 ± 1.4**	0.0 ± 0.0	0.0 ± 0.0	1.0 ±0.9	0.0 ±0.0
		antmaze-large-explore-v0(4x2)	**1.9 ± 1.8**	0.0 ± 0.0	0.0 ± 0.0	0.5 ±0.3	1.1 ±1.5
		antmaze-teleport-explore-v0(2x4)	**35.7 ± 4.8**	17.5 ± 7.5	2.6 ± 0.8	3.4 ±1.6	0.0 ±0.0
		antmaze-teleport-explore-v0(2x4d)	**34.2 ± 10.2**	13.6 ± 3.4	9.9 ± 0.7	2.9 ±0.2	0.0 ±0.0
		antmaze-teleport-explore-v0(4x2)	**36.6 ± 3.1**	13.0 ± 3.0	11.4 ± 2.1	2.5 ±0.7	0.0 ±0.0
antsoccer	navigate	antsoccer-arena-navigate-v0(2x4)	**34.8 ± 1.4**	10.6 ± 1.8	2.5 ± 0.4	0.0 ±0.0	0.0 ±0.0
		antsoccer-arena-navigate-v0(2x4d)	**37.4 ± 3.8**	12.3 ± 2.1	2.5 ± 0.8	0.0 ±0.0	0.0 ±0.0
		antsoccer-arena-navigate-v0(4x2)	**14.3 ± 1.5**	2.3 ± 1.2	0.6 ± 0.2	0.0 ±0.0	0.0 ±0.0
		antsoccer-medium-navigate-v0(2x4)	**7.1 ± 1.7**	0.5 ± 0.4	2.4 ± 0.9	0.0 ±0.0	0.0 ±0.0
		antsoccer-medium-navigate-v0(2x4d)	**8.1 ± 2.1**	1.1 ± 0.8	2.6 ± 1.0	0.0 ±0.0	0.0 ±0.0
		antsoccer-medium-navigate-v0(4x2)	**4.2 ± 4.1**	0.2 ± 0.2	0.9 ± 0.8	0.0 ±0.0	0.0 ±0.0
	stitch	antsoccer-arena-stitch-v0(2x4)	4.2 ± 1.1	0.2 ± 0.2	**13.5 ± 1.8**	0.0 ±0.0	0.0 ±0.0
		antsoccer-arena-stitch-v0(2x4d)	6.6 ± 1.3	0.2 ± 0.2	**14.6 ± 1.4**	0.0 ±0.0	0.0 ±0.0
		antsoccer-arena-stitch-v0(4x2)	1.4 ± 1.1	0.1 ± 0.2	**9.1 ± 1.6**	0.0 ±0.0	0.0 ±0.0
		antsoccer-medium-stitch-v0(2x4)	**2.0 ± 0.8**	0.0 ± 0.0	1.5 ± 0.6	0.0 ±0.0	0.0 ±0.0
		antsoccer-medium-stitch-v0(2x4d)	**3.3 ± 1.0**	0.0 ± 0.0	2.8 ± 0.7	0.0 ±0.0	0.0 ±0.0
		antsoccer-medium-stitch-v0(4x2)	**1.0 ± 0.4**	0.0 ± 0.0	1.1 ± 0.6	0.0 ±0.0	0.0 ±0.0

References

[1] 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. 1

[2] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021. 1

[3] 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. 1

[4] 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. 2

[5] 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. 1

[6] 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. 2

[7] 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, 2

[8] Seohong Park, Kevin Frans, Sergey Levine, and Aviral Kumar. Is value learning really the main bottleneck in offline rl? *Advances in Neural Information Processing Systems*, 37: 79029–79056, 2024. 1

[9] 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, 5

[10] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019. 1

[11] Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th International Conference on Machine Learning*, pages 745–750, 2007. 1

[12] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020. 2

[13] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, page 2085–2087, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems. 2

[14] 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. 2

[15] David Warde-Farley, Tom Van de Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, and Volodymyr Mnih. Unsupervised control through non-parametric discriminative rewards. In *International Conference on Learning Representations*, 2019. 1