Imitation Bootstrapped Multi-Agent Reinforcement Learning

Tony Zhang[1], Connor Mattson[1], Daniel S. Brown[1]
[1]Kahlert School of Computing, University of Utah, Salt Lake City, UT, USA
u1183156@utah.edu, c.mattson@utah.edu, daniel.s.brown@utah.edu

***Abstract*—Demonstrations are an effective way to accelerate single-agent reinforcement learning (RL), however, extending this idea to multi-agent reinforcement learning (MARL) is non-trivial: a lone human cannot realistically provide demonstrations for multiple agents simultaneously. Recent work on Round-Robin Behavior Cloning (R2BC) provides a solution by allowing a demonstrator to control a single agent at a time while the others execute their own current policies. In this work, we propose *Imitation Bootstrapped Multi-Agent Reinforcement Learning (IBMARL)*, a framework which extends IBRL to cooperative teams: at each time step every agent receives a proposal from its imitation policy and from its RL policy, and an ensemble of centralized critics selects among the per-agent combinations of the two. A gated imitation term pulls each RL actor toward the imitation policy only where the critics still prefer it, so that after training each agent acts alone. On three cooperative simulation tasks IBMARL surpasses its imitation policy 2.3–$3.7\times$ faster than demonstration-based baselines on two tasks and is the only method to surpass it on the third.**

***Index Terms*—multi-agent reinforcement learning, imitation learning, learning from demonstration, multi-robot systems**

I. Introduction

Cooperative multi-robot tasks are difficult for RL to solve: the joint-action space grows exponentially with the number of agents, credit assignment across agents is ambiguous, and rewards are often inherently sparse [13]. In single-agent RL, demonstrations reliably overcome similar obstacles—methods such as DQfD [3], DDPGfD [4], and DDPG + BC [5] seed the replay buffer or regularize the learner toward expert data, and Imitation Bootstrapped RL (IBRL) [2] goes further by keeping a behavior-cloned policy in the loop, using it to propose actions during exploration and to improve the bootstrap target. The obstacle to extending such frameworks to MARL is practicality rather than algorithm design: nearly all multi-agent imitation learning (MAIL) assumes joint demonstrations, in which an expert must simultaneously command all N agents—data a lone human operator cannot realistically provide for a physically coupled team [1], [7]. Round-Robin Behavior Cloning (R2BC) [1] resolves this by having the demonstrator cycle through each team member one at a time while the others execute their current learned policies, yielding decentralized cooperative policies from realistic single-agent demonstrations. However, R2BC inherits the classic limitations of behavior cloning: covariate shift, a performance ceiling at the demonstrator, and a prohibitive demonstration burden for the last increments of task performance.

This work combines MARL and MAIL. Assuming access to a ground-truth team reward, we ask: *Can policies trained from realistic single-agent demonstrations bootstrap MARL?* Our contributions are: (i) We formulate **IBMARL**, a generalization of IBRL to cooperative Dec-POMDPs in which the imitation learned (IL) policy is a set of N decentralized R2BC policies and a centralized critic ensemble arbitrates among per-agent combinations of IL and RL proposals. (ii) We introduce a **gated imitation term** that pulls each RL actor toward the IL proposal in proportion to the critics' preference for it, so that the actors alone match the executed policy and execution is decentralized. (iii) On three cooperative tasks in VMAS [12], IBMARL surpasses its R2BC initialization 2.3–$3.7\times$ faster than demonstration-based baselines on Navigation, $2.8\times$ faster on the physically coupled Buzzwire task, and is the only method to surpass it on Transport.

Related work. Closest to our setting is PegMARL [14], which shapes a sparse MARL reward from single-agent demonstrations given with no other agents present. IBMARL's imitation prior is instead trained from R2BC demonstrations collected online *in the presence of* teammates, and acts through action proposals, bootstrap targets and a critic-gated pull on the actor rather than reward shaping or a scheduled behavior-cloning penalty [5], [6].

II. Problem Formulation

Following R2BC [1], we model cooperative multi-robot learning as a decentralized partially observable Markov decision process (Dec-POMDP) [13] with N agents, a *shared* reward, discount factor γ and a fixed horizon T; agent i observes $o_i \in \Omega_i$ and selects $a_i \in \mathcal{A}_i$. We utilize two products of a R2BC training session: (i) a *frozen* decentralized imitation policy $\mu_\psi = (\mu_{\psi_1}, \ldots, \mu_{\psi_N})$, $\mu_{\psi_i} : \Omega_i \to \mathcal{A}_i$, the *IL policy*, trained by a demonstrator who cycles round-robin through the team, controlling one agent while the rest execute their current policies; and (ii) its demonstration set $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$ of per-agent transitions. The objective is to learn decentralized RL policies $\pi_\theta = (\pi_{\theta_1}, \ldots, \pi_{\theta_N})$ that surpass the IL policy with as little online environment interaction as possible.

We instantiate the problem on the *Navigation*, *Buzzwire* and *Transport* tasks of the Vectorized Multi-Agent Simulator (VMAS) [12] (Fig. 2). In *Navigation*, N=3 agents and their assigned goals spawn at random in a bounded 2-D workspace, each agent observing its own state, its goal's relative position, and a LiDAR scan of its teammates. In *Buzzwire*, N=2 agents are connected to a ball and must thread it through a narrow

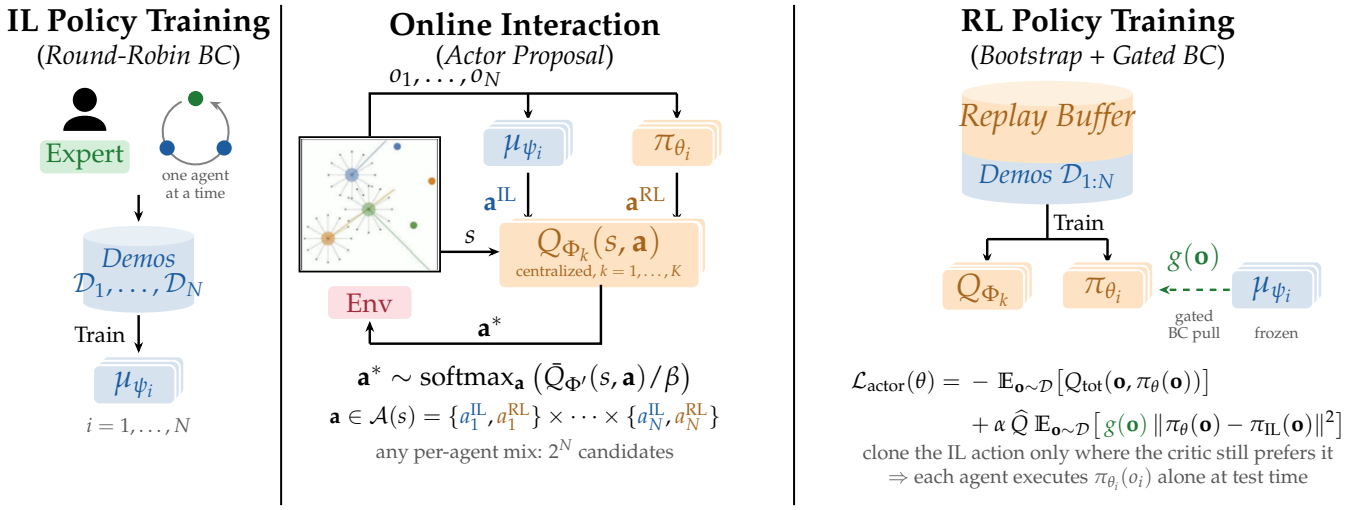


Fig. 1. IBMARL. Left: R2BC trains the decentralized IL policy μ_ψ from single-agent demonstrations. Middle: the agents' IL and RL proposals form 2^N joint candidates, scored by the centralized target critics and sampled by the arbiter. Right: the critics bootstrap through the arbiter and the actor loss adds the gated imitation term (π_{IL} denotes μ_ψ).

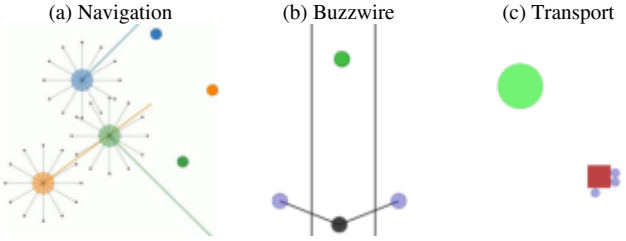


Fig. 2. Experimental Domains. (a) Navigation: three agents (LiDAR rays shown) reach their color-matched goals; adapted from [1]. (b) Buzzwire: two agents linked to a ball carry it along the corridor to the goal. (c) Transport: three agents push the package (red) onto the goal.

corridor to a goal without touching the walls; each agent observes its own position, velocity and relative position to the goal location. In *Transport*, N=3 agents must push a package far heavier than themselves onto a goal; each agent observes its own state and the package's state relative to the goal and to itself. In all three, agents act through continuous velocity commands over a fixed horizon T.

The native VMAS rewards of Navigation and Buzzwire are dense, which masks the exploration problem that demonstrations are meant to solve. We therefore replace them with a sparse variant, $r_{i,t} = \tilde{r}_{i,t}$ if $d_{i,t} < \delta$ and -1 otherwise, where $d_{i,t}$ is agent i's distance to its goal (Navigation) or the ball's distance to the goal, shared by both agents (Buzzwire), and $\tilde{r}_{i,t}$ is the small native VMAS shaping reward. Episodes run the full horizon, so an agent's return counts the steps spent outside the goal region, bounded below by $-T$. Navigation uses $T = 100$ and $\delta = 0.4$; Buzzwire uses $T = 200$ and $\delta = 0.1$, below the corridor half-width. Transport keeps its native dense reward (the per-step decrease of the package's distance to the goal) with $T = 500$; its difficulty is coordination, since one agent cannot move the package.

III. Imitation Bootstrapped MARL

IBMARL extends IBRL [2] to cooperative teams under centralized training with decentralized execution [8] (Fig. 1). The IL policy proposes actions during exploration and for the bootstrap target, pre-populates the replay buffer, and pulls the RL actors toward its actions through a gated imitation term; a critic-scored *arbiter*, not a schedule, decides when it is followed and gates the term.

A. Proposals and the Arbiter

Given a team observation $\mathbf{o} = (o_1, \ldots, o_N)$, each agent forms two candidate actions,

$$a_i^{\mathrm{RL}} = \pi_{\theta_i}(o_i) + \varepsilon_i, \qquad a_i^{\mathrm{IL}} = \mu_{\psi_i}(o_i) + \varepsilon_i', \tag{1}$$

with i.i.d. exploration noise $\varepsilon, \varepsilon' \sim \mathcal{N}(0, \sigma^2 I)$ on both proposals so that arbitration compares like with like. The arbiter's candidate set is the product of the per-agent choices, $\mathcal{A}(\mathbf{o}) = \{a_1^{\mathrm{IL}}, a_1^{\mathrm{RL}}\} \times \cdots \times \{a_N^{\mathrm{IL}}, a_N^{\mathrm{RL}}\}$, that is 2^N joint actions, so it may follow the IL policy for one agent and the RL policy for another; restricting it to the two pure joint actions is ablated in Sec. IV. Each agent maintains an ensemble of E=3 centralized critics $Q_{\Phi_i}^k(\mathbf{o}, \mathbf{a})$ with target copies $\bar{Q}_{\Phi_i}^k$. Following REDQ [11], the arbiter samples two target members K, $|K| = 2$, and scores each candidate $\mathbf{a} \in \mathcal{A}(\mathbf{o})$ by

$$\bar{Q}(\mathbf{o}, \mathbf{a}) = \sum_{i=1}^{N} \min_{k \in K} \bar{Q}_{\Phi_i}^k(\mathbf{o}, \mathbf{a}). \tag{2}$$

The executed action is sampled from a Boltzmann distribution over the candidates, $\mathbf{a}^* \sim \mathrm{softmax}_{\mathbf{a}\in\mathcal{A}(\mathbf{o})} \left(\bar{Q}(\mathbf{o}, \mathbf{a})/\beta\right)$ with temperature $\beta = 0.05$; evaluation takes the argmax.

B. Learning

Critics are trained with a TD(0) target that bootstraps through the arbiter: at the next observation $\mathbf{o}'$, the arbiter is applied to noise-free proposals from the target actors $\bar{\pi}_\theta$

and the IL policy, yielding a bootstrap action $\mathbf{a}'^{*}$, and every ensemble member regresses to

$$y_i = r_i + \gamma\,(1 - \mathbb{1}[\text{terminal}])\,\min_{k\in K} \bar{Q}^{k}_{\Phi_i}(\mathbf{o}', \mathbf{a}'^{*}), \qquad (3)$$

so members differ only by initialization.

Gated imitation term. The deterministic policy gradient of MADDPG [8], [9] maximizes $Q_{\text{tot}}(\mathbf{o}, \pi_\theta(\mathbf{o})) = \sum_i Q^{1}_{\Phi_i}(\mathbf{o}, \pi_\theta(\mathbf{o}))$ and can only climb the critic locally around the actor's own action. When the arbiter executes the IL policy, its successes raise the critics' value of the IL actions, but nothing moves the actor toward them: the RL actors lag the policy IBMARL actually executes (Sec. IV). We therefore train the actors with

$$\mathcal{L}(\theta) = -\mathbb{E}\big[Q_{\text{tot}}(\mathbf{o}, \pi_\theta(\mathbf{o}))\big] + \alpha\hat{Q}\,\mathbb{E}\big[g(\mathbf{o})\|\pi_\theta(\mathbf{o}) - \mu_\psi(\mathbf{o})\|_2^2\big], \qquad (4)$$

where the gate $g(\mathbf{o})$ is the arbiter's own preference for the IL policy at $\mathbf{o}$, the Boltzmann (sigmoid) probability of the pure IL joint action over the actor's noise-free joint action under the scoring of (2),

$$g(\mathbf{o}) = \sigma\big(\big[\bar{Q}(\mathbf{o}, \mu_\psi(\mathbf{o})) - \bar{Q}(\mathbf{o}, \pi_\theta(\mathbf{o}))\big]/\beta\big). \qquad (5)$$

$\hat{Q}$ is the detached minibatch mean of $|Q_{\text{tot}}|$, so that $\alpha = 0.4$ is dimensionless and transfers across tasks with different return scales, as in TD3+BC [6]. The term clones the IL action only where the critics still prefer it and switches itself off, state by state, as the actor overtakes the IL policy; it needs no annealing schedule and acts on the states the learner visits rather than on the demonstration set. Because the actors are trained toward the executed policy, each agent executes $\pi_{\theta_i}(o_i)$ alone at test time, without the critics or the IL policy.

C. Data and Implementation

A single replay buffer per team member is pre-loaded with all demonstration transitions in $\mathcal{D}$ and sampled uniformly, as in IBRL: online transitions gradually dilute the demonstrations. Updates begin once the buffer holds W online steps; on Buzzwire, where RL alone never reaches the goal, the IL policy's noised proposal is executed unconditionally during this warm-up, as in IBRL, so that the first arbitrated decisions follow observed successes. Actors are decentralized per-agent MLPs (two hidden layers of 256 units, $\tanh$); critics are centralized over the team's observations and actions. Training alternates F environment steps with G gradient steps at batch size 1024 (Adam, 10^{-4}, $\gamma = 0.99$, Polyak $\tau = 10^{-3}$, $\sigma = 0.1$, gradient-norm clipping at 1.0); $F/G/W$, replay capacity and budget are $1000/500/5{\times}10^3$, $2{\times}10^5$, $3{\times}10^5$ on Navigation, $8000/2000/4{\times}10^4$, $4{\times}10^5$, $6{\times}10^5$ on Buzzwire and $5000/500/5{\times}10^3$, 10^6, $4{\times}10^6$ on Transport.

IV. Experiments

IL policies and demonstrations. On each task the IL policy is the decentralized policy of one R2BC training session (one small MLP per agent), trained from demonstrations given through R2BC's round-robin protocol by a synthetic demonstrator: a MAPPO policy trained with the dense reward on Navigation, and scripted controllers that hold each agent at a fixed lateral offset while driving toward the goal on Buzzwire and line the agents up behind the package on Transport. The budget is intentionally small: 24 episodes on Navigation, 12 on Buzzwire, and 204 on Transport, where the demonstrator delivers the package in only 10% of its episodes. The frozen IL policies attain -48.7, -137.5 and 15.7. All demonstration-based methods consume the same transitions and IL policy.

Baselines share IBMARL's architectures, DDPG-style updates [9], exploration noise and hyperparameters, with a single critic per agent, so that *how* demonstrations are used is the only variable. **MADDPG** [8] is the demonstration-free reference. **RLfD** [4] keeps the demonstrations in a separate replay buffer from which each minibatch draws a fraction annealed linearly from 50% to 0% over training. **RFT** adds a behavior-cloning regularizer $\alpha\,\lambda(t)\,\|\pi_\theta(o^{\mathcal{D}}) - a^{\mathcal{D}}\|_2^2$ to the actor loss [5], [10], with $\alpha = 0.1$ and $\lambda(t)$ annealed from 1 to 0 over the first half of training (the whole run on Transport): the pull of (4) without the gate, on demonstration states and a fixed schedule.

Ablations. *No penalty* removes the gated imitation term ($\alpha = 0$); *No mixing* restricts the arbiter's candidate set to the two pure joint actions, all-IL or all-RL, as in IBRL.

Protocol. Policies are evaluated periodically without exploration noise on a dedicated environment. Every method, IBMARL included, is evaluated with its RL actors alone, the decentralized policy each agent would deploy; for IBMARL we also report the *executed* policy (greedy arbitration between the IL and RL proposals). Returns are smoothed with a centered window of 9, 3 and 21 evaluations; intervals are ± 1 s.e.m. over 5 seeds, finals average the last 10 evaluations, and steps to the IL policy are the first evaluation whose smoothed return exceeds it, median over seeds.

Navigation. IBMARL's actors surpass the IL policy after a median of 53k environment steps—$2.3\times$ fewer than RLfD (122k) and $3.7\times$ fewer than RFT (198k), while MADDPG reaches it in only 3 of 5 seeds (Fig. 3). With the full budget RLfD (-19.6 ± 0.5) and IBMARL (-20.5 ± 0.4) are comparable; RFT (-35.0 ± 4.9) and MADDPG (-43.9 ± 6.7) remain below. Both ablations match IBMARL here (54k and 55k): the RL actors never lag the executed policy (45k), so the gate has nothing to close.

Buzzwire. This task is considerably harder: MADDPG never reaches the goal in any seed. IBMARL's actors surpass the IL policy after 104k steps, $2.8\times$ fewer than RLfD and RFT (296k each), and finish at -68.7 ± 1.6 against -85.9 ± 8.1 for RLfD and -111.2 ± 4.9 for RFT. The gated imitation term is decisive. Without it (*No penalty*) the executed policy still crosses the IL policy at 136k steps and ends at -81.6 ± 6.2, but the RL actors alone cross in only 4 of 5 seeds and end at -111.6 ± 24.0, below RLfD: the actor lag of Sec. III. With the term the actor and executed curves coincide (Fig. 3), so the critics and the IL policy can be discarded at deployment. *No mixing* is indistinguishable from IBMARL (104k, -71.2 ± 3.0): with two agents, mixing adds only two candidates.

Transport. Within $4{\times}10^6$ steps none of the baselines leaves the starting level (finals -0.1, -0.1 and 0.4): pushing the

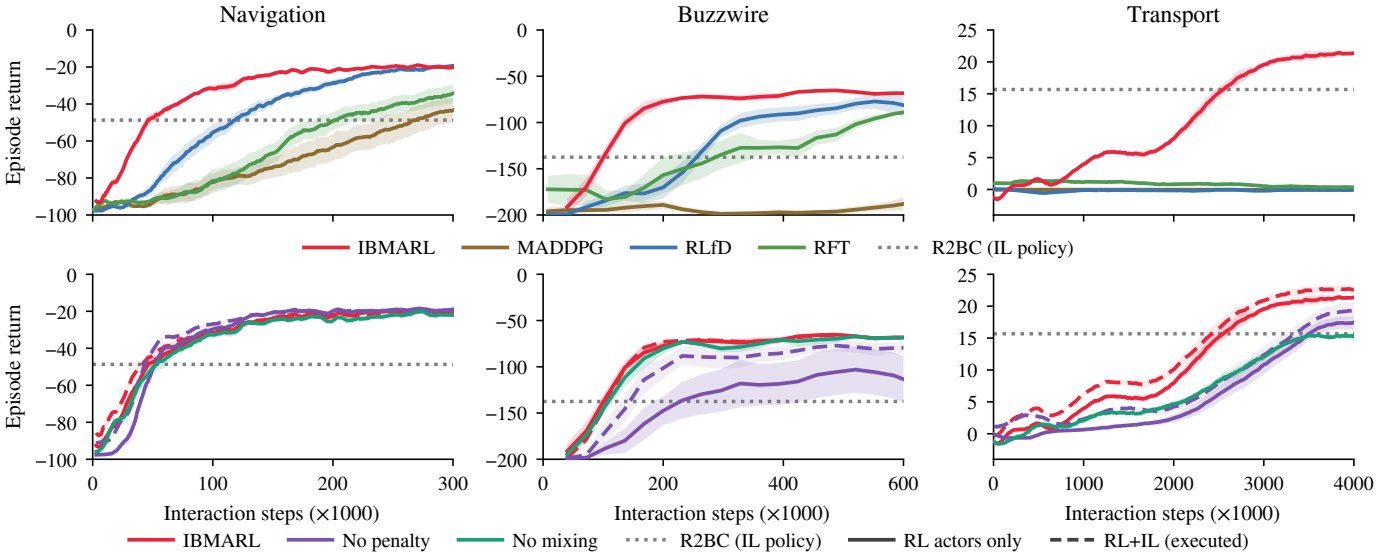


Fig. 3. Evaluation return (mean ± s.e.m., 5 seeds); dotted line: frozen IL policy. Top: RL actors of IBMARL and the baselines. Bottom: ablations; solid curves are the RL actors alone, dashed the executed (arbitrated) policy.

package requires all three agents to act together, and neither replaying nor cloning the demonstrations produces that. IBMARL's actors surpass the IL policy (15.7) after 2.6×10^6 steps and end at 21.3 ± 0.3 (executed policy 22.7 ± 0.6). Here both components carry weight, the term more: *No penalty* crosses in 3 of 5 seeds and ends at 17.3 ± 1.0, *No mixing* in 4 of 5 at 15.3 ± 0.6, the IL policy's own level. Unlike on Buzzwire, the executed policy stays ahead of the actors throughout (Fig. 3): here the term improves the executed policy itself, not only the actors.

V. Discussion and Limitations

By anchoring exploration, the bootstrap target and the actor update to IL proposals, IBMARL replaces MARL's joint-action search with a choice among coherent proposals. Once online interaction is plentiful, replay-based RLfD catches up on Navigation, so IBMARL's advantage there is sample efficiency, the binding constraint for physical multi-robot teams. The gated imitation term matters exactly where the RL actors cannot find success on their own: it is neutral on Navigation, closes the actor lag on Buzzwire, and is the difference between surpassing and merely matching the IL policy on Transport. Unlike RFT's scheduled regularizer, the weakest demonstration-based baseline on every task, its weight is set state by state by the critics, so it stops pulling toward a policy the learner has overtaken and guards against the suboptimal demonstrations R2BC and IBRL assume away. Per-agent mixing helps on the three-agent tasks and is neutral with two. Transport uses a dense reward, every arm rests on 5 seeds, and all tasks are simulated; human demonstrations and hardware validation are the natural next steps.

VI. Conclusion

We introduced IBMARL, in which a decentralized R2BC IL policy proposes actions during exploration, shapes bootstrap targets, pre-populates the replay buffer and, through a critic-gated imitation term, pulls the RL actors toward its actions wherever the critics still prefer them. IBMARL surpasses its IL policy 2.3–3.7× faster than demonstration-based baselines on Navigation and 2.8× faster on Buzzwire, is the only method to surpass it on Transport, and its actors execute alone: realistic single-agent demonstrations are a practical lever for sample-efficient MARL on physical multi-robot teams.

References

[1] C. Mattson, V. Raveendra, E. Novoseller, N. Waytowich, V. J. Lawhern, and D. S. Brown, "R2BC: Multi-agent imitation learning from single-agent demonstrations," in *ICRA*, 2026.

[2] H. Hu, S. Mirchandani, and D. Sadigh, "Imitation bootstrapped reinforcement learning," in *RSS*, 2024.

[3] T. Hester *et al.*, "Deep Q-learning from demonstrations," in *AAAI*, 2018.

[4] M. Večerík *et al.*, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," arXiv:1707.08817, 2017.

[5] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *ICRA*, 2018.

[6] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," in *NeurIPS*, 2021.

[7] E. Turco *et al.*, "Reducing cognitive load in teleoperating swarms of robots through a data-driven shared control approach," in *IROS*, 2024.

[8] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *NeurIPS*, 2017.

[9] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *ICLR*, 2016.

[10] A. Rajeswaran *et al.*, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," in *RSS*, 2018.

[11] X. Chen, C. Wang, Z. Zhou, and K. W. Ross, "Randomized ensembled double Q-learning: Learning fast without a model," in *ICLR*, 2021.

[12] M. Bettini, R. Kortvelesy, J. Blumenkamp, and A. Prorok, "VMAS: A vectorized multi-agent simulator for collective robot learning," in *DARS*, 2022.

[13] S. V. Albrecht, F. Christianos, and L. Schäfer, *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press, 2024.

[14] P. Yu *et al.*, "Beyond joint demonstrations: Personalized expert guidance for efficient multi-agent reinforcement learning," arXiv:2403.08936, 2025.