

R2BC: Multi-Agent Imitation Learning from Single-Agent Demonstrations

Connor Mattson¹, Varun Raveendra¹, Ellen Novoseller², Nicholas Waytowich²,
 Vernon J. Lawhern², Daniel S. Brown¹

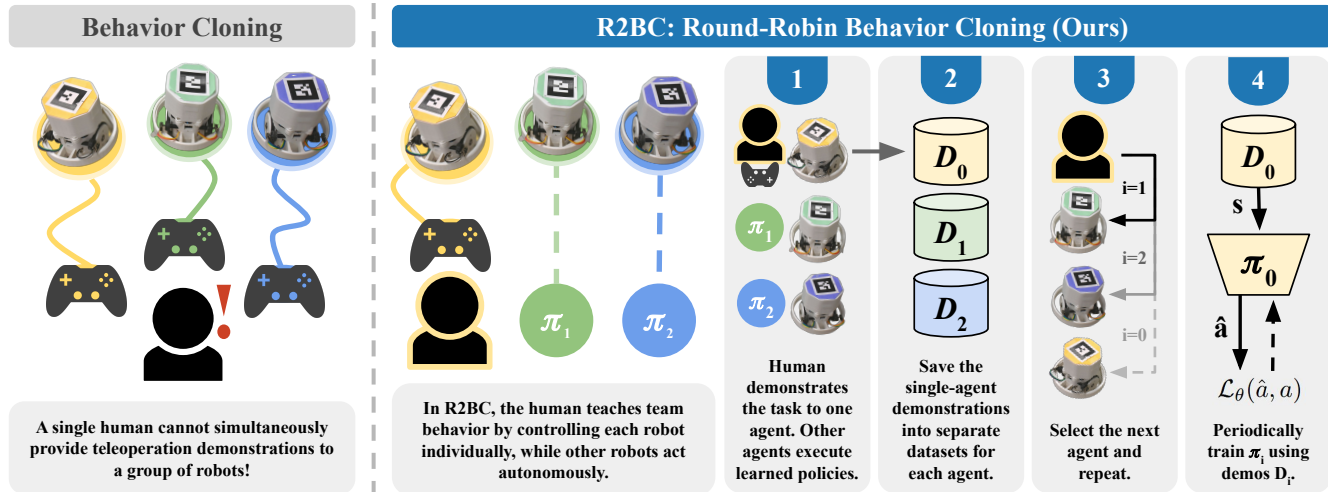


Fig. 1. **Round-Robin Behavior Cloning (R2BC):** Traditional Behavior Cloning (left) requires coordinated and centralized demonstrations, where an expert demonstrates actions near-optimally for all agents. In multi-agent domains, a single human cannot provide high-quality demonstrations due to underactuated control and increased cognitive load. Our method (right), R2BC, removes this restriction by letting the human control one agent at a time while the other agents execute their learned policies. This round-robin process collects realistic demonstrations and iteratively trains cooperative behavior.

Abstract—Imitation Learning (IL) is a natural way for humans to teach robots, particularly when high-quality demonstrations are easy to obtain. While IL has been widely applied to single-robot settings, relatively few studies have addressed the extension of these methods to multi-agent systems, especially in settings where a single human must provide demonstrations to a team of collaborating robots. In this paper, we introduce and study Round-Robin Behavior Cloning (R2BC), a method that enables a single human operator to effectively train multi-robot systems through sequential, single-agent demonstrations. Our approach allows the human to teleoperate one agent at a time and incrementally teach multi-agent behavior to the entire system, without requiring demonstrations in the joint multi-agent action space. We show that R2BC methods match—and in some cases surpass—the performance of an oracle behavior cloning approach trained on privileged synchronized demonstrations across four multi-agent simulated tasks. Finally, we deploy R2BC on two physical robot tasks trained using real human demonstrations. Videos, code, and supplemental materials can be found at <https://sites.google.com/view/r2bc/home>.

I. INTRODUCTION

Imitation Learning (IL) has become a cornerstone of robot learning, enabling robot agents to mimic human demonstrations without explicitly defined reward functions. In particular, multi-agent IL holds the potential to improve robot learn-

ing in a variety of applications involving multi-robot teaming and path planning, such as search and rescue [26], disaster response [11], surveillance [30], swarm robotics [18], and warehouse automation [5]. However, while single-agent IL has seen widespread success, the same cannot yet be said for multi-agent systems, where cooperation, coordination, and partial observability introduce unique challenges that are not easily addressed by simply scaling existing techniques.

Several studies have examined cases where multi-agent IL could greatly improve solutions to real-world problems [2], [4], [10]. However, these works make strong assumptions about the type of demonstrations that can be provided to a group of agents, relying on coordinated and synchronous demonstrations in which all agents take correct actions simultaneously. Such synchronized demonstrations reduce the multi-agent IL problem to a single-agent problem in the joint action space of the agents. These methods are unrealistic, and usually implausible, for real world deployment, as human operators cannot reliably teleoperate multiple robots at the same time to accomplish complex tasks.

In this paper, we address the following research question: *How can we extend imitation learning to multi-agent systems when humans can only feasibly provide demonstrations to one agent at a time?* Our work introduces a novel technique, **Round-Robin Behavior Cloning (R2BC)** (Fig. 1), which enables a human demonstrator to provide single-agent

¹Kahlert School of Computing, University of Utah, Salt Lake City, USA

²DEVCOM Army Research Laboratory, Aberdeen Proving Ground, USA

demonstrations by cycling through the agents, while at any given time, all non-teleoperated agents execute their current learned policies. The key idea is that demonstrations can be individually provided to agents online to iteratively improve a multi-agent policy, achieving parity with, or outperforming, the performance of traditional IL methods that require centralized demonstrations. The contributions of our work are as follows.

- We introduce a novel problem setting, *multi-agent imitation learning from single-agent demonstrations*, with the objective of training a collective policy using only individual agent demonstrations.
- We extend single-agent behavior cloning to this multi-agent problem setting by proposing Round-Robin Behavior Cloning (R2BC), which trains a policy that imitates the demonstrations while removing the unrealistic assumption in prior work of access to coordinated demonstrations.
- We show that R2BC is able to match or exceed the performance of behavior cloning trained on joint action coordinated demonstrations in 4 simulated multi-agent domains using a synthetic demonstrator.
- We deploy R2BC in navigation and block pushing physical robot tasks using real human demonstrations and demonstrate that R2BC outperforms centralized BC by 3.25x and 5.9x on the two tasks, respectively.

To the best of our knowledge, we are the first to propose and deploy a behavior cloning method for multi-agent systems that learns solely from online single-agent demonstrations.

II. RELATED WORK

A. Multi-Robot Teleoperation

Providing demonstrations to a team of coordinating robots is a challenging research problem, as the number of degrees of freedom (DoF) required to control the entire system often exceeds what one human can control at once. Existing methods for teleoperating multi-agent systems include having a single human operator manually switch between robots [9]—e.g. at any given time, all but one robot receive no-op actions—or having a team of human operators who each control a different individual agent [24].

Recent studies have shown that learned models can utilize low-dimensional control inputs to control high-DoF systems such as swarms [36] and manipulators [20]. While these models can effectively allow humans to control complex systems (and therefore provide demonstrations), they are often trained on task-specific data and are unlikely to generalize to unseen tasks, making them a costly approach to providing demonstrations in deployed settings. Instead, we consider the problem setting in which a human can control a subset of the control parameters, corresponding to the control of an individual agent, for each demonstration.

Similarly, HiTAB (Hierarchical Training of Agent Behaviors) [21], [33] provides an approach to multi-agent demonstration collection that learns a set of atomic “skills” that a human can use as a discrete action set to control a

group of heterogeneous agents. In this work, we assume that no preconceived skills library is available, and we provide demonstrations directly in the agent’s action space.

B. Multi-Agent Imitation Learning (MAIL)

The use of imitation learning (IL) to train a policy that mimics an expert task demonstrator has been widely studied in the single-agent literature [7], [12], [28], [29]. Extensions to multi-agent systems, namely *Multi-Agent Imitation Learning (MAIL)*, have shown that policies can be learned from demonstrations to solve problems in grid energy management [10], autonomous vehicle control [4], [14], and multi-agent path finding [2].

Current approaches successfully extend IL to the multi-agent setting, e.g. by modeling the spatial-temporal relationship between agents in a Graph Neural Network [19], [38], modeling latent structures of cooperation [17], modeling individual agent deviations from an optimized coordinated policy [35], combining LfD with RL [14], [25], leveraging large transformers [2], and extending single-agent IL methods such as adversarial IL [4], [8], [32] and inverse soft Q-learning [22] to the multi-agent regime.

Importantly, each of these proposed prior solutions involves training a policy from *joint demonstrations*, where expert trajectories involve all N agents working together simultaneously in noisily-optimal demonstrations. However, to obtain a dataset of joint demonstrations [2], one must assume unrealistic access to either synchronized human demonstrations for all agents or a pretrained RL demonstrator policy [38], planner [6], [18], [34], or model for simulating expert behavior [38], which may either remove the need to apply IL or be difficult to satisfy in real-world in-the-wild tasks. Furthermore, none of the discussed prior methods directly study how to allow a real human to provide demonstrations to the agents. Our work relaxes these assumptions, as a human is unlikely to be able to provide expert demonstrations to N agents simultaneously, but is capable of providing examples to individual agents.

Some work [13] also considers methods for fleet imitation learning where robots are operating independently and multiple human supervisors are scheduled to correct several single-agent IL instances. Our work addresses scenarios where a single human must teach multiple robots that interact, observe, and coordinate with each other. We are the first to show successful multi-agent IL under realistic assumptions about the human’s demonstration ability in multi-agent tasks. We show that a well-performing multi-agent policy can be trained by iteratively providing demonstrations to individual agents, one at a time, enabling training and deployment in settings with a single human supervisor.

III. PROBLEM FORMULATION

We seek to train a set of N robot agents to collaborate to successfully accomplish a cooperative task. We formulate the problem of multi-agent learning from demonstrations as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) represented via the tuple

$\langle \mathcal{I}, \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \{\mathcal{O}_i\}_{i \in \mathcal{I}}, \gamma \rangle$ consisting of a set of agents $\mathcal{I} = \{1, 2, \dots, N\}$, state space \mathcal{S} , and joint action space $\mathcal{A} = A_1 \times \dots \times A_N$. The environment transition dynamics, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$, represent the probability of transitioning from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ after the agents take joint action $a \in \mathcal{A}$. The joint agent observation space $\Omega = \Omega_1 \times \dots \times \Omega_N$ and corresponding observation function $\mathcal{O}_i : \mathcal{S} \times \mathcal{A} \times \Omega_i \rightarrow [0, 1]$ determines the probability distribution over the possible observations received by agent i after taking joint action a . As is standard in imitation learning, we do not learn directly from rewards (\mathcal{R}), but it is worth noting that we focus on games with *shared reward* (also called *common reward*), where all agents share the same collective objective.

We define a *joint* expert trajectory τ with finite horizon T as the sequence of states visited by an expert that controls all agents simultaneously, along with the associated joint expert actions taken in each state, $\tau = (s_0, a_0, s_1, a_1, \dots, s_T)$, where $s_t \in \mathcal{S}$ is the state at time t and $a_t \in \mathcal{A}$ is the joint action (i.e. concatenation over N individual agent actions) taken in state s_t .

Given a dataset of noisily-optimal demonstrations, \mathcal{D} , standard IL seeks to learn a joint policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that closely models the expert's state-action pairs and achieves similar task performance. Generally, the goal of IL is to find the parameters θ to a policy π_θ that minimize the error between the demonstrated expert's actions, $\pi^*(s)$, and the outputs of the learned policy, $\pi_\theta(s)$, expressed via the discrepancy loss:

$$\min_{\theta} \mathbb{E}_{\tau \sim \mathcal{D}} \left[\sum_{t=0}^{T-1} \mathcal{L}(\pi_\theta(s_t), \pi^*(s_t)) \right], \quad (1)$$

where \mathcal{L} is a loss function that penalizes divergence from the expert's actions, commonly the squared error.

Contrary to the single-agent IL literature, we assume that a single human *cannot* teleoperate the entire system with optimal actions, and therefore cannot provide actions in the joint action space, \mathcal{A} . For systems with any non-arbitrary number of agents (i.e. $N \geq 2$), this is a reasonable assumption given that (1) the amount of cognitive burden required to monitor the system increases with the number of agents [15], [36] and (2) teleoperation interfaces cannot control all degrees of freedom for complex systems without custom control schemes [1], [36]. Instead, we assume that the human is only capable of controlling small parts of the system, such as one individual agent at a time, which we call *single-agent demonstrations*. For agent $i \in \mathcal{I}$, a demonstration of length T takes the form

$$\tau_i = (o_{i,0}, a_{i,0}, o_{i,1}, a_{i,1}, \dots, o_{i,T}) \quad (2)$$

where $o_{i,t} \in \Omega_i$ is the i^{th} agent's local observation, and $a_{i,t} \in A_i$ is agent i 's *individual action*, rather than the joint action of all agents. We seek to train multi-agent policies that achieve sufficient collective performance using only N sets of single-agent demonstrations, $\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_N\}$, where \mathcal{D}_i is the set of single-agent demonstrations for agent i .

Algorithm 1 Round-Robin Behavior Cloning (R2BC)

Require: Number of agents N , expert policy π^* , initial agent policies $\{\pi_1, \dots, \pi_N\}$, demonstration buffers $\{\mathcal{D}_1, \dots, \mathcal{D}_N\}$, update frequency k , and task horizon T .

- 1: Initialize iteration counter $c \leftarrow 0$
- 2: **while** $c \leq \text{MAX_ITERS}$ **do**
- 3: **for** each agent $i = 1$ to N **do** ▷ round-robin
- 4: Reset environment to initial state $s_0 \sim \mathcal{S}_0$
- 5: **for** $t = 0$ to T **do**
- 6: Agent i receives expert action $a_{i,t} \leftarrow \pi^*(o_{i,t})$
- 7: **for** each agent $j \neq i$ **do**
- 8: Agent j computes action $a_{j,t} \leftarrow \pi_j(o_{j,t})$
- 9: **end for**
- 10: Execute joint action $a_t = (a_{1,t}, \dots, a_{N,t})$
- 11: Receive next observation $o_{i,t+1}$
- 12: Append $(o_{i,t}, a_{i,t})$ to \mathcal{D}_i
- 13: **end for**
- 14: **end for**
- 15: **if** $c \bmod k = 0$ **then**
- 16: Update π_i using BC on $\mathcal{D}_i \forall i \in \mathcal{I}$.
- 17: **end if**
- 18: $c \leftarrow c + 1$
- 19: **end while**

IV. ROUND-ROBIN BEHAVIOR CLONING

To teach cooperative tasks to teams of robots using single-agent demonstrations, we propose **Round-Robin Behavior Cloning (R2BC)**, an online imitation learning algorithm that enables a single human operator to sequentially provide demonstrations to a team of cooperative agents. Unlike traditional joint action behavior cloning, which requires the expert to simultaneously demonstrate actions for all agents (a setting infeasible for real-world human teleoperation), R2BC assumes that the expert can only control a single-agent at any given time. Our method, shown in Algorithm 1, cycles through the agents in a round-robin fashion, allowing the expert to demonstrate behaviors for one agent while the others operate their current learned policies. After an episode of demonstrating the task to agent i , the demonstrator switches to agent $i + 1 \pmod{N}$. Providing round-robin demonstrations ensures a uniform spread of demonstrations across all N agents during training.

In a decentralized fashion, each agent maintains its own policy $\pi_i : \Omega_i \rightarrow A_i$. Each agent i stores its demonstrations within a buffer \mathcal{D}_i , which is used only to train its policy, π_{θ_i} . At regular intervals (every k demonstrations), each agent's policy is updated using that agent's collected demonstration dataset using the standard behavior cloning loss:

$$L(\theta_i) = \sum_{(o_i, a_i) \sim \mathcal{D}_i} \|\pi_{\theta_i}(o_i) - a_i\|_2^2.$$

R2BC combines two extensions to traditional centralized behavior cloning. First, R2BC allows for full decentralization, both in policy execution and demonstrations. The main motivation of our work is to address realistic

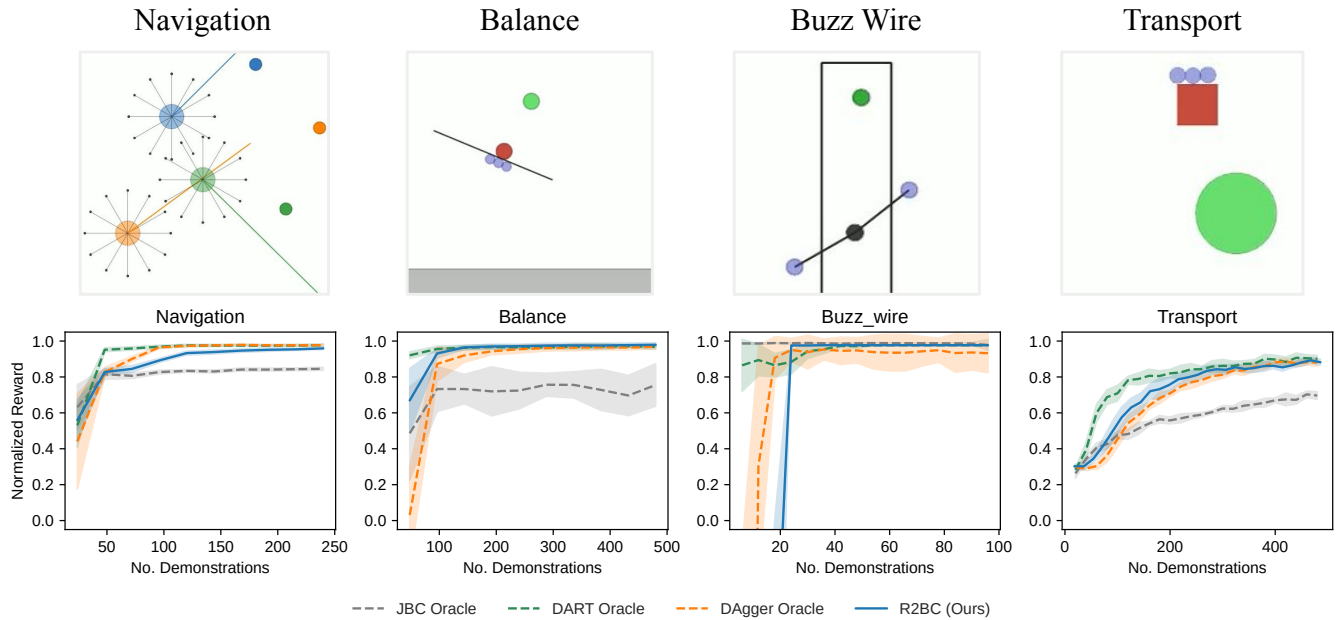


Fig. 2. **Simulation Results:** We compare the R2BC method to a set of three baselines that assume oracle (privileged) access to a joint action demonstrator. Results are averaged over 10 seeds with shaded regions indicating a 99% confidence interval. **Top:** Four multi-agent tasks selected from the Vectorized Multi-Agent Simulator (VMAS) [3]. **Bottom:** The performance of each method is shown with increasing number of expert demonstrations. In each case, R2BC outperforms or achieves parity with all baseline oracle methods, despite not having privileged access to joint action demonstrations. Performance is normalized between 0.0 (random policy) and 1.0 (oracle demonstrator performance).

demonstrator limitations by decentralizing demonstrations, allowing iterations of single-agent demonstration to teach behavior to multi-robot systems. Policy decentralization also addresses several real-world deployment considerations such as limited communication and partial observability. The full decentralization of R2BC promotes realistic demonstration collection from real humans and deployment on physical hardware in the real world.

Second, R2BC iteratively improves team performance in an online-learning manner, where demonstrations are provided online while other agents execute policies cloned on the demonstrations observed so far. Our method directly benefits from an online regime because demonstrators must teach a single-agent to perform appropriately alongside *existing actors*, i.e. the other agents that are executing autonomous policies. This ideally allows a demonstrator to teach an agent how to act amidst wide variability in the help received from other agents, resulting in greater demonstration diversity and downstream resilience to error and distribution shift. In Section VI, we examine several ablations of R2BC and highlight that both extensions—decentralization and online policy learning—are essential to the success of our method.

V. SIMULATED EXPERIMENTS

We compare R2BC to three oracle imitation learning methods on four cooperative tasks in simulation, each requiring coordination between 2-3 robots. First, we introduce the simulated scenarios and experimental details (Section V-A). Second, we highlight the surprising performance of R2BC compared to a privileged Joint Behavior Cloning (JBC)

baseline, where demonstrations are centralized, synchronous, and near-optimal (Section V-B). Finally, we offer intuition about the online benefits of R2BC by comparing our method to oracle implementations of DAgger [29] and DART [16] trained on online joint demonstrations (Section V-C).

A. Experiment Design

We evaluate R2BC and three baselines on four simulated cooperative tasks using the Vectorized Multi-Agent Simulator (VMAS) [3], shown in Figure 2 (top).

Navigation: Three agents are randomly positioned in a two-dimensional space each with a designated goal. Agents can utilize LiDAR-style sensors to sense other agents and avoid collisions *en route* to their objectives.

Balance: Three agents are placed under a freely rotating line carrying a spherical package. The agents must transport this package from the bottom to a goal at the top, without allowing the line or the package to fall.

Buzz Wire: Two agents are connected to an enclosed mass using rigid linkages. The agents are penalized for coming in contact with the enclosure and must take cooperative actions that push the mass through the hallway. This environment represents a system with coupled dynamics, where the actions of one agent can directly displace the other agent, reflecting difficult multi-agent transition dynamics.

Transport: Three agents cooperate to push a package into a designated goal region. The packages are heavy and cannot be pushed by individual agents alone, requiring agents to work together to transport the package.

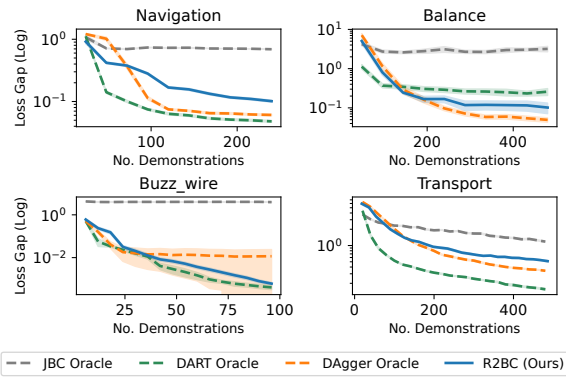


Fig. 3. **Covariate Shift Reduction:** The *Loss Gap* is defined as the difference between the training loss on the demonstration set and testing loss on evaluation trajectories (lower is better). Dashed lines indicate methods that require privileged centralized demonstrations, while the solid blue line is our method. R2BC reduces the train–test loss gap compared to JBC, while showing benefits comparable to other online imitation learning methods. Shaded regions indicate standard error across 10 seeds.

The methods are compared under a demonstration budget that varies between domains depending on the task difficulty. Each policy is trained using a MSE-Loss until the loss plateaus, using a fixed learning rate of 0.001 and a minibatch size of 256 state-action pairs. After training, we evaluate the learned policies on an identical set of 50 environment initialization seeds and compute the average environment reward. In all experiments, we normalize the task reward to the range $[0, 1]$, where a score of 0.0 corresponds to a random policy and 1.0 reflects the performance of the oracle expert demonstrator. As is standard in IL, the expert defines an upper bound that is typically not exceeded by the learner. Additional procedures, hyperparameters, and experiments can be found on our supplemental website¹.

B. Comparison to Joint Action Behavior Cloning (JBC)

JBC serves as an oracle behavior cloning method that learns from coordinated demonstrations provided in the joint action space. In practice, such demonstrations require complex learned teleoperation schemes or many-to-many teleoperation schemes where N human operators control N robots. Our experiments utilize a multi-agent reinforcement learning policy (MAPPO [31], [37]) trained on the ground truth environment reward to provide coordinated demonstrations to JBC for supervised imitation learning. Therefore, we consider this an oracle baseline with access to unrealistic data that reflects prior assumptions about coordinated demonstrations in multi-agent IL [2], [4], [10].

While it might seem that JBC would be an upper-bound for R2BC’s performance—due to JBC having oracle access to near-optimal, coordinated demonstrations—surprisingly, we found that in all simulated domains, R2BC outperforms or achieves parity with the oracle JBC method (Figure 2). Although JBC learns from coordinated demonstrations, its demonstration data solely contains examples of near-optimal

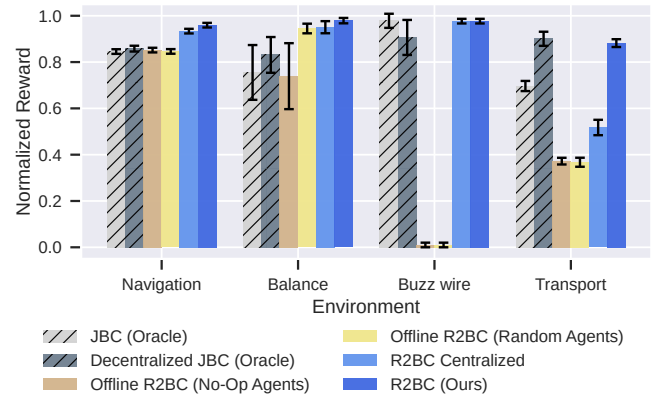


Fig. 4. **R2BC Ablations:** Comparison of R2BC to several ablations across four simulated environments. Across all environments, R2BC consistently outperforms the two offline variants and matches or exceeds the oracle JBC methods. Error bars indicate 99% confidence interval over 10 seeded runs.

behavior and lacks examples of how to correct erroneous behaviors or act in out-of-distribution states. Conversely, R2BC benefits from observation perturbations caused by the non-demonstrating agents, which act autonomously. The initially unpredictable actions of the other agents create situations where the single-agent demonstrator can show the robot how to correct wayward cooperative behavior or avoid a collision with another agent. Cases like these are much less prevalent in JBC demonstrations, in which the expert demonstrates fully-coordinated agent behaviors. Our empirical observations offer support for R2BC as a realistic and viable alternative to JBC, in several cases surpassing the performance of training on “super-human” JBC demonstrations using only single-agent demonstrations.

C. Comparison to Online Imitation Learning Baselines

We hypothesize that R2BC’s significant performance improvement over JBC results from the online paradigm, which exposes agents to diverse states and levels of cooperation from teammates. We compare R2BC to two popular single-agent online imitation learning methods specifically designed to combat compounding errors: (1) DAgger [29], where a joint action expert provides corrective labels to on-policy rollouts and (2) DART [16], where noise is injected into the demonstrator’s teleoperation online to vary the state distribution and allow for online corrections. Our implementation of these methods in multi-agent domains uses oracle centralized corrections generated from the same multi-agent RL policy described in the previous section.

Our simulated results for DART and DAgger indicate that R2BC achieves similar performance to both privileged online baselines (Figure 2). In all three cases, increasing the number of online demonstrations correlates with improved policy performance. These results provide evidence that R2BC yields a similar reduction in compounding errors to DART and DAgger, but importantly, R2BC achieves this without requiring N human demonstrators or a super-human joint demonstrator. We hypothesize that the observation variance

¹<https://sites.google.com/view/r2bc/home>

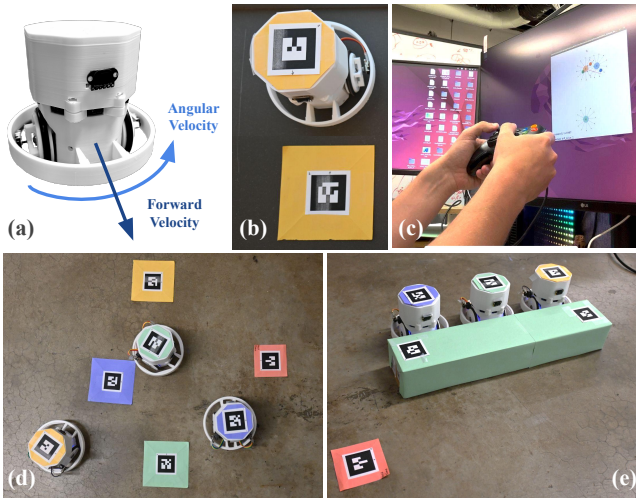


Fig. 5. **Real Robot Experiments:** (a) The HeRo+ Robot, a differential drive mobile robot that acts using a forward velocity and angular velocity, (b) AruCo markers used to track robot/goal states, (c) R2BC demonstrations are provided in simulation by a real human, (d) a real world instantiation of the navigation task (from Figure 2 top), where agents must navigate to their colored goals while avoiding collisions, (e) a block-pusher task where robots must transport a 0.5kg box across the environment. In both (d) and (e), the pink AruCo marker is used for camera-to-world registration.

induced by the non-demonstrating agents in R2BC acts as artificial noise injected into the demonstrator’s state, serving a similar function to the noise injected in DART, which was shown to reduce covariate shift in single-agent IL. We provide empirical support for covariate shift reduction in Figure 3, which shows that R2BC exhibits similar trends in the train-test loss gap as the two oracle online baselines, while achieving much better generalization than JBC.

VI. SIMULATED ABLATIONS

In Section IV, we highlighted the two extensions to traditional centralized behavior cloning introduced by our method: full decentralization and online learning. In this section, we perform several ablation experiments and show that both of these elements (decentralization and online learning) are essential to the training and deployment of our method. We evaluate the performance of these ablations and summarize the findings in Figure 4. All experiments shown use simulated human feedback.

Joint Behavior Cloning (JBC) most closely reflects traditional behavior cloning, where a policy maps states to the *joint actions* for all agents in a centralized manner. We compare this with **Decentralized JBC**, an approach that utilizes the same offline oracle joint action demonstrations as JBC, but clones actions into N independent networks. Notably, this method also decentralizes the state information, and the individual agents act solely based on their local (potentially partially observable) state. We find that Decentralized JBC performs on par with JBC in the navigation and buzz wire tasks, and significantly better than JBC in the balance and transport tasks. However, this oracle method still requires centralized near-optimal demonstrations, which a lone human demonstrator is unlikely to be able to provide.

Next, we evaluate two decentralized versions of R2BC that are offline (**Offline R2BC**). In these methods, the human cycles round-robin through each agent and demonstrates the task but, unlike in R2BC, the agents do not regularly update their policies and therefore do not learn during the demonstration phase. We test two versions of this: in the first, the non-demonstrating agents do not execute any actions (**No-Op Agents**), while in the second, the non-demonstrating agents uniformly sample random actions from the action space (**Random Agents**). We find that these approaches can occasionally perform on par with R2BC (e.g., the Random variant in Balance), but overall they are unreliable and prone to high variance across tasks, as seen in Buzz Wire.

Finally, we consider a centralized version of R2BC (**Centralized R2BC**). Here, instead of learning N decentralized policies, we maintain a single centralized network that outputs the joint action, and during training, we mask the loss to only update the output layer neurons corresponding to the demonstrated agent. We find that Centralized R2BC performs slightly worse than the decentralized version in navigation and balance, comparably in buzz wire, and significantly worse in transport.

These ablations highlight the importance of both decentralization and online learning during demonstration collection: removing either element leads to unreliable or degraded performance across tasks. Together, these findings demonstrate that the full R2BC framework is necessary to achieve robust multi-agent learning from single-agent demonstrations.

VII. PHYSICAL ROBOT DEPLOYMENT

We deploy both JBC and R2BC on three HeRo+ robots [23], [27], each with a two-dimensional action space consisting of forward/backward velocity and angular turning velocity (Figure 5a). A centralized Robot Operating System (ROS) server computes the instantaneous state of the environment using a Realsense D435i camera to track AruCo markers on agents and objects (Figure 5b), and passes either local observations to each agent’s policy (R2BC) or the full state to a centralized policy (JBC). Our real world experiments consist of 2 tasks:

Navigation (Figure 5d): A physical version of the simulated navigation task, as described in section V-A. Mock LiDAR values are computed instantaneously using the centralized video feed.

Block Pusher (Figure 5e): A physical variation of the simulated transport task. Three agents must coordinate to push an elongated green block to a goal location. Because of the block’s shape, agents must balance forces applied at different contact points to successfully transport the block.

To evaluate JBC and R2BC using real human demonstrations, the authors provided two sets of 240 demonstrations for each task by teleoperating agents in simulation using an Xbox360 joystick controller (Figure 5c). In R2BC, the demonstrator provided single-agent demonstrations and they were aware of which agent they would be controlling prior to the start of each episode. In JBC, the demonstrator used two game controllers and could control all agents simultaneously

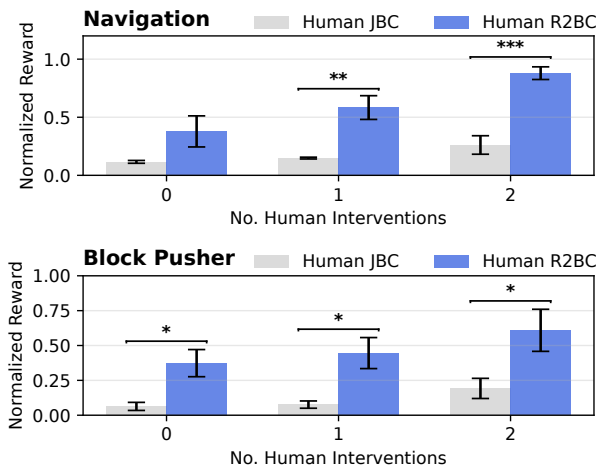


Fig. 6. **Real Robot Results:** JBC and R2BC policies trained from real human demonstrations in simulation are deployed directly onto the real robots in two tasks. To account for sim-to-real distribution shift, we report results with 0, 1, and 2 online interventions, where a human supervisor can briefly correct robot behavior. All results show mean cumulative episode reward with standard error over 5 randomized initial configurations. Stars (*) indicate one-tailed significance: $*p < 0.05$, $**p < 0.01$, $***p < 0.005$.

using 3 of the analog sticks. In practice, this often meant focusing on only 1–2 agents at once, or attempting to span the inputs by placing the controllers on a flat surface and manipulating the analog sticks with both hands, illustrating the difficulty of providing joint demonstrations in real time.

To support sim2real transfer, the simulated environments for both navigation and block pushing were customized to match the kinematics of our HeRo+ robots and physical workspace. Similarly, the action space in the simulator was clamped to realistic linear and angular velocities achievable by our robots in the real world.

For these physical tasks, we deployed the trained policies directly onto the real robots and evaluated the quantitative task performance over a fixed horizon. For the navigation task, we measured the cumulative inverse distance between agents and their respective goals (same reward metric used to evaluate policies in simulation) with a task horizon of 90 seconds. For the block pushing task, we measured the cumulative distance the package moved in the direction of the goal region with a task horizon of 30 seconds. Due to the distribution shift that accompanies sim2real deployment, deployed behaviors can sometimes get stuck in local minima, and agents are unable to complete the task. We account for this potential distribution shift in our real-world experiments by studying the effect of allocating 0, 1, or 2 human interventions. In each intervention, the human supervisor selects a robot that appears “stuck” and manually controls its actions for 3 seconds to correct the robot’s behavior while the other robots continue to execute their autonomous policies.

Our results (Figure 6) are averaged over 5 initial environment configurations, which were randomized but consistent across comparisons. To test for the directional hypothesis that R2BC performs significantly better than JBC, we employ a Welch two-sample t-test and convert the two-tailed p-value

to a single-tail p-value.

In the navigation task, we find that R2BC performs 3.25x better than JBC on average without supervisor interventions; however, the statistical tests do not indicate significance ($p = 0.0773$) due to the high performance variance that both methods experience under zero-shot sim2real deployment. However, significant differences between R2BC and JBC emerge when allowing both methods access to supervisor interventions. With one intervention allowed, R2BC outperforms JBC by 3.92x ($p = 0.0094$) and with two interventions allowed, R2BC outperforms JBC by 3.36x ($p < 0.005$). In the block pushing task, we find that R2BC performs 5.9x better than JBC ($p = 0.0126$) with no supervisor interventions. With one intervention allowed, R2BC outperforms JBC by 5.84x ($p = 0.0107$) and with two interventions allowed, R2BC outperforms JBC by 3.16x ($p = 0.0239$).

Ultimately, we find that training with real human demonstrations leads to improved performance for R2BC compared to JBC. As expected in sim2real transfer, the deployed policies encounter some local minima that can be easily corrected with just a few seconds of supervisor intervention. Notably, supervisor corrections help the performance of both methods, yet slight corrections to R2BC policies appear to have a much greater effect on performance compared to equivalent corrections in JBC. This reinforces the notion that R2BC is a learning paradigm that leads to high performing policies and can easily be adopted by teachers, removing the need to collect joint action demonstrations. Our results confirm that Round-Robin Behavior Cloning can reliably outperform traditional centralized behavior cloning methods both in simulation and when trained with real human demonstrations and deployed on physical hardware.

VIII. CONCLUSION & FUTURE WORK

Our paper highlights a novel research question targeting the deployment of multi-agent systems: *How can we extend imitation learning to multi-agent systems when humans can only feasibly provide demonstrations to one agent at a time?* While multi-agent IL methods assume unrealistic joint demonstrations, our algorithm, R2BC, utilizes round-robin single-agent demonstrations to gather examples of what each agent should do with respect to the environment and other agents. Our results show that R2BC can outperform JBC in both simulated domains with synthetic demonstrations and physical robot deployment with real human demonstrations.

One underexplored aspect of this work is evaluating the human factors associated with R2BC. In future work, we plan to conduct user studies with non-experts to evaluate the cognitive burden and qualitative experience of our method. Although the empirical evidence for R2BC is strong, developing theoretical guarantees for convergence and covariate shift reduction would further reinforce the intuition provided in this work. Overall, R2BC provides a practical path toward teaching multi-agent systems via realistic human demonstrations, bridging the gap between single-agent IL and scalable multi-agent deployment.

ACKNOWLEDGMENTS

This work was completed in the Aligned, Robust, and Interactive Autonomy (ARIA) Lab at the University of Utah. ARIA Lab research is supported in part by the NSF (IIS-2310759, IIS2416761), the NIH (R21EB035378), ARPA-H, Coefficient Giving, and the ARL STRONG program.

REFERENCES

- [1] Marco Aggravi, Giuseppe Sirignano, Paolo Robuffo Giordano, and Claudio Pacchierotti. Decentralized control of a heterogeneous human-robot team for exploration and patrolling. *IEEE Transactions on Automation Science and Engineering*, 2021.
- [2] Anton Andreychuk, Konstantin Yakovlev, Aleksandr Panov, and Alexey Skrynnik. Mapf-gpt: Imitation learning for multi-agent pathfinding at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.
- [3] 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*. Springer, 2022.
- [4] Raunak P Bhattacharyya, Derek J Phillips, Blake Wulfe, Jeremy Morton, Alex Kuefler, and Mykel J Kochenderfer. Multi-agent imitation learning for driving simulation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.
- [5] Ali Bolu and Ömer Korçak. Adaptive task planning for multi-robot smart warehouse. *Ieee Access*, 2021.
- [6] Lin Chen, Yaonan Wang, Zhiqiang Miao, Yang Mo, Mingtao Feng, Zhen Zhou, and Hesheng Wang. Transformer-based imitative reinforcement learning for multirobot path planning. *IEEE Transactions on Industrial Informatics*, 2023.
- [7] Bin Fang, Shidong Jia, Di Guo, Muhua Xu, Shuhuan Wen, and Fuchun Sun. Survey of imitation learning for robotic manipulation. *International Journal of Intelligent Robotics and Applications*, 2019.
- [8] Zheng Fang, Tianhao Chen, Tian Shen, Dong Jiang, Zheng Zhang, and Guangliang Li. Multi-agent generative adversarial interactive self-imitation learning for auv formation control and obstacle avoidance. *IEEE Robotics and Automation Letters*, 2025.
- [9] Ildar Farkhatdinov and Jee-Hwan Ryu. Teleoperation of multi-robot and multi-property systems. In *2008 6th IEEE International Conference on Industrial Informatics*. IEEE, 2008.
- [10] Shuhua Gao, Yizhuo Xu, Zhaoqian Zhang, Zhengfang Wang, Xiaoyu Zhou, and Jing Wang. Multi-agent imitation learning based energy management of a microgrid with hybrid energy storage and real-time pricing. *IEEE Internet of Things Journal*, 2025.
- [11] Jason Gregory, Jonathan Fink, Ethan Stump, Jeffrey Twigg, John Rogers, David Baran, Nicholas Fung, and Stuart Young. Application of multi-robot systems to disaster-relief scenarios with limited communication. In *Field and service robotics: results of the 10th international conference*. Springer, 2016.
- [12] Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S Brown, and Ken Goldberg. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. In *Conference on Robot Learning*, 2021.
- [13] Ryan Hoque, Lawrence Yunliang Chen, Satvik Sharma, Karthik Dharmarajan, Brijen Thananjeyan, Pieter Abbeel, and Ken Goldberg. Fleet-dagger: Interactive robot fleet learning with scalable human supervision. In *Conference on Robot Learning*, pages 368–380. PMLR, 2023.
- [14] Chang Huang, Junqiao Zhao, Hongtu Zhou, Hai Zhang, Xiao Zhang, and Chen Ye. Multi-agent decision-making at unsignalized intersections with reinforcement learning from demonstrations. In *IEEE Intelligent Vehicles Symposium (IV)*, 2023.
- [15] Julian Kaduk, Müge Cavdan, Knut Drawing, and Heiko Hamann. From one to many: How active robot swarm sizes influence human cognitive processes. In *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*. IEEE, 2024.
- [16] Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on robot learning*. PMLR, 2017.
- [17] Hoang M Le, Yisong Yue, Peter Carr, and Patrick Lucey. Coordinated multi-agent imitation learning. In *International Conference on Machine Learning*. PMLR, 2017.
- [18] Jie Li and Ying Tan. A two-stage imitation learning framework for the multi-target search problem in swarm robotics. *Neurocomputing*, 2019.
- [19] Kai Li, Zhao Ma, Liang Li, and Shiyu Zhao. Collective behavior clone with visual attention via neural interaction graph prediction. *arXiv preprint arXiv:2503.06869*, 2025.
- [20] Dylan P Losey, Krishnan Srinivasan, Ajay Mandlekar, Animesh Garg, and Dorsa Sadigh. Controlling assistive robots with learned latent actions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- [21] Sean Luke and Vittorio Amos Ziparo. Learn to behave! rapid training of behavior automata. In *Proceedings of adaptive and learning agents workshop at aamas*. Citeseer, 2010.
- [22] Tien Mai, Thanh Nguyen, et al. Inverse factorized soft q-learning for cooperative multi-agent imitation learning. *Advances in Neural Information Processing Systems*, 2024.
- [23] Connor Mattson, Varun Raveendra, Ricardo Vega, Cameron Nowzari, Daniel S Drew, and Daniel S Brown. Discovery and deployment of emergent robot swarm behaviors via representation learning and real2sim2real transfer. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, 2025.
- [24] Jayam Patel, Tyagaraja Ramaswamy, Zhi Li, and Carlo Pinciroli. Transparency in multi-human multi-robot interaction. *arXiv preprint arXiv:2101.10495*, 2021.
- [25] Xinyue Qi, Jianhang Tang, Jiangming Jin, and Yang Zhang. Diffusion-based multi-agent reinforcement learning with communication. In *Asia Pacific Wireless Communications Symposium*, 2024.
- [26] Jorge Pena Queralta, Jussi Taipalmaa, Bilge Can Pullinen, Victor Kathan Sarker, Tuan Nguyen Gia, Hannu Tenhunen, Moncef Gabbouj, Jenni Raitoharju, and Tomi Westerlund. Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *Ieee Access*, 2020.
- [27] Paulo Rezeck, Héctor Azpúrua, Mauricio FS Correa, and Luiz Chaimowicz. Hero 2.0: A low-cost robot for swarm robotics research. *Autonomous Robots*, 2023.
- [28] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *International conference on artificial intelligence and statistics*, 2010.
- [29] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, 2011.
- [30] Jürgen Scherer and Bernhard Rinner. Multi-robot persistent surveillance with connectivity constraints. *IEEE Access*, 2020.
- [31] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [32] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. Multi-agent generative adversarial imitation learning. *Advances in neural information processing systems*, 2018.
- [33] William G Squires and Sean Luke. Lfd training of heterogeneous formation behaviors. In *AAAI Spring Symposia*, 2018.
- [34] Mohit Srinivasan, Ankush Chakrabarty, Rien Quirynen, Nobuyuki Yoshikawa, Toshisada Mariyama, and Stefano Di Cairano. Fast multi-robot motion planning via imitation learning of mixed-integer programs. *IFAC-PapersOnLine*, 2021.
- [35] Jingwu Tang, Gokul Swamy, Fei Fang, and Steven Z Wu. Multi-agent imitation learning: Value is easy, regret is hard. *Advances in Neural Information Processing Systems*, 2024.
- [36] Enrico Turco, Chiara Castellani, Valerio Bo, Claudio Pacchierotti, Domenico Prattichizzo, and Tommaso Lisini Baldi. Reducing cognitive load in teleoperating swarms of robots through a data-driven shared control approach. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024.
- [37] Chao Yu, Akash Velu, Eugene Vangitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 2022.
- [38] Siyu Zhou, Mariano J Phielipp, Jorge A Sefair, Sara I Walker, and Heni Ben Amor. Clone swarms: Learning to predict and control multi-robot systems by imitation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.