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

Connor Mattson
Kahlert School of Computing
University of Utah
Salt Lake City, Utah
c.mattson@utah.edu

Daniel S. Brown
Kahlert School of Computing
University of Utah
Salt Lake City, Utah
daniel.s.brown@utah.edu

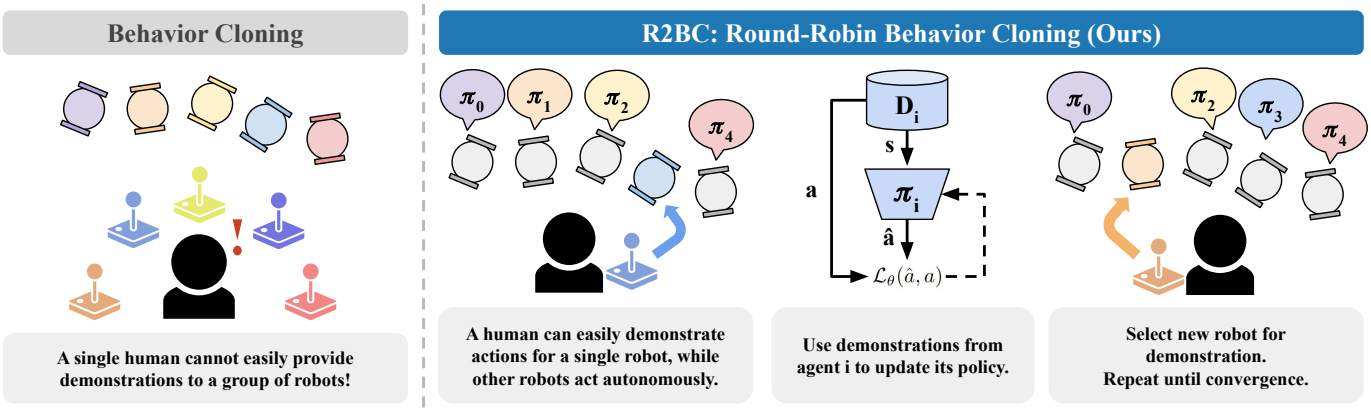

Fig. 1. **Round-Robin Behavior Cloning (R2BC):** Traditional Behavior Cloning (**left**) require coordinated and centralized demonstrations, where an expert demonstrates actions near-optimally for all agents. A lone human operator is unlikely to be able to provide high-quality demonstrations due to underactuated control and increased cognitive burden. Our method (**right**), R2BC, extends imitation learning to the multi-agent domain and only requires the human operator to provide demonstrations to one agent at a time. As the expert cycles through demonstrations for each agent, the learned policies cloned from the existing single-agent demonstrations are being executed on the other robots, diversifying the distribution of observations seen by the expert's agent and iteratively improving the cooperative behavior.

***Abstract*—Imitation Learning (IL) is a natural and intuitive way for humans to teach robots, particularly when high-quality demonstrations are easy to obtain. While many studies have explored IL in the context of single-agent robotic tasks, relatively few have addressed how to extend 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 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 leverages the human operator's ability to control one agent at a time and enables the operator to iteratively teach 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 traditional behavior cloning trained on an oracle set of privileged synchronized demonstrations across four multi-agent tasks.**

I. Introduction

Imitation Learning (IL) has become a cornerstone of robot learning, enabling agents to mimic human demonstrations without explicitly defined reward functions. From manipulation in cluttered environments [22, 29, 5] to mobile navigation [12], IL has shown that it can distill expert behavior into performant policies with minimal intervention. However, while single-agent imitation learning 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 imitation learning could greatly improve solutions to real-world problems [1, 8, 3], but they make strong assumptions about the type of demonstrations that can be provided to a group of agents. Specifically, these studies rely on coordinated and synchronous demonstrations, where all agents are taking correct actions simultaneously, reducing 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 adaptation as human operators can not 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 provide demonstrations to one agent a time?* Our work introduces a novel technique, **R**ound-**R**obin **B**ehavior **C**loning (**R2BC**) (Fig. 1). The key idea is that demonstrations can be individually provided to agents online to iteratively improve a multi-agent policy that achieves parity with, or outperforms, the performance of traditional imitation

learning 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), removing previous unrealistic assumptions of access to coordinated demonstrations in order to imitate demonstrations to learn a policy.
- 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 show that two round-robin variants of DAgger [24] and DART [15] achieve performance parity with the original algorithms trained with joint-action coordinated demonstrations.

To the best of our knowledge, we are the first to propose and test 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 degrees of freedom required to control the entire system is often more than one human can control at once. Existing methods for teleoperating multi-agent systems include a single-human operator manualing switching between robots [7], or having multiple human operators controlling individual agents simultaneously [20].

Recent studies have shown that learned models can map low-dimensional control inputs to control high-DoF systems such as swarms [28] and manipulators [18]. 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 where the human can control a subset of the control parameters, corresponding to the control of an individual agent, for each demonstration.

Similarly, HiTAB (Heirarchical Training of Agent Behaviors) [27, 19] provides an approach to multi-agent demonstration collection that learns a set of atomic "skills" that a human can use as a discrete set of actions to control a set of heterogeneous agents. 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 to train a policy that mimics an expert task demonstrator has been widely studied in single agent literature [24, 9, 23, 13, 10, 4, 6]. 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 [8], autonomous vehicle control [3, 11], and multi-agent path finding [1].

Current approaches successfully extend learning from demonstrations (LfD) to the multi-agent setting by modeling the spatial-temporal relationship between agents in a Graph Neural Network [30, 17], modeling latent structures of coorpoeration [16], combining LfD with RL [11, 21], leveraging large transformers [1], and introducing parameter sharing techniques for traditional single-agnet IL methods [3, 26].

Each of these proposed solutions involve training a policy from joint demonstrations, where expert trajectories involve all N agents working together simultaneously in noisily-optimal demonstrations. However, a dateset of joint demonstrations [1] assumes unrealistic access to synchronized demonstrations for all agents, or a pretrained RL demonstrator policy [30], in which case we have no need to deploy imitation learning. These assumptions may be difficult to satisfy in real-world in-the-wild tasks and none of the discussed 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. Our work is the first to show successful multi-agent IL under realistic assumptions about the human's ability to demonstrate multi-agent tasks. Our work shows that a sufficient policy can be trained by iteratively providing demonstrations to individual agents, one at a time, enabling deployment to lone-instructor tasks.

III. Problem Formulation

We seek to train a set of N robot agents to collaborate while successfully accomplishing a task. Following existing nomenclature, we formulate the multi-agent learning from demonstrations problem as a Markov (Stochastic) Game represented as the tuple $\langle \mathcal{I}, \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T} \rangle$ consisting of a set of agents $\mathcal{I} = \{1, 2, \ldots, N\}$, states $\mathcal{S}$, and joint action-space $\mathcal{A} = A_1 \times A_2 \times \cdots \times A_N$. In this work, we do not learn directly from rewards, but it is worth noting that we focus on games with *shared reward* (also called *common reward*), where all agents have the same goals and receive the same reward signal at time t. The transition dynamics of the environment, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$, represents the probability of transitioning from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ after agents take joint action $a \in \mathcal{A}$.

Let an expert trajectory with finite horizon, T, be defined as the sequence of states visited by the expert and the associated action taken at that state, $\tau = (s_0, a_0, s_1, a_1, \ldots, s_T)$, 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 nosily-optimal demonstrations, $\mathcal{D}$, imitation learning seeks to learn a 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 imitation learning is

to find the parameters, θ, to a policy, π_θ, such as to minimize the error between the demonstrated expert's actions, $\pi^*(s)$, and the outputs of the learned policy, $\pi_\theta(s)$, given by the objective function

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

where $\mathcal{L}$ is a loss function that penalizes divergence from the experts actions, commonly the squared error $||\pi_\theta(s_t) - \pi^*(s_t)||^2$.

Contrary to the single-agent literature, we assume that a single human *cannot* teleoperate the entire system with optimal actions, therefore they 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 this system increases with the number of agents [14] or (2) teleoperation interfaces cannot control all degrees of freedom for complex systems. However, we assume that the human is 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

$$\tau_i = (s_0, a_{i,0}, s_1, a_{i,1}, \ldots, s_T) \tag{2}$$

where $a_{i,t} \in A_i$ is an *individual agent's action*, not the joint action of all agents. Note that the state remains unchanged here, and the other agents will still be operating in the environment, which the expert actions should account for.

Given N sets of single-agent demonstrations, $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N\}$, we seek to learn a policy that achieves sufficient collective performance, using *only* individual demonstrations.

IV. Methods

In this section, we introduce Round-Robin Behavior Cloning (R2BC) (Fig. 1), a method that enables a single human operator to cycle through each agent individually and provide a demonstration of the task. At regular intervals, the policies for all agents are updated using the respective buffer of demonstration data provided to each agent. To iteratively learn a collaborative policy, the demonstrator continues to provide actions (in a round-robin fashion) for each agent while the remaining agents execute the learned policy,

First, we introduce our method and the learning paradigm for both centralized and independent policy architectures (Section IV-A). Then, we offer some theoretical insight into how R2BC can produce more robust imitation policies than traditional behavior cloning (Section IV-C).

A. Round-Robin Behavior Cloning

Round-Robin Behavior Cloning (R2BC) is 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 all agents to be simultaneously demonstrated by an expert (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 using their current learned policies. This setup enables the realistic collection of diverse, on-policy training data across a wide distribution of states.

Algorithm 1 Round-Robin Behavior Cloning (R2BC)

Require: Number of agents N, expert policy π^*, initial agent policies $\{\pi_1, \ldots, \pi_N\}$, demonstration buffers $\{D_1, \ldots, D_N\}$, update frequency k

Initialize iteration counter $c \leftarrow 0$
while not converged **do**
  for each agent $i = 1$ to N **do**
    Reset environment to initial state s_0
    for $t = 0$ to T **do**
      Agent i receives expert action $a_{i,t} \leftarrow \pi^*(s_t)$
      for each agent $j \neq i$ **do**
        Agent j takes action $a_{j,t} \leftarrow \pi_j(s_t)$
      end for
      Execute joint action $a_t = (a_{1,t}, \ldots, a_{N,t})$
      Observe next state s_{t+1}
      Append $(s_t, a_{i,t})$ to D_i
    end for
  end for
  if $c \bmod k = 0$ **then**
    Update policies $\pi_{1\ldots N}$ using BC on $D_{1\ldots N}$.
  end if
  $c \leftarrow c + 1$
end while

R2BC is designed to terminate when the demonstrator indicates that task performance has converged. In practice, we cycle over agents multiple times, collecting trajectories for each and performing policy updates every k iterations for a specified number of single-agent demonstrations.

B. Centralized vs Independent Variants

While Algorithm 1 is described assuming independent policies, where each agent i has a distinct policy π_i, our method is also compatible with a centralized architecture that models the joint action distribution. In this section, we describe how R2BC accommodates both policy structures.

In the independent case, each agent maintains its own policy $\pi_i : \mathcal{O}_i \to \mathcal{A}_i$, where $\mathcal{O}_i \subseteq \mathcal{S}$ is the local observation space. Demonstrations for each agent are stored separately, and only the policy π_{θ_i} is updated using its corresponding buffer D_i:

$$L(\theta_i) = \sum_{(s,a_i)\sim D_i} \|\pi_{\theta_i}(s) - a_i\|^2.$$

In the centralized case, a single policy $\pi_\theta : \mathcal{S} \to \mathcal{A}$ predicts the joint action from the global state. During training, we apply

the loss only to the subset of output dimensions corresponding to the demonstrated agent. Specifically, we define:

$$L(\theta) = \sum_{i=1}^{N} \sum_{(s,a_i)\sim D_i} \|\pi_\theta(s)[i] - a_i\|^2,$$

where $\pi_\theta(s)[i]$ denotes the predicted sub-action for agent i (i.e., the output slice of the joint action vector corresponding to agent i).

Both implementations share the same round-robin data collection process, where only one agent is demonstrated at a time and the others act using their most recent learned policies. Centralized models benefit from access to full state information and tighter coordination, while independent policies scale more easily and are suitable for partially observable settings. In our experiments, we study both variants of R2BC to assess the relative performance across diverse multi-agent tasks.

C. Reduced Covariate Shift

Compared to behavior cloning, we hypothesize that R2BC implicitly reduces the covariate shift by diversifying the behavior of the other agents while demonstrating optimal actions to the i-th agent. A centralized behavior cloning paradigm would require coordinated near-optimal actions jointly taken by all agents, limiting the training states to only states where all agents are acting optimally. R2BC relaxes this distribution and allows states where only the demonstrating agent is acting optimally.

While we do not provide a theoretical foundation for reduced covariate shift in this work, we provide empirical evidence for this claim in our experiments and plan to prove this in future work. We believe our method may have similar theoretical improvements as other online imitation learning methods such as DAGGER [24] and demonstrator noise injection methods like DART [15].

V. Experiments

We demonstrate the efficacy of round-robin behavior cloning approaches on 4 multi-agent tasks in simulation. Our experiments were designed to test the following hypotheses:

H1: Both centralized and independent R2BC will match or exceed the performance of joint-action IL methods, using the same number of demonstrations.

H2: An independent round-robin variant of DAgger [24], R2DAgger, will match or exceed the performance of DAgger using an oracle joint-action demonstrator.

H3: An independent round-robin variant of DART [15], R2DART, will match or exceed the performance of DART using an oracle joint-action demonstrator.

H4: R2BC methods will empirically reduce the covariate shift compared to offline joint-action behavior cloning, indicating a better ability to generalize to the deployment state distribution.

We design two experiments across 4 multi-agent tasks in simulation to test our hypotheses. First, we evaluate the performance of our R2BC agents under the withheld ground-truth reward to measure task performance and test H1, H2 and H3. Second, we compare behavior cloning loss metrics from the training data to the state of states observed in the evaluation environments to test H4 and provide empirical support for the theoretical hypothesis formed in section IV-C. The full details of our experimental setup are described in section V-B.

A. Environments

We evaluate our methods in the Vectorized Multi-Agent Simulator (VMAS) [2] on 4 cooperative tasks.

- **Navigation**: *N* agents are randomly positioned in a two-dimensional space each with a designated goal. Agents can utilize LiDAR-style sensors to avoid colliding with each other *en route* to their objectives.
- **Balance**: *N* 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**: *N* agents cooperate to push package(s) having a set shape set and mass into a designated goal region. By default, packages are significantly heavier than the individual agents can push on their own, requiring the agents to work together to transport the package.

B. Methods and Baselines

We compare four variants of our R2BC method to a set of behavior cloning baselines that have access to an oracle demonstrator. For each environment, the methods are compared under the same number of total demonstrations, which vary between domains depending on the difficulty of the task. Each policy is trained using a MSELoss until convergence with a fixed learning rate of 1×10^{-3} and a minibatch size of 256 state-action pairs.

After training, we evaluate the learned policies on the same identical 50 initialization seeds and compute the average environment reward. One benefit of testing in simulated environments with synthetic demonstrators is that we can measure how well our policies are able to generalize to the evaluation states by measuring the deviation between between the experts actions and the learned policy for each testing state. To do this, we measure the loss on the training dataset and the loss between the learned policy's actions and the experts action for each test trajectory in the evaluation environments and compare the gap both losses (Fig. 2, bottom).

1) Joint Behavior Cloning (JBC): JBC serves as an oracle behavior cloning method that learns from coordinated demonstrations provided in the joint-action space. Recall from our problem statement, that demonstrations of this form require infeasible teleoperation degrees of freedom or a many-to-many

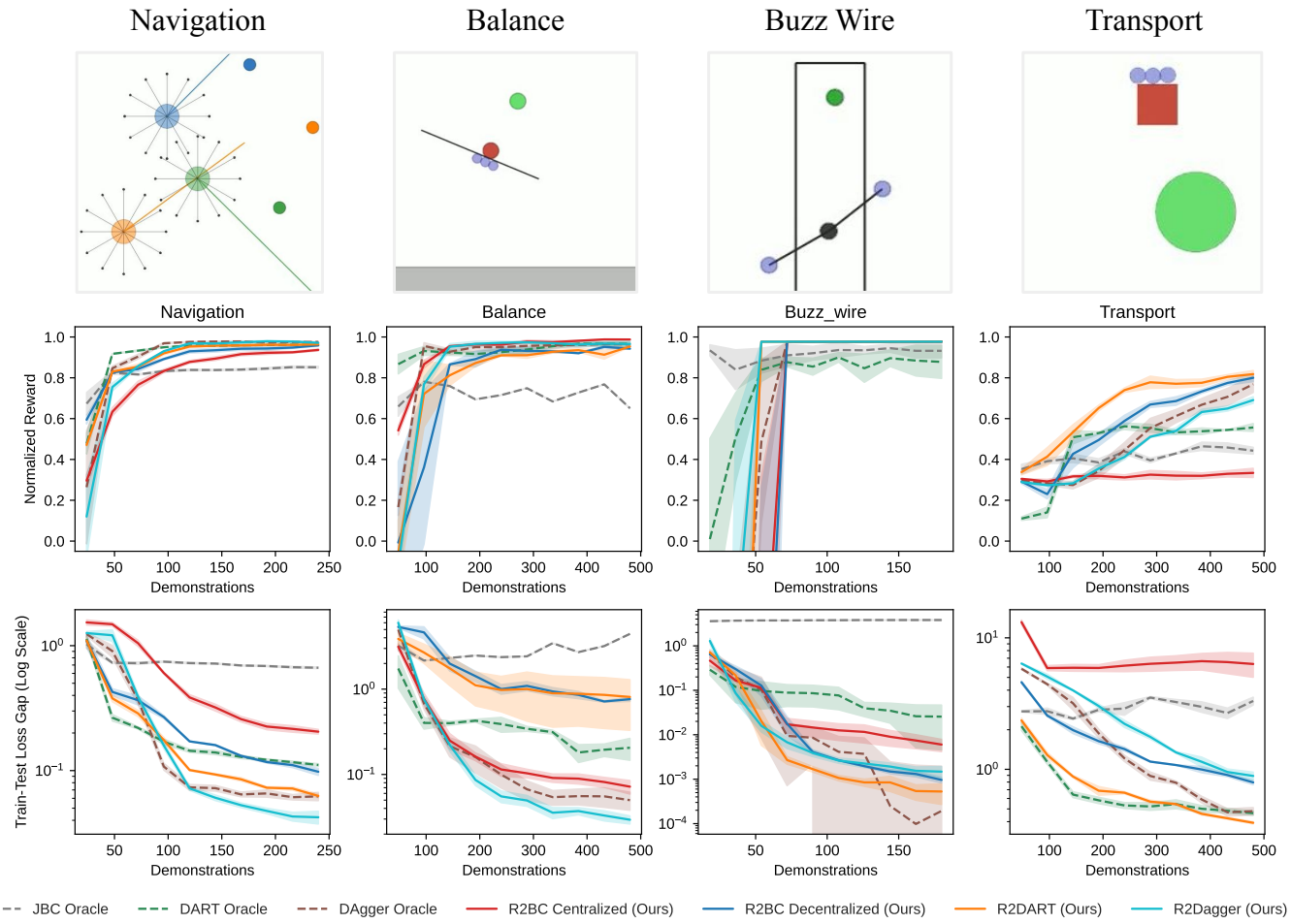

Fig. 2. **Main Results:** We compare 4 R2BC methods to a set of 3 baselines that assume oracle (privileged) access to a joint-action demonstrator. Results are averaged over 3 seeds with error bars indicating standard error. **Top:** 4 multi-agent tasks selected from the Vectorized Multi-Agent Simulator (VMAS) [2] including Navigation ($N = 3$), Balance ($N = 3$), Buzz Wire ($N = 2$), and Transport ($N = 3$). **Middle:** The performance of each method is shown with increasing number of expert demonstrations. In each case, one of our R2BC implementations matches or outperforms the baseline method. The rewards are normalized between 0.0 (random policy) and 1.0 (demonstrator performance). **Bottom:** The difference between the training loss on the demonstration set and testing loss on the evaluation trajectories. Lower values are considered better as the gap between training and evaluation loss closes.

teleoperation scheme where N human operators control N robots. Therefore, we consider this an oracle method that has access to unrealistic data, but reflects the currently proposed assumptions about coordinated demonstration in multi-agent imitation learning [1, 8, 3].

2) DAgger w/ Oracle Actions: A centralized multi-agent implementation of DAgger [24] where a joint-action expert provides corrective labels to on-policy rollouts.

3) DART w/ Oracle Actions: An centralized multi-agent implementation of DART [15] where noise is injected into the demonstrators control online to vary the state distribution and allow the demonstrator to correct the robot in bad states.

4) Centralized-R2BC: A centralized policy implementation of our approach. Demonstrations are provided for one agent at a time, reflecting more realistic assumptions about human teleoperators. The loss is applied over the output indicies corresponding to the i-th agent receiving demonstrations.

5) Independent-R2BC: Same as Centralized-R2BC but with an independent policy implementation with local observations as input instead of the global state.

6) R2DAgger: A decentralized round-robin implementation of DAgger [24], where the demonstrator provides a corrective actions for only one agent in each demonstration.

7) R2DART: A decentralized round-robin implementation of DART [15]. This is identical to Independent-R2BC with noise injected into the demonstrator's policy.

C. Expert Demonstrations

To quickly evaluate our approach and perform an analysis of covariate shift, we used a synthetic demonstrator trained using reinforcement learning. For all tasks except Transport, we trained a centralized PPO policy [25] for 6 million timesteps using the VMAS default hyperparameters [2]. The learned policy takes the state and input and outputs a joint action. This satisfies the requirements of the JBC baseline, which requires

coordinated and centralized demonstrations. For R2BC, we use the output logits corresponding to the ith agent to get a single-agent action. For the Transport task, we found that the VMAS heuristic policy performed better than any RL policy which is what we used as the demonstrator for that task. Modeling expert policies as centralized allows us to give demonstrations to the entire set of agents, and to individual agents (by indexing into the joint action), ensuring that the same expert demonstrator is used across all methods. In the continuation of this work, we plan to add experiments with a real human providing demonstration to individual agents via teleoperation.

VI. Results

A. Task Performance

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 expert demonstrator. As standard in imitation learning, the expert sets an upper bound that is typically not exceeded by the learned policy.

In *navigation*, *balance*, and *buzz wire*, both the centralized and decentralized R2BC methods achieve better performance than JBC after 96, 144, and 54 single-agent demonstrations, respectively. Notably, in *transport*, only the decentralized R2BC method is able to surpass the performance of JBC, while the centralized method fails to improve with more demonstrations. We hypothesize that this is due to the challenging nature of the transport task, which has a much larger variance in starting configurations than the other tasks, making regression over the 6 dimensional centralized action space more difficult from single-agent demonstrations than it is in the independent and oracle counterparts. These results indicate strong support for H1 in the context of independent policies, with some support for H1 in the centralized case. In each environment, at least one variant of R2BC achieves parity with, or surpasses, JBC—despite JBC having access to unrealistic, fully coordinated joint-action demonstrations. This observation supports our claim that R2BC offers a viable and scalable alternative for teaching multi-agent systems with only single-agent expert supervision.

Across *navigation*, *balance*, and *buzz wire*, R2DAgger performs significantly better than DAgger and in *transport* achieves similar performance as DAgger. This indicates strong support for H2. For *navigation* and *balance*, R2DART achieves parity with the oracle DART method with increasing demonstrations. Surprisingly, in both *buzz wire* and *transport*, R2DART performs significantly better than DART, which appears to plateau in performance early on (after 54 and 144 demonstrations, respectively). We believe this improvement is a result of the compounded diversity experienced under R2DART which observes varying state in the other agents (R2) in addition to control noise injected into the demonstrators actions (DART). This indicates strong support for H3.

B. Covariate Shift

To empirically measure covariate shift (H4), we compare the training and testing loss gap of each method, as shown in the bottom row of Figure 2. Following previous experiments studying empirical measures of covariate shift [15], we interpret the convergence of the testing loss toward the training loss as evidence of improved generalization and reduced covariate shift.

We find that, in almost all cases, the four round-robin methods all exhibit a similar decrease in the train-test gap as the two online oracle baselines (DART and DAgger) and exhibit significantly less evaluation loss compared to the offline oracle behavior cloning (JBC). The one exception to this is the centralized R2BC method in the transport task, which very quickly plateaus in the loss graph indicating a failure to uncover new data that leads to improved generalization. While H4 is not yet theoretically supported, these results are compelling evidence that round-robin methods are inherently served by similar online data diversity benefits as the oracle DAgger and DART methods, leading to improved test time generalization. Therefore our results carry some empirical support for H4.

Another notable trend emerges from our empirical data—in environments where the test loss converges to the training loss, we observe a corresponding spike in task performance—often approaching the expert level. This empirical correlation suggests that the alignment between training and deployment distributions can serve as a useful diagnostic for policy quality in imitation learning, particularly in multi-agent settings where covariate shift can be severe.

VII. Conclusion

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 provide demonstrations to one agent a time?* Our algorithm, R2BC, utilizes round-robin single-agent demonstrations to gather examples of what each agent should do with respect to both the environment and the other autonomous agents. Using these online demonstrations, we show that a synthetic demonstrator can actually achieve greater task performance using our algorithm than when providing joint-action coordinated demonstrations for all N agents simultaneously. We hypothesize that the online nature of R2BC is able to implicitly reduce the covariate shift compared to JBC and therefore correctly infer actions in unseen states at runtime.

We are eager to continue this work and enable the real-world deployment of R2BC. While our initial results are promising, our results indicate that the reliability of our method may depend on the type of task/environment that R2BC is deployed in. Therefore, we plan to conduct additional analysis to determine which domains and tasks R2BC is best suited for, including testing on a more diverse set of environments. We also plan to deploy this method for real-world mobile robotics tasks and evaluate the cognitive burden on humans providing R2BC feedback online to robots in the real world.

References

[1] 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*, volume 39, pages 23126–23134, 2025.

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

[3] 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)*, pages 1534–1539. IEEE, 2018.

[4] Daniel Brown, Scott Niekum, and Marek Petrik. Bayesian robust optimization for imitation learning. *Advances in Neural Information Processing Systems*, 33: 2479–2491, 2020.

[5] Thanpimon Buamanee, Masato Kobayashi, Yuki Uranishi, and Haruo Takemura. Bi-act: Bilateral control-based imitation learning via action chunking with transformer. In *2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 410–415. IEEE, 2024.

[6] 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*, 3(4):362–369, 2019.

[7] Ildar Farkhatdinov and Jee-Hwan Ryu. Teleoperation of multi-robot and multi-property systems. In *2008 6th IEEE International Conference on Industrial Informatics*, pages 1453–1458. IEEE, 2008.

[8] 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.

[9] 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. *arXiv preprint arXiv:2109.08273*, 2021.

[10] Ryan Hoque, Ashwin Balakrishna, Carl Putterman, Michael Luo, Daniel S Brown, Daniel Seita, Brijen Thananjeyan, Ellen Novoseller, and Ken Goldberg. Lazydagger: Reducing context switching in interactive imitation learning. In *2021 IEEE 17th international conference on automation science and engineering (case)*, pages 502–509. IEEE, 2021.

[11] 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 *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6. IEEE, 2023.

[12] Ahmed Hussein, Eyad Elyan, Mohamed Medhat Gaber, and Chrisina Jayne. Deep imitation learning for 3d navigation tasks. *Neural computing and applications*, 29:389–404, 2018.

[13] Zaynah Javed, Daniel S Brown, Satvik Sharma, Jerry Zhu, Ashwin Balakrishna, Marek Petrik, Anca Dragan, and Ken Goldberg. Policy gradient bayesian robust optimization for imitation learning. In *International Conference on Machine Learning*, pages 4785–4796. PMLR, 2021.

[14] Julian Kaduk, Müge Cavdan, Knut Drewing, 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)*, pages 1207–1212. IEEE, 2024.

[15] Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on robot learning*, pages 143–156. PMLR, 2017.

[16] Hoang M Le, Yisong Yue, Peter Carr, and Patrick Lucey. Coordinated multi-agent imitation learning. In *International Conference on Machine Learning*, pages 1995–2003. PMLR, 2017.

[17] 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.

[18] 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)*, pages 378–384. IEEE, 2020.

[19] 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.

[20] Jayam Patel, Tyagaraja Ramaswamy, Zhi Li, and Carlo Pinciroli. Transparency in multi-human multi-robot interaction. *arXiv preprint arXiv:2101.10495*, 2021.

[21] Xinyue Qi, Jianhang Tang, Jiangming Jin, and Yang Zhang. Diffusion-based multi-agent reinforcement learning with communication. In *2024 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, pages 1–6. IEEE, 2024.

[22] Ilija Radosavovic, Xiaolong Wang, Lerrel Pinto, and Jitendra Malik. State-only imitation learning for dexterous manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7865–7871. IEEE, 2021.

[23] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Confer-

ence Proceedings, 2010.

[24] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[25] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[26] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. Multi-agent generative adversarial imitation learning. *Advances in neural information processing systems*, 31, 2018.

[27] William G Squires and Sean Luke. Lfd training of heterogeneous formation behaviors. In *AAAI Spring Symposia*, 2018.

[28] 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)*, pages 4731–4738. IEEE, 2024.

[29] Jan Ole von Hartz, Tim Welschehold, Abhinav Valada, and Joschka Boedecker. The art of imitation: Learning long-horizon manipulation tasks from few demonstrations. *IEEE Robotics and Automation Letters*, 2024.

[30] 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)*, pages 4092–4099. IEEE, 2019.