Hierarchical and Heterogeneous MARL for Coordinated Multi-Robot Box-Pushing Environment

Ebasa Temesgen, Sarah Boelter, Maria Gini

University of Minnesota Twin-Cities
100 Union St SE, Minneapolis, MN 55455, USA
temes021@umn.edu, boelt072@umn.edu, gini@umn.edu

Abstract

Consider an environment where at least two robotic agents are surrounded by small and large-sized objects that have to be moved from their current locations to designated goal locations. Small objects can be pushed by one robotic agent while large objects need to be pushed by two robotic agents. The objective is to minimize the total time required to move all the boxes to their respective goal locations. In this paper, we explore a solution using a cooperative Partially Observable Markov Decision Process (POMDP) utilizing Proximal Policy Optimization algorithms in a simulated environment.

Introduction

Achieving robust cooperation among multiple autonomous robotic agents is a central challenge in Multi-Agent Reinforcement Learning (MARL). Tasks intuitive for humans, such as coordinated object manipulation, are challenging to robotic agents, requiring agents to effectively share loads, navigate cluttered environments, and adapt their behaviors to diverse embodiments. While substantial progress has been made in this field, key performance gaps and theoretical unsolved problems remain in MARL.

The primary goal of this paper is to study the problem of multi-agent cooperation with MARL using Proximal Policy Optimization (PPO) (**?**) methods and translate the findings into a multi-robot system.

Recent work (Bettini et al. 2024) on the Vectorized Multi-Agent Simulator (VMAS) benchmark suite provides information on challenges that centralized training methods face in certain environments. Specifically, the authors report that only Independent Proximal Policy Optimization (IPPO) successfully converges to the optimal policy in the transport scenario, as shown in Figure 1. This is attributed to the exploration difficulties that arise when using centralized critics, as they must process the concatenated observations from all agents. This high-dimensional input leads to a large variance in possible joint states and a correspondingly low likelihood of revisiting similar states. Consequently, centralized methods often fail to generalize, while IPPO, relying solely on local observations, proves to be more adept at handling such complexity.

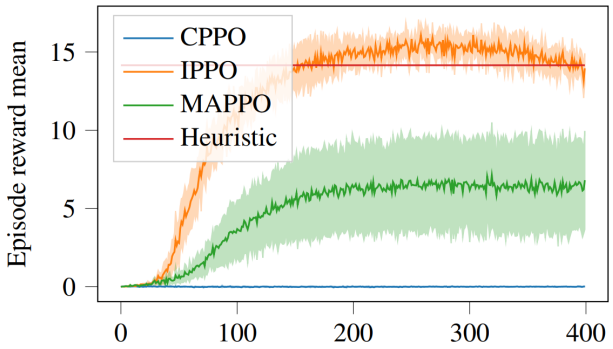

Figure 1: Performance of multiple MARL algorithms, where IPPO is the only algorithm that successfully converges to the optimal policy (Bettini et al. 2024).

This motivates the core question of our work: Why does IPPO outperforms Multi-Agent Proximal Policy Optimization (MAPPO) (Yu et al. 2022) in this cooperative box-pushing scenario? If IPPO's decentralized training paradigm excels due to simpler exploration dynamics or better handling of partial observability, these could inform new approaches to overcome MAPPO's limitations.

Building on this, we consider two avenues to enhance MARL systems further. First, we explore heterogeneity. Real-world teams of robots often differ in their dynamics, sensing modalities, and capabilities. Existing MARL algorithms frequently assume homogeneous agents or rely on heuristics to differentiate behaviors. By introducing Heterogeneous Graph Neral Network Policy Optimization (HetGPPO) (Bettini, Shankar, and Prorok 2023) an algorithm designed to handle heterogeneous agents explicitly, our goal is to understand how agent differences impact the interplay between decentralized and centralized strategies. Comparing IPPO, MAPPO, and HetGPPO in controlled settings will help isolate factors such as agent specialization and diverse sensing that influence overall performance and stability.

Second, there is interest in exploring hierarchical structures within MARL frameworks, such as those proposed in recent works on hierarchical multiagent reinforcement learning (HRL) (Hong et al. 2024). While IPPO might inform us how local and decentralized intelligence aids explo-

ration and adaptation, hierarchical decomposition can further break down complex manipulation tasks into manageable subtasks. High-level controllers can set strategic goals, such as selecting objects or navigating obstacles, while mid-level and low-level policies execute these objectives. The combination of insights from IPPO's advantages, HetGPPO's handling of heterogeneity, and hierarchical control promises a more principled design of MARL solutions that scale to cluttered environments and extended planning horizons.

In parallel, the demand for more realistic scenarios has led to experiments with heterogeneous teams of robots and varied motion models (Boroson and Ayanian 2019). While initial studies may focus on homogeneous platforms in simplified 2D domains like VMAS, bridging the gap to real-world applications requires considering more diverse hardware platforms and environments. For instance, combining differential-drive mobile bases and quadrupedal robots in physically rich 3D simulations (e.g., IsaacGym (Makoviychuk et al. 2021)) can provide critical insights into how well-learned policies generalize across embodiments. Such heterogeneity also motivates approaches like HetGPPO, which leverage structured representations of agent differences, such as distinct morphologies or sensory modalities to facilitate learning robust cooperative strategies.

In this paper we present a mathematical formulation of the methods we will use. We later outline a plan for the experimental work that we will do to analyze in simulation the performance of the three MARL algorithms (MAPPO, IPPO, and HetGPPO) in a custom-controlled scenario.

Related Work

Collaborative MARL

Coordination by robotic agents to complete a single task has real-world applications in a wide variety of domains including distribution warehouses, search and rescue, and spacecraft collaboration and communication (Adams et al. 2024). Our focus area primarily fits a warehouse-like environment, where robotic agents need to carry or push boxes or crates from one starting location to a goal location, usually with obstacles taking the form of other boxes or other agents. Specific to this sort of scenario, there have been several advances in multi-agent collaboration for task completion.Recently, pairs of quadruped robots were used for long-horizon pushing of boxes from a starting location to a goal location while remaining aware of obstacles (Feng et al. 2024). Others have used larger numbers of fully decentralized puck robots to push objects much larger in size to goal locations using whole body (Chen et al. 2015). Fully decentralized systems of robots have also used manipulation to transport much larger objects with no prior knowledge of the shape or size of the object, relying fully on angular velocity and center of mass measurements to carry the object from the start to the goal location (Culbertson and Schwager 2018).

In addition, traditional MARL frameworks cannot explicitly accommodate policy heterogeneity and typically constrain agents to share neural network parameters. This causes the agents' models to be identical and, thus, homogeneous. While this is beneficial to speed up training, it can prevent learning in scenarios that require truly heterogeneous behavior. While heterogeneous policies for robots with different goal objectives are fairly common practice, there is a scarcity of work dealing with heterogeneous policies single single-objective problems. This would allow robots with similar hardware to exhibit different behaviors despite having similar inputs (Bettini, Shankar, and Prorok 2023).

Proximal Policy Optimization

PPO is a group of policy gradient methods for reinforcement learning that alternate between sampling data through interaction with the environment and optimizing a "surrogate" objective function using stochastic gradient ascent. PPO uses an objective function that enables multiple epochs of mini-batch updates (**?**). Alternative methods have drawbacks. For example, Q-Learning with function approximation is not well understood and does not perform well outside of game environments. Vanilla policy gradient methods have poor data efficiency and robustness, and trust region policy optimization can be complicated and not compatible with noisy architectures. (**?**)

The academic focus on MARL has been on utilizing off-policy learning frameworks like Multi-Agent Deep Deterministic Policy Gradient (MADDPG) (Lowe et al. 2017) and Q-learning (Phan et al. 2024) (Sunehag et al. 2017). However, recently there has been a focus on PPO, an on-policy algorithm underutilized in MARL (Yu et al. 2022). PPO-based multi-agent algorithms achieve surprisingly strong performance in four popular multi-agent testbeds: the particle-world environments, the StarCraft multi-agent challenge, Google Research Football, and the Hanabi challenge, with minimal hyperparameter tuning and without any domain-specific algorithmic modifications or architectures. Compared to competitive off-policy methods, PPO often achieves competitive or superior results in both final returns and sample efficiency(Yu et al. 2022).

Two additional MARL models of interest are the homogeneous Graph Neural Network Proximal Policy Optimization (GPPO) and its heterogeneous counterpart Heterogeneous HetGPPO (Bettini, Shankar, and Prorok 2023) GPPO builds upon Independent Proximal Policy Optimization (Bettini et al. 2024). GPPO overcomes the limitations of IPPO while maintaining its benefits. It uses a graph neural network (GNN) (Scarselli et al. 2008) communication layer, allowing agents to share information in neighborhoods to coordinate and overcome partial observability.

Preliminaries

Decentralized Partially Observable Markov Decision Process

Before we can formulate the cooperative multi-agent problem as a Decentralized Partially Observable Markov Decision Process (DEC-POMDP) with shared rewards (Oliehoek and Amato 2016), we must define our environment and agents.

Let $\mathcal{A} = \{a_1, a_2, \ldots, a_N\}$ denote the set of N agents. Let $\mathcal{B} = \{b_1, b_2, \ldots, b_M\}$ denote the set of M boxes, each box with the size of the attributes, denoted by $s_b \in \{\text{small}, \text{large}\}$ and mass, denoted by $m_b \in \mathbb{R}^+$.

Time in our environment is measured by time t, the environment state at time t is given by S_t, encompassing the states of all agents and boxes.

A DEC-POMDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, P, R, n, \gamma)$, where $\mathcal{S}$ is a set of global states S_t, $\mathcal{A}$ is the action space for each agent, with the joint action space $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$, $\mathcal{O}$ is the observation space for each agent, where $o_i^t = \mathcal{O}(S_t; i)$ is the local observation for agent a_i at time t. P is the state transition probability function $P(S_{t+1}|S_t, A_t)$, with $A_t = (a_1^t, \ldots, a_n^t)$ being the joint action of all agents. R is the shared reward function $R(S_t, A_t)$, n is the number of agents, and $\gamma \in [0, 1)$ is the discount factor.

At each time step t, each agent a_i receives a local observation o_i^t based on the global state S_t. Agents select actions a_i^t according to their individual policies $\pi_{\theta_i}(a_i^t|o_i^t)$, parameterized by θ_i. The environment transitions to a new state S_{t+1} according to the transition function P. All agents receive a shared reward $R_t = R(S_t, A_t)$. The agents aim to maximize the expected cumulative discounted reward:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t R(S_t, A_t)\right],$$

where $\pi_\theta = \{\pi_{\theta_1}, \ldots, \pi_{\theta_n}\}$ represents the set of all agent policies.

Methods

In this work, we address the cooperative object manipulation task in a DEC-POMDP setting, where multiple robotic agents must coordinate to push objects of varying sizes and masses to designated goal locations. Our approach centers on leveraging PPO-based multi-agent extensions, primarily IPPO, to achieve efficient and scalable learning of complex collaborative strategies. We organize our approach into three key components: (1) environment design and state representation, (2) learning algorithms and network architectures, and (3) training protocols and evaluation frameworks.

Multi-Agent Proximal Policy Optimization

Our approach employs MAPPO (Yu et al. 2022) within the Centralized Training with Decentralized Execution (CTDE) framework. In MAPPO, we have policies, where each agent a_i has a policy $\pi_{\theta_i}(a_i^t|o_i^t)$ that depends only on its local observation o_i^t, and a centralized value function, a shared value function $V_\phi(S_t)$ parameterized by ϕ, which takes the global state S_t as input and is used for variance reduction during training. Our clipping function can be defined as c.

The surrogate objective for MAPPO is formulated as:

$$L(\theta) = \mathbb{E}_t\left[\sum_{i=1}^{n} \min\left(r_i^t(\theta)\hat{A}_i^t,\ \mathrm{c}\left(r_i^t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_i^t\right)\right]$$

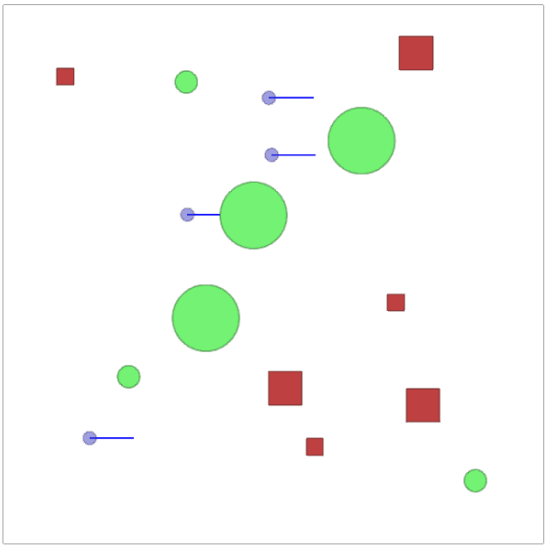

Figure 2: Visualization of a multi-agent environment showing the four circular agents (purple circles) with their orientation represented by blue directional lines. The environment includes three small boxes and three large boxes (red squares), each with corresponding goal positions (green circles).

where $r_i^t(\theta) = \frac{\pi_{\theta_i}(a_i^t|o_i^t)}{\pi_{\theta_i^{\text{old}}}(a_i^t|o_i^t)}$ is the probability ratio between the current and previous policies, $\hat{A}_i^t$ is the advantage estimate for agent a_i at time t, computed using the centralized value function $V_\phi(S_t)$, and ϵ is the clipping parameter to limit the policy update step size.

In practice, MAPPO's centralized critic often improves coordination and leads to stronger overall performance than purely independent training methods, especially in complex tasks where global context aids in the credit assignment.

Independent Proximal Policy Optimization IPPO involves each agent independently learning both its policy and value function based solely on local observations o_i^t, without access to global state information even during training. In our fully cooperative setting with shared rewards, MAPPO's use of a centralized value function provides better coordination among agents and often leads to improved performance (Yu et al. 2022). While IPPO shows that decentralized methods can excel, it may struggle in settings with heterogeneous agents or partial observability.

Heterogeneous Graph Neural Network PPO HetGPPO addresses these challenges by integrating graph neural network (GNN) layers into the PPO framework. Agents communicate their local embeddings through a graph structure, enabling richer information exchange and improved coordination without requiring full global state access.

HetGPPO allows heterogeneity in agent morphologies, capabilities, and sensors by letting each agent process its unique observation space and share encodings with neigh-

bors. The GNN-based communication mitigates the non-stationarity and enhances cooperation in complex, multi-robot scenarios where differing embodiments and roles are integral. In essence, HetGPPO inherits IPPO's simplicity and scalability by adding structured communication to handle heterogeneous teams and challenging partially observable environments.

Mathematical Formulation

With the objective function, the goal is to find an optimal policy π^* that maximizes the expected cumulative reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{T} \gamma^t R_t \right]$$

For state representation, the state S_t includes the positions of all agents: $\{\mathbf{x}_i^t \in \mathbb{R}^2 \mid i = 1, \ldots, N\}$, positions and velocities of all boxes: $\{\mathbf{y}_b^t, \mathbf{v}_b^t \in \mathbb{R}^2 \mid b = 1, \ldots, M\}$, and goal positions for each box $\{\mathbf{g}_b \in \mathbb{R}^2 \mid b = 1, \ldots, M\}$

Each agent a_i has an action space $\mathcal{A}_i$ consisting of movement actions:

$$\mathcal{A}_i = \{\mathbf{u}_i \in \mathbb{R}^2 \mid \|\mathbf{u}_i\| \leq u_{\max}\}$$

The state-space S_t includes the positions $\mathbf{x}_i^t$ of agents and the positions and velocities $\mathbf{y}_b^t, \mathbf{v}_b^t$ of boxes. Each agent's action space $\mathcal{A}_i$ consists of 2D movement or force commands constrained by maximum bounds. The boxes move according to the summed forces exerted by agents in contact, governed by Newtonian dynamics:

$$m_b \frac{d\mathbf{v}_b^t}{dt} = \sum_{i \in \mathcal{C}_b^t} \mathbf{f}_i^t$$

where $\mathcal{C}_b^t$ is a set of agents in contact with box b at time t and $\mathbf{f}_i^t$ is the force applied by agent a_i.

The reward function R_t is designed firstly to encourage agents to reduce their distance to the goal:

$$R_{\text{dist}}^t = -\sum_{b=1}^{M} \left(\delta_b^t - \delta_b^{t-1} \right)$$

where $\delta_b^t = \|\mathbf{y}_b^t - \mathbf{g}_b\|$ is the Euclidean distance from box b to its goal at time t. Secondly, encourage cooperation for large boxes:

$$R_{\text{coop}}^t = \sum_{b \in \mathcal{B}_{\text{large}}} \beta \cdot \mathbb{I} \left(|\mathcal{C}_b^t| \geq 2 \right)$$

where $\mathcal{B}_{\text{large}}$ is a set of large boxes, $\beta > 0$ is a cooperation bonus, and $\mathbb{I}(\cdot)$ is an indicator function. Thirdly the reward function is designed to encourage agents toward a goal completion bonus:

$$R_{\text{goal}}^t = \sum_{b=1}^{M} \gamma_b \cdot \mathbb{I} \left(\delta_b^t = 0 \right)$$

where $\gamma_b > 0$ is the goal completion bonus for box b.

To improve exploration, we introduce a heuristic reward R_{explore}^t that encourages agents to try diverse pushing strategies or occupy distinct positions around the box's perimeter. For instance, an occlusion-based heuristic (Chen et al. 2015) can guide agents to approach the box from different sides:

$$R_{\text{explore}}^t = \alpha \sum_{i=1}^{N} f(o_i^t, S_t),$$

where $f(\cdot)$ could measure how "original" or "varied" the contact points are compared to recently visited states, or how well agents position themselves to push the box toward subgoals set by a high-level planner. $\alpha > 0$ is a small weight that ensures this term does not overshadow task completion but still aids in preventing premature convergence to suboptimal strategies. The total reward at time t is:

$$R_t = R_{\text{dist}}^t + R_{\text{coop}}^t + R_{\text{goal}}^t + R_{\text{explore}}^t$$

Conclusions

To summarize the above, we propose the following structured research plan for the remaining work:

1. We will begin by evaluating three MARL algorithms (MAPPO, IPPO, and HetGPPO) in a custom-controlled scenario where agents must push boxes of varying sizes, some requiring only one agent, and others demanding two to identify the conditions driving IPPO's strong performance. This will help us gain insight into why the decentralized approach performs better and help us reaffirm the results reported in the VMAS (Bettini et al. 2024).
2. We will then adopt a Hierarchical MARL Framework that integrates local decentralized policies capitalizing on the strengths of IPPO at the lower levels, improving exploration and adaptability, while employing global, high-level goal-setting policies to address long-horizon planning. By separating decision-making across multiple temporal and abstraction scales, we will handle the complexity of navigation, coordination, and long-horizon planning in cluttered environments. In later stages, we will incorporate differential-drive and quadrupedal robots, exploring the generalization of these hierarchical policies to more complex, heterogeneous platforms.
3. Next we will conduct rigorous testing with heterogeneous robots across a varying number of boxes, from simplified 2D VMAS scenarios to 3D IsaacGym simulations. We will then look at how we can extend it to a heterogeneous robot team. We plan to ultimately move toward physical experiments.

Although this study is in the early stages, our approach is grounded in well-motivated research questions and leverages a comprehensive set of tools. By introducing heuristic rewards to encourage exploration and designing experiments that transition progressively from 2D VMAS scenarios to complex 3D IsaacGym environments, we anticipate a thorough evaluation of each proposed methodology. Overall, our work aspires to yield MARL frameworks that not only explain IPPO's edge in challenging settings but also produce practical, robust multi-robot policies capable of handling diverse tasks with real-world constraints.

References

Adams, C.; Iatauro, M.; Kempa, B.; Levinson, R.; Frank, J.; Gridnev, S.; and Lassiter, C. 2024. Advancing Autonomy in Distributed Space Systems: Results From the Distributed Spacecraft Autonomy Experiment on Starling 1.0. In *38th Annual Small Satellite Conference*.

Bettini, M.; Kortvelesy, R.; Blumenkamp, J.; and Prorok, A. 2024. VMAS: A Vectorized Multi-agent Simulator for Collective Robot Learning. In Bourgeois, J.; Paik, J.; Piranda, B.; Werfel, J.; Hauert, S.; Pierson, A.; Hamann, H.; Lam, T. L.; Matsuno, F.; Mehr, N.; and Makhoul, A., eds., *Distributed Autonomous Robotic Systems*, volume 28, 42–56. Cham: Springer Nature Switzerland. Series Title: Springer Proceedings in Advanced Robotics.

Bettini, M.; Shankar, A.; and Prorok, A. 2023. Heterogeneous multi-robot reinforcement learning. arXiv:2301.07137.

Boroson, E. R.; and Ayanian, N. 2019. 3D keypoint repeatability for heterogeneous multi-robot SLAM. In *2019 International Conference on Robotics and Automation (ICRA)*, 6337–6343. IEEE.

Chen, J.; Gauci, M.; Li, W.; Kolling, A.; and Groß, R. 2015. Occlusion-Based Cooperative Transport with a Swarm of Miniature Mobile Robots. *IEEE Transactions on Robotics*, 31(2): 307–321.

Culbertson, P.; and Schwager, M. 2018. Decentralized Adaptive Control for Collaborative Manipulation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 278–285. Brisbane, QLD: IEEE.

Feng, Y.; Hong, C.; Niu, Y.; Liu, S.; Yang, Y.; Yu, W.; Zhang, T.; Tan, J.; and Zhao, D. 2024. Learning Multi-Agent Loco-Manipulation for Long-Horizon Quadrupedal Pushing. arXiv:2411.07104.

Hong, C.; Feng, Y.; Niu, Y.; Liu, S.; Yang, Y.; Yu, W.; Zhang, T.; Tan, J.; and Zhao, D. 2024. Learning multi-agent collaborative manipulation for long-horizon quadrupedal pushing. arXiv:2411.07104v2 [cs.RO].

Lowe, R.; WU, Y.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; and Mordatch, I. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30.

Makoviychuk, V.; Wawrzyniak, L.; Guo, Y.; Lu, M.; Storey, K.; Macklin, M.; Hoeller, D.; Rudin, N.; Allshire, A.; Handa, A.; et al. 2021. Isaac gym: High performance GPU-based physics simulation for robot learning. arXiv:2108.10470.

Oliehoek, F. A.; and Amato, C. 2016. *A Concise Introduction to Decentralized POMDPs*. SpringerBriefs in Intelligent Systems. Cham: Springer International Publishing. ISBN 978-3-319-28927-4 978-3-319-28929-8.

Phan, D. N.; Hytla, P.; Rice, A.; and Nguyen, T. N. 2024. Cognitive Multi-agent Q-Learning for Cooperation in Competitive Environments.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2008. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1): 61–80.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; and Graepel, T. 2017. Value-decomposition networks for cooperative multi-agent learning. arXiv:1706.05296 [cs].

Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; and Wu, Y. 2022. The surprising effectiveness of PPO in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35: 24611–24624.