To Centralize or Decentralize? Examining Cyber-Physical Control on the Manufacturing Shop Floor

Akash Agrawal
Department of Mechanical Engineering,
Carnegie Mellon University,
5000 Forbes Ave, Pittsburgh, PA 15213
e-mail: aragrawa@andrew.cmu.edu

Sung Jun Won
Naval Architecture and Marine Engineering,
University of Michigan,
500 S. State Street, Ann Arbor, MI 48109
e-mail: sungjun@umich.edu

Tushar Sharma
Faculty of Computer Science,
Dalhousie University,
Halifax, Nova Scotia, Canada B3H 4R2
e-mail: tushar@Dal.Ca

Mayuri Deshpande
Premier Health Partners,
12550 120th Ave NE, Kirkland, WA 98034
e-mail: mvdeshpand@premierhealth.com

Christopher McComb
Department of Mechanical Engineering,
Carnegie Mellon University,
5000 Forbes Ave, Pittsburgh, PA 15213
e-mail: ccm@cmu.edu

ABSTRACT

Multi-agent Reinforcement Learning (RL) frameworks for job scheduling and navigation control of autonomous mobile robots are becoming increasingly common as a means of increasing productivity in manufacturing. Centralized and decentralized frameworks have emerged as the two dominant archetypes for these systems. However, the tradeoffs of these competing archetypes in terms of efficiency, stability, robustness, accuracy, generalizability, and scalability are not well-understood. This work investigates the time efficiency, learning stability, and robustness to operational disruptions of an exemplar decentralized RL framework in comparison to a centralized RL framework. Specifically, several policies with increasing computational budgets are trained using both frameworks and then evaluated on the throughput and safety of the shop floor in static and dynamic tests. We observe that the decentralized framework yields a high performing policy at a significantly lower training budget than the centralized one. However, the centralized framework exhibits superior learning stability as well as robustness to the initialization of robot locations in static testing. Furthermore, we compare the robustness of the frameworks in dynamic tests to find that the decentralized framework provides better compensation for processing delays and failures in real time.

Keywords: Artificial intelligence, Machine learning, Multi-agent reinforcement learning, Manufacturing systems, Intelligent manufacturing, Autonomous mobile robots, Job scheduling, Robotics

1. INTRODUCTION

Industry 4.0 is an information-intensive transformation in manufacturing, powered by advanced technologies such as the internet of things, robotics, and artificial intelligence [1–5], with the goal of enhancing productivity, customization, and safety [6]. Aligned with the Industry 4.0 design principles of decentralization and real-time capability [7], there is an increasing interest in adopting Autonomous Mobile Robots (AMRs) controlled by multi-agent deep Reinforcement Learning (RL) algorithms [8–12] in shop floors. However, the tradeoffs of these decentralized frameworks, in terms of efficiency, stability, robustness, accuracy, generalizability, and scalability, as compared to competing centralized frameworks, are not well-understood. To address this gap, this work investigates the time efficiency, learning stability, and robustness to operational disruptions of an exemplar decentralized RL framework in comparison to a centralized RL framework.

Intelligent manufacturing systems are composed of multiple interacting autonomous agents involved in tasks such as enterprise-level collaboration, process planning, job scheduling, and shop floor control [13–15]. When properly designed, such multi-agent systems can provide benefits like increased flexibility, a higher degree of modularity, better reconfigurability, and more robust adaptability. However, they face distinct challenges as well, including agent organization, agent coordination, and negotiation. Deep RL can be used to overcome several of these challenges, primarily due to the potential to learn coordinated behavior between agents, without an engineer having to encode specific desirable system behaviors [11,16–21].

The agents that make up a multi-agent intelligent manufacturing system can represent a variety of entities, including decision-making software, human operators, and autonomous machines. In a job scheduling task, which is the focus in this work, these entities are tasked with collaborating to complete a variety of parallel and sequential jobs through effective shop floor control. In doing so they must contend with limited resources and timing constraints. Competing approaches for accomplishing job scheduling can generally be categorized as centralized or decentralized frameworks [11,22,23]. A centralized framework holds the advantage of possessing comprehensive system information, enabling a holistic approach to decision-making. However, it can lead to high computational complexity and is prone to deadlocks triggered by a single point of failure. In contrast, a well-designed decentralized framework can be more scalable and enable better local decision-making capabilities in response to dynamic changes in the task environment. Such changes are common in manufacturing environments and may include rush jobs, processing time delays, and even failures like machine breakdowns and robot malfunctions.

This work investigates the efficiency, learning stability, and robustness of a decentralized RL framework for an autonomous mobile robot driven shop floor in comparison to a centralized framework. The rest of the paper is organized as follows. In Section 2, we discuss relevant background and prior literature. In Section 3, we review the specific multi-agent system from a prior work that serves as a testbed for comparing decentralized and centralized RL frameworks for mobile robot driven shop floors. This is followed by the methodology for conducting the comparison. Section 4 details the results of the decentralized versus centralized frameworks' performance on time efficiency, learning stability, and robustness to robot placement and dynamic challenges like delays and failures. Section 5 summarizes the contributions of the work and proposes future directions.

2. BACKGROUND

2.1 Intelligent Manufacturing with Autonomous Mobile Robots

Intelligent manufacturing is actively evolving through the integration of cyber-physical agents [24–27] like autonomous mobile robots [8–12] on the shop floor. The cyber portion of these robots consists of software entities that handle perception, action, and communication to undertake intelligent real-time decision-making [28]. The mobile robots are often tasked with transporting material across machines that process it to finished or semi-finished parts. Human workers perform monitoring on the shop floor by reporting anomalies and doing quality checks on the machines. This multi-agent system works towards meeting various shop floor objectives involving throughput, equipment utilization, energy consumption, maintenance cost, and safety. The various characteristics of such an autonomous mobile robot driven shop floor and the associated Industry 4.0 principles are illustrated in Figure 1 and discussed hereinafter.

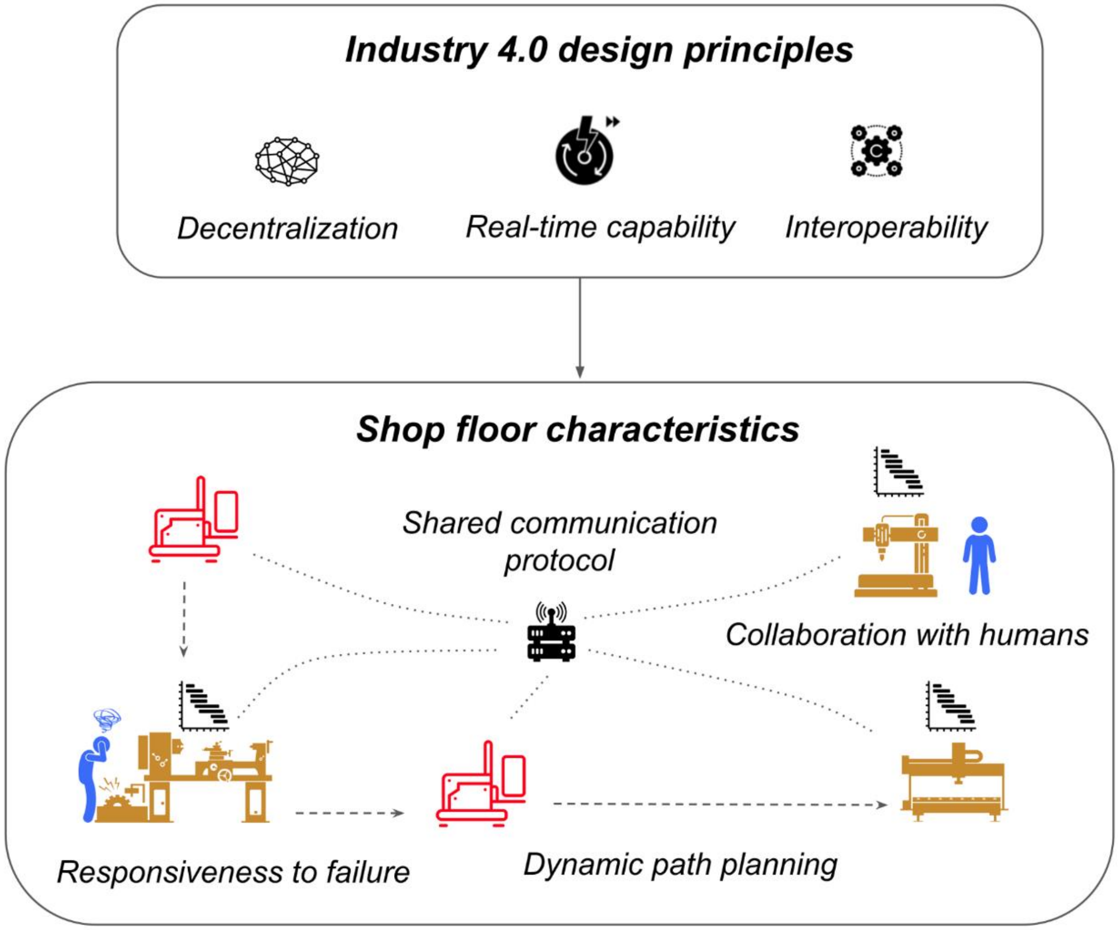

FIGURE 1: CHARACTERISTICS OF AUTONOMOUS MOBILE ROBOT DRIVEN SHOP FLOORS

Autonomous mobile robots, embodying the Industry 4.0 principles of decentralization and real time capability [7], can dynamically plan their path based on sensory information from the shop floor as well as communicated information from other agents. This enables them to undertake decentralized material handling and job scheduling at various machines that process this material. With their physical ability to navigate and handle material freely and cyber ability to make decisions independently, dynamic changes like machine processing delays and rush jobs, and failures like machine breakdowns and robot malfunctions are handled more robustly than traditional manufacturing systems. This reduces congestion and material queues ultimately leading to an improvement in the throughput of the shop floor. Further, all cyber-physical assets communicate with each other through standard protocols. This facilitates interoperability [7] across systems from different manufacturers, a fundamental principle of Industry 4.0 design.

The successful integration of human workers with cyber-physical systems also plays a vital role in the advancement of intelligent manufacturing [29–31]. Humans can work alongside robots, monitoring shop floor activities and reporting any irregularities, thus enhancing the system's real-time capability [7]. Ensuring the safety of human workers in close proximity to robots is a critical objective for such shop floors. Mobile robots should adapt their behavior to compensate for the stochastic nature of human workers' activity in its proximity. The integrated job scheduling and navigation control of mobile robots to meet the objectives of throughput and safety in this context is a challenging problem in intelligent manufacturing systems.

2.2 Reinforcement Learning for Intelligent Manufacturing

Reinforcement learning (RL) [32–36] is a machine learning approach where an agent or multi-agent system learns to make decisions in an environment by trial-and-error, aiming to maximize long-term rewards. It is applicable to problems that can be modeled as Markov decision processes (MDP) [34]. Further, RL can be classified into single-agent RL or multi-agent RL. A single-agent RL framework can involve either a single entity, or multiple entities sending their perceptions and receiving actions from a centralized agent. In the latter case, the centralized agent essentially integrates all the entities as a single combined agent [37]. A single-agent RL framework for a system with multiple entities can provide a strong global optimization capability as the decisions of the agent are based on system-wide information. However, such frameworks can pose challenges involving computational expense and can be more sensitive to unanticipated behavior in stochastic systems [38]. Multi-agent RL frameworks [18,20], consisting of multiple homogeneous interacting agents coexisting in an environment, can learn different regions of a high-dimensional policy space and share their experience to reduce memory and computational cost. Moreover, they are more tolerant to failures as the agents can compensate for each other's failures.

The use of RL is transforming several operational paradigms of manufacturing, including order selection, process planning, job scheduling, robotic manipulation, and motion planning [16,21]. This work focusses on the application of RL for job scheduling and navigation control of autonomous mobile robots. Within this context, the efficiency, stability, robustness, accuracy, generalizability, and scalability of the RL framework is vital to advance its applicability to real-world scenarios [39–42]. Efficiency is characterized by aspects like time efficiency (speed of learning), sample efficiency (training samples needed) and resource utilization (computing systems needed) [40,42,43]. Stability of the learning algorithm entails the consistency of the algorithm's performance as it learns over time, ensuring that overall improvements in decision-making does not introduce variance in the process [41,42]. Robustness [39–41] in the algorithms ensures resilience against diverse shop floor states as well as dynamic changes or failures, allowing robots to re-route and re-schedule tasks without significant delays [8]. Accuracy, particularly in navigation, is essential to avoid collisions and guarantee materials are delivered safely. The generalizability [39] of the algorithms is its ability to cater across various shop floors layouts, machine configurations, and job requirements without extensive retraining [44]. Lastly, scalability [40] ensures that the algorithm can handle an increasing number of robots and machines as manufacturing demands evolve. This work focusses on investigating the time efficiency, learning stability, and robustness to robot locations and operational disruptions of a decentralized RL framework in comparison to a centralized one. The subsequent sections will delve into the background of RL's applicability for job scheduling and navigation in mobile robots, further motivating the investigation of efficiency, stability, and robustness

2.3 Reinforcement Learning for Job Scheduling and Robot Navigation

Job scheduling in manufacturing, aimed at the efficient production of finished or semi-finished parts from raw materials, can be solved using various techniques including heuristics, meta-heuristics, and knowledge-based systems [45–47]. In contrast to these approaches, machine learning-based scheduling approaches [48–52] leverage insights from historical scheduling data and utilize them to outperform heuristic and meta-heuristic approaches. Furthermore, for shop floors involving automated guided vehicles (AGVs), RL algorithms have showcased enhanced performance, efficiency, and robustness than heuristic and metaheuristic approaches [53–56]. In another work, Mayer et al. [23] highlight the generalizability of RL in an AGV driven shop floor to diverse manufacturing scenarios.

Many of the scheduling techniques based on the above algorithms are centralized or hierarchical in nature, leading to problems like high computational effort, high communication cost, lack of adaptability, and lack of recovery when the equipment of the centralized agent malfunctions. Decentralized multi-agent RL frameworks present advantages including flexibility, modularity, reconfigurability, and adaptability. For instance, Dittrich and Fohlmeister [57] have shown that a co-operative multi-agent scheduling system based on RL can achieve better and comparable performances than sequential (round-robin) scheduling and centralized capacity-based scheduling respectively. In AGV driven shop floors, multi-agent RL frameworks showcase enhanced efficiency and robustness as compared to centralized genetic algorithms and RL frameworks [58,59]. More recently, Fang et al. [60] successfully integrated an adaptive mechanism within a digital twin-driven scheduling system by utilizing decentralized RL to counteract manufacturing disruptions like processing delays and machine breakdowns.

Robot navigation can also be achieved using several techniques including classical algorithms, heuristics, and learning-based approaches [61]. Like job scheduling, learning-based approaches offer greater adaptability to dynamic environments by mapping raw sensory observations to navigation controls. More recently, RL is being extensively used for the navigation of ground robots for obstacle avoidance in dynamic environments [62–67]. This approach is suitable for the navigation control of mobile robots to various machines on a manufacturing shop floor. However, similar to centralized job scheduling, centralized navigation control of robots also suffers from computational limitations. For instance, Li et al. [68] investigated a centralized navigation approach that cannot meet real-time demands (0.1 seconds to return planning results) for a large-scale system of AGVs. In response to such challenges, multi-agent approaches to robot navigation have gained traction. For instance, a multi-agent RL algorithm has shown promise for the integrated target assignment and path planning of aerial vehicles [69]. Multi-agent RL has also been used to train aerial vehicles to cooperatively perform field coverage making it suitable for surveillance in intelligent manufacturing shop floors [70]. Further, deep RL has enabled ground and aerial vehicles to form a coalition that is complementary and cooperative for completing tasks that they are incapable of achieving alone [71,72].

When integrated, job scheduling and navigation of autonomous mobile robots is often formulated as a decentralized task allocation problem [73–75]. Like the individual tasks of job scheduling and robot navigation, the combined task also benefits from decentralized approaches. For instance, in multi-robot additive manufacturing involving both scheduling and navigation, a swarm principle-based decentralized algorithm outperforms a centralized method (based on genetic and A* algorithms), in terms of efficiency, robustness, and scalability [76]. Malus et al. [9] focused on an autonomous mobile robot shop floor wherein decentralized RL agents bid on individual transport jobs based on the start location of the current job, the location of the mobile robot, the number of jobs assigned and the end location of the last assigned job.

This work specifically builds on an earlier investigation by Agrawal et al. [8] in which mobile robot actions were mapped to their own local observations with only a limited set of communicated observations from other agents. While such a fine granularity enhances robustness and manufacturing productivity, it introduces a compromise in the learning efficiency of the framework. In the current work, we specifically explore the efficiency, stability, and robustness distinctions between decentralized and centralized RL frameworks in mobile robot driven shop floors.

3. METHODOLOGY

The aim of this work is to compare the efficiency, learning stability, and robustness of a decentralized RL framework to that of a centralized RL framework in an autonomous mobile robot driven shop floor. The decentralized framework that is utilized in this work is adapted from prior work by Agrawal et al. [8] and is illustrated in Figure 2. In this framework, robots and machines are represented by agents that partially perceive and autonomously act upon the shop floor environment through embedded sensors and controllers. Human workers are represented as operator agents that also perceive the shop floor and act based on their own intelligence. All agents communicate with each other through the communication server agent. This multi-agent system aims to maximize throughput and safety.

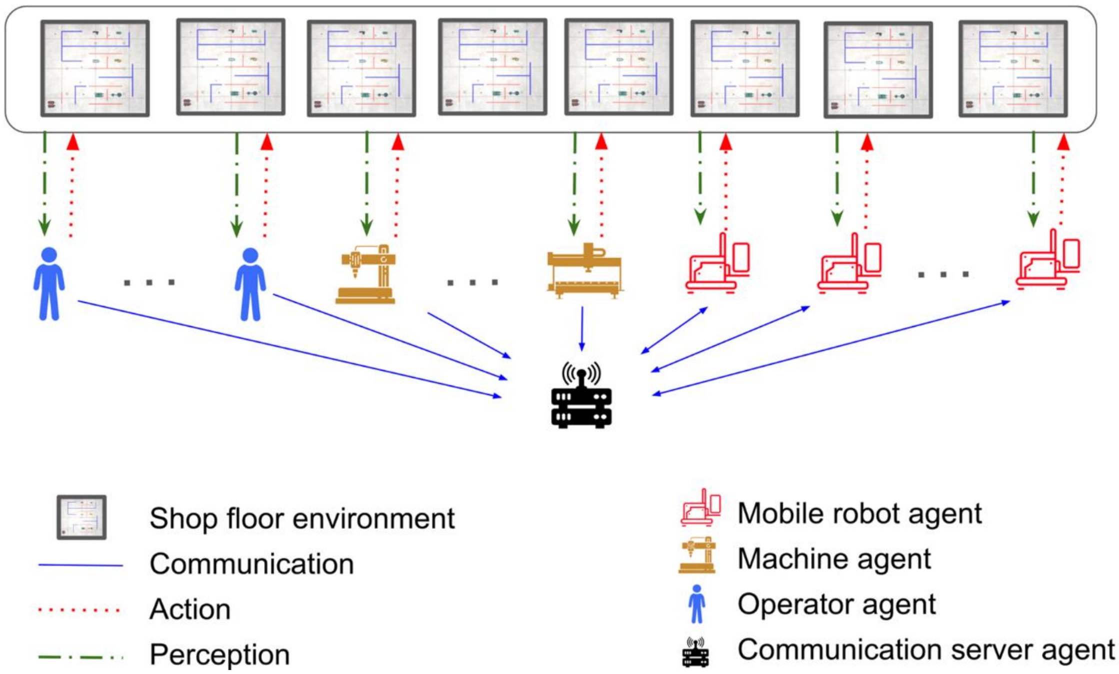

FIGURE 2: MULTI-AGENT SYSTEMS FRAMEWORK
(adapted from Agrawal et al. [8]*)*

The learning problem for the decentralized framework is modelled as an MDP with communication amongst homogenous robot agents sharing a common policy network, π during training. This is illustrated in Figure 3 and is often referred to as parameter sharing [77,78]. The observation, action, and rewards for the framework are adapted from prior work by Agrawal et al. [8]

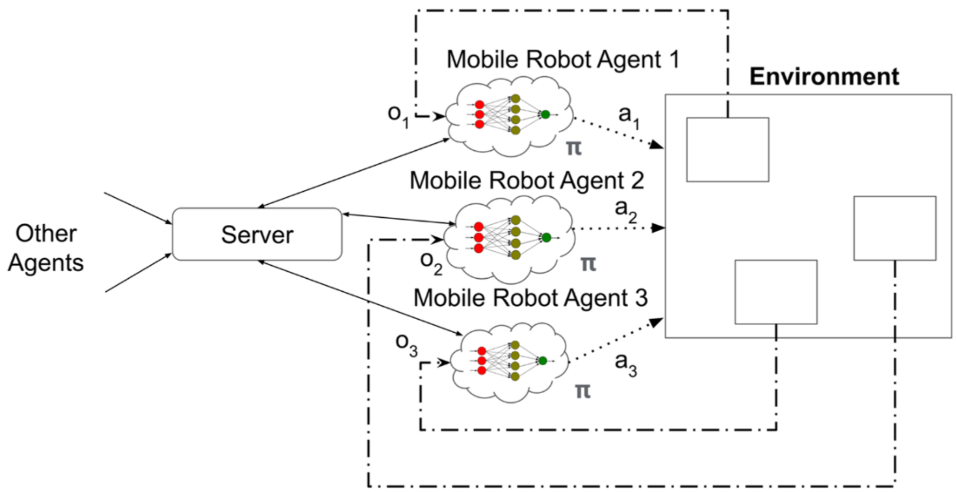

FIGURE 3: DECENTRALIZED TRAINING SCENARIO

For the centralized framework, the MDP assumes a joint model for the actions and states [78]. It involves a single agent with a policy network, π_0 that maps the joint observations of the robots to a joint action as shown in Figure 4. The observation of this agent is the union of the observations of each agent from the decentralized training scenario. The action space of the agent is the cartesian product of the action spaces from the decentralized training scenario. To make this case comparable with the previous one, the reward function is one-third the sum of the rewards each agent would have received in the decentralized scenario.

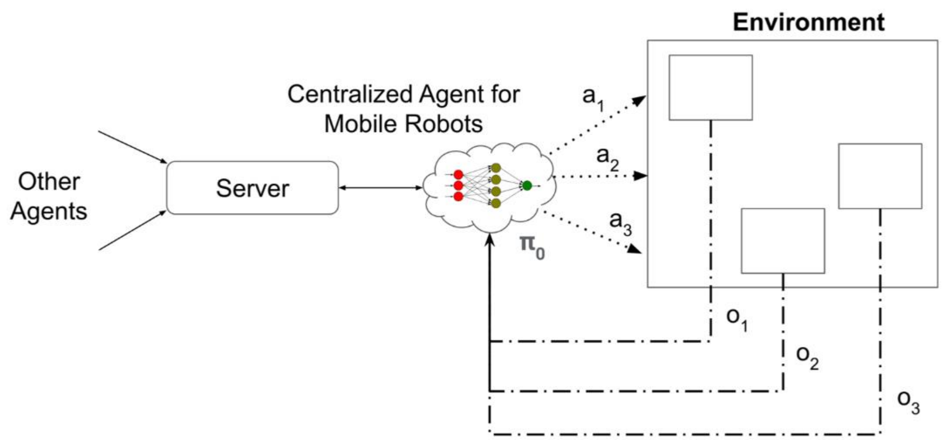

FIGURE 4: CENTRALIZED TRAINING SCENARIO

A proximal policy optimization algorithm with curiosity driven exploration [79,80] is used for training both frameworks. We make use of Unity's built in ML Agents Toolkit for our reinforcement learning algorithm [81], but this work can be translated to other simulators and algorithmic implementations. Specifically, the training is performed on an NVIDIA GeForce GTX 1650 with 8 instances of the shop floor environment running in parallel. Each instance involves random initialization for the location of the mobile robots.

To compare the efficiency, stability, and robustness of both the frameworks, several policies with increasing time budgets are trained using both the frameworks. The following tests are conducted to assess the performance of all the trained policies:

1. Static testing: The trained policies are evaluated for a fixed period (referred to as an evaluation run) with repeated machine cycles, with each cycle having the same processing time. To test the robustness of the policies to initialization, several such evaluation runs are performed by randomly varying the initial position of robots on the shop floor in each run.

2. Dynamic testing: To test the robustness of the policies in dynamic scenarios and failures, the trained policies are evaluated with machine processing time variations of up to 30%. Moreover, two failures are introduced in the form of a machine breakdown and a robot malfunction after a few cycles. Specifically, Machine 5 gets stuck in the processing state and Robot 2 gets stuck in the halt state. Similar to static testing, several evaluation runs are performed by randomly varying the initial position of robots on the shop floor in each run.

For each evaluation run, we compute the resulting throughput (total number of jobs completed across all machines) and safety (total number of collisions amongst all entities on the shop floor).

4. RESULTS AND DISCUSSION

The cumulative reward received by the agents trained using the centralized and decentralized frameworks with various compute budgets is shown in Figure 5. We observe that the reward increases over time in both scenarios with the decentralized cases gaining rewards equivalent to the centralized cases at a lower computational expense. This shows that the decentralized framework is more efficient than the centralized framework. However, we observe higher fluctuations in the cumulative reward in the training runs of decentralized framework, indicating a lower stability in the learning process.

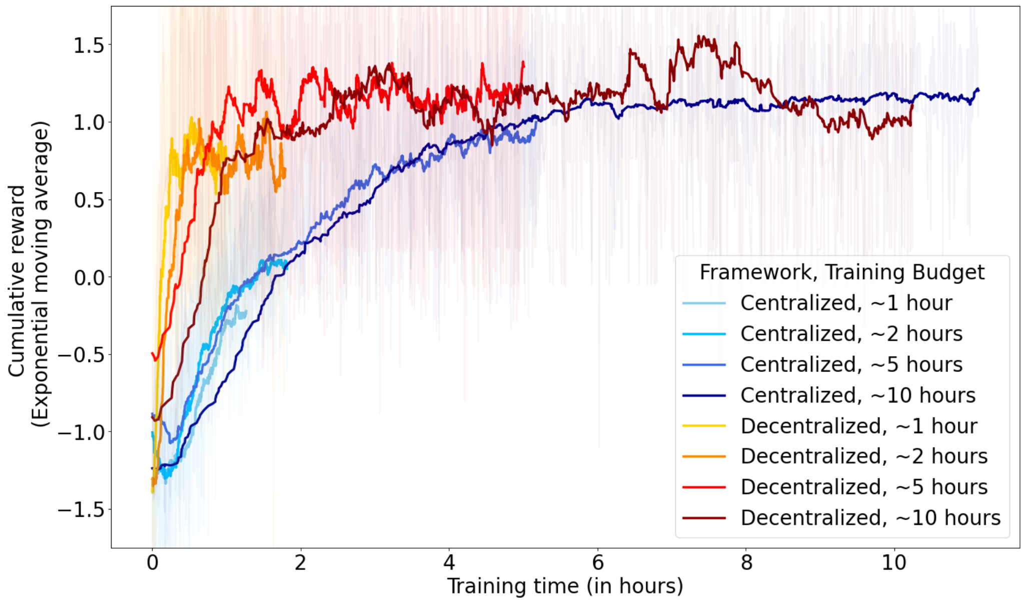

FIGURE 5: CUMULATIVE REWARD COMPARISON ACROSS COMPUTE BUDGETS

To further investigate the efficiency and robustness of the decentralized and centralized frameworks, several static and dynamic tests are conducted using all the policies that were trained across the set of budgets and evaluated for throughput and safety. In each of these runs, the initial location of the mobile robots is randomly assigned on the shop floor. Figure 6 shows the results of all the testing scenarios.

We observe that both frameworks yield high performance (i.e., throughput and safety) in static testing when trained using a high compute budget. Specifically, this corresponds to a high number of jobs scheduled and low number of collisions at a training time of approximately 10 hours in Figure 6(a) and 6(b), respectively. However, the decentralized framework yields a higher performing policy than the centralized one at a reduced budget in the static testing. Specifically, this corresponds to the policies that are trained for approximately 5 hours. This emphasizes that the decentralized framework is more efficient than the centralized one. Further, the variance in the number of jobs scheduled at the highest budget (around 10 hours) is lower for the centralized case as compared to the decentralized case. This indicates that the centralized case is more robust to initialization than the decentralized one, albeit at a lower computational efficiency.

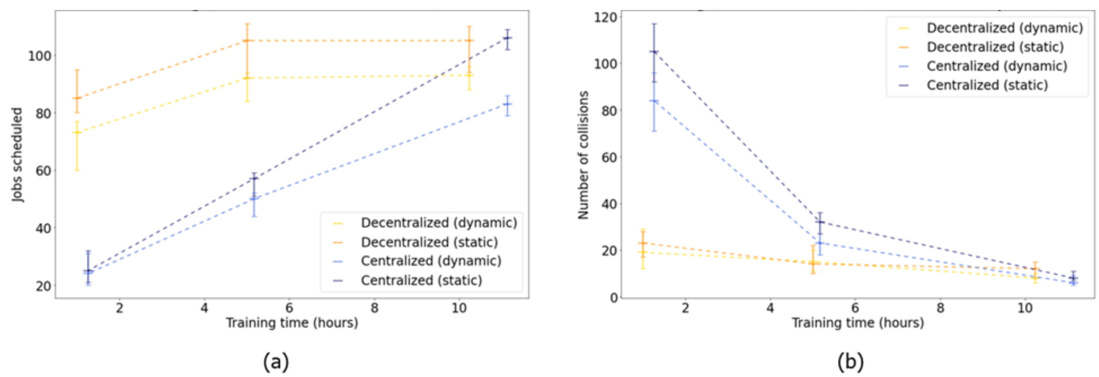

FIGURE 10: RESULTS OF STATIC AND DYNAMIC TESTING

When comparing the effect of dynamic and static testing for the decentralized framework, we observe that the number of jobs scheduled are reduced upon introducing dynamic changes across all budgets. However, the number of collisions does not change significantly upon introducing processing time variations and failures. This is because the robots still navigate around obstacles in a similar manner despite their altered trajectories.

When comparing the effect of dynamic and static testing for the centralized framework, we observe that the number of jobs scheduled are reduced upon introducing dynamic changes like the previous comparison. However, the reduction is more prominent in the high-performance case trained at a budget of approximately 10 hours. The number of collisions does not change significantly for the dynamic case at this budget, like the previous comparison. However, for the low-performance policies, the number of collisions is reduced upon introduction of dynamic changes. This is attributed to the failure of robot 2 which contributes less to the overall number of collisions on the shop floor. This is mainly observed in the low-performance cases as the robots have not learnt to effectively navigate around obstacles to reach the machines.

When comparing the centralized and decentralized frameworks in dynamic testing, the results of static testing at a high budget serve as a suitable baseline performance. Although the number of jobs completed in both frameworks is comparable under these static conditions, the introduction of processing time variations and failures leads to a more significant performance reduction in the centralized framework. This shows that the decentralized framework is more robust to dynamic changes than the centralized one.

Overall, the results of this study reveal that the decentralized framework outperforms the centralized one in terms of efficiency and robustness to dynamic changes and failures. The decentralized framework's advantage in efficiency can be attributed to its distributed computing capabilities. Instead of mapping joint observations to joint actions, each robot maps its own local observations and a limited set of communicated observations from other robots to its own local actions, leading to reduced computational complexity and a more efficient solution. Furthermore, its local decision-making capabilities allow for enhanced robustness to dynamic changes such as processing time delays, machine breakdowns, and robot malfunctions. Specifically, each robot can make decisions independently to maximize its contribution to the overall throughput even with such dynamic events. Further, upon extensive training, the centralized framework is also able to yield a high-performance shop floor. Moreover, a reduced variance across several evaluations runs is indicative of its robustness to initialization of robot locations. This enhanced robustness is credited to the centralized agent's comprehensive access to shop floor information and its control over all the robots. Furthermore, the decentralized framework's training exhibits higher fluctuations, signalling a lower stability in the learning process as compared to the centralized one. This is attributed to multiple robots sharing parameters while training the same policy despite making independent decisions. While this setup improves the learning speed by pooling experiences, the variability in individual experiences coupled with collective policy training results in fluctuations in learning.

Importantly, the extensive training required for the centralized framework to achieve high performance is often impractical. The time and computational resources needed for such training can be prohibitive in high-fidelity simulators, making the decentralized framework more valuable, particularly given its inherent robustness to dynamic changes and failures.

5 Conclusion

This work compares a decentralized RL framework for the integrated job scheduling and navigation control of a system involving autonomous mobile robots with a comparable centralized framework when learning a control policy. On one hand, we observe that the centralized framework exhibits higher learning stability and is more robust to the initial location of the robots on the shop floor when comparing the policies trained with the highest budget. On the other hand, the decentralized framework showcases superior time efficiency and robustness to dynamic changes like processing time delays, machine breakdowns and robot malfunctions. This study underscores the decentralized framework's superiority over centralized approaches in managing autonomous mobile robot-driven shop floors, highlighting its efficiency, and robustness to dynamic changes and failures as key factors that make it a more practical and effective solution for real-world applications.

Future work should aim to extend the investigation beyond the metrics of efficiency, stability, and robustness to include a comprehensive analysis of accuracy, generalizability, and scalability of decentralized frameworks in comparison to centralized ones. This could involve an evaluation of the accuracy of robot actions in a high-fidelity simulator using both the frameworks. An extension to this work could also be an investigation of the generalizability across different shop floors with diverse layouts. Additionally, assessing the scalability of these systems with a larger number of robots and machines will be key to determining their viability for large-scale industrial applications. Addressing these dimensions will provide a more holistic view of the potential and limitations of decentralized frameworks, thereby guiding the development of more effective and adaptable autonomous mobile robot-driven shop floors.

REFERENCES

[1] Rai, R., Tiwari, M. K., Ivanov, D., and Dolgui, A., 2021, "Machine Learning in Manufacturing and Industry 4.0 Applications," Int J Prod Res, **59**(16), pp. 4773–4778.

[2] Dogan, A., and Birant, D., 2021, "Machine Learning and Data Mining in Manufacturing," Expert Syst Appl, **166**.

[3] Arinez, J. F., Chang, Q., Gao, R. X., Xu, C., and Zhang, J., 2020, "Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook," Journal of Manufacturing Science and Engineering, Transactions of the ASME, **142**(11).

[4] Malhan, R. K., and Gupta, S. K., 2023, "The Role of Deep Learning in Manufacturing Applications: Challenges and Opportunities," J Comput Inf Sci Eng, pp. 1–11.

[5] Kabir, H., Tham, M. L., and Chang, Y. C., 2023, "Internet of Robotic Things for Mobile Robots: Concepts, Technologies, Challenges, Applications, and Future Directions," Digital Communications and Networks, **9**(6), pp. 1265–1290.

[6] Wichmann, R. L., Eisenbart, B., and Gericke, K., 2019, "The Direction of Industry: A Literature Review on Industry 4.0," *Proceedings of the International Conference on Engineering Design, ICED*, Cambridge University Press, pp. 2129–2138.

[7] Hermann, M., Pentek, T., and Otto, B., 2016, "Design Principles for Industrie 4.0 Scenarios," *Proceedings of the Annual Hawaii International Conference on System Sciences*, IEEE Computer Society, pp. 3928–3937.

[8] Agrawal, A., Won, S. J., Sharma, T., Deshpande, M., and McComb, C., 2021, "A Multi-Agent Reinforcement Learning Framework for Intelligent Manufacturing with Autonomous Mobile Robots," Proceedings of the Design Society, **1**, pp. 161–170.

[9] Malus, A., Kozjek, D., and Vrabič, R., 2020, "Real-Time Order Dispatching for a Fleet of Autonomous Mobile Robots Using Multi-Agent Reinforcement Learning," CIRP Annals, **69**(1), pp. 397–400.

[10] Fragapane, G., de Koster, R., Sgarbossa, F., and Strandhagen, J. O., 2021, "Planning and Control of Autonomous Mobile Robots for Intralogistics: Literature Review and Research Agenda," Eur J Oper Res, **294**(2), pp. 405–426.

[11] Wesselhöft, M., Hinckeldeyn, J., and Kreutzfeldt, J., 2022, "Controlling Fleets of Autonomous Mobile Robots with Reinforcement Learning: A Brief Survey," Robotics, **11**(5).

[12] Bai, Y., Lv, Y., and Zhang, J., 2023, "Smart Mobile Robot Fleet Management Based on Hierarchical Multi-Agent Deep Q Network towards Intelligent Manufacturing," Eng Appl Artif Intell, **124**.

[13] Monostori, L., Váncza, J., and Kumara, S. R. T., 2006, "Agent-Based Systems for Manufacturing," CIRP Ann Manuf Technol, **55**(2), pp. 697–720.

[14] Leitão, P., 2009, "Agent-Based Distributed Manufacturing Control: A State-of-the-Art Survey," Eng Appl Artif Intell, **22**(7), pp. 979–991.

[15] Parente, M., Figueira, G., Amorim, P., and Marques, A., 2020, “Production Scheduling in the Context of Industry 4.0: Review and Trends,” Int J Prod Res, **58**(17), pp. 5401–5431.
[16] Li, C., Zheng, P., Yin, Y., Wang, B., and Wang, L., 2023, “Deep Reinforcement Learning in Smart Manufacturing: A Review and Prospects,” CIRP J Manuf Sci Technol, **40**, pp. 75–101.
[17] Hernandez-Leal, P., Kartal, B., and Taylor, M. E., 2019, “A Survey and Critique of Multiagent Deep Reinforcement Learning,” Auton Agent Multi Agent Syst, **33**(6), pp. 750–797.
[18] Oroojlooy, A., and Hajinezhad, D., 2022, “A Review of Cooperative Multi-Agent Deep Reinforcement Learning,” Applied Intelligence.
[19] Li, Y., 2018, “Deep Reinforcement Learning.”
[20] Zhou, Z., Liu, G., and Tang, Y., 2023, “Multi-Agent Reinforcement Learning: Methods, Applications, Visionary Prospects, and Challenges,” IEEE International Conference on Program Comprehension, **2022-March**, pp. 36–47.
[21] Panzer, M., and Bender, B., 2022, “Deep Reinforcement Learning in Production Systems: A Systematic Literature Review,” Int J Prod Res, **60**(13), pp. 4316–4341.
[22] Schwung, A., Schwung, D., Sharafath, M., and Hameed, A., *Cooperative Robot Control in Flexible Manufacturing Cells: Centralized vs. Distributed Approaches*.
[23] Mayer, S., Classen, T., and Endisch, C., 2021, “Modular Production Control Using Deep Reinforcement Learning: Proximal Policy Optimization,” J Intell Manuf, **32**(8), pp. 2335–2351.
[24] Lee, J., Bagheri, B., and Kao, H.-A., 2015, “A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems,” Manuf Lett, **3**, pp. 18–23.
[25] Wang, L., Törngren, M., and Onori, M., 2015, “Current Status and Advancement of Cyber-Physical Systems in Manufacturing,” J Manuf Syst, **37**, pp. 517–527.
[26] Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., and Ueda, K., 2016, “Cyber-Physical Systems in Manufacturing,” CIRP Annals, **65**(2), pp. 621–641.
[27] Pivoto, D. G. S., de Almeida, L. F. F., da Rosa Righi, R., Rodrigues, J. J. P. C., Lugli, A. B., and Alberti, A. M., 2021, “Cyber-Physical Systems Architectures for Industrial Internet of Things Applications in Industry 4.0: A Literature Review,” J Manuf Syst, **58**, pp. 176–192.
[28] Jennings, and Wooldridge, 1998, *Agent Technology*, Springer Berlin Heidelberg.
[29] Wang, B., Zheng, P., Yin, Y., Shih, A., and Wang, L., 2022, “Toward Human-Centric Smart Manufacturing: A Human-Cyber-Physical Systems (HCPS) Perspective,” J Manuf Syst, **63**, pp. 471–490.
[30] Zhou, J., Zhou, Y., Wang, B., and Zang, J., 2019, “Human–Cyber–Physical Systems (HCPSs) in the Context of New-Generation Intelligent Manufacturing,” Engineering, **5**(4), pp. 624–636.
[31] Cimini, C., Pirola, F., Pinto, R., and Cavalieri, S., 2020, “A Human-in-the-Loop Manufacturing Control Architecture for the next Generation of Production Systems,” J Manuf Syst, **54**, pp. 258–271.
[32] Williams, R. J., 1992, “Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning,” Mach Learn, **8**(3–4), pp. 229–256.

[33] Watkins, C. J. C. H., and Dayan, P., 1992, "Q-Learning," Mach Learn, **8**(3–4), pp. 279–292.
[34] Sutton, R. S., and Barto, A. G., 2018, *Reinforcement Learning : An Introduction*.
[35] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D., 2015, "Human-Level Control through Deep Reinforcement Learning," Nature, **518**(7540), pp. 529–533.
[36] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D., 2016, "Mastering the Game of Go with Deep Neural Networks and Tree Search," Nature, **529**(7587), pp. 484–489.
[37] Stone, P., and Veloso, M., 2000, *Multiagent Systems: A Survey from a Machine Learning Perspective*.
[38] Canese, L., Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M., and Spanò, S., 2021, "Multi-Agent Reinforcement Learning: A Review of Challenges and Applications," Applied Sciences (Switzerland), **11**(11).
[39] Xu, M., Liu, Z., Huang, P., Ding, W., Cen, Z., Li, B., and Zhao, D., 2022, "Trustworthy Reinforcement Learning Against Intrinsic Vulnerabilities: Robustness, Safety, and Generalizability," IEEE International Conference on Program Comprehension, **2022-March**, pp. 36–47.
[40] Shi, L., 2023, *Provable Algorithms for Reinforcement Learning*.
[41] Pullum, L. L., *Review of Metrics to Measure the Stability, Robustness and Resilience of Reinforcement Learning*.
[42] Dong, H., Ding, Z., Zhang, S., Fundamentals, E., and Research, A., *Deep Reinforcement Learning*.
[43] Agrawal, A., and McComb, C., 2023, "Reinforcement Learning for Efficient Design Space Exploration With Variable Fidelity Analysis Models," J Comput Inf Sci Eng, **23**(4).
[44] Kaven, L., Huke, P., Göppert, A., and Schmitt, R. H., 2024, "Multi Agent Reinforcement Learning for Online Layout Planning and Scheduling in Flexible Assembly Systems," J Intell Manuf.
[45] Ouelhadj, D., and Petrovic, S., 2009, "A Survey of Dynamic Scheduling in Manufacturing Systems," Journal of Scheduling, **12**(4), pp. 417–431.
[46] Mohan, J., Lanka, K., and Rao, A. N., 2019, "A Review of Dynamic Job Shop Scheduling Techniques," *Procedia Manufacturing*, Elsevier B.V., pp. 34–39.
[47] Priore, P., Gómez, A., Pino, R., and Rosillo, R., 2014, "Dynamic Scheduling of Manufacturing Systems Using Machine Learning: An Updated Review," Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, **28**(1), pp. 83–97.
[48] Yu, Q., Yang, H., Lin, K. Y., and Li, L., 2020, "A Predictive Dispatching Rule Assisted by Multi-Layer Perceptron for Scheduling Wafer Fabrication Lines," J Comput Inf Sci Eng, **20**(3).

[49] Zhang, L., Feng, Y., Xiao, Q., Xu, Y., Li, D., Yang, D., and Yang, Z., 2023, “Deep Reinforcement Learning for Dynamic Flexible Job Shop Scheduling Problem Considering Variable Processing Times,” J Manuf Syst, **71**, pp. 257–273.

[50] Kayhan, B. M., and Yildiz, G., 2023, “Reinforcement Learning Applications to Machine Scheduling Problems: A Comprehensive Literature Review,” J Intell Manuf, **34**(3), pp. 905–929.

[51] Liu, Z., Mao, H., Sa, G., Liu, H., and Tan, J., 2024, “Dynamic Job-Shop Scheduling Using Graph Reinforcement Learning with Auxiliary Strategy,” J Manuf Syst, **73**, pp. 1–18.

[52] Ping, Y., Liu, Y., Zhang, L., Wang, L., and Xu, X., 2023, “Enterprise and Service−level Scheduling of Robot Production Services in Cloud Manufacturing with Deep Reinforcement Learning,” J Intell Manuf.

[53] Cals, B., Zhang, Y., Dijkman, R., and van Dorst, C., 2020, “Solving the Order Batching and Sequencing Problem Using Deep Reinforcement Learning.”

[54] Chang, X., Jia, X., Fu, S., Hu, H., and Liu, K., 2023, “Digital Twin and Deep Reinforcement Learning Enabled Real-Time Scheduling for Complex Product Flexible Shop-Floor,” Proc Inst Mech Eng B J Eng Manuf, **237**(8), pp. 1254–1268.

[55] Zhang, L., Yang, C., Yan, Y., and Hu, Y., 2022, “Distributed Real-Time Scheduling in Cloud Manufacturing by Deep Reinforcement Learning,” IEEE Trans Industr Inform, **18**(12), pp. 8999–9007.

[56] Li, Y., Gu, W., Yuan, M., and Tang, Y., 2022, “Real-Time Data-Driven Dynamic Scheduling for Flexible Job Shop with Insufficient Transportation Resources Using Hybrid Deep Q Network,” Robot Comput Integr Manuf, **74**.

[57] Dittrich, M. A., and Fohlmeister, S., 2020, “Cooperative Multi-Agent System for Production Control Using Reinforcement Learning,” CIRP Annals, **69**(1), pp. 389–392.

[58] Zhang, Y., Zhu, H., Tang, D., Zhou, T., and Gui, Y., 2022, “Dynamic Job Shop Scheduling Based on Deep Reinforcement Learning for Multi-Agent Manufacturing Systems,” Robot Comput Integr Manuf, **78**.

[59] Zhou, T., Tang, D., Zhu, H., and Zhang, Z., 2021, “Multi-Agent Reinforcement Learning for Online Scheduling in Smart Factories,” Robot Comput Integr Manuf, **72**.

[60] Fang, W., Zhang, H., Qian, W., Guo, Y., Li, S., Liu, Z., Liu, C., and Hong, D., 2023, “An Adaptive Job Shop Scheduling Mechanism for Disturbances by Running Reinforcement Learning in Digital Twin Environment,” J Comput Inf Sci Eng, **23**(5).

[61] Loganathan, A., and Ahmad, N. S., 2023, “A Systematic Review on Recent Advances in Autonomous Mobile Robot Navigation,” Engineering Science and Technology, an International Journal, **40**.

[62] Sun, H., Zhang, W., Yu, R., and Zhang, Y., 2021, “Motion Planning for Mobile Robots - Focusing on Deep Reinforcement Learning: A Systematic Review,” IEEE Access, **9**, pp. 69061–69081.

[63] Xiao, X., Liu, B., Warnell, G., and Stone, P., 2022, “Motion Planning and Control for Mobile Robot Navigation Using Machine Learning: A Survey,” Auton Robots, **46**(5), pp. 569–597.

[64] Fan, T., Long, P., Liu, W., and Pan, J., 2020, "Distributed Multi-Robot Collision Avoidance via Deep Reinforcement Learning for Navigation in Complex Scenarios," International Journal of Robotics Research, **39**(7), pp. 856–892.

[65] Liu, X., and Jin, Y., 2020, "Reinforcement Learning-Based Collision Avoidance: Impact of Reward Function and Knowledge Transfer," Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, **34**(2), pp. 207–222.

[66] Han, R., Chen, S., and Hao, Q., 2020, *Cooperative Multi-Robot Navigation in Dynamic Environment with Deep Reinforcement Learning; Cooperative Multi-Robot Navigation in Dynamic Environment with Deep Reinforcement Learning*.

[67] Aizat, M., Qistina, N., and Rahiman, W., 2023, "A Comprehensive Review of Recent Advances in Automated Guided Vehicle Technologies: Dynamic Obstacle Avoidance in Complex Environment Toward Autonomous Capability," IEEE Trans Instrum Meas, **73**, pp. 1–25.

[68] Li, B., Liu, H., Xiao, D., Yu, G., and Zhang, Y., 2017, "Centralized and Optimal Motion Planning for Large-Scale AGV Systems: A Generic Approach," Advances in Engineering Software, **106**, pp. 33–46.

[69] Qie, H., Shi, D., Shen, T., Xu, X., Li, Y., and Wang, L., 2019, "Joint Optimization of Multi-UAV Target Assignment and Path Planning Based on Multi-Agent Reinforcement Learning," IEEE Access, **7**, pp. 146264–146272.

[70] Pham, H. X., La, H. M., Feil-Seifer, D., and Nefian, A., 2018, "Cooperative and Distributed Reinforcement Learning of Drones for Field Coverage."

[71] Zhang, J., Yu, Z., Mao, S., Periaswamy, S. C. G., Patton, J., and Xia, X., 2020, "IADRL: Imitation Augmented Deep Reinforcement Learning Enabled UGV-UAV Coalition for Tasking in Complex Environments," IEEE Access, **8**, pp. 102335–102347.

[72] Dinelli, C., Racette, J., Escarcega, M., Lotero, S., Gordon, J., Montoya, J., Dunaway, C., Androulakis, V., Khaniani, H., Shao, S., Roghanchi, P., and Hassanalian, M., 2023, "Configurations and Applications of Multi-Agent Hybrid Drone/Unmanned Ground Vehicle for Underground Environments: A Review," Drones, **7**(2).

[73] Li, M., Guo, B., Zhang, J., Liu, J., Liu, S., Yu, Z., Li, Z., and Xiang, L., 2021, "Decentralized Multi-AGV Task Allocation Based on Multi-Agent Reinforcement Learning with Information Potential Field Rewards."

[74] Association for Computing Machinery., and Institute of Electrical and Electronics Engineers., *2019 Winter Simulation Conference (WSC)*.

[75] Orr, J., and Dutta, A., 2023, "Multi-Agent Deep Reinforcement Learning for Multi-Robot Applications: A Survey," Sensors, **23**(7).

[76] Poudel, L., Elagandula, S., Zhou, W., and Sha, Z., 2023, "Decentralized and Centralized Planning for Multi-Robot Additive Manufacturing," Journal of Mechanical Design, **145**(1).

[77] Terry J K, Nathaniel Grammel, Ananth Hari, Luis Santos, and Benjamin Black, 2021, *Revisiting Parameter Sharing in Multi-Agent Deep Reinforcement Learning*.

[78] Gupta, J. K., Egorov, M., and Kochenderfer, M., 2017, "Cooperative Multi-Agent Control Using Deep Reinforcement Learning," pp. 66–83.

[79] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., 2017, "Proximal Policy Optimization Algorithms," ArXiv, pp. 1–12.

[80] Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T., 2017, “Curiosity-Driven Exploration by Self-Supervised Prediction,” 34th International Conference on Machine Learning, ICML 2017, **6**, pp. 4261–4270.

[81] Juliani, A., Berges, V.-P., Teng, E., Cohen, A., Harper, J., Elion, C., Goy, C., Gao, Y., Henry, H., Mattar, M., and Lange, D., 2018, “Unity: A General Platform for Intelligent Agents,” pp. 1–28.