Coordinating Search with Foundation Models and Multi-Agent Reinforcement Learning in Complex Environments

Christopher Allred*†, Jacob Haight†, Chandler Justice†, Isaac Peterson†
Rosario Scalise§, Ted Hromadka¶, Jason Pusey‡, Mario Harper†

***Abstract*—We present a multi-agent system simulation designed for efficient coordination and collaboration among multiple robots, particularly suited for search operations. This simulation reflects unstructured and complex outdoor scenarios where significant obstruction and occluded terrain surfaces cause difficulties in search. The software utilizes reinforcement learning (RL) and a centralized Multi-Agent Transformer (MAT) to enable autonomous agents to collect, process, and integrate data into the MAT. Simulated robots can effectively search in dynamic and unstructured environments. The project code and videos can be found at https://github.com/DIRECTLab/Coordinating-MAT-Env**

***Index Terms*—Isaac Sim, Multi-Agent Simulation, Reinforcement Learning, Unstructured Environment, Simulation, Software**

I. Introduction

Coordinating large-scale search and exploration tasks in complex, dynamic environments presents significant challenges, particularly when multiple autonomous robots are involved. These environments, often characterized by unpredictable terrain, obstacles, and occluded areas, require flexible, adaptable software systems that enable robots to collaborate effectively in real-time. Traditional methods for robotic coordination often lack the ability to dynamically adapt to such evolving conditions, limiting their effectiveness in outdoor, unstructured environments.

*Corresponding author: C. Allred, Christopher.Allred@usu.edu.

† Supported by the Department of Computer Science, Utah State University; ‡ , DEVCOM Army Research Lab (ARL), Autonomous Systems Branch under AEOP Contract No. W9115R-15-2-0001; § Department of Computer Science, University of Washington; ¶ NVIDIA.

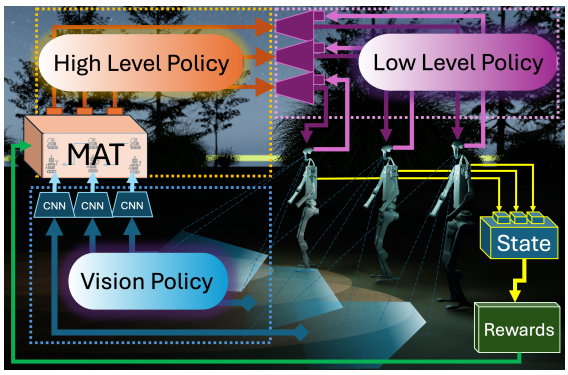

Fig. 1: Coordinated mission conducted by a fleet of homogeneous robots in a forested area, The diagram illustrates a hierarchical control architecture for robotic movement. The High-Level Policy generates strategic decisions that are refined by a pertained foundation locomotion model(RL) Low-Level Policy to control the robot's actions. The Vision Policy, using Convolutional Neural Networks (CNNs), processes visual inputs to guide the Multi-Agent Transformer (MAT), which integrates sensory data for decision-making. This system dynamically interacts with the all of the robot's State, influencing its actions and subsequent Rewards, optimizing behavior through continuous feedback.

Traditional methods often lack the adaptability required to handle such dynamic conditions. In response to these challenges, we present a software system specifically designed to coordinate multi-robot search operations in complex environments. The system integrates advanced tech-

nologies, including the Multi-Agent Transformer (MAT) architecture, for high-level coordination and decision-making, alongside reinforcement learning (RL)-based modules for low-level control and autonomous behavior. By leveraging the NVIDIA Isaac-Sim platform and extending its capabilities through GRUtopia, our Multi-Agent Reinforcement Learning (MARL) software enables high-fidelity simulation and real-time decision-making in challenging scenarios.

This software framework allows for the integration of various robotic platforms, and supports a wide range of tasks, from autonomous search to environmental coverage. Its modular architecture ensures scalability, enabling the deployment of large fleets of robots while maintaining efficient coordination. The system's flexibility makes it adaptable to different algorithms, robotic platforms, and environmental setups, making it a valuable tool for research and real-world applications.

In this paper, we focus on the design and implementation of this software system, highlighting its architecture, modularity, and the simulation environments it supports. We demonstrate how the software can be applied to multi-robot coordination tasks, with a particular emphasis on unstructured search environments. The effectiveness of the system is showcased through its ability to dynamically manage complex robotic tasks and its potential for future applications in autonomous systems.

Reinforcement Learning (RL) has been successfully applied to complex coordination problems, including games with strategy-based tasks such as DOTA [1] and Starcraft [2]. Inspired by this, we built this software on NVIDIA's Isaac-Sim [3], a simulation platform that enables high-fidelity simulation of complex robotic environments, the simulation is shown in Figure 1. By utilizing RL-based models for locomotion and control [4]–[6], our simulation environment supports biped legged robots. These models ensure robust coordination and flexibility in real-time operations, enabling the robots to dynamically adapt to changing conditions.

II. Methodology

This work focuses on coverage applications using a fleet of multiple robots, an operator will be able to set bounds on an environment which they believe the desired target to be located within, and the fleet coordinates their strategy autonomously and begin searching in a coordinated manner until the area has been fully covered or a separate condition is met (IE, searching a sufficient proportion of the area within the bounds, or finding the desired object). A high level illustration of this can be found at 1.

We developed a coverage algorithm within a MARL framework, utilizing a centralized MAT architecture. This algorithm was implemented in the GRUtopia simulation environment to coordinate multiple heterogeneous robotic agents in efficiently covering a specified area. The primary objective was to maximize environmental coverage by enabling agents to explore as many unique grid cells as possible within a fixed number of steps.

A. Environment Configuration

We utilized several tools for the research and development of this system including NVIDIA's Isaac-Sim [3], a special built tool for developing and simulating robot technologies developed under the umbrella of the Omniverse project. Additionally, we utilize an integration into Isaac-Sim primarily developed by OpenRobotLab called GRUtopia [7], which allows for interactive simulation of interactive 3D environments.

To further enhance the realism and stress-test the search algorithms, we incorporated NVIDIA's Blast Destruction to simulate debris fields that robots must navigate through during missions, see Figure 2. This tool enables us to accurately model the physics of rubble and structural collapses, providing a more challenging and dynamic environment for the agents. We also utilized NVIDIA Flow to simulate smoke during search operations, stressing the robot's sensors and decision-making processes in low-visibility conditions. These additional simulations act as digital twins for stress testing the search methods, ensuring robustness in various disaster scenarios.

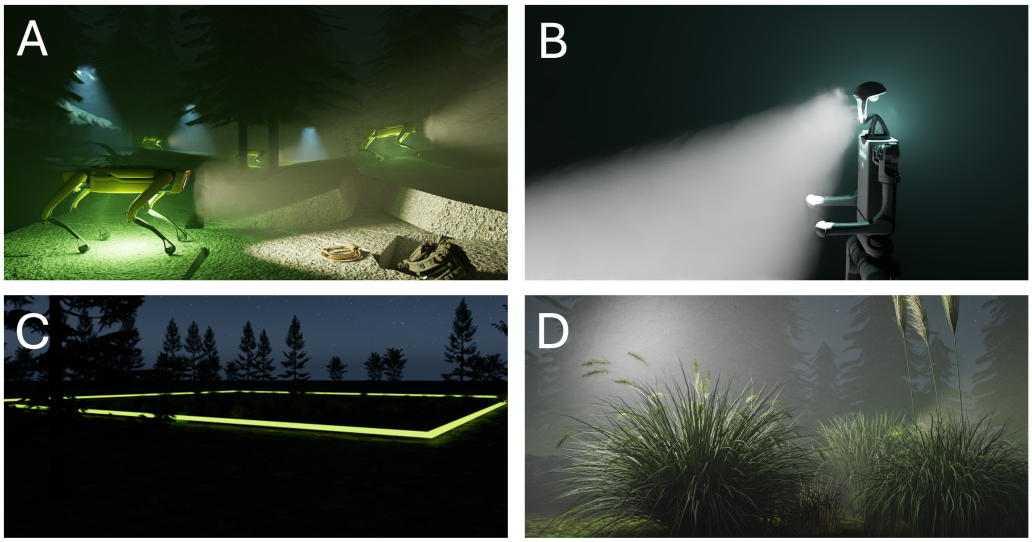

Fig. 2: Simulation environment demonstrating various capabilities: (A) NVIDIA's Blast integration allowing flexible use of robot platforms, including quadrupeds and drones; (B) volumetric lighting effects in Isaac Sim, enhancing realistic robotic perception; (C) defined search area highlighting mission boundaries; and (D) dense vegetation representing complex natural obstacles for testing navigation and obstacle avoidance.

To facilitate coordination within the Isaac-Sim environment, we utilized a modified instance of the GRUtopia toolset. These modifications are crucial for enabling the multi-agent communication strategies discussed in the previous section. The simulation environment is designed to mimic real-world scenarios, with a generic forest scene that reflects one of the many potential applications of this technology. The Isaac-Sim environment allows for large-scale training that would be impractical in the real world, providing a robust and complex simulation that closely emulates real-world conditions. This enables the training data obtained from Isaac-Sim and GRUtopia to be directly applicable to real-world deployments of heterogeneous robotic fleets.

A coverage grid (`coverage_grid`) is maintained to record the cells visited by the agents, with the goal of collectively maximizing the number of unique cells explored. Agents are required to remain within predefined boundaries; exceeding these boundaries results in penalties to encourage operation within the designated area.

B. Observation and Action Spaces

Each agent receives observations that include its current position and orientation within the environment, represented as a vector, and simulated camera data providing visual perception of the surroundings. The action space for each agent is continuous, represented by a three-dimensional vector that dictates movement in different directions, allowing for fine-grained control over navigation.

C. Reward Function Design

The reward function is designed to incentivize efficient coverage while penalizing undesirable behaviors. Agents receive a positive reward when they visit a new, unvisited cell in the coverage grid, encouraging exploration. Penalties are imposed when agents revisit cells, move out of bounds, or fail to contribute to area coverage. A small negative reward is assigned at each time step to encourage agents to complete the coverage task promptly. Additionally,

a completion bonus is granted if the agents achieve full coverage before reaching the maximum number of steps (N_{max}), scaled inversely with the time taken to further promote efficiency.

Mathematically, the immediate reward $r_{i,j}(t)$ for agent j in environment i at time step t is defined as:

$$r_{i,j}(t) = \begin{cases} +1, & \text{if a new cell is visited} \\ -\Delta\text{ROI}, & \text{if out of bounds} \\ 0, & \text{if revisiting a cell} \end{cases} \quad (1)$$

Where ΔROI represents the deviation from the region of interest.

To encourage quicker completion of the coverage task, a final reward is scaled inversely with the time taken, providing an incentive for agents to maximize efficiency. The time-based scaling factor, denoted by α, measures the proportion of remaining steps relative to the maximum number of allowed steps: $\alpha = (N_{\text{max}} - N_{\text{current}})N_{\text{max}}$, where N_{current} is the number of steps taken so far.

The total final reward R_{final} is calculated by multiplying the achieved coverage proportion C, the time bonus α, and a scaling factor k, which is set to 10 in this context but can be adjusted as a hyperparameter. Additionally, the variable ε represents the time penalty per step, which is set to -0.01. This final reward is added to the cumulative reward:

$$r_{\text{total}} = \sum_{t} r_{i,j}(t) + C\alpha k\varepsilon \quad (2)$$

This reward mechanism effectively encourages agents not only to maximize the coverage area but also to complete the task in fewer steps, promoting efficient exploration and coordination among agents. The format for equation 2 is inspired from the work of [8].

D. Multi-Agent Transformer

The Multi-Agent Transformer (MAT) [9] is an architecture that applies the principles of sequence modeling, commonly used in natural language processing, to multi-agent reinforcement learning (MARL). MAT views the actions of agents as a sequence of interrelated events, using an encoder-decoder structure to process observations and generates actions for each agent. Through masked attention, MAT ensures that agents only base their decisions on previous actions in the sequence. This allows the software to model agent interactions in a highly coordinated and efficient manner. Trained on-policy, MAT adapts dynamically to real-world environments, making it ideal for tasks requiring high levels of collaboration, such as multi-agent robotics and search operations.

III. RESULTS

The core contribution of our system is its modular architecture, enabling integration of multiple robotic platforms, control algorithms, and decision-making models. It leverages a centralized Multi-Agent Transformer (MAT) for high-level coordination, translating strategic instructions into low-level commands specific to each robot. This flexible architecture supports both centralized and decentralized decision-making, balancing global coordination with individual agent autonomy. The system can be extended to incorporate new robotic models, control policies, or task-specific algorithms without significant reconfiguration.

Our software includes several key components. The MAT serves as a centralized entity that processes high-level strategies and distributes contextualized, low-level commands to individual agents, improving coordination in dynamic environments and ensuring that each robot's actions are aligned with the overall mission goals. Additionally, the software incorporates robust multi-agent communication protocols that allow real-time data sharing and decision synchronization across the robotic fleet. This capability ensures smooth coordination even in environments with dynamic obstacles and occlusions.

The MAT processes high-level strategies and distributes context-specific commands, improving coordination in dynamic environments. Robust multi-agent communication protocols ensure real-time data sharing and synchronization across the fleet, enabling smooth coordination even in challenging environments with dynamic obstacles. The simu-

lation environment, built on NVIDIA's Isaac-Sim and extended via GRUtopia, offers high-fidelity emulation of real-world conditions, modeling various terrains, including forests, urban areas, and unstructured terrains with dynamic lighting and visibility conditions. It supports complex environments, including debris fields using NVIDIA BLAST, and operates at 215 fps (on an RTX 4090) when optimized. Scalability is a key strength, with support for both small teams and large fleets using mesh geometry instancing, making it suitable for testing individual and group coordination strategies.

IV. Conclusion

Our software system represents a significant advancement in multi-agent robotic coordination, offering a flexible, scalable, and high-fidelity platform for both simulation and real-world applications. By providing an adaptable and robust framework, the system empowers researchers and developers to tackle complex search and exploration tasks across a wide range of robotic platforms and environments.

The software system offers a valuable platform for benchmarking and improving multi-agent search and coordination strategies. Its design supports flexible experimentation in complex simulated environments, providing a baseline for researchers and developers to test their algorithms on key performance metrics like coverage efficiency, scalability, and real-time adaptability. By efficiently assigning exploration tasks to individual robots, the system minimizes redundancy, making it an ideal tool for testing and refining RL approaches aimed at optimizing search performance. Additionally, the platform's ability to simulate real-time adaptation in dynamic scenarios, such as low-visibility environments, allows users to evaluate how different strategies maintain coordination and communication across a fleet. The system's scalability, supporting both small teams and larger fleets of up to 20 agents, ensures that it can accommodate research aimed at solving real-world challenges requiring multi-agent coordination.

As part of our commitment to fostering collaboration and further research, the software system, including the simulation environments and MAT integration, has been open-sourced. This provides researchers and developers access to a powerful tool for testing multi-robot coordination strategies in complex environments. The modular design of the system enables users to tailor the platform to their specific research requirements easily. Future work aims to integrate this system with recent advancements from Isaac-Lab [10], enhancing high-level coordination capabilities and providing robust solutions for multi-agent coordination in complex and challenging environments.

Acknowledgment

This work was supported by the U.S. Army Research Fellowship Program administered by the AEOP through a cooperative agreement with the Autonomous Systems Branch, Army Research Lab (ARL). Research was sponsored by ARL and was accomplished under Contract Number No. W9115R-15-2-0001. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the DEVCOM Army Research Laboratory or the US Government.

References

[1] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[2] M. Samvelyan, T. Rashid, C. S. De Witt, G. Farquhar, N. Nardelli, T. G. Rudner, C.-M. Hung, P. H. Torr, J. Foerster, and S. Whiteson, "The starcraft multi-agent challenge," *arXiv preprint arXiv:1902.04043*, 2019.

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

[4] K. Caluwaerts, A. Iscen, J. Kew, W. Yu, T. Zhang, D. Freeman, K.-H. Lee, L. Lee, S. Saliceti, V. Zhuang, N. Batchelor, S. Bohez, F. Casarini, J. E. Chen, O. Cortes, E. Coumans, A. Dostmohamed, G. Dulac-Arnold, A. Escontrela, E. Frey, R. Hafner, D. Jain, B. Jyenis, Y. Kuang, E. Lee, L. Luu, O. Nachum, K. Oslund, J. Powell, D. Reyes, F. Romano, F. Sadeghi, R. Sloat, B. Tabanpour, D. Zheng, M. Neunert, R. Hadsell, N. Heess, F. Nori, J. Seto, C. Parada, V. Sindhwani, V. Vanhoucke, and

J. Tan, "Barkour: Benchmarking animal-level agility with quadruped robots," *ArXiv*, vol. abs/2305.14654, 2023.
[5] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," *ArXiv*, vol. abs/2309.14341, 2023.
[6] D. Hoeller, N. Rudin, D. V. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *ArXiv*, vol. abs/2306.14874, 2023.
[7] H. Wang, J. Chen, W. Huang, Q. Ben, T. Wang, B. Mi, T. Huang, S. Zhao, Y. Chen, S. Yang, P. Cao, W. Yu, Z. Ye, J. Li, J. Long, Z. Wang, H. Wang, Y. Zhao, Z. Tu, Y. Qiao, D. Lin, and P. Jiangmiao, "Grutopia: Dream general robots in a city at scale," in *arXiv*, 2024.
[8] Valve Corporation, "Portal," 2007, platforms: PC, Xbox 360, PlayStation 3. [Online]. Available: https://www.thinkwithportals.com/
[9] M. Wen, J. G. Kuba, R. Lin, W. Zhang, Y. Wen, J. Wang, and Y. Yang, "Multi-agent reinforcement learning is a sequence modeling problem," *arXiv preprint arXiv:2205.14953*, 2022.
[10] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.