

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Multi-Agent Reinforcement Learning Approaches to Leverage Distributed Problem Solving

Inês Matos de Oliveira



Master in Informatics and Computing Engineering

Supervisor: Prof. António Costa

Co-Supervisor: Prof. Rosaldo Rossetti

July 18, 2025

Multi-Agent Reinforcement Learning Approaches to Leverage Distributed Problem Solving

Inês Matos de Oliveira

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

President: Prof. Ana Paula Rocha

Examiner: Prof. Pedro Campos

Supervisor: Prof. António Costa

July 18, 2025

Resumo

Os cenários de evacuação de emergência apresentam desafios complexos de coordenação, nos quais múltiplos agentes autônomos devem navegar em ambientes dinâmicos sob pressão temporal, evitando congestionamentos e otimizando a segurança coletiva. As abordagens convencionais de modelação de evacuação, abrangendo os Modelos de Força Social e os Modelos Baseados em Agentes com regras fixas, são suscetíveis a comportamentos predeterminados que não conseguem capturar a natureza adaptativa das situações de emergência reais. A presente dissertação investiga a aplicação de Aprendizagem por Reforço Multi-Agente (MARL) para desenvolver estratégias de evacuação descentralizadas eficazes em ambientes dinâmicos e parcialmente observáveis.

A investigação formaliza o problema de evacuação como um processo de decisão de Markov Parcialmente Observável Descentralizado (Dec-POMDP) e emprega o paradigma de Treino Centralizado com Execução Descentralizada (CTDE) para permitir aprendizagem coordenada mantendo execução autônoma. Três algoritmos MARL proeminentes são avaliados sistematicamente: Deep Q-Network (DQN), Proximal Policy Optimization (PPO), e Advantage Actor-Critic (A2C), selecionados com base na sua eficácia comprovada em ambientes de ação discreta, característicos de cenários de evacuação.

Procedeu-se ao desenvolvimento de um ambiente de simulação personalizado, recorrendo ao framework PettingZoo, o qual permitiu a extensão do Multi-Agent Particle Environment para modelar dinâmicas realistas de evacuação. A avaliação experimental abrange cinco cenários de complexidade crescente, incluindo múltiplas saídas, obstáculos estáticos, restrições de congestionamento e observabilidade parcial. O ambiente suporta interações heterogêneas entre agentes e implementa estruturas de recompensa sofisticadas que encorajam tanto o sucesso individual na evacuação como a eficiência de coordenação coletiva.

Os resultados demonstram a superioridade consistente do PPO em todos os cenários testados, revelando comportamentos emergentes como formação espontânea de filas e coordenação descentralizada. A investigação identifica limitações significativas do DQN em cenários complexos, incluindo o fenómeno de catastrophic forgetting. Métricas inovadoras de coordenação e eficiência de comunicação foram desenvolvidas para avaliar quantitativamente o desempenho dos agentes. Os resultados contribuem para o avanço de sistemas inteligentes de evacuação, demonstrando como o MARL pode superar as limitações das abordagens clássicas através de aprendizagem adaptativa e coordenação descentralizada, aproveitando a resolução distribuída de problemas.

Abstract

Emergency evacuation scenarios represent complex coordination challenges where multiple autonomous agents must navigate dynamic environments under time pressure while avoiding congestion and optimizing collective safety. Traditional evacuation modeling approaches, including Social Force Models and rule-based Agent-Based Models, rely on predetermined behavioral rules that fail to capture the adaptive nature of real-world emergency situations. This dissertation investigates the application of Multi-Agent Reinforcement Learning (MARL) to develop effective decentralized evacuation strategies in dynamic and partially observable environments.

The research formalizes the evacuation problem as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) and employs the Centralized Training with Decentralized Execution (CTDE) paradigm to enable coordinated learning while maintaining autonomous execution. Three prominent MARL algorithms are systematically evaluated: Deep Q-Network (DQN), Proximal Policy Optimization (PPO), and Advantage Actor-Critic (A2C), each selected for their proven effectiveness in discrete action environments characteristic of evacuation scenarios.

A custom simulation environment was developed using the PettingZoo framework, extending the Multi-Agent Particle Environment to model realistic evacuation dynamics. The experimental evaluation encompasses five scenarios of increasing complexity, including multiple exits, static obstacles, congestion constraints, and partial observability. The environment supports heterogeneous agent interactions and implements sophisticated reward structures that encourage both individual evacuation success and collective coordination efficiency.

The findings demonstrate PPO's consistent superior performance across all tested scenarios, revealing emergent behaviors such as spontaneous queue formation and decentralized coordination. The research identifies significant limitations of DQN in complex scenarios, including catastrophic forgetting phenomena. Novel coordination and communication efficiency metrics were developed to quantitatively assess agent performance. The results contribute to the advancement of intelligent evacuation systems by demonstrating how MARL can overcome the limitations of classical approaches through adaptive learning and decentralized coordination, leveraging distributed problem solving. The research provides a comprehensive framework for developing evacuation policies that can generalize across different spatial layouts and agent populations while maintaining computational efficiency suitable for real-time applications.

Keywords: Multi-Agent Reinforcement Learning, Emergency Evacuation, Decentralized Coordination, Multi-Agent systems, Partially Observable Environments.

UN Sustainable Development Goals

The Sustainable Development Goals (SDGs) of the United Nations provide a global framework to achieve a better and more sustainable future for all. It includes 17 goals to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice. These goals are designed to be a blueprint for achieving a more equitable and sustainable society by 2030, promoting prosperity while protecting the planet.

The specific Sustainable Development Goals mentioned have the following names:

SDG 3 Good Health and Well-being

SDG 11 Sustainable Cities and Communities

SDG 16 Peace, Justice and Strong Institutions

SDG	Target	Contribution	Performance Indicators and Metrics
3	3.6	Developing intelligent evacuation systems can significantly reduce casualties and injuries during emergency situations by optimizing escape routes and coordination.	Reduction in evacuation time and casualty rates in emergency scenarios
	3.d	Enhancing emergency response capabilities through MARL contributes to strengthening health emergency preparedness and risk reduction.	Improvement in emergency response efficiency metrics
11	11.5	Multi-agent evacuation systems help reduce deaths and economic losses caused by disasters by enabling more efficient and coordinated emergency responses.	Percentage improvement in evacuation success rates and reduction in disaster-related losses
	11.b	Contributing to disaster risk reduction strategies through intelligent coordination systems that can adapt to various emergency scenarios and building layouts.	Number of adaptive evacuation policies developed for different urban environments
16	16.7	Developing transparent and accountable decision-making systems for emergency management that ensure inclusive and participatory evacuation strategies.	Implementation of fair and inclusive evacuation algorithms that consider vulnerable populations

Acknowledgements

I would like to express my sincere gratitude to all those who supported and guided me throughout this journey. Writing this thesis has been both a significant challenge and a personal milestone, and I am deeply thankful to everyone who contributed to its completion in their unique way.

I owe special thanks to my supervisor, Professor António Costa, for his invaluable guidance, patience, and unwavering support. The transformation of a complex concept into a concrete and meaningful project was made possible through your expertise and encouragement. I am particularly grateful for your trust in my work and the autonomy and responsibility you granted me throughout this process. This experience has shaped my academic development and my approach to tackling complex challenges.

To my family, I extend my deepest gratitude for being my foundation. Your love, encouragement, and unwavering belief in me have been a constant source of strength. I want to thank my parents, Cristina and Pedro, for always inspiring me to strive for excellence and for teaching me the importance of perseverance, regardless of the obstacles we face. I thank my sisters, Rita and Mariana, for being my first role models; their example, wisdom, and resilience shaped the values and ambition I carry with me today. And I thank my nephew and godson, Martim, who brought light into my life at the right moment. His smile made difficult days brighter and reminded me daily of what truly matters. To the little one who is on the way, you have already filled our hearts with joy and anticipation, reminding me that the best chapters are yet to be written. This achievement belongs to all of you as much as it does to me.

To my friends, who made this trip unforgettable. To Bruna, Choca, Lulu, Malva, Phranklin, Tixa, Afonso, Ávila, and Dinis, my first university family, thank you for sharing the most stressful moments, such as exam periods, but also the unforgettable memories that I will carry with me for the rest of my life. To Kiko and Toni, my steadfast companions who never let me give up and reminded me why I belonged here.

To Carol, Sofia, Tança, and Bia, my dearest friends since elementary school, who taught me what a true friendship means and have been my anchor through every chapter of life. To Ricky, Dinis, and Luís, who believed in me before I believed in myself and convinced me that FEUP was the right path for me, this journey began the day you submitted my application. To Mariana, *Pintos* and to all my JuniFEUP friends, who made every moment memorable and reminded me that the best part of university isn't what you learn in classrooms, but the bonds you forge along the way.

To all of you, my heartfelt thanks.

Inês Matos de Oliveira

“It always seems impossible until it’s done.”

Nelson Mandela

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation	2
1.3	Objectives	2
1.4	Research Questions	2
1.5	Document Structure	3
2	Background Knowledge	4
2.1	Foundations of Reinforcement Learning	4
2.1.1	The Markov Decision Process Framework	5
2.1.2	Policies and Value Functions	6
2.1.3	Returns and the Objective Function	7
2.1.4	Taxonomy of RL Algorithms	8
2.1.5	Exploration Strategies	9
2.2	Multi-Agent Reinforcement Learning	9
2.2.1	Formal Framework: Decentralized Partially Observable MDPs	10
2.2.2	Fundamental Challenges in MARL	10
2.2.3	Learning Paradigms in MARL	11
2.3	CTDE and Representative MARL Algorithms	12
2.3.1	Centralized Training with Decentralized Execution Framework	12
2.4	Learning and Communication in MARL	13
2.4.1	Communication Paradigms and Message Types	14
2.4.2	Learning Communication Protocols	14
2.4.3	Communication in Centralized Training with Decentralized Execution . .	15
2.4.4	Applications to Evacuation Scenarios	16
2.5	Simulation Environments for MARL	17
2.5.1	Taxonomy of MARL Simulation Platforms	18
2.5.2	Established MARL Simulation Frameworks	19
2.5.3	High-Performance and Scalable Simulation	19
2.5.4	Domain-Specific Simulation for Evacuation Scenarios	19
2.6	Crowd Modelling and Evacuation Dynamics	20
2.6.1	Emergent Phenomena in Crowd Evacuations	21
2.6.2	MARL Applications in Evacuation Scenarios	22
3	Literature Review	24
3.1	Methodology for Literature Review	24
3.1.1	Keywords	25
3.1.2	Search Query and Criteria	25

3.2	Traditional Approaches to Distributed Problem Solving	26
3.3	MARL-Based Approaches to Distributed Problem Solving	27
3.4	Hybrid and Comparative Approaches	29
3.5	Coordination Challenges in Multi-Agent Systems	31
3.6	Summary and Research Gaps	32
3.7	Conclusions	34
4	Methodological Framework	36
4.1	Problem Statement	36
4.1.1	Scope	37
4.2	Problem Formalization	37
4.2.1	Emergency Evacuation as a Coordination Problem	37
4.2.2	Decentralized Partially Observable Markov Decision Process (Dec-POMDP)	38
4.3	Simulation Environment	42
4.3.1	Framework: PettingZoo and MPE	43
4.3.2	Custom Multi-Agent Evacuation Environment	43
4.3.3	Environment Dynamics and Agent Observations	45
4.4	Solution Architecture	46
4.4.1	CTDE for Multi-Agent Evacuation	46
4.4.2	Multi-Agent Learning Algorithms	46
4.4.3	Neural Network Architecture	47
4.4.4	Technologies and Dependencies	47
4.5	Limitations of Classical Models and the Case for MARL	48
4.6	Constraints and Limitations	49
4.7	Summary	51
5	Experimental Results	52
5.1	Experimental Setup	52
5.1.1	Algorithms	52
5.1.2	Training Parameters	53
5.1.3	Scenarios	54
5.1.4	Reproducibility	57
5.1.5	Environment Flexibility	57
5.2	Comparative Algorithm Performance and Reward Function Effectiveness	58
5.2.1	Algorithm Performance Hierarchy and Learning Stability	58
5.2.2	Reward Function Impact on Coordination Emergence	60
5.2.3	Generalization Capabilities and Scalability Implications	61
5.2.4	Scenario Complexity Impact on Learning Dynamics	61
5.2.5	Implications for Real-World Deployment	62
5.3	Emergent Coordination Behaviors	63
5.3.1	Implicit vs. Explicit Coordination	63
5.3.2	Coordination Index: Methodology and Implementation	64
5.3.3	Scalability and Generalization	65
5.3.4	Empirical Results and Analysis	67
5.3.5	Implications for Real-World Evacuation Systems	68
5.4	Environmental Complexity Analysis	68
5.4.1	Impact of Spatial Constraints and Obstacles	69
5.4.2	Communication in Complex Spatial Environments	70
5.4.3	Algorithm-Specific Responses to Environmental Complexity	71

5.4.4	Reward Function Calibration Across Complexity Levels	72
5.4.5	Scalability Implications of Environmental Complexity	73
5.4.6	Implications for Real-World Deployment	73
5.5	Comprehensive Performance Evaluation	74
5.5.1	Evacuation Success and System Reliability	74
5.5.2	Temporal Efficiency and Evacuation Dynamics	75
5.5.3	Spatial Navigation Efficiency	77
5.5.4	Safety Performance and Collision Analysis	78
5.5.5	Integrated Performance Assessment	79
5.6	Comparison with Traditional Approaches	79
5.6.1	Advantages of MARL	79
5.6.2	Identified Limitations	81
5.6.3	Implications for Practical Deployment	84
5.7	Summary	84
6	Conclusions	88
6.1	Main Contributions	88
6.2	Future Work	89
	References	92
A	Algorithm Hyperparameters	98
A.1	General Training Configuration	98
A.2	Algorithm Implementation Details	99
A.2.1	Proximal Policy Optimization (PPO)	99
A.2.2	Advantage Actor-Critic (A2C)	100
A.2.3	A2C Hyperparameters for Communication Scenarios (4 and 5)	100
A.2.4	Deep Q-Network (DQN)	101
A.2.5	DQN Hyperparameters for Scenario1	102
A.2.6	DQN Hyperparameters for Scenarios 2 to 5	103
A.3	Network Architecture	104
A.4	SuperSuit Wrapper Configuration	105
B	Experiment Results	106
B.1	Scenario 1	106
B.2	Scenario 2	108
B.3	Scenario 3	110
B.4	Scenario 4	112
B.5	Scenario 5	114

List of Figures

2.1	The agent-environment interaction loop in a Markov Decision Process. At each time step t , the agent observes state S_t , selects action A_t , receives reward R_{t+1} , and transitions to new state S_{t+1} . This cyclical process continues until a terminal state is reached or a maximum time horizon is exceeded [58].	6
2.2	Comprehensive taxonomy of reinforcement learning algorithms, showing the hierarchical organization from high-level paradigms to specific algorithmic implementations. The taxonomy illustrates the evolution from classical tabular methods to modern deep reinforcement learning approaches.	8
2.3	Communication autoencoder architecture in multi-agent reinforcement learning. The left side shows the communication topology between agents, where each agent generates messages $c_t^{(i)}$ and actions $a_t^{(i)}$ based on their observations $o_t^{(i)}$. The right side details the internal architecture of the communication autoencoder, which processes visual observations through an image encoder and incoming messages through a message encoder to generate coordinated actions and outgoing communications.	15
3.1	Overview of the Systematic Literature Review Process.	25
4.1	Architecture of the custom evacuation environment. Agents interact with a continuous 2D space populated with obstacles and meeting points. Each agent perceives the world locally and must evacuate efficiently under congestion constraints. . . .	44
5.1	Scenario 1: Free navigation with 2 agents (blue circles) moving toward the visible meeting point (green circle) in an open environment without obstacles.	54
5.2	Scenario 2: Corridor navigation with 6 agents (blue circles) navigating through bottlenecks created by walls (gray circles) toward the meeting point (green circle).	55
5.3	Scenario 3: Complex navigation with 6 agents (blue circles) facing both corridor walls (gray circles) and multiple obstacles (gray circles in the middle) toward the meeting point (green circle).	55
5.4	Scenario 4: Communication-based evacuation with 6 agents (blue circles) discovering and sharing the meeting point (green circle) location through inter-agent communication.	56
5.5	Scenario 5: Communication-based evacuation with 6 agents (blue circles) discovering and sharing the meeting point (green circle) location while navigating around obstacles (gray circles).	57

5.6	Learning curves with moving average and confidence intervals for PPO, A2C, and DQN algorithms across five evacuation scenarios. The x-axis represents training steps (0-700k), the y-axis shows cumulative reward, and shaded areas indicate reward variability across multiple training runs.	59
5.7	Emergent queue formation behavior in Scenario 2 (corridor navigation) showing 6 agents (blue circles) naturally organizing into a single-file formation when approaching the bottleneck created by wall constraints (dark blue big circles). The agents demonstrate spontaneous coordination without explicit queue-forming rules, with the leading agent positioned closest to the meeting point (green circle) and subsequent agents maintaining orderly spacing behind.	64
5.8	Queue formation scalability demonstration in Scenario 2 with 12 agents. The agents automatically organizing into stable queue structures, illustrating how emergent coordination behaviors remain consistent across different population sizes.	66
5.9	Coordination index per evaluation episode across all scenarios and algorithms. The index ranges from 0 to 1, where values approaching 1 indicate high spatial coordination (agents clustered near the centroid) and values near 0 indicate dispersion.	67
5.10	Number of collisions with obstacles per episode in Scenarios 2, 3, and 5 for all algorithms. Each panel shows the evolution of collisions throughout evaluation episodes, where lower values indicate better learning of collision prevention strategies.	70
5.11	Spatial distribution of agents using PPO with a collision penalty of 0.3. Agents remain stationary in the center, avoiding the corridor.	72
5.12	Evacuation success rates across all algorithms and scenarios, where darker colors indicate higher success rates. Perfect evacuation (1.0) appears in yellow, while partial failures show darker colors, like dark blue.	75
5.13	Mean evacuation time per episode across all scenarios and algorithms. Lower values indicate superior temporal efficiency, with shaded regions representing variability across evaluation episodes.	76
5.14	Path efficiency across all scenarios, where values near 1.0 indicate optimal navigation. Values slightly above 1.0 may occur due to reward structure design when agents approach meeting points.	77
B.1	Average number of collisions per agent across episodes in Scenario 1. PPO consistently outperforms other algorithms, maintaining near-zero collision rates. DQN exhibits a significant number of collisions, with increasing variance over time.	106
B.2	Comparison of the average distance travelled by agents per episode. A2C leads in initial performance, but PPO and DQN gradually converge towards similar values by the final episodes.	106
B.3	Total number of collisions in the environment during each episode. The trend mirrors the per-agent analysis, with DQN yielding the highest collision counts. PPO and A2C maintain low collision frequencies.	107
B.4	Cumulative reward achieved per episode by each algorithm. PPO and DQN demonstrate relatively higher rewards compared to A2C, although all algorithms show a decreasing trend across episodes.	107
B.5	Average number of collisions per agent throughout episodes in Scenario 2. DQN shows the highest collision rate, while PPO remains the most stable and exhibits the lowest values, suggesting better coordination among agents.	108

B.6	Average number of steps required to discover the goal. A2C appears less efficient, requiring more steps on average. PPO shows improved efficiency, stabilizing around fewer discovery steps.	108
B.7	Comparison of the average distance travelled by agents. A2C consistently covers longer distances, potentially reflecting more exploration, whereas PPO remains the most conservative across episodes.	108
B.8	Mean number of stationary steps per episode, indicating inactivity or indecision. All algorithms remain relatively stable, with PPO displaying minimal fluctuation, which may suggest consistent policy execution.	108
B.9	Total collisions recorded in the environment across episodes. A similar trend to the per-agent collision analysis is observed, with DQN and A2C experiencing more frequent incidents compared to PPO.	109
B.10	Cumulative reward per episode for each algorithm. PPO maintains the highest average reward, while A2C receives consistently lower values, possibly due to excessive exploration or higher penalty accumulation.	109
B.11	Number of collisions per agent over time. A2C shows a significant spike in collision rate, whereas PPO and DQN maintain consistent and low collision frequencies throughout all episodes.	110
B.12	Average number of steps required to discover the target in Scenario 3. While PPO and DQN remain stable, A2C shows a marked increase in discovery steps, particularly in later episodes, indicating performance degradation.	110
B.13	Mean distance travelled by agents in each episode. A2C shows a steady increase, possibly due to inefficient exploration, while PPO demonstrates more conservative and consistent mobility patterns.	110
B.14	Average number of stationary steps per episode, reflecting inactivity. PPO and DQN maintain stable behavior, while A2C increasingly idles over time, suggesting instability in decision-making.	110
B.15	Total collisions in the environment throughout episodes. A2C shows an exponential increase in collision count from episode 3 onward, while DQN and PPO demonstrate minimal and stable collision behavior.	111
B.16	Cumulative reward per episode for each learning algorithm. A2C experiences a sharp decline in total reward after episode 4, likely due to increased collisions and inefficiencies. PPO performs most consistently.	111
B.17	Average number of collisions per agent in Scenario 4. A2C suffers a sharp increase after episode 3, indicating a collapse in behavior. PPO remains relatively stable, while DQN displays a slight downward trend.	112
B.18	Steps needed to discover the target. A2C experiences a significant jump from episode 4, consistent with the rising collisions and lower rewards, while PPO remains efficient throughout.	112
B.19	Mean distance travelled by agents. A2C agents travel increasingly larger distances, likely inefficiently, while DQN and PPO maintain steady, low-mobility patterns.	112
B.20	Average number of inactive steps. DQN shows an upward trend in idle time, while PPO and A2C maintain near-constant low levels.	112
B.21	Total number of collisions across all agents. A2C shows an explosive increase, whereas PPO maintains a low and stable collision rate. DQN improves slightly over time.	113

B.22	Total reward obtained by agents. A2C's performance collapses drastically after episode 3, reaching values below -100. PPO and DQN remain robust, sustaining high reward values.	113
B.23	Total time spent exploring per episode. A2C sees a large spike aligned with rising discovery steps and poor rewards, suggesting inefficient searching.	113
B.24	Proportion of exploratory actions. PPO stabilizes above 0.75, suggesting balanced behavior. A2C drops below 0.6 in episode 3 and recovers slightly, whereas DQN shows a gradual decline.	113
B.25	Mean collisions per agent across episodes. DQN suffers from significantly higher collision rates, while PPO consistently maintains minimal collisions. A2C remains stable but suboptimal.	114
B.26	Average number of steps required to discover a target in Scenario 5. DQN starts with better performance but converges to similar values as A2C by episode 6. PPO consistently shows the lowest discovery steps.	114
B.27	Cumulative distance travelled by agents in Scenario 5. DQN and A2C consistently outperform PPO, suggesting more active exploration or longer trajectories. . . .	114
B.28	Average number of stationary steps in Scenario 5. PPO exhibits minimal inactivity, while DQN increases stationary behavior over time. A2C remains stable.	114
B.29	Total number of collisions in the environment. DQN experiences a high and consistent number of collisions. PPO sustains minimal collisions across all episodes.	115
B.30	Total reward received by agents across episodes. PPO significantly outperforms the others, showing a more stable and less negative reward trajectory.	115
B.31	Time spent exploring during each episode. PPO shows the lowest exploration time, while DQN and A2C increase over time, possibly due to less efficient target acquisition.	115
B.32	Proportion of exploratory behavior over time. PPO maintains the highest exploration ratio, which gradually increases. A2C and DQN both start lower and plateau.	115

List of Tables

3.1	Traditional approaches to distributed problem solving.	27
3.2	MARL-based approaches to distributed problem solving.	29
3.3	Hybrid and comparative approaches in multi-agent coordination.	30
3.4	Representative strategies addressing coordination challenges in MARL.	32
4.1	Project technological requirements and versions.	48
5.1	Training parameters used across all experiments.	54
A.1	Common training parameters across all algorithms	98
A.2	Algorithm implementation configuration	99
A.3	PPO default hyperparameters (Stable-Baselines3)	99
A.4	A2C default hyperparameters (Stable-Baselines3)	100
A.5	A2C hyperparameters optimized for communication scenarios 4 and 5.	100
A.6	DQN default hyperparameters (Stable-Baselines3)	101
A.7	DQN hyperparameters for Scenario 1 multi-agent coordination.	102
A.8	DQN hyperparameters optimized for coordination scenarios 2 to 5.	103
A.9	Default network architecture specifications	104
A.10	SuperSuit transformation pipeline	105

Acronyms and Abbreviations

A2C	Advantage Actor-Critic
AI	Artificial Intelligence
API	Application Programming Interface
CTDE	Centralized Training with Decentralized Execution
Dec-POMDP	Decentralized Partially Observable Markov Decision Process
DQN	Deep Q-Network
MARL	Multi-Agent Reinforcement Learning
MAS	Multi-Agent System
MDP	Markov Decision Process
MDPS	Multi-Agent Distributed Problem Solving
ML	Machine Learning
MLP	Multilayer Perceptron
MPE	Multi-Agent Particle Environment
POMDP	Partially Observable Markov Decision Process
PPO	Proximal Policy Optimization
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
RL	Reinforcement Learning
RQ	Research Question
SAC	Soft Actor-Critic

Chapter 1

Introduction

This chapter introduces the context, motivation, and research objectives of the dissertation, which investigates how Multi-Agent Reinforcement Learning (MARL) can optimize distributed problem solving, particularly in emergency evacuation scenarios. Section 1.1 presents the fundamental concepts of Multi-Agent Distributed Problem Solving (MDPS) and its relevance in dynamic and decentralized environments. Section 1.2 outlines traditional coordination approaches' limitations and highlights MARL's adaptive potential. Section 1.3 defines the main research objectives, while Section 1.4 presents the research questions on which this dissertation was based. Finally, Section 1.5 summarizes the structure of this document.

1.1 Context

Multi-Agent Distributed Problem Solving (MDPS) is a subfield of Artificial Intelligence (AI) and Multi-Agent Systems (MAS) [2] and addresses the problem of how autonomous agents collaborate to solve complex problems in a decentralized and scalable manner. In these systems, agents operate with partial information, making independent decisions and coordinating actions to achieve shared objectives.

This paradigm is especially relevant to domains where decentralization and adaptability under uncertainty are crucial, such as disaster response, autonomous vehicle coordination, and large-scale infrastructure systems [36]. However, achieving effective cooperation requires addressing fundamental challenges, including coordination under partial observability, communication constraints, and robustness in dynamic environments.

A critical application of MDPS is emergency evacuation, where agents representing individuals must navigate dynamic and partially observable environments. These include multiple exits, obstacles, and bottlenecks. Furthermore, success in evacuation depends on the agents' ability to adaptively coordinate, avoid congestion, and maximize throughput [70].

Despite its potential, MDPS still faces challenges, like communication bottlenecks, conflicts among agents with divergent goals, limited scalability, and poor adaptability. These limitations underline the need for more adaptive, learning-based coordination mechanisms like MARL.

1.2 Motivation

Traditional MDPS techniques often rely on predefined coordination protocols or centralized planning, which may fail when faced with unpredictable and fast-changing scenarios. This is particularly problematic in evacuation settings, where emergent behaviors, like herding, clogging, or panic, can lead to dangerous or suboptimal outcomes.

Reinforcement Learning (RL) enables agents to learn behaviors through trial-and-error interaction within the environment. In multi-agent settings, MARL facilitates the distributed learning of coordination strategies without explicit programming. Over time, agents can discover cooperative behaviors, adapt to varying agent densities, and adjust to different exit configurations.

Recent studies show that MARL shows potential in addressing large-scale coordination problems, including emergency evacuations and smart city applications [12, 36]. However, there is a need to explore further the impact of reward structures, agent interaction models, and learning dynamics in such settings.

This thesis leverages MARL frameworks to simulate, train, and evaluate agent behavior under varying environment configurations and reward mechanisms to improve evacuation performance.

1.3 Objectives

Given the dynamic and uncertain nature of emergency evacuation scenarios, this dissertation explores the application of multi-agent reinforcement learning (MARL) to optimize evacuation strategies in partially observable environments. The primary objective is to assess the effectiveness of various MARL algorithms in reducing evacuation times while enhancing coordination among agents. Additionally, the work aims to investigate the impact of reward shaping and different environmental configurations on the learning process, particularly how these factors influence policy convergence and behavioral stability. Finally, the study seeks to perform a comparative analysis between MARL-based approaches and traditional rule-based strategies, focusing on their relative adaptability, scalability, and robustness under diverse operating conditions.

1.4 Research Questions

The following research questions guided this work:

1. How can Multi-Agent Reinforcement Learning (MARL) be applied to develop effective decentralized evacuation strategies in dynamic and partially observable environments?
2. Which MARL algorithms and reward shaping techniques most effectively minimize evacuation time and congestion in multi-agent evacuation scenarios?
3. How do MARL-based evacuation strategies compare to traditional heuristic or rule-based methods in terms of adaptability, scalability, and coordination efficiency?

1.5 Document Structure

This dissertation is organized into six chapters that systematically develop a Multi-Agent Reinforcement Learning approach to emergency evacuation. Chapter 2 establishes the theoretical foundations of RL and multi-agent systems. Chapter 3 presents a systematic literature review following PRISMA guidelines. Chapter 4 formalizes the evacuation problem as a Dec-POMDP and describes the CTDE solution architecture. Chapter 5 presents experimental results comparing PPO, A2C, and DQN algorithms across multiple evacuation scenarios. Finally, Chapter 6 synthesizes the findings and outlines future research directions.

Chapter 2

Background Knowledge

This chapter provides the foundational concepts and frameworks that underpin this dissertation. Section 2.1 introduces the core principles of Reinforcement Learning, including the Markov Decision Process formalism and value functions. Subsection 2.1.4 classifies major families of RL algorithms, distinguishing between value-based, policy-based, and actor-critic methods. Section 2.2 extends these concepts to Multi-Agent Reinforcement Learning, detailing the challenges and formalisms unique to multi-agent settings. Section 2.3 presents the Centralized Training with a Decentralized Execution paradigm, a cornerstone of modern MARL strategies. Subsection 2.2.3 surveys representative MARL algorithms. Section 2.4 explores the interplay between learning and communication in decentralized environments. Section 2.5 reviews key simulation platforms for MARL experimentation, focusing on evacuation scenarios. Finally, Section 2.6 discusses crowd modeling and evacuation dynamics, highlighting how MARL can contribute to adaptive evacuation strategies.

2.1 Foundations of Reinforcement Learning

Researchers have been fascinated by the challenge of creating computational systems capable of learning through interaction with their environment for more than half a century, much like biological organisms adapt and optimize their behavior through experience [58]. This pursuit has led to the development of Reinforcement Learning (RL). This paradigm fundamentally differs from machine learning approaches by emphasizing learning through trial-and-error interaction rather than supervised instruction or unsupervised pattern discovery [62].

Reinforcement Learning represents a computational framework for sequential decision-making in which autonomous agents learn to optimize their behavior by interacting with an environment and receiving feedback in the form of rewards [58]. Unlike supervised learning, where labeled examples provide explicit guidance, or unsupervised learning, where patterns are discovered in static data, RL is grounded in the dynamic interplay between action and consequence. An agent perceives the current state of its environment, selects an action based on its current policy, receives

a numerical reward signal, observes the resulting environmental change, and iteratively refines its decision-making strategy to maximize long-term cumulative reward.

This trial-and-error learning paradigm enables agents to autonomously discover optimal policies even in complex, stochastic environments where explicit models or supervision are unavailable [58]. The power of RL lies in its ability to balance immediate rewards with long-term consequences, a challenge that mirrors many real-world decision-making scenarios from financial portfolio management to autonomous navigation and, as we will explore in this dissertation, coordinated emergency evacuation [2].

2.1.1 The Markov Decision Process Framework

The mathematical foundation of RL rests upon the Markov Decision Process (MDP), a formal framework that captures the essential elements of sequential decision-making under uncertainty [51]. An MDP provides a clean mathematical abstraction for modeling the interaction between an agent and its environment, enabling theoretical analysis and practical algorithm development [58].

Formally, an MDP is defined as a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where:

- $\mathcal{S}$ is the state space, representing all possible configurations of the environment
- $\mathcal{A}$ is the action space containing all actions available to the agent
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the state transition function, where $\mathcal{P}(s'|s, a)$ specifies the probability of transitioning to state s' when taking action a in state s
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the reward function, providing immediate feedback for state transitions
- $\gamma \in [0, 1]$ is the discount factor, determining the relative importance of immediate versus future rewards

The Markov property is central to this formulation, stating that the probability of transitioning to any future state depends only on the current state and action, not on the complete history of past states and actions [51]. Mathematically, this is expressed as:

$$P(S_{t+1} = s' | S_t = s, A_t = a, S_{t-1}, A_{t-1}, \dots, S_0, A_0) = P(S_{t+1} = s' | S_t = s, A_t = a) \quad (2.1)$$

This memoryless property significantly simplifies the learning problem by ensuring that optimal decisions can be made based solely on current state information, without requiring the agent to maintain or reason about its complete interaction history [58].

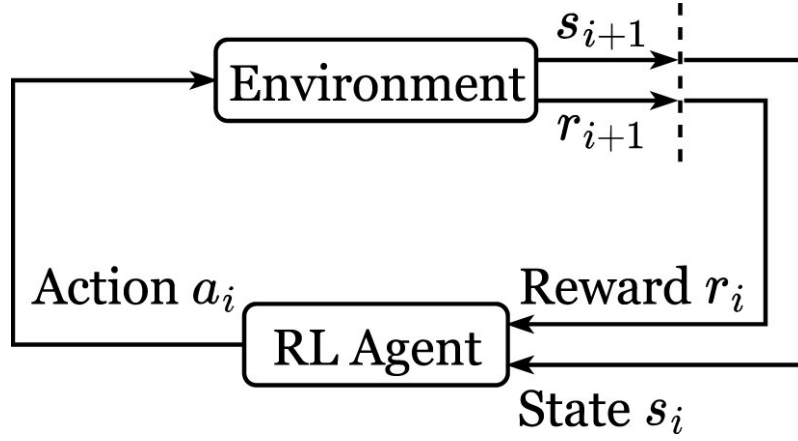


Figure 2.1: The agent-environment interaction loop in a Markov Decision Process. At each time step t , the agent observes state S_t , selects action A_t , receives reward R_{t+1} , and transitions to new state S_{t+1} . This cyclical process continues until a terminal state is reached or a maximum time horizon is exceeded [58].

The agent-environment interaction unfolds as a sequence of discrete time steps, as illustrated in Figure 2.1. At each time step t , the agent observes the current state $S_t \in \mathcal{S}$, selects an action $A_t \in \mathcal{A}$ according to its policy, receives an immediate reward $R_{t+1} \in \mathbb{R}$, and observes the next state S_{t+1} [58]. This generates a trajectory of experience:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \dots \quad (2.2)$$

In evacuation scenarios, the state space $\mathcal{S}$ includes agent positions, crowd density observations, and exit visibility. The action space $\mathcal{A}$ consists of movement directions (north, south, east, west, wait). The transition function $\mathcal{P}$ captures movement success probabilities (higher in less crowded areas), while the reward function $\mathcal{R}$ provides positive rewards for reaching exits and negative rewards for remaining in dangerous areas [47].

2.1.2 Policies and Value Functions

Central to RL is the concept of a policy, which defines the agent's decision-making strategy. A policy $\pi(a|s)$ represents the probability distribution over actions given a state, encapsulating the agent's learned behavior [58]. Recent advances in policy representation have explored neural network parameterizations that enable continuous action spaces and complex behavioral patterns [43].

Reinforcement L employs value functions that estimate expected future rewards to evaluate policy effectiveness. The state-value function $V^\pi(s)$ quantifies the expected return from state s under policy π :

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right] \quad (2.3)$$

Similarly, the action-value function $Q^\pi(s, a)$ estimates the expected return from taking action a in state s and following policy π thereafter:

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right] \quad (2.4)$$

These functions satisfy the fundamental Bellman equations, which express the recursive relationship between current and future values [58]:

$$V^\pi(s) = \sum_a \pi(a|s) \sum_{s'} \mathcal{P}(s'|s, a) [\mathcal{R}(s, a, s') + \gamma V^\pi(s')] \quad (2.5)$$

$$Q^\pi(s, a) = \sum_{s'} \mathcal{P}(s'|s, a) \left[\mathcal{R}(s, a, s') + \gamma \sum_{a'} \pi(a'|s') Q^\pi(s', a') \right] \quad (2.6)$$

Recent theoretical work has extended these foundations to address modern challenges in deep RL, including function approximation guarantees and convergence properties in high-dimensional spaces [35].

In evacuation scenarios, the state-value function $V^\pi(s)$ might represent the expected time to safety from location s . In contrast, the action-value function $Q^\pi(s, a)$ estimates the expected evacuation time when choosing movement direction a from location s . Negative rewards for remaining in dangerous areas and positive rewards for reaching exits naturally shape these value estimates toward efficient evacuation policies.

2.1.3 Returns and the Objective Function

The fundamental objective in RL is to maximize the expected cumulative reward, known as the return [58]. However, the definition of return depends on whether the task is episodic (with natural termination points) or continuing (potentially infinite duration).

For episodic tasks, the return from the time step t is simply the sum of all subsequent rewards [58]:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T \quad (2.7)$$

where T is the terminal time step. However, for continuing tasks or to ensure mathematical convergence, we introduce the discount factor γ to define the discounted return [58]:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (2.8)$$

The discount factor serves multiple purposes [58]. When $\gamma < 1$, it ensures that the infinite sum converges to a finite value, provides a mathematical preference for immediate rewards over distant ones (reflecting uncertainty about the future), and offers computational advantages by bounding the magnitude of returns.

An important recursive relationship emerges from this definition [58]:

$$G_t = R_{t+1} + \gamma G_{t+1} \quad (2.9)$$

This recursive structure is fundamental to many RL algorithms, as it allows value functions to be updated incrementally based on immediate rewards and estimates of future value [58].

2.1.4 Taxonomy of RL Algorithms

Reinforcement Learning algorithms can be categorized along several key dimensions relevant to our evacuation modeling approach. The most fundamental distinction separates **model-free** from **model-based** approaches, based on whether the algorithm explicitly learns environment dynamics [58].

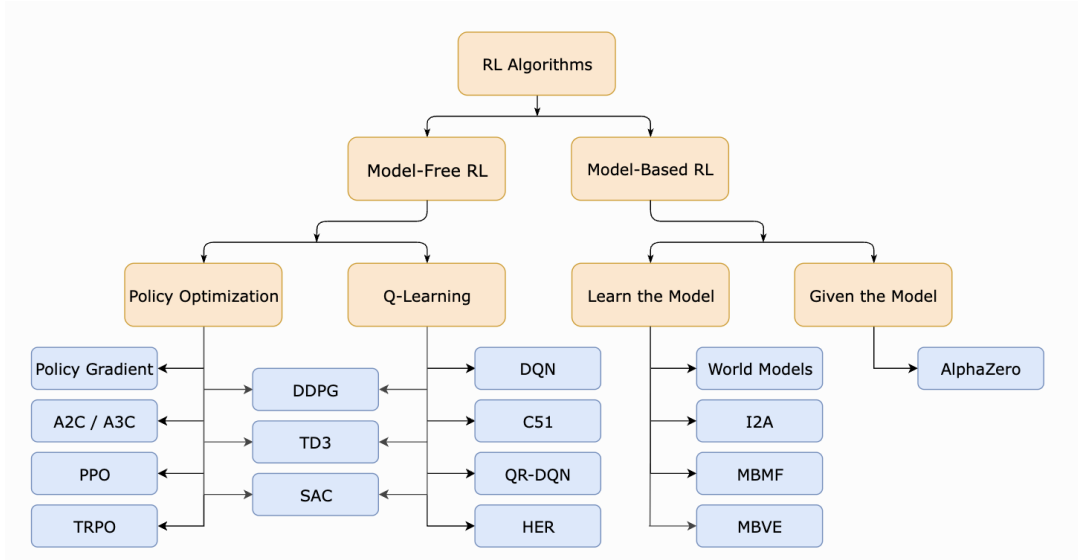


Figure 2.2: Comprehensive taxonomy of reinforcement learning algorithms, showing the hierarchical organization from high-level paradigms to specific algorithmic implementations. The taxonomy illustrates the evolution from classical tabular methods to modern deep reinforcement learning approaches.

Within model-free methods, we distinguish between value-based algorithms that learn action-value functions (such as DQN), policy-based methods that directly optimize parameterized policies, and actor-critic approaches that combine both paradigms (such as PPO and A2C) [6]. Additionally, algorithms can be classified as on-policy or off-policy depending on whether they learn from data generated by the current policy or can utilize data from different behavioral policies.

This taxonomical framework provides the foundation for understanding how DQN (off-policy, value-based), PPO (on-policy, actor-critic), and A2C (on-policy, actor-critic) address the core challenges of sequential decision-making in evacuation scenarios where agents must learn navigation policies while coordinating with others.

2.1.5 Exploration Strategies

Compelling exploration remains fundamental in RL, particularly in evacuation environments with sparse rewards and large state spaces. The exploration-exploitation dilemma requires agents to balance leveraging known good actions and discovering potentially better alternatives [58].

Epsilon-greedy exploration provides a simple approach, selecting random actions with probability ϵ and greedy actions otherwise. While effective in many domains, this approach can be inefficient in complex spatial environments where more sophisticated exploration strategies are required [2].

Entropy regularization encourages exploration by adding an entropy bonus to the reward function, promoting stochastic policies that maintain uncertainty over actions. This approach has proven particularly effective in continuous control domains and forms the basis of PPO's policy optimization [17].

The choice of exploration strategy significantly impacts learning efficiency in evacuation scenarios, where agents must discover effective navigation paths while avoiding dangerous areas and congestion.

2.2 Multi-Agent Reinforcement Learning

Traditional Reinforcement Learning (RL) frameworks assume a single agent interacting with a stationary environment. However, many real-world scenarios are inherently multi-agent, involving multiple autonomous entities that operate concurrently, interact, and influence each other's learning processes [2]. Examples include collaborative robots in manufacturing, intelligent vehicles navigating intersections, autonomous agents managing emergency evacuations, and distributed resource allocation in smart grids. These applications require decentralization, cooperation, and dynamic coordination, aspects that are poorly captured by the classical single-agent RL setting.

Multi-Agent Reinforcement Learning (MARL) extends RL to systems involving multiple learning agents, where the environment includes other learning agents whose Behavior changes over time, making it non-stationary from any individual agent's perspective [23]. This breaks key assumptions of standard RL, including the Markov property and stationary reward distributions, introducing fundamental challenges not present in single-agent systems while enabling the modeling of complex distributed problem-solving behaviors through interaction and coordination.

The significance of MARL lies in its ability to address problems that are inherently distributed, where centralized control is either impossible or undesirable. In evacuation scenarios, for instance, individual agents (representing people) must make local decisions based on partial information while implicitly coordinating to achieve global objectives such as minimizing evacuation time and avoiding dangerous congestion [69].

2.2.1 Formal Framework: Decentralized Partially Observable MDPs

A formal description of MARL is commonly framed in terms of the Decentralized Partially Observable Markov Decision Process (Dec-POMDP), which generalizes the single agent MDP to multi-agent settings with partial observability [44]. A Dec-POMDP is defined by the tuple:

$$\mathcal{M} = \langle N, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, \mathcal{P}, \mathcal{R}, \{\mathcal{O}_i\}_{i=1}^N, \mathcal{Z}, \gamma \rangle \quad (2.10)$$

where:

- N is the number of agents
- $\mathcal{S}$ is the global state space
- $\mathcal{A}_i$ is the action space for agent i , with joint action space $\mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_N$
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the state transition function based on joint actions
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the reward function (potentially shared or individual)
- $\mathcal{O}_i$ is the observation space for agent i
- $\mathcal{Z} : \mathcal{S} \times \mathcal{A} \times \mathcal{O}_1 \times \dots \times \mathcal{O}_N \rightarrow [0, 1]$ is the joint observation function
- $\gamma \in [0, 1]$ is the discount factor

Each agent i maintains a local policy $\pi_i(a_i|o_i)$ that maps its local observation $o_i \in \mathcal{O}_i$ to an action $a_i \in \mathcal{A}_i$. The joint policy $\pi = (\pi_1, \dots, \pi_N)$ determines the collective behavior of all agents.

The objective in cooperative MARL is to find the joint policy that maximizes the expected cumulative reward:

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R_t \right] \quad (2.11)$$

In evacuation scenarios, the global state $s \in \mathcal{S}$ includes positions of all agents, environmental hazards, and exit availability. Each agent i observes only local information o_i such as nearby crowd density, visible exits, and local hazards. Actions a_i include movement directions and waiting. The reward function $\mathcal{R}$ provides positive rewards for reaching exits safely and negative rewards for remaining in dangerous areas or contributing to congestion.

2.2.2 Fundamental Challenges in MARL

MARL introduces several fundamental challenges that distinguish it from single-agent RL, making algorithm design and convergence analysis significantly more complex.

Non-stationarity represents the most significant challenge in MARL. Since all agents are learning simultaneously, the environment as perceived by any individual agent becomes dynamic and unpredictable [23]. This violates the stationarity assumption underlying most RL convergence

guarantees and makes experience replay problematic, as stored experiences may no longer reflect current environment dynamics.

Formally, if we consider agent i 's perspective, the effective transition function becomes:

$$\mathcal{P}_i^{eff}(s'|s, a_i) = \sum_{\mathbf{a}_{-i}} \prod_{j \neq i} \pi_j(a_j|o_j) \cdot \mathcal{P}(s'|s, a_i, \mathbf{a}_{-i}) \quad (2.12)$$

where $\mathbf{a}_{-i}$ represents the actions of all agents except i . As other agents' policies π_j evolve during training, $\mathcal{P}_i^{eff}$ changes continuously, creating a moving target for learning.

Credit Assignment becomes particularly challenging in cooperative settings with global rewards. When multiple agents contribute to a shared outcome, determining which agents' actions were responsible for success or failure is non-trivial [65]. This problem is exacerbated in scenarios with delayed rewards, where the consequences of actions may not be immediately apparent.

Scalability issues arise as the joint action space grows exponentially with the number of agents [2]. For N agents each with $|\mathcal{A}_i|$ actions, the joint action space has size $\prod_{i=1}^N |\mathcal{A}_i|$, making centralized value function estimation computationally intractable for large agent populations.

Coordination challenges emerge when agents must align their behaviors to achieve collective objectives. Without explicit coordination mechanisms, agents may converge to suboptimal equilibria or exhibit conflicting behaviors that impede overall performance.

Partial Observability compounds these difficulties by limiting each agent's knowledge of the global state and other agents' intentions. Agents must make decisions based on incomplete information while reasoning about the likely actions of others.

2.2.3 Learning Paradigms in MARL

MARL algorithms can be distinguished by how they structure learning and coordination across agents. These paradigms have significant implications for evacuation scenarios, where scalability, coordination, and deployment constraints are critical.

One of the most straightforward approaches is independent learning, in which each agent learns its own policy as if operating in isolation, treating other agents as part of the environment [59]. This paradigm offers simplicity and scalability, avoiding centralized components and allowing for decentralized updates. However, it introduces a significant challenge: the non-stationarity of the environment from each agent's perspective due to the concurrent learning and adaptation of other agents. As a result, convergence becomes difficult, and coordination is often suboptimal, particularly in tasks that require synchronized actions or shared objectives.

To address these limitations, the centralized training with decentralized execution (CTDE) paradigm has become the dominant framework in MARL research [44]. CTDE enables agents to access global state information and other agents' actions during training, facilitating the learning of stable and coordinated policies. At execution time, however, each agent operates solely based on local observations, aligning with the practical constraints of decentralized systems. This structure allows for leveraging global coordination signals during training without violating the locality constraints imposed in real-time deployment.

The CTDE approach can be formalized through policy gradient methods, where each agent i optimizes its own policy $\pi_i(a_i|o_i)$ using a centralized critic that evaluates joint actions:

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E} [\nabla_{\theta_i} \log \pi_i(a_i|o_i) \cdot Q(\mathbf{s}, \mathbf{a})] \quad (2.13)$$

Here, the policy depends only on local observation o_i , while the critic $Q(\mathbf{s}, \mathbf{a})$ conditions on the whole state and joint action during training. This separation allows agents to benefit from global coordination signals while still producing locally executable policies.

2.3 CTDE and Representative MARL Algorithms

The practical deployment of multi-agent reinforcement learning systems faces a fundamental tension between the need for coordination during learning and the constraints of decentralized execution in real-world environments. This challenge has led to the emergence of Centralized Training with Decentralized Execution (CTDE) as the dominant paradigm in modern Multi-Agent Reinforcement Learning (MARL), providing a principled framework that balances algorithmic sophistication with deployment feasibility [44].

CTDE addresses the core limitations of both fully centralized and fully decentralized approaches. While theoretically optimal, centralized methods suffer from exponential scaling in the joint action space and require global communication during execution, making them impractical for large-scale or distributed systems. Conversely, independent learning approaches treat other agents as part of the environment, leading to non-stationarity and poor convergence properties due to the constantly changing Behavior of other learning agents [59].

The CTDE paradigm resolves this tension by leveraging global information during training to learn stable value functions and coordination mechanisms while ensuring that the resulting policies depend only on local observations during execution. This approach enables agents to benefit from centralized coordination signals during learning while maintaining the scalability and robustness required for real-world deployment [13].

2.3.1 Centralized Training with Decentralized Execution Framework

The CTDE framework formalizes the idea of learning decentralized policies $\pi_i(a_i|o_i)$ for each agent i , where decisions rely solely on local observations while leveraging a centralized critic $Q(\mathbf{s}, \mathbf{a})$ or $V(\mathbf{s})$ during training. This design enables agents to benefit from global state and joint action information to stabilize learning without violating the decentralization requirements imposed during deployment.

Formally, the policy gradient under CTDE can be expressed as:

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E}_{\tau} [\nabla_{\theta_i} \log \pi_i(a_i|o_i; \theta_i) \cdot Q(\mathbf{s}, \mathbf{a})] \quad (2.14)$$

where τ denotes the trajectory distribution sampled from the environment, θ_i the parameters of agent i 's policy, and $Q(\mathbf{s}, \mathbf{a})$ the centralized critic that evaluates the quality of joint actions given

the global state. Importantly, while training benefits from global awareness, the policy $\pi_i(a_i|o_i; \theta_i)$ remains conditioned exclusively on the agent’s local observation o_i , ensuring compatibility with decentralized execution constraints:

$$\pi_i(a_i|o_i) = \pi_i(a_i|\mathcal{O}_i^{local}) \quad (2.15)$$

Here, $\mathcal{O}_i^{local}$ refers strictly to the locally observable information available to agent i at runtime, excluding access to the global state or private observations from other agents.

This framework provides several advantages in cooperative multi-agent settings. First, centralized critics improve learning stability by offering consistent value estimates based on full-state information, mitigating the non-stationarity inherent in independent learning approaches. Furthermore, the centralized perspective enhances credit assignment by enabling better attribution of rewards to individual agent actions, accelerating convergence and improving collaborative task performance. During execution, the decentralization constraint ensures that computational and communication overheads remain feasible, supporting scalability in complex environments. The modular nature of the approach also permits the potential integration of new agents without necessitating complete retraining, provided the learned policies are sufficiently generalizable.

Nevertheless, CTDE is not without limitations. The assumption that global information is always accessible during training may not be realistic in certain domains, particularly when data is limited or partially observable even during simulation. Moreover, the framework does not entirely resolve the issue of non-stationarity, especially if agents are updated asynchronously or if there is policy drift over time. The quality of the resulting policies is also heavily influenced by the centralized critic’s representational power and the mismatch between training and deployment conditions, which can introduce unintended biases or instability in real-world scenarios [13].

2.4 Learning and Communication in MARL

In Multi-Agent Reinforcement Learning (MARL), the fundamental challenge extends beyond individual policy optimization to encompass the coordination of multiple learning agents operating under partial observability and dynamic interdependencies. Unlike single-agent settings where the environment remains stationary, multi-agent systems introduce non-stationarity as each agent’s learning process affects the environment perceived by others [74]. This complexity necessitates sophisticated information sharing and coordination mechanisms, making communication a cornerstone of effective multi-agent learning.

Communication in MARL serves multiple critical functions: it broadens agents’ observational scope beyond their local sensors, enables coordination of joint actions, facilitates the sharing of learned knowledge, and supports the emergence of collaborative strategies that would be impossible through independent learning alone [74]. Integrating communication mechanisms into learning

algorithms represents a fundamental shift from treating other agents as environmental noise to explicitly modeling them as collaborative or competitive partners whose behaviors can be influenced and predicted through information exchange.

The significance of communication becomes particularly pronounced in scenarios requiring tight coordination, such as emergency evacuation, where agents must navigate complex spatial constraints while avoiding congestion and ensuring collective safety. In such contexts, the ability to share intentions, plans, and situational awareness can mean the difference between successful coordination and catastrophic failure [30].

2.4.1 Communication Paradigms and Message Types

In MARL, communication expands agents' perceptual range, facilitates coordination, and supports joint decision-making. Capabilities especially critical in evacuation scenarios [74]. Observation sharing allows agents to disseminate local information about crowd density, exits, or hazards, enhancing collective situational awareness. Intention sharing enables agents to broadcast planned movements, supporting predictive coordination and mitigating congestion [30]. The effectiveness of communication also depends on the topology: while broadcast maximizes coverage, it may overload bandwidth, whereas targeted communication restricts messages to relevant recipients based on proximity or urgency. Designing communication protocols for evacuation thus involves balancing information utility with latency, selectivity, and robustness under high-stress, bandwidth-limited conditions.

2.4.2 Learning Communication Protocols

Developing communication protocols in MARL requires agents to learn individual policies and coordination strategies simultaneously. This dual learning problem is particularly challenging in evacuation scenarios where communication effects are often indirect and delayed.

Modern approaches to learning communication in MARL typically employ neural architectures that can process environmental observations and other agents' messages. Figure 2.3 illustrates a representative communication framework where agents learn to encode relevant information into messages while decoding received communications to inform their decision-making process.

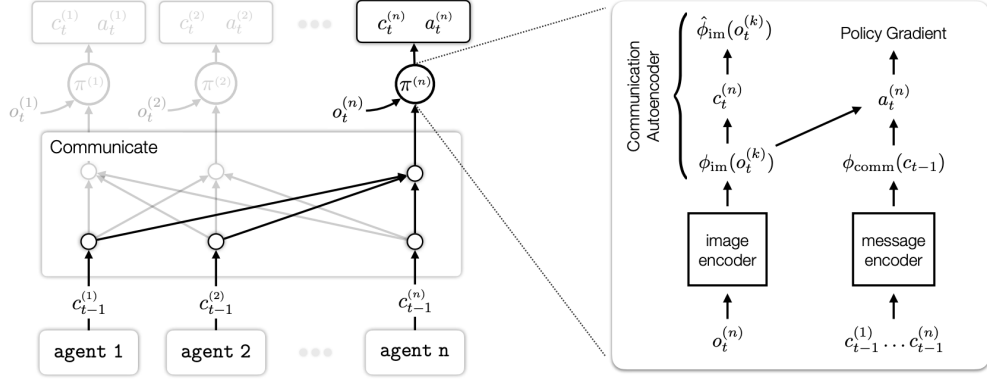


Figure 2.3: Communication autoencoder architecture in multi-agent reinforcement learning. The left side shows the communication topology between agents, where each agent generates messages $c_t^{(i)}$ and actions $a_t^{(i)}$ based on their observations $o_t^{(i)}$. The right side details the internal architecture of the communication autoencoder, which processes visual observations through an image encoder and incoming messages through a message encoder to generate coordinated actions and outgoing communications.

The communication autoencoder architecture shown in Figure 2.3 demonstrates how agents can learn to compress relevant environmental information into compact messages $c_t^{(n)}$ while simultaneously processing incoming messages $c_{t-1}^{(1)} \dots c_{t-1}^{(n)}$ from other agents. The image encoder ϕ_{im} processes visual observations $o_t^{(n)}$, while the message encoder ϕ_{comm} handles communication inputs, with both streams contributing to policy gradient updates for coordinated Behavior.

Attention-Based Communication allows agents to dynamically determine which information to share and prioritize based on current context [24]. The attention mechanism can be formulated as:

$$\alpha_{ij} = \frac{\exp(f_{att}(h_i, h_j))}{\sum_{k=1}^N \exp(f_{att}(h_i, h_k))} \quad (2.16)$$

where α_{ij} represents the attention weight that agent i assigns to agent j 's message, and h_i, h_j are the hidden representations of the respective agents.

In evacuation scenarios, learned communication protocols can significantly improve coordination by enabling agents to share critical information about exit accessibility, hazard locations, and crowd conditions. The autoencoder framework allows agents to learn compact representations of complex spatial information, facilitating efficient message passing even under bandwidth constraints typical of emergencies.

2.4.3 Communication in Centralized Training with Decentralized Execution

The integration of communication mechanisms within the Centralized Training with Decentralized Execution (CTDE) framework introduces both significant opportunities and unique challenges.

During the training phase, agents have access to global information. They can develop communication protocols in a coordinated manner, taking advantage of centralized critics to evaluate the overall impact of communication strategies on team performance. However, during execution, agents must operate solely based on local observations and any messages received from peers, making the effectiveness of the learned communication policies a critical factor for success.

In the training phase, global state information allows learning communication policies that optimize team-wide objectives. The centralized critic plays a pivotal role in this context by assessing the joint utility of action-communication combinations, offering valuable learning signals to optimize communication protocols. An extended policy gradient formulation can capture this:

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E}_{\tau} [\nabla_{\theta_i} \log \pi_i(a_i, m_i | o_i, \mathbf{m}_{-i}; \theta_i) \cdot Q(\mathbf{s}, \mathbf{a}, \mathbf{m})] \quad (2.17)$$

where m_i denotes the message emitted by agent i , $\mathbf{m}_{-i}$ represents the set of messages received from the other agents, and $Q(\mathbf{s}, \mathbf{a}, \mathbf{m})$ is a centralized critic that evaluates the effectiveness of the joint action and communication strategy in a given global state.

During decentralized execution, agents must rely exclusively on local observations and received messages, operating under realistic communication constraints such as limited bandwidth, latency, message loss, or asynchronous updates. The communication protocols learned during training must be robust enough to remain effective under these limitations despite the absence of centralized coordination or full observability.

Recent work on fully independent communication has further highlighted the potential of decentralized communication learning. In this setting, agents learn to communicate effectively without parameter sharing or explicit coordination during training, thereby increasing robustness and transferability across domains [50]. While this independent learning paradigm introduces greater challenges due to the increased non-stationarity and lack of shared structure, it has shown promise in enabling agents to develop communication strategies that generalize better and remain effective in dynamic and heterogeneous multi-agent environments.

2.4.4 Applications to Evacuation Scenarios

In evacuation scenarios, communication is vital in ensuring safety and efficiency, especially when agents must navigate constrained spaces under time pressure and uncertainty. Agents representing evacuees must coordinate their movements, share information about the environment, such as exit availability and hazard locations, and adapt their strategies dynamically as the situation evolves. These challenges are exacerbated in conditions involving blocked passages, shifting crowd densities, or limited visibility [47]

A central aspect of effective communication in this context is situational awareness. By sharing local observations regarding exit accessibility and crowd density, agents can collectively avoid overloading specific escape routes and promote more balanced usage of the available exits. This type of information exchange can significantly reduce congestion and improve evacuation times

[72]. However, communication bandwidth limitations and the rapidly changing environment constrain what and when information should be shared. Determining the relevance and urgency of shared data thus becomes a non-trivial aspect of the communication strategy.

Another key communication modality involves intention broadcasting, whereby agents inform nearby peers of their intended movements. This foresight enables predictive coordination and the avoidance of local conflicts. For instance, if multiple agents approach a narrow corridor simultaneously, pre-emptive communication of their trajectories allows them to stagger their entry and reduce the likelihood of bottlenecks or gridlock, enhancing overall throughput [25].

In more critical situations, emergency signaling becomes necessary. This involves transmitting urgent messages alerting others to blocked exits, hazardous areas, or emergent threats. Given the importance of timely propagation, communication protocols must incorporate prioritization mechanisms that elevate the urgency of such signals above routine updates. Ensuring that life-saving information reaches agents quickly and reliably is essential for system-wide responsiveness.

Communication success in evacuation tasks hinges on several factors: the robustness of the protocols under stress conditions, the scalability of the approach to large agent populations, and the adaptability to diverse architectural layouts and emergency scenarios. Empirical evidence suggests that agents with learned communication capabilities outperform their non-communicative counterparts, particularly in complex environments with multiple exits, uncertain hazards, and dynamic obstacles. These improvements are more pronounced when coordination and adaptation are required in real-time.

Ultimately, the integration of communication learning into multi-agent reinforcement learning frameworks marks a significant advancement in the ability of agents to operate effectively under uncertainty and partial observability. As research continues to focus on safety-critical domains such as evacuation, developing robust, efficient, and generalizable communication protocols will remain a pivotal challenge and a promising area for impactful contributions.

2.5 Simulation Environments for MARL

Developing and evaluating Multi-Agent Reinforcement Learning (MARL) algorithms critically depend on robust, scalable, and realistic simulation environments. In contrast to single-agent settings, where complexity arises primarily from environmental dynamics, multi-agent scenarios introduce additional layers of intricacy due to inter-agent interactions, emergent behaviors, and coordination requirements. Simulation environments in this context serve not only as training grounds for algorithm development but also as essential platforms for studying collective phenomena, validating theoretical assumptions, and bridging the gap between algorithmic design and real-world deployment [34].

Designing simulation environments for MARL introduces unique challenges. As the number of agents increases, computational scalability becomes a primary concern, with complexity often growing superlinearly with the agent population. These environments must also support diverse interaction modalities, including physical proximity, communication protocols, and shared

information structures. Moreover, they should accommodate asynchronous behaviors, varying observation frequencies, and heterogeneous action spaces, all while ensuring reproducibility and consistency across experimental runs.

Modern simulation platforms must strike a careful balance between realism, efficiency, flexibility, and standardization. Realism requires high-fidelity modeling of physical dynamics, sensory limitations, and structural constraints in real-world environments. Efficiency demands optimized computation to support large-scale experiments and fast training cycles. Flexibility requires modular architectures representing varied agent types, reward formulations, and task configurations. Standardization, meanwhile, enables reproducible research by defining common APIs, evaluation metrics, and benchmark tasks.

Beyond algorithmic validation, simulation environments are vital in safety-critical applications such as evacuation, disaster response, and autonomous systems. These scenarios often involve dangerous or ethically sensitive conditions that cannot be tested directly in the real world. Simulations allow for controlled experimentation with extreme or rare events, including high-density crowds, structural failures, or panic-induced behaviors, providing a safe yet informative proxy for system evaluation.

2.5.1 Taxonomy of MARL Simulation Platforms

Simulation environments in MARL can be categorized along several key dimensions that reflect their abstraction level, computational architecture, interaction paradigms, and standardization. At one end of the spectrum, highly abstract environments such as grid worlds, matrix games, and particle systems prioritize computational efficiency and clarity of analysis. These environments are ideal for theoretical exploration, rapid prototyping, and algorithm comparison but often fail to capture the intricacies of real-world deployment [61]. On the opposite end, realistic environments integrate detailed physics, sensor models, and rich environmental features that increase the transferability of learned policies at the cost of higher computational demands [27].

The underlying computational architecture further distinguishes simulation platforms. Traditional CPU-based environments offer flexibility and ease of development, but their scalability is often limited. In contrast, GPU-accelerated simulators exploit parallelism to support large agent populations and intensive computations, enabling studies of emergent behavior at scale [8, 34].

Another dimension concerns the modes of interaction supported by the environment. Some platforms focus on spatial interactions, where agents navigate physical spaces and manage collisions, particularly relevant for robotics or evacuation tasks. Others emphasize communication and decision-sharing protocols, capturing distributed coordination in information-rich domains [30]. Hybrid environments offer both modalities, facilitating integrated physical and informational coordination studies.

Standardization is also a distinguishing factor. While custom-built environments afford maximum flexibility for specific domains, they often sacrifice reproducibility and interoperability. Standardized platforms with consistent APIs and evaluation pipelines, such as PettingZoo, facilitate cross-comparison of algorithms and promote best practices in experimental design [2].

2.5.2 Established MARL Simulation Frameworks

Several frameworks have emerged as key tools in the MARL research landscape. PettingZoo, for example, extends the OpenAI Gym interface to support multi-agent interactions through Agent Environment Cycle (AEC) and parallel execution modes [61]. Its included environments, such as the Multi-Agent Particle Environment (MPE), enable basic coordination, communication, and competition studies in 2D physics-based settings. PettingZoo’s primary strength lies in its accessibility and standardization, though its environments are relatively simple and abstract.

Unity ML-Agents integrates reinforcement learning with the Unity game engine, enabling 3D simulations with realistic physics and rendering [27]. Supporting multiple observation types and both discrete and continuous action spaces, Unity ML-Agents is suited for complex tasks involving spatial awareness and interaction. Its architecture separates simulation from learning, allowing distributed training across multiple instances and making it highly versatile for research and applied scenarios.

VMAS (Vectorized Multi-Agent Simulator) addresses the scalability challenge by providing a fully GPU-accelerated simulation framework capable of running thousands of agents in parallel [8]. Supporting continuous control, custom reward structures, and efficient physics modeling, VMAS enables the investigation of large-scale coordination and emergent behaviors that are computationally infeasible in traditional simulators.

2.5.3 High-Performance and Scalable Simulation

As MARL research increasingly targets large-scale systems, scalability has become critical. GPU-accelerated simulation frameworks such as Vogue leverage parallel computing architectures to simulate tens of thousands of agents with high throughput and real-time performance [34]. These systems implement agent logic entirely on GPU hardware, exploiting SIMD parallelism to avoid the latency and synchronization overheads of CPU-based simulations.

Distributed simulation architectures push scalability further by partitioning the environment across multiple GPUs or computing nodes. In such systems, spatial decomposition is used to divide the global environment into non-overlapping subregions, each managed independently but with consistency maintained across boundaries. This approach enables the simulation of city-scale systems, including traffic networks or evacuation scenarios involving millions of agents.

Memory-efficient representations are also essential in large-scale environments. Techniques such as sparse encoding, hierarchical data structures, and dynamic memory allocation help manage memory consumption. At the same time, streaming architectures ensure that only the relevant portions of the environment are maintained in active memory.

2.5.4 Domain-Specific Simulation for Evacuation Scenarios

Evacuation scenarios impose unique simulation demands that go beyond general-purpose MARL frameworks. Accurate modeling of pedestrian behavior, physical constraints, and emergency dynamics is essential for realistic and actionable insights. Crowd dynamics models, such as the

Social Force Model [19], represent individual motion as the result of goal-directed forces, social interactions, and environmental constraints:

$$m_i \frac{d\mathbf{v}_i}{dt} = \mathbf{f}_i^{goal} + \sum_{j \neq i} \mathbf{f}_{ij}^{social} + \sum_w \mathbf{f}_{iw}^{wall} \quad (2.18)$$

This formulation enables realistic reproduction of pedestrian flows, bottleneck formation, and collective motion and can be integrated with decision-making policies learned via [47].

Dynamic hazards, including fire spread, smoke propagation, and structural degradation, further complicate evacuation simulations. Environments must model time-varying risks and enforce constraints that evolve as agents act and conditions change. The ability of agents to adapt policies in real-time becomes crucial in these settings [72].

Heterogeneity among agents adds another layer of realism. Variability in mobility, knowledge, roles, and risk tolerance affects evacuation strategies and group behavior. Simulating this diversity enables the study of inclusive evacuation strategies that account for vulnerable populations and differing objectives.

Validation of such simulations remains challenging due to the limited availability of real-world evacuation data. However, empirical studies, evacuation drills, and historical incidents offer calibration points. Methods such as sensitivity analysis and uncertainty quantification can strengthen the credibility of simulation results.

Integrating MARL with domain-specific evacuation simulators opens promising avenues for developing adaptive, robust, and efficient evacuation systems. These platforms enable agents to learn context-sensitive policies that optimize global outcomes while navigating uncertain, dynamic, and high-risk environments [36].

2.6 Crowd Modelling and Evacuation Dynamics

Evacuation scenarios represent one of the most challenging and socially critical applications of multi-agent systems, where the coordination of large numbers of individuals under stress, uncertainty, and time pressure can determine the difference between life and death. Unlike conventional multi-agent coordination problems, evacuation dynamics are characterized by high-stakes decision-making, emergent collective behaviors, and the complex interplay between individual rationality and group welfare [19]. The computational modeling of these scenarios serves multiple purposes: understanding the fundamental mechanisms underlying crowd behavior, designing safer built environments, developing emergency response protocols, and creating intelligent systems capable of real-time evacuation assistance.

The complexity of evacuation dynamics arises from the intersection of physical constraints, psychological factors, and social interactions operating simultaneously across multiple scales. At the individual level, evacuees must navigate spatial environments while processing incomplete information about exit locations, hazard distribution, and crowd density. At the interpersonal level, social forces such as imitation, leadership, and competitive Behavior influence movement patterns

and decision-making. At the collective level, emergent phenomena such as bottleneck formation, crowd waves, and spontaneous coordination patterns emerge from the aggregate Behavior of many individuals [33].

Traditional approaches to evacuation modeling have relied primarily on deterministic rule-based systems and physics-inspired models that treat crowds as continuous fluids or collections of particles subject to social forces. While these approaches have provided valuable insights into crowd dynamics, they often fail to capture the adaptive, learning-based behaviors that characterize human decision-making under uncertainty. Integrating Multi-Agent Reinforcement Learning (MARL) into evacuation modeling represents a paradigm shift toward intelligent, adaptive agents capable of learning optimal evacuation strategies through experience and interaction [47].

2.6.1 Emergent Phenomena in Crowd Evacuations

Evacuation scenarios exhibit a rich spectrum of emergent behaviors that arise from the complex interactions between individual decision-making and collective dynamics. Understanding these phenomena is crucial for developing realistic simulation models and effective evacuation strategies.

Herding Behavior represents one of the most pervasive and well-documented phenomena in evacuation scenarios. Individuals tend to follow others rather than independently assess the environment, particularly when facing uncertainty about exit locations or optimal routes [11]. Recent empirical studies have demonstrated that herding behavior is particularly pronounced at critical decision points such as intersections and exit choices, where individuals seek additional information by observing others' actions [33]. While herding can facilitate coordination and information sharing, it often leads to suboptimal outcomes, such as exit underutilization and increased evacuation times.

The mathematical modeling of herding behavior can be incorporated into MARL frameworks through social influence terms in the reward function:

$$R_i^{social}(s, a) = \alpha \sum_{j \in N_i} \mathbb{I}[a_i = a_j] \cdot w_{ij} \quad (2.19)$$

Where α controls the strength of social influence, N_i represents agent i 's neighborhood, $\mathbb{I}[a_i = a_j]$ is an indicator function for action alignment, and w_{ij} represents the social weight between agents based on factors such as proximity, familiarity, or perceived expertise.

The Faster-is-Slower Effect constitutes a counterintuitive phenomenon where increased individual urgency leads to decreased collective evacuation efficiency. This effect emerges when high-stress conditions cause individuals to move more aggressively, resulting in increased physical interactions, bottleneck formation, and coordination failures [56]. The faster-is-slower effect demonstrates the fundamental tension between individual rationality and collective optimality in evacuation scenarios.

Recent research has revealed that the faster-is-slower effect operates within specific parameter ranges and can be overcome under extreme pressure conditions, where very high-stress levels

actually improve evacuation performance by breaking deadlock situations [56]. This suggests that the relationship between individual Behavior and collective outcomes is more complex than initially understood, with multiple equilibria and phase transitions characterizing the evacuation dynamics.

Arching and Clogging occur when evacuees form semi-circular clusters near narrow exits, creating temporary blockages that impede flow and increase evacuation times. These formations arise from the combination of spatial constraints, social forces, and competitive Behavior and represent a classic example of how local interactions can produce global inefficiencies [19].

The formation of arches can be modeled in MARL through spatial reward shaping that penalizes high-density configurations:

$$R_i^{spatial}(s) = -\beta \cdot \rho_i(s) \cdot \mathbb{I}[\text{near_exit}(s_i)] \quad (2.20)$$

where β is a penalty coefficient, $\rho_i(s)$ represents the local density around agent i , and $\mathbb{I}[\text{near_exit}(s_i)]$ indicates proximity to exit areas.

2.6.2 MARL Applications in Evacuation Scenarios

Applying Multi-Agent Reinforcement Learning (MARL) to evacuation scenarios offers distinct advantages over traditional rule-based or optimization-based models. Chief among these is the ability to adapt behavior in real-time, coordinate emergently without explicit programming, and handle environments characterized by partial observability and non-stationary dynamics.

A particularly relevant application is adaptive route selection. In evacuation contexts, agents must make movement decisions based on dynamic observations of crowd density, exit accessibility, and the presence of hazards. Through MARL, agents learn to choose evacuation paths that optimize their safety and contribute to collective efficiency. Unlike fixed routing algorithms, MARL agents can adapt to unforeseen conditions and uncover non-obvious but effective strategies in real-time [47]. This decision-making process is naturally framed as a decentralized, partially observable Markov decision process (Dec-POMDP), where each agent relies on local information to act under uncertainty and time pressure. A core challenge in this formulation lies in balancing the exploration of alternative routes with the exploitation of previously successful paths, especially in scenarios where delays can be critical.

Another key benefit of MARL in evacuation scenarios is its capacity to generate emergent coordination. When agents are trained with shared reward functions that reflect global performance, coordination behaviors emerge even without explicit communication. These behaviors may include alternating movement patterns at bottlenecks, self-organized queuing, or the emergence of spontaneous leadership. This can be formalized through a shared reward function that combines multiple evacuation objectives:

$$R_{shared} = w_1 \cdot R_{safety} + w_2 \cdot R_{efficiency} + w_3 \cdot R_{equity} \quad (2.21)$$

Here, R_{safety} penalizes risky or unsafe states, $R_{efficiency}$ rewards timely evacuation, and R_{equity} encourages fair outcomes across the agent population.

Communication and information sharing further enhance the effectiveness of MARL in evacuation settings. Access to accurate and timely information becomes a bottleneck in environments where visibility is limited or hazards evolve dynamically. Incorporating learned communication policies allows agents to share critical observations such as discovering new exits, identifying blocked paths, or reports of high-density congestion. This collective awareness of the situation improves decision-making and fosters more coordinated and informed movement throughout the environment. Empirical evidence shows that communication-aware MARL models consistently outperform their non-communicating counterparts, particularly in scenarios involving multiple exits, dynamic obstacles, or limited observability [47].

Finally, evacuation contexts benefit from modeling heterogeneity among agents. In real-world emergencies, evacuees vary in mobility, environmental familiarity, risk tolerance, and roles (e.g., leaders, followers, or dependents). MARL frameworks can capture this diversity through agent-specific policies and observations. Formally, this can be represented as:

$$\pi_i(a_i|o_i; \theta_i, \phi_i) \quad (2.22)$$

where θ_i defines the policy parameters and ϕ_i encodes the individual characteristics of agent i , such as limited mobility or incomplete knowledge. Considering heterogeneity, MARL systems can better reflect real evacuation behavior and develop more inclusive, robust, and effective evacuation strategies.

Chapter 3

Literature Review

This chapter presents a structured literature review of distributed problem-solving strategies in multi-agent systems, with a particular focus on emergency evacuation scenarios. The review is grounded on a systematic methodology, as outlined in Section 3.1, which ensures comprehensive and reproducible coverage of relevant academic work. The main body of the chapter is organized into four thematic sections. Section 3.2 surveys traditional modeling and routing approaches, including physics-based models and rule-based heuristics. Section 3.3 discusses recent advancements in Multi-Agent Reinforcement Learning (MARL), highlighting key algorithms and applications to evacuation contexts. Section 3.4 reviews hybrid and comparative approaches that integrate MARL with heuristics or structured planning. Section 3.5 explores coordination challenges inherent to MARL, such as non-stationarity and credit assignment, and examines strategies developed to address them. Finally, Section 3.6 summarizes the findings and identifies research gaps that inform the design decisions and contributions of this thesis.

3.1 Methodology for Literature Review

To ensure a rigorous and unbiased survey of relevant research, a systematic literature review was conducted, following the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) guidelines. The process comprised five main steps: definition of keywords, construction of search queries, removal of duplicates, screening of results, and full-text analysis. The overall workflow is illustrated in Figure 3.1.

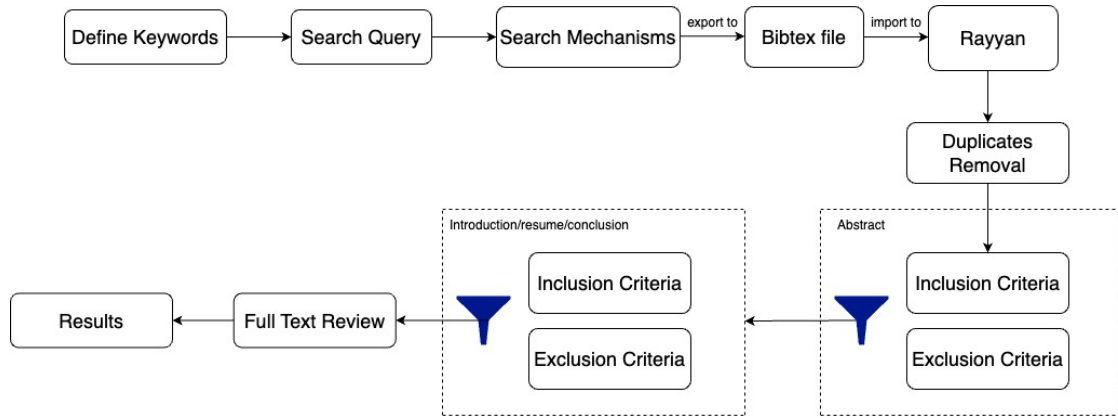


Figure 3.1: Overview of the Systematic Literature Review Process.

The review targeted peer-reviewed articles published between 2013 and 2024, with particular emphasis on works from 2019 onward, reflecting the most recent advances in Multi-Agent Reinforcement Learning (MARL) and distributed problem-solving. The databases used for retrieval included IEEE Xplore, ACM Digital Library, Scopus, Web of Science, and arXiv.

3.1.1 Keywords

To guide the search and ensure coverage of relevant areas, keywords were grouped into three thematic categories:

- **General Concepts:** “multi-agent reinforcement learning”, “emergency evacuation”, “crowd simulation”, “distributed problem solving”
- **Techniques:** “actor-critic”, “policy gradient”, “deep Q-learning”, “decentralized coordination”
- **Challenges:** “partial observability”, “scalability”, “communication overhead”, “dynamic environments”, “reward shaping”

3.1.2 Search Query and Criteria

A comprehensive Boolean search string was formulated to combine the selected terms:

("emergency evacuation" OR "crowd simulation" OR "multi-agent pathfinding") AND ("multi-agent" OR "decentralized") AND ("reinforcement learning" OR "MARL" OR "Multi-agent Reinforcement Learning") AND ("coordination" OR "partial observability" OR "dynamic environments")

The inclusion criteria required that publications be written in English, published from 2017 onward, and relevant to domains such as artificial intelligence, machine learning, or multi-agent systems, particularly in the context of distributed coordination. This time frame ensures that the review captures recent developments in MARL and distributed AI.

In contrast, exclusion criteria were applied to discard papers without full text access, duplicate entries retrieved from multiple databases, and preliminary or incomplete works such as abstracts, workshop summaries, tutorials, and invited talks. Only studies providing sufficient methodological detail and experimental validation were retained for analysis.

In total, 195 papers were retrieved. After removal of duplicates and selection by title and abstract, 24 articles were selected for full text review. Of these, 12 articles were retained for in-depth analysis based on relevance, methodological clarity, and experimental completeness.

3.2 Traditional Approaches to Distributed Problem Solving

Traditional approaches to distributed problem solving, especially in multi-agent scenarios such as emergency evacuation, have historically relied on physics-based models, rule-based heuristics, and deterministic path-planning algorithms. These methods, although foundational, are limited in their ability to handle dynamic and uncertain environments and coordinate decentralized decision-making at scale.

One of the most established techniques is the Social Force Model proposed by Helbing and Johansson [20], which simulates pedestrian dynamics based on attractive and repulsive forces between agents and their environment. This framework captures emergent crowd behaviors such as lane formation and clogging and has been widely applied in evacuation modeling due to its interpretability and physical realism. However, its fixed-force formulation makes it unsuitable for scenarios that require adaptability or strategic learning.

Agent-Based Models (ABMs) offer an alternative, where each agent follows a set of predefined rules, often calibrated from empirical data. Almeida et al. [3, 4], for example, used serious games to collect behavioral data under simulated evacuations and then used those data to inform rule-based agent decisions in crowd simulations. Despite their flexibility, ABMs typically encode behavior statically, limiting their responsiveness to unexpected conditions or evolving contexts. Moreover, as the number of agents grows, these systems often exhibit scalability challenges due to combinatorial rule interactions and coordination bottlenecks.

Classical routing algorithms such as A* and Dijkstra have played a significant role in multi-agent navigation, particularly in structured environments where global map knowledge is available. These approaches are efficient for determining shortest paths under static conditions. However, they are limited by their assumption of static environments and lack mechanisms for adaptive behaviors or real-time congestion handling. Even with extensions like Flow-Aware A* or Potential Fields, these methods remain fundamentally centralized and reactive. Recent research highlights the need for adaptive and decentralized solutions to address dynamic and congested scenarios in multi-agent systems [46].

In general, while traditional approaches have proven effective in constrained and well-understood domains, they exhibit major limitations in complex dynamic settings. These include the inability to learn or improve from experience, limited coordination mechanisms in decentralized contexts, and poor generalization to unforeseen scenarios. These gaps have motivated the growing interest

in data-driven and learning-based approaches, particularly Multi-Agent Reinforcement Learning (MARL), which promises to address these issues through autonomous adaptation and scalable coordination. In this context, algorithms such as Proximal Policy Optimization (PPO) [54], Soft Actor-Critic (SAC) [17], and the more recent Advantage Actor-Critic (A2C) [41] have emerged as robust alternatives, offering improved sample efficiency, robustness to stochasticity, and the ability to learn from partial observability, all of which are essential in real-world evacuation and distributed problem solving scenarios.

Study	Method / Strategy	Contribution or Key Insight	Remarks
Helbing and Johansson [20]	Social Force Model	Simulates pedestrian dynamics using attractive/repulsive forces; captures emergent behaviors like lane formation and clogging	Limited adaptability; fixed-force formulation
Almeida et al. [3, 4]	Agent-Based Modeling	Uses data from serious games to calibrate rule-based agent behavior in evacuation scenarios	Static rules; scalability and responsiveness issues
Panayotov et al. [46]	Adaptive decentralized planning	Highlights the limitations of classical planners under congestion; proposes adaptive coordination models	Emphasizes real-time, decentralized control

Table 3.1: Traditional approaches to distributed problem solving.

3.3 MARL-Based Approaches to Distributed Problem Solving

In recent years, Multi-Agent Reinforcement Learning (MARL) has emerged as a powerful paradigm for solving distributed problems in complex, dynamic environments. Unlike traditional approaches, MARL allows agents to autonomously learn policies through interaction with their surroundings, enabling decentralized coordination, continuous adaptation, and strategic reasoning under uncertainty. This has proven especially valuable in domains such as emergency evacuation, traffic routing, swarm robotics, and resource allocation, where coordination among agents and responsiveness to environmental changes are critical.

The appeal of MARL lies in its ability to model decentralized decision making while still benefiting from collective optimization. Early MARL models suffered from stability issues, non-stationarity, and sparse reward signals. However, advances in deep reinforcement learning algorithms have improved training convergence and policy generalization. In particular, actor-critic

methods such as Proximal Policy Optimization (PPO) [54], Soft Actor-Critic (SAC) [17], and Advantage Actor-Critic (A2C) [41] have become standard in MARL literature due to their balance between performance and robustness. PPO offers clipped surrogate objectives that prevent large, destabilizing policy updates, SAC introduces entropy regularization to promote exploration, and A2C simplifies the policy update process by computing advantage estimates using a synchronous actor-critic architecture, which improves stability and sample efficiency.

In the context of evacuation or pathfinding, MARL enables agents to learn policies that go beyond shortest-path routing, taking into account congestion, dynamic obstacles, and emergent cooperation. Xu et al. [66] demonstrated that MARL agents can learn to distribute themselves efficiently across multiple exits, minimizing overall evacuation time. Similarly, Li et al. [36] proposed a hybrid MARL model combining deep reinforcement learning with heuristic guidance to improve convergence in large-scale simulations. These studies highlight the potential of MARL to outperform traditional methods not only in final performance but also in robustness to novel scenarios.

Another important strand of work focuses on reward design and communication mechanisms. Since MARL agents often operate under partial observability and local feedback, shaping the reward function is critical to encourage collective behavior. Some models penalize proximity to congested zones or reward agents for proximity to less-used exits. In addition, explicit communication has been explored through message passing architectures or shared observation models. While these approaches can enhance coordination, they also introduce additional complexity and require careful synchronization to avoid instability during training.

Despite their promise, MARL approaches still face open challenges. These include the high sample complexity of training, the difficulty of hyperparameter tuning in non-stationary environments, and the lack of standard evaluation benchmarks across different domains. Nevertheless, the recent trajectory of research indicates growing maturity in the field, with an increasing number of studies demonstrating successful applications to real-world and large-scale simulated environments. The techniques discussed here form the methodological foundation of the empirical work developed in this thesis, which explores how MARL can be used to optimize distributed problem solving in emergency scenarios through the use of actor-critic algorithms and attention-aware coordination mechanisms.

Study	Method / Strategy	Contribution or Key Insight	Remarks
Xu et al. [66]	Deep RL (multi-exit evacuation)	Agents learn to distribute across multiple exits, reducing congestion and evacuation time	Demonstrates MARL outperforming rule-based methods
Li et al. [36]	Hybrid MARL + heuristics	Integrates heuristic guidance in MARL for faster convergence and robust coordination	Improves learning efficiency and scalability
Wang et al. [63]	Multi-task MARL	Demonstrates shared decision-making knowledge across different multi-agent coordination tasks	Improves transferability and reduces training costs

Table 3.2: MARL-based approaches to distributed problem solving.

3.4 Hybrid and Comparative Approaches

Hybrid approaches that combine Multi-Agent Reinforcement Learning (MARL) with domain-specific heuristics, classical planning methods, or auxiliary learning signals have emerged as promising alternatives to overcome the limitations of purely reinforcement learning-based models. By integrating the adaptability and learning capacity of MARL with the structure and interpretability of traditional methods, these approaches enhance both the efficiency of the learning process and the robustness of solutions in complex, dynamic environments.

A common strategy is to incorporate heuristic guidance into the training of MARL agents. Rather than learning entirely from scratch, agents are provided with prior knowledge in the form of hand-crafted policies, potential fields, or expert trajectories, which serve as reference signals during learning. For example, Li et al. [36] proposed an adaptive learning approach for cooperative multi-agent systems, integrating decentralized reinforcement learning with heuristic adjustments to focus learning on high-impact decisions. This method accelerates convergence and improves coordination in environments with sparse rewards.

Multi-task learning techniques have also been employed to mitigate the instability of early-stage exploration in multi-agent environments. In these methods, agents are initially trained in simplified scenarios and gradually transfer their policies to more complex contexts, reducing the risk of converging to suboptimal policies and improving generalization. Recent studies, such as Wang et al. [63], demonstrate that shared learning architectures can extract common coordination patterns and enhance agent performance across multiple related tasks.

Systematic comparisons between MARL algorithms have revealed important differences in their suitability for various tasks. For instance, studies like Schulman et al. [54] and Haarnoja et al. [17] show that Soft Actor-Critic (SAC) performs better in continuous action spaces due to entropy regularization, while Proximal Policy Optimization (PPO) remains a strong baseline in discrete environments thanks to stable policy updates. Advantage Actor-Critic (A2C) stands out for its computational efficiency in low-dimensional observations and dense reward settings. Therefore, the choice of algorithm should consider specific domain characteristics such as action space continuity, observability, stochasticity, and coordination requirements.

Comparative studies between MARL methods and classical approaches, such as centralized A* or rule-based decision trees, indicate that MARL often outperforms traditional methods in dynamic scenarios, particularly when adaptability and scalability are critical. However, in static and fully observable environments, classical methods can still be more efficient. This has motivated the development of hybrid architectures capable of switching between learned and heuristic behaviors according to environmental conditions, maximizing both performance and robustness.

Emerging trends also point to the integration of architectures based on graph neural networks (GNNs) and inter-agent communication protocols, which act as intermediaries between structured knowledge and learned policies. Such solutions have shown improvements in scalability, coordination, and robustness, especially in tasks involving large agent populations and highly uncertain environments [12].

In summary, the hybridization of methods, whether through heuristic guidance, curriculum learning, or advanced architectural integration, represents a promising direction for advancing distributed problem solving in multi-agent systems, offering more adaptive, interpretable, and efficient solutions.

Study	Method / Strategy	Contribution or Key Insight	Remarks
Li et al. [36]	Adaptive learning with heuristic adjustments	Accelerated convergence and improved coordination in sparse reward environments	Combines prior knowledge with MARL updates
Wang et al. [63]	Multi-task learning with shared layers	Extracts common decision-making knowledge across different coordination tasks	Enables knowledge transfer between related scenarios
Fan et al. [12]	Attention-based graph MARL	Enhances coordination by leveraging inter-agent attention mechanisms	Improves scalability and robustness in dynamic environments

Table 3.3: Hybrid and comparative approaches in multi-agent coordination.

3.5 Coordination Challenges in Multi-Agent Systems

Although Multi-Agent Reinforcement Learning (MARL) has shown promising results in solving distributed problems, particularly in dynamic environments like evacuation scenarios, several fundamental challenges related to coordination continue to hinder its broader applicability and scalability. These challenges include non-stationarity, partial observability, reward assignment complexity, communication constraints, and poor generalization to unseen environments [2].

One of the core issues in MARL is the non-stationarity of the environment as perceived by each agent. Since all agents learn simultaneously, the policy of one agent changes the environment for the others, violating the Markov assumption on which most reinforcement learning algorithms rely. This results in unstable learning dynamics and slow convergence. Strategies like Centralized Training with Decentralized Execution (CTDE) have been introduced to mitigate this problem, allowing agents to learn from a centralized critic with access to the global state during training, while maintaining decentralized policies during deployment [13, 53].

Another major challenge stems from partial observability. In real-world evacuation settings, agents often lack access to global information due to occlusions, limited sensory ranges, or environmental complexity. This leads to uncertainty in decision-making, especially in high-density environments. Techniques based on recurrent neural networks have been proposed to address this limitation [24], enabling agents to maintain an internal belief state. Nevertheless, these approaches increase computational requirements and are difficult to scale. Alternatively, communication-based methods such as attention-driven message passing [25, 29] can facilitate coordination under uncertainty, but introduce synchronization overhead and may not be robust in failure-prone environments.

The credit assignment problem also remains a key barrier to effective learning in MARL. In many cooperative tasks, including evacuation, rewards are global and delayed, making it difficult to identify which agent decisions contributed positively or negatively to the outcome. This often leads to inefficient learning or suboptimal policies. Solutions such as counterfactual baselines [13], difference rewards, and value decomposition networks [57] have been explored, but are still limited in environments with large agent populations and sparse feedback signals.

Scalability is another persistent concern. As the number of agents increases, the joint action space grows exponentially, complicating the learning process and often degrading policy performance. Methods like mean field MARL [67] and graph-based coordination [26] attempt to address this by approximating agent interactions locally or through communication topologies, yet these still rely on simplifying assumptions that may not hold in large-scale, dynamic evacuation scenarios.

Finally, generalization remains a key limitation in the deployment of MARL systems. Policies trained in specific layouts, population densities, or obstacle configurations often perform poorly when exposed to new, unseen environments. This restricts the practical utility of MARL in real-world applications where variability is the norm. Emerging approaches such as curriculum learning, domain randomization [49], and modular policy design aim to improve transferability, though

they are still underexplored in high-stakes domains like evacuation.

In summary, while MARL offers a compelling framework for decentralized decision-making and coordination, it continues to face significant challenges related to non-stationarity, observability, communication, scalability, and generalization. These limitations underline the need for hybrid approaches and more robust learning frameworks, which are explored in the following section.

Study	Method / Strategy	Contribution or Key Insight	Remarks
Foerster et al. [13]	Counterfactual baselines (CTDE)	Stabilizes multi-agent learning by isolating individual agent contributions in cooperative settings	Mitigates non-stationarity; requires global information during training
Iqbal and Sha [24]	Recurrent actor-critic (A2C-RNN)	Improves policy learning under partial observability using recurrent neural networks	Increases complexity; limited scalability
Jiang and Lu [25]	Attention-based communication	Enables agents to focus on relevant messages for better coordination	Improves cooperation; adds communication overhead
Sunehag et al. [57]	Value decomposition networks	Decomposes global rewards into individual contributions to improve credit assignment	Effective in small-scale tasks
Yang et al. [67]	Mean field MARL	Approximates many-agent interactions using local neighborhood averages	Improves scalability; assumes local homogeneity
Wang et al. [63]	Shared layer architecture	Leverages common coordination patterns across multiple tasks for improved learning	Demonstrates transferable multi-agent cognition

Table 3.4: Representative strategies addressing coordination challenges in MARL.

3.6 Summary and Research Gaps

This chapter presented a comprehensive review of distributed problem-solving strategies in multi-agent systems, with a particular focus on emergency evacuation contexts. The discussion was structured into four main parts: traditional methods, learning-based approaches using Multi-Agent

Reinforcement Learning (MARL), core coordination challenges in multi-agent settings, and recent hybrid or comparative strategies.

Traditional approaches such as the Social Force Model, rule-based Agent-Based Models (ABMs), and classical routing algorithms like A* and Dijkstra have laid the foundation for evacuation modeling. These methods, while interpretable and computationally efficient, are inherently limited in their adaptability to dynamic, decentralized, and uncertain environments.

MARL has emerged as a promising alternative, enabling agents to learn decentralized policies through interaction with their environment. Actor-critic methods such as PPO, SAC, and A2C have demonstrated strong performance under partial observability and complex coordination constraints. However, MARL introduces its own set of challenges, notably non-stationarity, sparse rewards, and sensitivity to hyperparameters.

To address these issues, recent research has focused on the specific coordination challenges that arise in MARL settings, particularly in high-density and dynamic environments. These include scalability with increasing agent populations, the credit assignment problem, and the need for effective decision-making under partial observability and limited communication. Several studies have explored targeted mechanisms such as counterfactual reasoning, recurrent architectures, attention-based communication, and mean field approximations to mitigate these barriers.

Hybrid strategies further extend MARL by integrating heuristic rules, curriculum learning, or graph-based attention mechanisms. These combinations seek to improve learning efficiency, stability, and generalization, particularly in scenarios where full learning from scratch is impractical or too costly.

Despite this progress, to the best of our knowledge, several research gaps persist:

- **Scalability to large populations:** MARL frameworks often degrade with increasing agent counts due to coordination overhead and joint action space explosion.
- **Generalization to unseen environments:** Many models struggle to transfer learned policies to novel spatial layouts, obstacle configurations, or density distributions, limiting their applicability in real-world deployments.
- **Robust coordination under limited communication:** While some approaches use explicit messaging, there is a need for lightweight, implicit coordination mechanisms that are robust to communication failure or bandwidth constraints.
- **Reward design under uncertainty:** Shaping appropriate reward functions remains difficult in dynamic environments where global objectives must emerge from local observations and actions.
- **Benchmarking and reproducibility:** A lack of standardized simulation environments and performance metrics continues to hinder comparative evaluation across methods.

These challenges motivate the technical contributions of this thesis, which explore how actor-critic MARL algorithms, attention-based coordination, and hybrid learning strategies can be leveraged to optimize decentralized evacuation strategies under realistic conditions of uncertainty and partial observability.

3.7 Conclusions

This chapter provided a structured literature review on distributed problem-solving approaches in multi-agent systems, with a focus on emergency evacuation. The analysis covered classical methods, recent advances in Multi-Agent Reinforcement Learning (MARL), coordination challenges, and hybrid techniques, laying the theoretical foundation for this dissertation. The following considerations directly address the research questions guiding this work:

RQ1 - How can Multi-Agent Reinforcement Learning (MARL) be applied to develop effective decentralized evacuation strategies in dynamic, partially observable environments?

Section 2.2 demonstrates that MARL is particularly suited for environments where agents must learn to act based on local observations while contributing to a collective objective. Actor-critic methods such as PPO [54] and SAC [17] are recurrently used to handle the stochasticity and partial observability inherent to evacuation contexts. Moreover, the reviewed works (e.g., [66]) confirm that MARL agents can learn to coordinate evacuations without requiring centralized control. The empirical gains over rule-based systems, particularly in dynamic layouts with congestion, support MARL as a viable paradigm for decentralized evacuation.

RQ2 - Which MARL algorithms and reward shaping techniques most effectively minimize evacuation time and congestion in multi-agent evacuation scenarios?

As explored in Sections 2.2 and 3.5, actor-critic algorithms (notably PPO, SAC, and A2C) balance sample efficiency with policy robustness. Table 3.2 summarizes successful applications of these methods to crowd dynamics, while Section 3.5 highlights how techniques such as entropy regularization [17], counterfactual reasoning [13], and multi-agent specialization approaches can shape more effective behaviors. Reward shaping, like penalizing congestion or rewarding dispersion across exits, emerges as a key tool for improving convergence and reducing evacuation time, particularly under sparse feedback.

RQ3 - How do MARL-based evacuation strategies compare to traditional heuristic or rule-based methods in terms of adaptability, scalability, and coordination efficiency?

Section 3.2 identifies the limitations of classical approaches like Social Force Models [20] and A* planning, which struggle with adaptability and real-time coordination under uncertainty. In contrast, MARL approaches reviewed in Section 2.2 and summarized in Table 3.2 consistently outperform these baselines in complex, dynamic scenarios. Nonetheless, as detailed in Section 3.6, MARL faces scalability issues as agent count increases, and generalization across layouts remains

a challenge. Hybrid approaches (Section 3.4) and communication-aware models (Table 3.4) show promise in mitigating these limitations, suggesting a trade-off between interpretability and flexibility.

In summary, the literature review supports the thesis hypothesis that MARL, particularly actor-critic variants enhanced through reward shaping and coordination mechanisms, is a promising framework for developing adaptive and decentralized evacuation strategies. The identified gaps, including those related to scalability, generalization, and robust coordination, directly inform the design choices explored in the following chapters.

Chapter 4

Methodological Framework

This chapter presents the methodological framework for developing and evaluating emergency evacuation policies based on Multi-Agent Reinforcement Learning. The methodological approach is structured into seven main sections that establish the theoretical and practical foundations of the research. Section 4.1 presents the problem statement, contextualizing coordination challenges in emergency evacuation scenarios and justifying the adoption of MARL methods over classical approaches. Section 4.2 formally defines the problem as a Dec-POMDP, including the specification of state, action, and observation spaces, as well as the justification for selecting algorithms suitable for discrete action spaces. Section 4.3 describes the custom simulation environment developed on the PettingZoo framework, detailing environment dynamics and test scenarios. Section 4.4 presents the solution architecture based on the CTDE (Centralized Training with Decentralized Execution) paradigm, including neural network specifications and the training pipeline. Section 4.5 analyzes the limitations of classical models and provides theoretical justification for the need of MARL approaches in evacuation problems. Section 4.6 identifies and discusses the main limitations and constraints of the proposed approach.

4.1 Problem Statement

Emergency evacuation represents one of the most challenging coordination problems in safety-critical systems, where multiple autonomous agents must navigate complex environments under time pressure while avoiding collisions and congestion [47, 72]. These scenarios are characterized by inherent uncertainty, limited individual perception, and the need for rapid collective decision-making without centralized control [45]. Traditional evacuation modeling has predominantly relied on predetermined behavioral rules and static optimization techniques that fail to capture the adaptive and stochastic nature of real-world emergency situations.

By conducting a thorough analysis of existing evacuation modeling approaches, we identified that classical methods such as social force models and agent-based model-based rules are insufficient to handle dynamic coordination challenges because they cannot learn from experience or adapt their strategies based on environmental feedback [20, 66]. This fundamentally limits their

effectiveness in scenarios where optimal evacuation strategies must emerge through agent interaction rather than being predefined by human designers.

The growing interest in Multi-Agent Reinforcement Learning for emergency management has highlighted the potential for agents to develop sophisticated coordination behaviors through trial-and-error learning [36]. Recent advances in actor-critic algorithms, particularly PPO, SAC, and A2C, have demonstrated superior performance in continuous control tasks and partial observability settings that are characteristic of evacuation scenarios [17, 54]. However, the application of these techniques to evacuation modeling remains underexplored, with most existing studies focusing on simplified grid environments rather than realistic continuous spaces with complex physical constraints [72].

Moreover, current research lacks comprehensive evaluation frameworks that systematically compare different MARL algorithms under varying evacuation conditions, including different agent densities, exit configurations, and obstacle layouts. This gap prevents practitioners from understanding which coordination mechanisms are most effective for specific evacuation scenarios and limits the practical deployment of learning-based evacuation systems.

4.1.1 Scope

This dissertation focuses specifically on cooperative multi-agent reinforcement learning algorithms applied to continuous 2D evacuation environments, acknowledging that real-world implementation involves additional complexities beyond simulation scope. Within the emergency evacuation domain, we investigate how Centralized Training with Decentralized Execution paradigms can enable effective coordination without requiring communication infrastructure during deployment [13]. The research concentrates on developing adaptive evacuation policies that can generalize across different spatial layouts and agent populations while maintaining computational efficiency suitable for real-time applications. Extensions to multi-floor buildings, heterogeneous agent capabilities, and integration with existing emergency response systems are considered beyond the current scope but represent important directions for future work.

4.2 Problem Formalization

In this section, the main problem addressed by this dissertation is formally introduced. Firstly, it is approached from an application domain perspective concerning the emergency evacuation problem and its considered features and characteristics. In the second subsection, the evacuation problem is formalized as a dynamic sequential decision-making problem, and its characteristics are presented in the form of its corresponding Dec-POMDP formulation.

4.2.1 Emergency Evacuation as a Coordination Problem

Emergency evacuation scenarios represent complex coordination challenges where multiple autonomous agents must navigate from initial positions to designated safe zones while avoiding

obstacles, minimizing congestion, and optimizing collective evacuation time. Unlike traditional path planning problems, evacuation scenarios are characterized by several critical properties that make them particularly suitable for Multi-Agent Reinforcement Learning approaches.

The primary objective of the evacuation problem is to find optimal decentralized policies for all agents that minimize the total evacuation time while conforming to a particular set of constraints, such as avoiding collisions, respecting exit capacity limits, and maintaining safe navigation behaviors. The evacuation process must account for dynamic interactions between agents, partial observability of the environment, and the emergence of bottlenecks at exit points.

In the following subsections, the formal mathematical framework and constraints considered for the evacuation problem are introduced and discussed in detail.

4.2.2 Decentralized Partially Observable Markov Decision Process (Dec-POMDP)

The emergency evacuation scenario is formalized as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP), which extends the classical MDP framework to multi-agent settings under partial observability. The Dec-POMDP tuple is defined as:

$$\mathcal{M} = \langle \mathcal{I}, \mathcal{S}, \{\mathcal{A}_i\}_{i \in \mathcal{I}}, \{\mathcal{O}_i\}_{i \in \mathcal{I}}, \mathcal{T}, \mathcal{R}, \{\Omega_i\}_{i \in \mathcal{I}}, \gamma, h \rangle \quad (4.1)$$

where each component is formally defined as follows:

- $\mathcal{I} = \{1, 2, \dots, N\}$ is the finite set of agent indices
- $\mathcal{S}$ is the finite set of global environment states
- $\mathcal{A}_i$ is the discrete action space for agent i : $\mathcal{A}_i = \{0, 1, 2, 3, 4\}$
- $\mathcal{O}_i$ is the local observation space for agent i
- $\mathcal{T} : \mathcal{S} \times \mathcal{A}_1 \times \dots \times \mathcal{A}_N \rightarrow \Delta(\mathcal{S})$ is the state transition function
- $\mathcal{R} : \mathcal{S} \times \mathcal{A}_1 \times \dots \times \mathcal{A}_N \rightarrow \mathbb{R}$ is the global reward function
- $\Omega_i : \mathcal{S} \rightarrow \Delta(\mathcal{O}_i)$ is the observation function for agent i
- $\gamma \in [0, 1]$ is the discount factor
- h is the finite planning horizon

4.2.2.1 State Space

The global state $s_t \in \mathcal{S}$ at time step t encodes the complete configuration of the evacuation environment, including agent positions, velocities, and obstacle configurations. Formally, the state is represented as:

$$s_t = \{(\mathbf{p}_{i,t}, \mathbf{v}_{i,t})_{i=1}^N, \mathbf{O}, \mathbf{L}\} \quad (4.2)$$

where:

- $\mathbf{p}_{i,t} = (x_{i,t}, y_{i,t}) \in \mathbb{R}^2$ is the position of agent i at time t
- $\mathbf{v}_{i,t} = (v_{x,i,t}, v_{y,i,t}) \in \mathbb{R}^2$ is the velocity of agent i at time t
- $\mathbf{O} = \{\mathbf{o}_k\}_{k=1}^{N_o}$ represents static obstacle positions
- $\mathbf{L} = \{\mathbf{l}_m\}_{m=1}^{N_L}$ represents meeting point positions for post-evacuation navigation

4.2.2.2 Action Space and Algorithm Selection

Each agent i operates in a discrete action space representing directional movement commands. The action space is defined as:

$$\mathcal{A}_i = \{0, 1, 2, 3, 4\} \quad (4.3)$$

where each action corresponds to:

- 0: Wait (no movement)
- 1: Move north
- 2: Move south
- 3: Move east
- 4: Move west

The action $a_{i,t} \in \mathcal{A}_i$ at time step t represents the movement direction selected by agent i . The physics engine maps these discrete actions to acceleration vectors according to a predefined mapping function and integrates them to update agent velocities and positions:

$$\mathbf{v}_{i,t+1} = \mathbf{v}_{i,t} + \Delta t \cdot \text{map_action}(a_{i,t}) \quad (4.4)$$

$$\mathbf{p}_{i,t+1} = \mathbf{p}_{i,t} + \Delta t \cdot \mathbf{v}_{i,t+1} \quad (4.5)$$

where $\text{map_action}(a)$ converts discrete actions to unit direction vectors:

$$\text{map_action}(a) = \begin{cases} (0.0, 0.0) & \text{if } a = 0 \text{ (stay)} \\ (0.0, 1.0) & \text{if } a = 1 \text{ (north)} \\ (0.0, -1.0) & \text{if } a = 2 \text{ (south)} \\ (1.0, 0.0) & \text{if } a = 3 \text{ (east)} \\ (-1.0, 0.0) & \text{if } a = 4 \text{ (west)} \end{cases} \quad (4.6)$$

The system remains subject to maximum velocity constraints $\|\mathbf{v}_{i,t}\| \leq v_{\max}$ and collision handling mechanisms.

The choice of discrete action space aligns with the discrete nature of human decision-making in emergency situations, where individuals typically choose between a limited set of directional options rather than fine-grained continuous movements. This design choice also ensures compatibility with the selected MARL algorithms (DQN, PPO, A2C), which are specifically designed for discrete action environments.

4.2.2.3 Observation Space

Agent i receives partial observations $o_{i,t} \in \mathcal{O}_i$ that include local information about its state and nearby environment entities. The observation vector is structured as:

$$o_{i,t} = [\mathbf{p}_{i,t}, \mathbf{v}_{i,t}, \mathbf{r}_{i,t}^{\text{agents}}, \mathbf{r}_{i,t}^{\text{obstacles}}, \mathbf{r}_{i,t}^{\text{meeting point}}] \quad (4.7)$$

where:

- $\mathbf{r}_{i,t}^{\text{agents}}$ contains relative positions and velocities of other agents
- $\mathbf{r}_{i,t}^{\text{obstacles}}$ contains relative positions of static obstacles
- $\mathbf{r}_{i,t}^{\text{meeting points}}$ contains relative positions of post-evacuation meeting points

The choice of discrete action space has direct implications for algorithm selection in MARL. While continuous control methods such as Soft Actor-Critic (SAC) offer theoretical advantages in terms of policy expressiveness, they are not directly compatible with discrete action environments without significant modifications [10, 60].

In evacuation scenarios, discrete actions (e.g., move up, down, left, right, or wait) provide a more interpretable and computationally efficient representation of agent movement decisions. This design choice aligns with the discrete nature of human decision-making in emergency situations, where individuals typically choose between a limited set of directional options rather than fine-grained continuous movements [47].

Consequently, this work employs algorithms specifically designed for discrete action spaces, namely Deep Q-Network (DQN), Proximal Policy Optimization (PPO), and Advantage Actor-Critic (A2C), which have demonstrated superior performance and stability in discrete multi-agent environments [16].

4.2.2.4 Reward Function

The global reward function is designed to encourage agents to efficiently reach the meeting point (landmark) while penalizing undesirable behaviors such as collisions, remaining stationary, or leaving the environment boundaries. At each time step t , the reward is computed as:

$$r(t) = \sum_{i=1}^N \left[-\alpha - r_i^{\text{collision-obstacle}}(t) - r_i^{\text{collision-agent}}(t) - r_i^{\text{stationary}}(t) - r_i^{\text{boundary}}(t) + r_i^{\text{landmark}}(t) + r_i^{\text{scenario}}(t) \right] \quad (4.8)$$

The individual reward components are defined as follows:

$$r_i^{\text{collision-obstacle}}(t) = \begin{cases} P_{\text{obstacle}} & \text{if agent } i \text{ collides with an obstacle or wall at } t \\ 0 & \text{otherwise} \end{cases} \quad (4.9)$$

$$r_i^{\text{collision-agent}}(t) = \begin{cases} P_{\text{agent}} & \text{if agent } i \text{ collides with another agent at } t \\ 0 & \text{otherwise} \end{cases} \quad (4.10)$$

$$r_i^{\text{stationary}}(t) = \begin{cases} P_{\text{stationary}} & \text{if agent } i \text{ moves less than a threshold at } t \\ 0 & \text{otherwise} \end{cases} \quad (4.11)$$

$$r_i^{\text{boundary}}(t) = \sum_{p=1}^2 \text{bound}(|p_{i,t}^{(p)}|) \quad (4.12)$$

$$r_i^{\text{landmark}}(t) = \begin{cases} R_{\text{landmark}} & \text{if agent } i \text{ reaches the meeting point } (\|\mathbf{p}_{i,t} - \mathbf{l}\| < 0.1) \\ 0 & \text{otherwise} \end{cases} \quad (4.13)$$

$$r_i^{\text{scenario}}(t) = \text{scenario-specific reward (see below)} \quad (4.14)$$

where the boundary penalty function is

$$\text{bound}(x) = \begin{cases} 0 & x < 0.9 \\ 10(x - 0.9) & 0.9 \leq x < 1.0 \\ \min(\exp(2x - 2), 10) & x \geq 1.0 \end{cases} \quad (4.15)$$

- $P_{\text{obstacle}} = 2.0$ (collision with obstacle or wall)
- $P_{\text{agent}} = 0.01$ (collision with another agent)
- $P_{\text{stationary}} = 0.05$ (if $\|\Delta \mathbf{p}_{i,t}\| < 0.01$)
- $\alpha = 0.01$ (small time-step penalty)
- R_{landmark} is a positive value (scenario-dependent)

Scenario-specific rewards:

- *Scenarios 1 & 2:* Agents receive a positive incremental reward proportional to the reduction in distance to the meeting point:

$$r_i^{\text{scenario}}(t) = 5.0 \times (\text{prev_dist} - \text{curr_dist})$$

- *Scenario 4 (Communication)*: Agents receive a large reward (+50) for discovering the landmark, and further incremental rewards for reducing the distance to the average communicated position.

4.2.2.5 Episode Termination and Constraints

Episodes terminate under the following conditions:

- **Success Termination**: All agents reach their designated meeting points
- **Time Limit**: Maximum episode length $T = 150$ steps is reached
- **Failure Termination**: System reaches an infeasible state (optional)

The environment enforces several constraints to maintain realistic evacuation dynamics:

$$\text{Velocity Bound: } \|\mathbf{v}_{i,t}\| \leq v_{\max} \quad \forall i, t \quad (4.16)$$

$$\text{Collision Avoidance: } \|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\| \geq d_{\min} \quad \forall i \neq j \quad (4.17)$$

4.2.2.6 Policy and Objective

Each agent i learns a stochastic policy $\pi_i : \mathcal{O}_i \rightarrow \Delta(\mathcal{A}_i)$ that maps local observations to action distributions. The joint policy is denoted as $\pi = \{\pi_1, \dots, \pi_N\}$.

The objective is to find the optimal joint policy that maximizes the expected cumulative discounted reward:

$$J(\pi) = \mathbb{E}_{\pi} \left[\sum_{t=0}^h \gamma^t r(t) \right] \quad (4.18)$$

This formulation provides the mathematical foundation for applying Centralized Training with Decentralized Execution (CTDE) algorithms to learn coordinated evacuation policies under partial observability constraints.

4.3 Simulation Environment

This section describes the simulation environment developed for training and evaluating Multi-Agent Reinforcement Learning (MARL) evacuation policies. The environment is based on a customized extension of the Multi-Agent Particle Environment (MPE), accessed through the Petting-Zoo API [61], and adapted to model evacuation dynamics in a 2D continuous space. This simulation environment supports heterogeneous agent interactions, decentralized observations, dynamic reward functions, and congestion-sensitive exit behavior [40].

4.3.1 Framework: PettingZoo and MPE

The simulation environment is built using the `PettingZoo` library, a Python-based framework for standardized multi-agent reinforcement learning environments [61]. `PettingZoo` provides a consistent agent-environment interface, supporting both sequential (AEC) and parallel environments, and integrates seamlessly with `Stable-Baselines3`[52], `RLlib`[37, 38], and other MARL libraries.

At its core, the environment uses the Multi-Agent Particle Environment (MPE), a lightweight 2D physics simulator designed for rapid prototyping of MARL algorithms[40]. MPE supports continuous control, collision detection, meeting point dynamics, and customizable observation/action spaces. Its modular structure enables precise control over simulation logic, making it well-suited for extending into more domain-specific environments such as evacuation modeling.

4.3.2 Custom Multi-Agent Evacuation Environment

To study emergent coordination in evacuation scenarios, we developed a custom environment named `simple_evacuation`, implemented as a subclass of the MPE base environment. This environment models a bounded 2D evacuation area populated with agents, static obstacles, and optional post-evacuation meeting points. Each agent operates independently, observing only local information, and is tasked with exiting the domain in the shortest possible time while avoiding collisions [73].

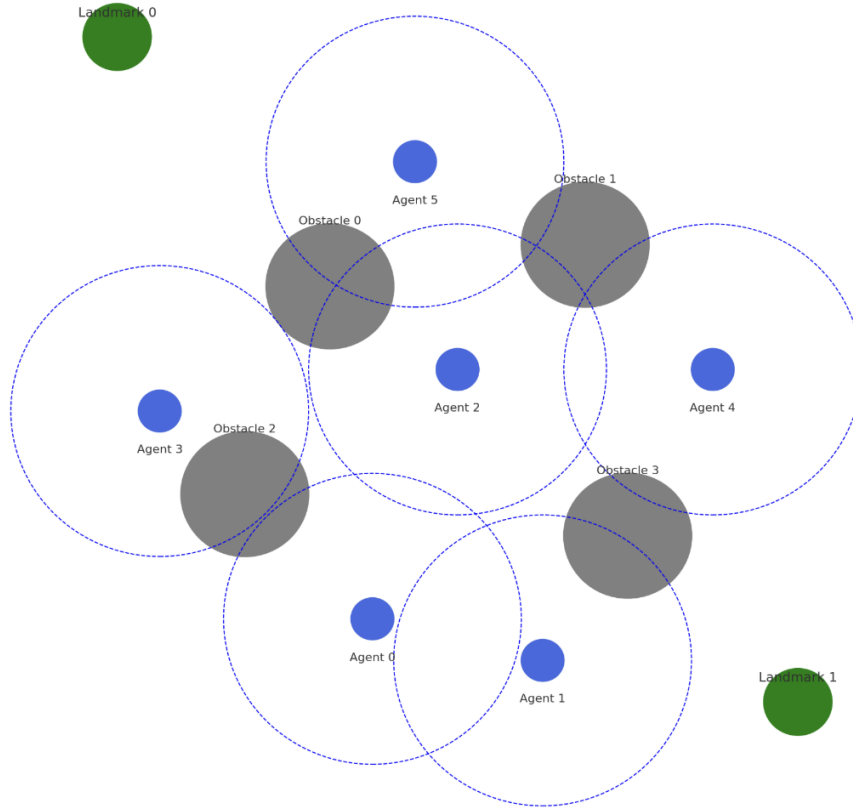


Figure 4.1: Architecture of the custom evacuation environment. Agents interact with a continuous 2D space populated with obstacles and meeting points. Each agent perceives the world locally and must evacuate efficiently under congestion constraints.

Key features of the `simple_evacuation` environment include:

- **Decentralized Observations:** Each agent perceives its position, velocity, and relative distances to nearby obstacles, meeting points, and other agents. Observations are partial and noisy, reflecting realistic constraints of limited sensing [45].
- **Discrete Actions:** Agents act through discrete directional commands (north, south, east, west, wait) that are mapped to acceleration vectors, subject to maximum velocity limits and collision penalties.

Although the experimental configuration utilizes small-scale environments, such as rooms or offices with a limited number of agents, due to hardware limitations and computational constraints, the architecture of both the environment and the underlying algorithms was intentionally designed with scalability considerations in mind. The framework accommodates increases in agent and obstacles count, and the results suggest potential scalability to larger spaces with higher population

densities, as the trained policies demonstrate generalization capabilities even when trained with fewer agents (as further discussed in Chapter 5). However, it is important to note that scalability evaluation would also be subject to hardware and computational power limitations. Consequently, while the results presented here validate the effectiveness of the model in constrained contexts, they indicate promising potential for adaptability to more complex and realistic scenarios, pending further validation in larger-scale environments.

4.3.3 Environment Dynamics and Agent Observations

Each episode begins with agents positioned according to scenario-specific configurations within a square domain of size $[-1.0, +1.0]^2$. The positioning follows predefined patterns for each scenario rather than uniform distribution. For example, in scenario 1, agents are positioned randomly, while in scenario 2, agents are placed within a specific corridor area with x-range $(-0.3, +0.9)$ and y-range $(-0.9, +0.9)$.

Agents receive a continuous observation vector $o_i(t)$ at each time step t with a **fixed size of 52 elements**, which includes:

- Agent i 's own velocity $(v_{x,i}, v_{y,i})$ and position (x_i, y_i) (4 elements)
- Relative position of the main meeting points $(x_j - x_i, y_j - y_i)$ when visible (2 elements)
- Relative positions of up to 10 nearby obstacles within distance 0.5 (20 elements, padded with zeros)
- Relative positions and velocities of up to 5 other agents (20 elements, padded with zeros)
- Communication data when enabled (2 elements, padded with zeros if communication is disabled)

The agent's action $a_i(t)$ is a discrete value representing directional movement commands. The physics engine maps these discrete actions to acceleration vectors, then integrates these to update positions with damping coefficient 0.2, time step 0.1, maximum acceleration 3.0, and maximum speed 1.0.

The reward function at time step t is defined as:

$$r_i(t) = -0.01 + r_i^{\text{goal}}(t) + r_i^{\text{movement}}(t) - r_i^{\text{collision}}(t) - r_i^{\text{boundary}}(t) \quad (4.19)$$

Where:

- -0.01 is a time penalty encouraging faster completion
- $r_i^{\text{goal}}(t)$ includes rewards for reaching meeting points (up to 100.0) and scenario-specific objectives
- $r_i^{\text{movement}}(t)$ provides positive rewards for approaching targets and penalties for staying stationary (-0.05 if movement < 0.01)

- $r_i^{\text{collision}}(t)$ penalizes collisions with obstacles/walls (-3.0) and other agents (-0.01)
- $r_i^{\text{boundary}}(t)$ penalizes agents for approaching or exceeding domain boundaries using an exponential penalty function

Episodes terminate after a fixed time horizon of 150 steps (configurable via `max_cycles` parameter). The environment does not automatically terminate when agents reach goals, but agents who reach meeting points have their velocities reduced and are marked as having achieved their goal.

4.4 Solution Architecture

To effectively address the decentralized nature of the evacuation task while leveraging centralized training information during simulation, the solution architecture follows the *Centralized Training with Decentralized Execution* (CTDE) paradigm[14]. Under CTDE, agents are trained with access to global state information (e.g., full positions of other agents and obstacles), but execute learned policies using only local observations at test time. This hybrid strategy enables agents to develop robust and coordinated behaviors while remaining scalable and fully autonomous at deployment.

4.4.1 CTDE for Multi-Agent Evacuation

The choice of CTDE is motivated by the need to balance learning stability with decentralization[32]. During training, a centralized critic benefits from access to the full joint state-action configuration, allowing for more accurate gradient estimation and alleviating issues caused by environment non-stationarity[14, 59]. Meanwhile, each agent maintains its own actor network, which is conditioned only on local observations.

This setup promotes scalability: policies generalize across different numbers of agents and exit configurations, and can be deployed independently without requiring centralized coordination or communication[53]. CTDE also mitigates the credit assignment problem, by allowing the critic to reason about the joint behavior, and enables more stable optimization in cooperative settings[57, 64].

4.4.2 Multi-Agent Learning Algorithms

The selection of reinforcement learning algorithms in this work was informed by the discrete nature of the action space and the cooperative multi-agent setting of the evacuation problem. Three algorithms were ultimately chosen for comparative evaluation, each offering distinct methodological strengths aligned with the requirements of the task.

Deep Q-Network (DQN) was adopted as the baseline value-based method, owing to its proven effectiveness in discrete environments. Its integration of experience replay and target networks

supports more stable learning dynamics, which is particularly valuable in non-stationary multi-agent settings[42]. In parallel, Proximal Policy Optimization (PPO) was selected due to its reliable performance in multi-agent domains and its policy gradient formulation, which inherently accommodates stochastic policies, an important characteristic for fostering emergent coordination among agents[54]. To complement these approaches, Advantage Actor-Critic (A2C) was included as a representative actor-critic method[41]. Its hybrid structure balances value-based and policy-based learning and offers a relatively lightweight computational footprint, making it suitable for multi-agent scenarios that require efficient training.

Initially, Soft Actor-Critic (SAC) was also considered, particularly due to its entropy-regularized framework and sample efficiency, both advantageous in high-dimensional problems[17]. However, SAC’s standard implementations, such as those available in Stable-Baselines3, are tailored for continuous action spaces. Although discrete-action variants like Soft Actor-Critic for Discrete Actions (SAD-SAC) have been proposed, they remain largely experimental and lack the empirical maturity of the selected algorithms. Furthermore, given the domain-specific need for interpretable and clearly defined discrete actions in evacuation planning, these limitations weighed heavily in the final selection.

This choice is further substantiated by recent empirical studies, which position DQN, PPO, and A2C as robust and widely accepted baselines for multi-agent learning in discrete control environments.

4.4.3 Neural Network Architecture

Each agent is parameterized by a deep actor network $\pi_{\theta_i}(a_i | o_i)$ and is trained using gradients provided by a centralized critic $Q_{\phi}(s, \mathbf{a})$. The actor receives as input the local observation o_i (as defined in Section 4.3.3) and outputs a mean vector μ_i for a multivariate Gaussian policy, from which actions are sampled. An entropy term is added to the reward objective (as in Soft Actor-Critic) to encourage exploration[17].

The critic takes as input the full global state s concatenated with the joint action vector $\mathbf{a} = (a_1, \dots, a_n)$ and predicts the expected cumulative reward. Both networks consist of two hidden layers of 128 and 64 units, respectively, with ReLU activations. Gradient updates are computed via backpropagation through the actor-critic pipeline.

4.4.4 Technologies and Dependencies

The implementation leverages Python as the primary programming language due to its extensive ecosystem for machine learning and multi-agent systems. Table 4.1 presents the main technological requirements and their respective versions.

Table 4.1: Project technological requirements and versions.

Library	Version	Purpose
Python	3.11.9	Core programming language
PyGame	2.6.1	Python Game Development
PettingZoo	1.25.0	Multi-agent environment API
Stable-Baselines3	2.6.0	MARL algorithm implementations
SuperSuit	3.10.0	Wrappers for Gymnasium and PettingZoo
PyTorch	2.7.1	Deep learning framework
NumPy	2.3.1	Numerical computing
Gymnasium	1.1.1	RL environment interface
Matplotlib	3.10.3	Visualization and plotting
Pandas	2.3.0	Data manipulation and analysis

4.5 Limitations of Classical Models and the Case for MARL

Emergency evacuation is a dynamic, decentralized, and partially observable process in which multiple autonomous agents must reason under uncertainty to navigate toward safety. Traditionally, evacuation modeling has relied on frameworks such as Cellular Automata (CA), Social Force Models (SFM), and rule-based Agent-Based Models (ABMs). These approaches have been widely adopted due to their simplicity, interpretability, and computational efficiency[21]. However, their reliance on fixed behavioral heuristics and global assumptions imposes severe limitations when applied to complex, real-world scenarios.

One major limitation lies in their lack of adaptability[18]. Agents in classical models typically follow deterministic, pre-scripted behaviors that do not evolve in response to environmental feedback. This rigidity is problematic in dynamic settings where crowd density, exit accessibility, or obstacle layout may shift unpredictably. For example, if an exit becomes temporarily congested or blocked, agents governed by predefined rules may continue to attempt to reach it, leading to delays, bottlenecks, or even system-wide failure. Adjusting such models to new scenarios requires manual parameter tuning or redesigning rule sets, tasks that are labor-intensive and prone to human bias [48].

Furthermore, most classical approaches assume complete or near-complete observability and often rely on global coordination mechanisms [47]. In practice, individuals in evacuation scenarios operate with local, limited, and often noisy information. They make decisions based on personal observations, partial views of obstacles, and estimates of crowd dynamics. Classical models do

not adequately capture these decentralization and information asymmetry constraints, leading to unrealistic predictions or oversimplified representations of real human behavior[39].

In contrast, Multi-Agent Reinforcement Learning (MARL) offers a compelling alternative grounded in learning from experience, stochastic policies, and distributed decision-making[71]. MARL agents do not require handcrafted rules. Instead, they learn behavior through direct interaction with the environment, guided by carefully designed reward functions. Over time, they develop decentralized strategies that adapt to dynamic environments and react to unforeseen configurations. Their policies generalize across scenarios, agent counts, and layout variations, which makes them significantly more robust and scalable than traditional models[1].

One particularly advantageous aspect of MARL is the emergence of coordination without explicit communication[15]. Through reward shaping and repeated trial-and-error, agents begin to anticipate the actions of others, adjusting their paths to avoid collisions. This emergent behavior is difficult to engineer manually in rule-based models but arises naturally in MARL systems trained under collective objectives.

Moreover, MARL supports continuous control and operates directly in realistic 2D physics-based environments, enabling smooth and reactive motion policies. This contrasts with the often grid-constrained or discretized assumptions of CA and ABM models. The use of CTDE (Centralized Training with Decentralized Execution), discussed in Section 4.4, further enhances MARL's suitability by allowing access to full environmental state during training while preserving local autonomy during deployment[24].

From a scientific standpoint, MARL aligns with current trends in machine learning, robotics, and collective AI systems, where adaptability, decentralization, and data-driven policy synthesis are increasingly prioritized[7]. In the context of evacuation modeling, MARL provides a means to synthesize intelligent behavior that is not only reactive but strategic, responsive to emergent bottlenecks, and optimized over time for throughput and safety.

In sum, while classical models have laid a foundational understanding of evacuation dynamics, they fall short in contexts that demand scalability, adaptability, and robustness[9]. MARL bridges these gaps by offering a flexible, learning-based framework that captures the stochastic, distributed, and interactive nature of real-world evacuations. Its application in this work is both theoretically justified and practically motivated by the limitations of existing methods and the complexity of the problem domain.

4.6 Constraints and Limitations

While the proposed approach offers a flexible and data-driven framework for modeling decentralized evacuation, it is not without limitations. Understanding the constraints under which the system operates is essential both for interpreting results and for guiding future extensions.

The first and most significant limitation arises from the simulation environment itself. The `simple_evacuation` environment, developed atop the PettingZoo MPE framework, offers a clean and efficient platform for prototyping multi-agent coordination tasks. However, it remains

a simplified abstraction of real-world evacuation scenarios[47]. The simulation assumes a two-dimensional, continuous plane with idealized agent dynamics governed by basic Newtonian motion. Agents are modeled as discs with uniform size, mass, and speed constraints, and they move frictionlessly within the environment. These simplifications abstract away many physical phenomena that are critical in real evacuation dynamics, such as inter-agent pressure, psychological stress, crowd panic, or variable mobility across different individuals[55].

Furthermore, the environment does not account for complex architectural features often present in real buildings, such as multiple floors, staircases, or variable exit widths, and simplifies obstacle interactions to static, non-deformable barriers. As a result, while the learned policies exhibit coordination and adaptability within the simulation, their direct applicability to physical deployment or real-world crowd modeling must be interpreted with caution[47].

Another constraint lies in the observation model. Each agent receives partial, localized observations that include the positions of nearby agents, obstacles, and meeting points. While this is a realistic approximation of individual perception, it is still idealized. In practice, human perception is affected by occlusion, delayed information processing, and limited reaction time [19]. Agents in the current model respond instantly to environment updates, without modeling such latency or sensory uncertainty.

From the learning perspective, the MARL algorithms employed, based on actor-critic methods and CTDE, require extensive training time and numerous environment interactions to converge toward efficient policies[22]. Although simulation provides the opportunity to scale training without real-world risk, the computational cost is non-negligible. Additionally, hyperparameter tuning and stability remain challenges, particularly in cooperative multi-agent settings where the reward signal is sparse and policy updates are interdependent[28].

The reward function design, while carefully shaped to encourage evacuation and discourage congestion or collision, is still handcrafted[5]. It reflects a set of human-defined priorities (e.g., evacuation speed, path efficiency, collision avoidance), which may not fully capture more complex or ethically nuanced evacuation criteria.

Finally, this work abstracts away from the cognitive and emotional aspects of crowd behavior. Factors such as panic propagation, herding effects, or role-based behavior differentiation (e.g., leaders versus followers) are not explicitly modeled [55]. While MARL provides a powerful framework for emergent coordination, its capacity to model sociopsychological dynamics remains limited unless such features are explicitly incorporated into the observation space, reward function or agent architecture.

Despite these limitations, the simulation setup remains well-suited for the study of core coordination mechanisms in decentralized evacuation [47]. It provides a controlled environment to evaluate MARL strategies under varying layout complexity, agent density, and obstacle configurations. Nonetheless, further work is required to bridge the gap between simulation and reality, including transfer learning to more realistic simulators or hybrid models that integrate empirical behavioral data with reinforcement learning components.

4.7 Summary

This chapter established a comprehensive methodological framework for applying Multi-Agent Reinforcement Learning to emergency evacuation scenarios. The framework formalizes the problem as a Dec-POMDP, designs a discrete action space aligned with human decision making, and develops a custom simulation environment based on PettingZoo. The solution architecture employs Centralized Training with Decentralized Execution (CTDE) to balance learning stability with deployment autonomy.

The methodology addresses key limitations of classical evacuation models by enabling adaptive, decentralized coordination under partial observability. The systematic approach to problem formalization, environment design, and solution architecture establishes the foundation for experimental evaluation and provides the tools necessary to investigate MARL's practical applicability in evacuation scenarios.

Chapter 5

Experimental Results

This chapter presents a comprehensive experimental evaluation of multi-agent reinforcement learning approaches for emergency evacuation scenarios, demonstrating the practical effectiveness of distributed coordination strategies. The experimental framework ensures reproducible and comparable results across different algorithmic approaches and environmental complexities. The chapter is organized into six main sections. Section 5.1 establishes the experimental framework, including algorithm selection, training parameters, and scenario design. Section 5.2 analyzes the reward function design and its impact on learning dynamics. Section 5.3 investigates emergent coordination behaviors through novel metrics. Section 5.4 examines how environmental complexity affects multi-agent coordination and navigation strategies. Section 5.5 presents a comprehensive performance evaluation. Finally, Section 5.6 provides a comparison with traditional approaches, followed by an integrated discussion that synthesizes the findings and establishes practical implications for real-world evacuation systems.

5.1 Experimental Setup

This section provides an overview of the experimental framework developed to evaluate the proposed multi-agent reinforcement learning (MARL) approach. The experimental setup includes the algorithms used, the training parameters, the simulated evacuation scenarios, the reproducibility measures adopted, and the overall flexibility of the environment.

5.1.1 Algorithms

The experiments were carried out using three widely adopted reinforcement learning algorithms implemented via the *Stable-Baselines3* library: *Proximal Policy Optimization* (PPO), *Advantage Actor-Critic* (A2C), and *Deep Q-Network* (DQN). These algorithms were selected due to their proven performance in multi-agent domains, contrasting learning paradigms (on-policy vs. off-policy), and compatibility with the *PettingZoo* multi-agent environment used throughout this work.

Extensive comparative studies in reinforcement learning literature support the choice of these three algorithms. De la Fuente and Vidal Guerra (2024) conducted a comprehensive analysis of DQN, PPO, and A2C performance, demonstrating that DQN exhibits remarkable resilience across different hyperparameter configurations due to its experience replay mechanism, while PPO shows superior stability through its clipped objective function that prevents destructive policy updates [16]. Their findings indicate that A2C requires more careful hyperparameter tuning but can achieve competitive performance when properly configured.

In multi-agent evacuation scenarios, specifically, recent research has shown promising results with these algorithms. Zhang et al. successfully applied DQN combined with social force models for emergency evacuation, demonstrating the algorithm’s ability to efficiently eliminate exits and obstacles [73]. Similarly, studies on multi-agent reinforcement learning for evacuation planning have shown that MARL approaches using PPO-based algorithms can outperform single-agent methods in complex scenarios with multiple agents and dynamic constraints [68].

All agents were trained using a multilayer perceptron (MLP) policy network with standard activation functions. Policy and value networks shared parameters unless otherwise specified. Observations and actions were processed vectorized, taking advantage of the parallel execution tools provided by the *Supersuit* library to enable efficient training on CPU. This shared architecture approach is consistent with recent MARL implementations that have succeeded in evacuation and coordination tasks [68].

PPO was implemented using the default hyperparameters from Stable-Baselines3 to establish a baseline performance reference. However, A2C and DQN required scenario-specific hyperparameter tuning to achieve optimal performance across the different evacuation scenarios. The complete hyperparameter specifications for A2C and DQN across all scenarios are detailed in Appendix A, including the rationale for each adjustment and the specific values used for different scenario complexities. This approach ensures reproducibility while allowing for fair comparison across different algorithmic paradigms, with PPO serving as the stable baseline and A2C/DQN demonstrating their performance potential when properly tuned.

5.1.2 Training Parameters

The training configuration varied according to the complexity of each scenario but followed a consistent methodology to ensure comparability across algorithms and environments. Agents were trained over 700,000 steps, with each episode limited to 150 simulation cycles (`max_cycles`) to accommodate different task lengths.

The agents ranged from 2 in simple settings to 6 in high-density scenarios. Similarly, the number and placement of obstacles and walls were scenario-dependent. Table 5.1 summarizes the key training settings used throughout the experiments.

Parameter	Value(s)	Remarks
Algorithms	PPO, A2C, DQN	Selected for diversity in policy updates
Total training steps	700,000	Consistent across all scenarios
Max cycles per episode	150	Fixed episode duration
Number of agents	2-6	Scales with environment density
Obstacles	0 or 3	Used to shape navigation constraints
Walls	0 or 2	Create corridors and narrow passages
Policy architecture	MLP (2 hidden layers)	Shared policy across agents
Hardware	CPU (no GPU)	Consistent across all runs

Table 5.1: Training parameters used across all experiments.

5.1.3 Scenarios

Five custom scenarios were implemented to emulate evacuation dynamics under varying degrees of spatial and informational complexity. All scenarios were based on a modified version of the PettingZoo `simple_spread` environment, adapted to include obstacles, constrained passages, and support for agent communication.

Scenario 1 – Free Navigation (Figure 5.1): Agents must move toward a visible meeting point in a fully open space with no obstacles or restrictions. This baseline scenario tests basic goal-directed behavior and serves as a foundation for comparison with more complex environments.

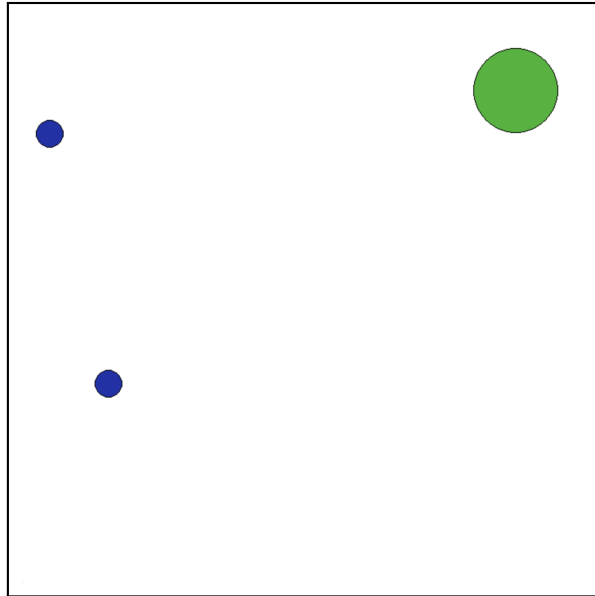


Figure 5.1: Scenario 1: Free navigation with 2 agents (blue circles) moving toward the visible meeting point (green circle) in an open environment without obstacles.

Scenario 2 – Corridor with Walls (Figure 5.2): A narrow corridor created by wall obstacles restricts the direct path to the meeting point, which requires agents to learn the coordinated passage through the bottleneck. This scenario tests spatial coordination and collision avoidance in constrained environments.

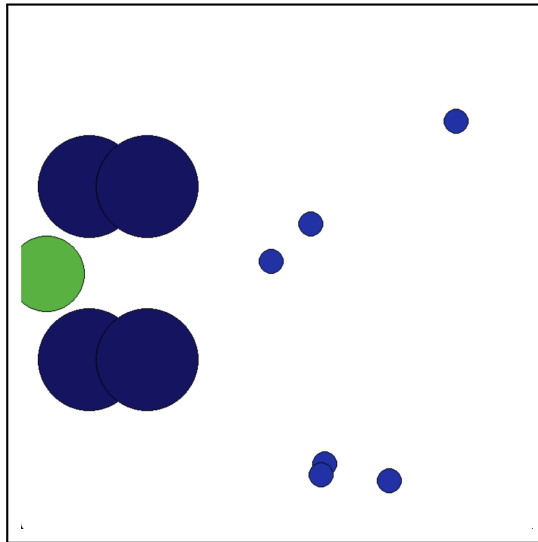


Figure 5.2: Scenario 2: Corridor navigation with 6 agents (blue circles) navigating through bottlenecks created by walls (gray circles) toward the meeting point (green circle).

Scenario 3 – Multiple Obstacles with Corridor (Figure 5.3): The environment combines the corridor from Scenario 2 with additional static obstacles, creating multiple navigation challenges. Agents must coordinate to avoid wall constraints and obstacle collisions while reaching the meeting point efficiently.

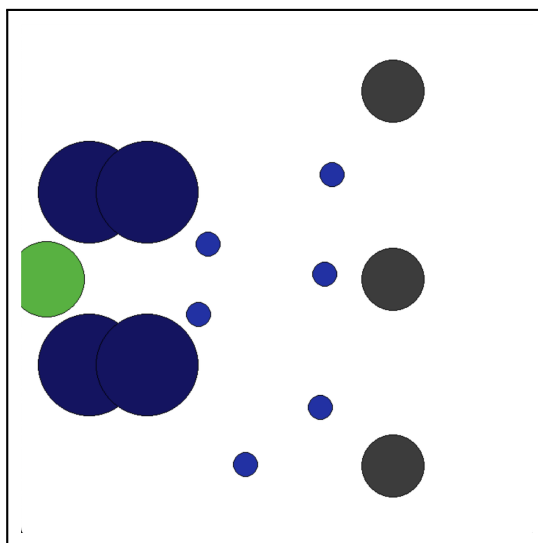


Figure 5.3: Scenario 3: Complex navigation with 6 agents (blue circles) facing both corridor walls (gray circles) and multiple obstacles (gray circles in the middle) toward the meeting point (green circle).

Scenario 4 – Partial Observability with Communication (Figure 5.4): The meeting point is not visible from the agents' initial positions, agent discovery, and inter-agent communication. Successful evacuation depends on collaborative exploration and information sharing to locate and navigate to the hidden meeting point.

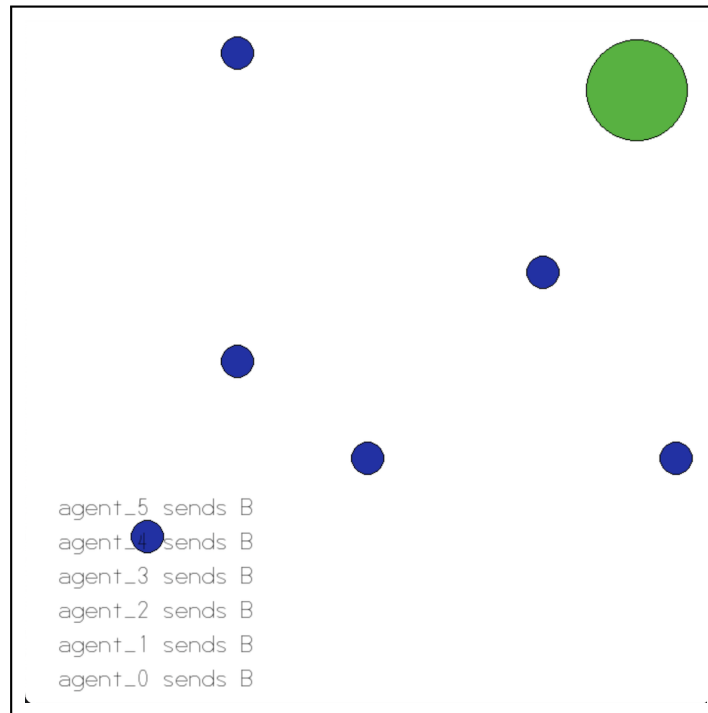


Figure 5.4: Scenario 4: Communication-based evacuation with 6 agents (blue circles) discovering and sharing the meeting point (green circle) location through inter-agent communication.

Scenario 5 – Partial Observability with Communication and Obstacles (Figure 5.5): This scenario combines the communication challenges from Scenario 4 with the spatial complexity of static obstacles. The meeting point remains hidden from agents' initial positions, requiring collaborative exploration and information sharing, but agents must now navigate around strategically placed obstacles while communicating the meeting point location. This scenario tests the most complex form of multi-agent coordination, where agents must simultaneously manage exploration, communication, obstacle avoidance, and cooperative navigation.

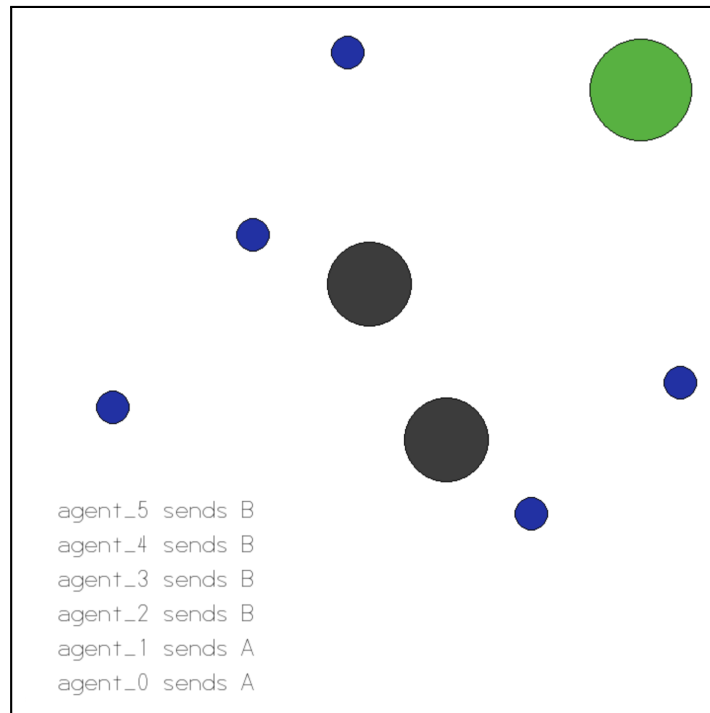


Figure 5.5: Scenario 5: Communication-based evacuation with 6 agents (blue circles) discovering and sharing the meeting point (green circle) location while navigating around obstacles (gray circles).

Each scenario was designed to evaluate different aspects of multi-agent coordination, from basic navigation to complex cooperative behavior under partial observability and spatial constraints.

5.1.4 Reproducibility

To ensure full reproducibility of results, random seeds were fixed across all training and evaluation runs. All model checkpoints were saved with timestamps, and evaluation logs were exported in CSV format for further analysis. The experiments were conducted using the technologies and versions specified in Section 4.4.4. All training and inference procedures were executed on CPU to ensure consistency across different hardware configurations.

The training and evaluation pipeline was designed to be fully modular and repeatable. Scripts for scenario generation, agent configuration, training, and evaluation were isolated into distinct modules to support independent experimentation and facilitate future research extensions.

5.1.5 Environment Flexibility

The environment was implemented with a high degree of modularity to support testing of diverse configurations. Scenarios were defined via structured configuration dictionaries specifying the number and position of agents, obstacles, meeting point, and walls. This parameterization enabled rapid instantiation of environments with varying difficulty and topology.

The simulation code allowed toggling features such as communication channels, meeting point visibility, and observation ranges. This flexibility made it possible to isolate the effects of specific design choices, such as reward shaping or observability constraints, on the agents' learning performance, laying the foundation for the comprehensive analyses presented in the following sections.

5.2 Comparative Algorithm Performance and Reward Function Effectiveness

The reward function design serves as the fundamental mechanism through which multi-agent reinforcement learning algorithms learn to balance competing objectives in emergency evacuation scenarios. Rather than simply encouraging individual goal achievement, the implemented reward structure must simultaneously promote collective coordination, safety through collision avoidance, and time-efficient evacuation strategies. The comprehensive analysis of how different algorithms respond to this reward structure across varying scenario complexities reveals critical insights into both the effectiveness of the reward design and the relative capabilities of each algorithmic approach in handling distributed coordination challenges.

5.2.1 Algorithm Performance Hierarchy and Learning Stability

The experimental results demonstrate a consistent performance hierarchy across all five evacuation scenarios, with Proximal Policy Optimization (PPO) emerging as the most robust and adaptable algorithm for multi-agent evacuation tasks. This superiority manifests not only in terms of final performance metrics but also in the stability and consistency of the learning process itself. PPO's clipped surrogate objective function proves particularly well-suited to the non-stationary environment created by multiple simultaneously learning agents, as it prevents the destructive policy updates that can occur when agents must adapt to the changing behaviors of their peers while maintaining their own learned coordination strategies.

The learning curves (Figure 5.6) reveal that PPO maintains superior average cumulative rewards across all scenarios while exhibiting the most stable learning trajectory, particularly evident in the narrower confidence intervals observed during training. This stability becomes increasingly valuable as scenario complexity escalates, with PPO demonstrating remarkable resilience even when faced with the combined challenges of partial observability, communication requirements, and complex spatial navigation in Scenario 5. The algorithm's ability to maintain consistent performance despite these compounding difficulties suggests that its policy gradient approach, enhanced by the clipped objective mechanism, provides an optimal balance between exploration and exploitation in multi-agent settings where coordination emerges through trial-and-error learning rather than explicit programming.

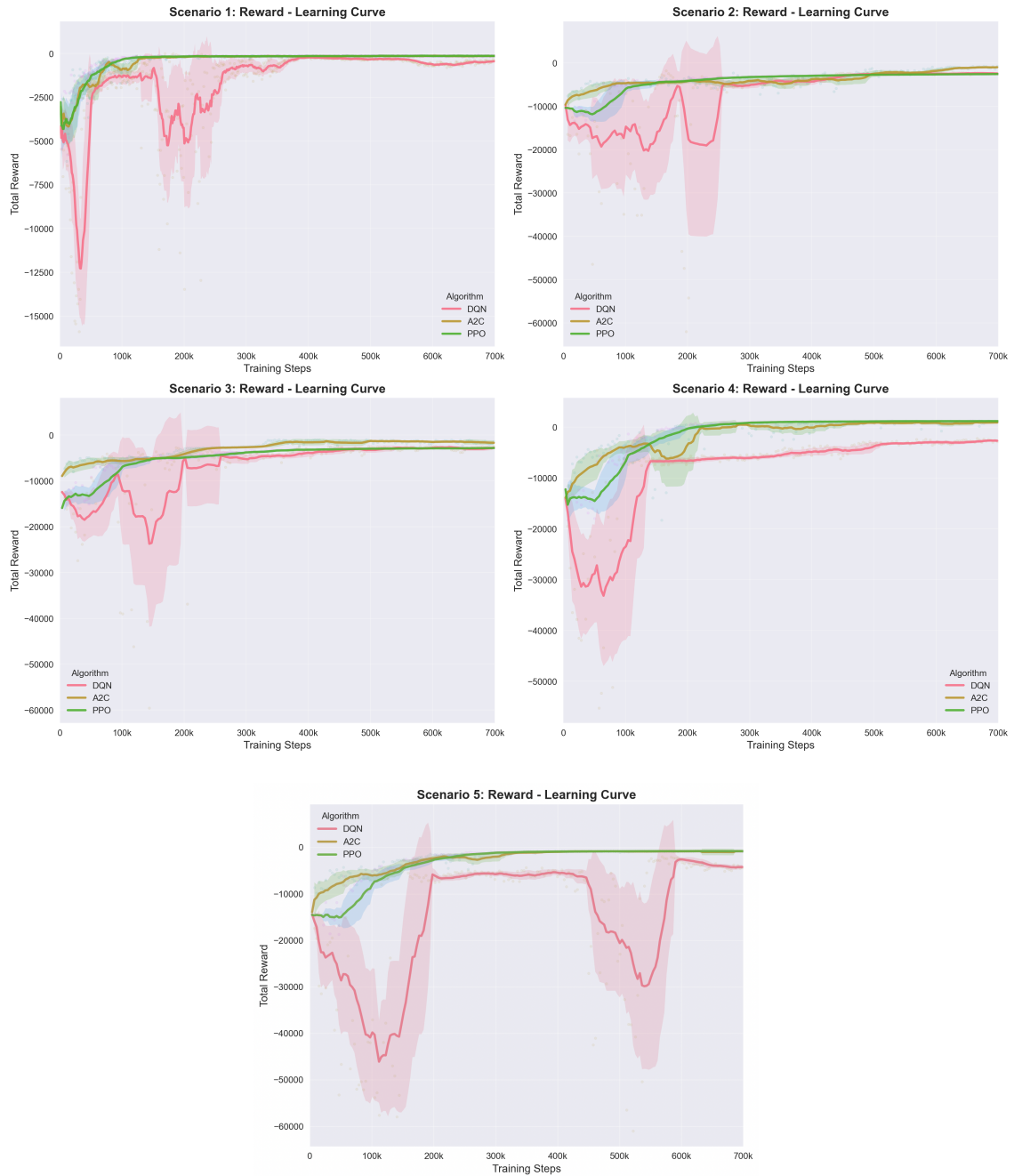


Figure 5.6: Learning curves with moving average and confidence intervals for PPO, A2C, and DQN algorithms across five evacuation scenarios. The x-axis represents training steps (0-700k), the y-axis shows cumulative reward, and shaded areas indicate reward variability across multiple training runs.

In contrast, Advantage Actor-Critic (A2C) demonstrates competent performance in simpler environments but exhibits a progressive decline in effectiveness as environmental complexity increases. While A2C maintains relatively stable learning trajectories in the initial scenarios, its performance becomes increasingly volatile as coordination demands intensify. This degradation pattern suggests that while actor-critic methods can effectively balance value-based and policy-

based learning in moderate coordination scenarios, they may lack the stability mechanisms necessary to handle the complex interdependencies that emerge when multiple agents must simultaneously learn navigation, communication, and coordination strategies. The algorithm’s performance in scenarios with communication requirements particularly highlights this limitation, as the additional complexity of learning when and how to share information appears to overwhelm A2C’s learning capacity.

Deep Q-Networks (DQN) present the most concerning learning dynamics among the evaluated algorithms, with performance deteriorating rapidly as environmental demands increase. The algorithm’s fundamental reliance on experience replay and off-policy learning creates inherent challenges in multi-agent settings where the environment becomes non-stationary due to the concurrent learning of other agents. This non-stationarity violates key assumptions underlying DQN’s convergence guarantees and leads to the most dramatic manifestation of learning instability observed in Scenario 5, where the algorithm experiences catastrophic forgetting [31] around the 500,000 training step mark. This phenomenon, characterized by a sudden and dramatic loss of previously learned capabilities followed by only partial recovery, illustrates DQN’s structural limitations in handling complex multi-agent coordination tasks where maintaining previously learned behaviors while adapting to new challenges becomes critical for success.

5.2.2 Reward Function Impact on Coordination Emergence

The reward function’s effectiveness in promoting desired behaviors varies significantly across algorithms and scenario complexities, revealing important insights about how different learning paradigms respond to the same incentive structures. The multi-component reward design, which integrates time penalties, collision avoidance incentives, goal-directed rewards, and scenario-specific bonuses, proves most effective when paired with algorithms capable of stable policy updates in non-stationary environments. PPO’s consistent ability to internalize these diverse reward signals and translate them into coherent coordination behaviors demonstrates the importance of algorithmic stability in multi-agent learning contexts.

The analysis reveals that the reward function’s collision avoidance components successfully reduce collision rates across all algorithms, indicating that agents can indeed learn to internalize safety constraints as part of their decision-making processes. However, the effectiveness of this learning varies considerably, with PPO and DQN achieving significantly lower collision rates than A2C, particularly in scenarios with complex spatial constraints. This differential response suggests that the reward function’s safety-promoting components require stable learning dynamics to be fully effective, as algorithms experiencing high learning variance may struggle to consistently apply learned safety behaviors across different environmental configurations.

More critically, the reward function’s ability to promote emergent coordination behaviors becomes increasingly strained as scenario complexity escalates. While the basic goal-directed and collision avoidance components remain effective across all scenarios, the emergence of sophisticated coordination strategies such as queue formation and synchronized movement patterns

appears to require learning stability that only PPO consistently provides. The communication-enabled scenarios particularly highlight this limitation, as the reward structure must simultaneously encourage information sharing, coordinate spatial navigation, and maintain individual goal pursuit objectives. DQN's catastrophic forgetting episode in Scenario 5 suggests that the current reward design may lack the structured guidance necessary to maintain complex learned behaviors when agents must simultaneously master multiple coordination mechanisms.

5.2.3 Generalization Capabilities and Scalability Implications

One of the most significant findings regarding reward function effectiveness concerns the differential generalization capabilities exhibited by the three algorithms when trained policies are evaluated with varying numbers of agents. PPO and A2C demonstrate remarkable robustness in this regard, maintaining high performance and coordination effectiveness even when the number of agents during evaluation exceeds the training population. This generalization capability suggests that these algorithms successfully learn coordination principles rather than merely memorizing specific multi-agent configurations, indicating that the reward function effectively promotes the development of scalable coordination strategies.

The ability of PPO and A2C to generalize across different agent populations without requiring retraining represents a crucial advantage for real-world deployment scenarios where the number of evacuees may vary unpredictably. This generalization success indicates that the reward function's design successfully captures the essential elements of coordination behavior in a way that transcends specific population sizes. The algorithms appear to learn fundamental principles of spatial coordination, collision avoidance, and goal-directed movement that remain applicable regardless of the specific number of agents present in the environment.

DQN's failure to demonstrate similar generalization capabilities reveals a critical limitation that extends beyond simple algorithmic performance to fundamental questions about scalability in multi-agent reinforcement learning systems. The algorithm's marked decline in efficiency when evaluated with larger agent populations than those used during training suggests that its learning process becomes overly specialized to specific environmental configurations. This limitation appears to stem from DQN's heightened sensitivity to hyperparameters, which require careful tuning as environmental complexity increases. Without such adjustments, DQN struggles to maintain the delicate balance between exploration and exploitation necessary for effective coordination, leading to degraded performance that undermines the practical applicability of learned policies.

5.2.4 Scenario Complexity Impact on Learning Dynamics

The progressive increase in scenario complexity reveals fundamental insights about how different algorithmic approaches handle the escalating coordination demands inherent in realistic evacuation scenarios. The transition from simple goal-directed navigation in Scenario 1 to the complex

integration of communication, obstacle avoidance, and coordination in Scenario 5 creates a natural experimental gradient that exposes the relative strengths and limitations of each algorithm’s learning paradigm.

In the initial scenarios, characterized by relatively straightforward spatial navigation requirements, all algorithms demonstrate rapid learning with steep performance improvements occurring within the first 100,000 training steps. These environments, with their reduced interaction dependencies and minimal planning requirements, allow policies to converge efficiently regardless of the specific algorithmic approach. However, as complexity intensifies through the introduction of spatial constraints, communication requirements, and partial observability, the learning process becomes markedly slower and more fragile, with effective policy emergence often requiring extended training horizons that stretch throughout the entire training window.

The increasing complexity burden particularly affects coordination requirements, as later scenarios necessitate strategic path planning and behavioral alignment across agents, further complicated by the need for timely and context-aware information exchange. Scenario 5 exemplifies this compounded difficulty, where agents must simultaneously master spatial navigation around obstacles, develop effective communication protocols for sharing critical information, and maintain coordination behaviors that prevent congestion and conflicts. Under these demanding conditions, the differences in algorithmic robustness become especially pronounced, with PPO maintaining stable performance trajectories while A2C exhibits growing instability and DQN experiences the dramatic performance collapse characteristic of catastrophic forgetting.

This complexity-performance relationship suggests that the reward function’s effectiveness becomes increasingly dependent on the underlying algorithmic stability as coordination demands escalate. While the reward structure provides adequate guidance for basic behaviors across all algorithms, the emergence and maintenance of sophisticated coordination strategies appears to require the kind of stable learning dynamics that only PPO consistently provides throughout the experimental evaluation.

5.2.5 Implications for Real-World Deployment

The comprehensive analysis of reward function effectiveness and algorithmic performance across varying complexity levels provides crucial insights for the practical deployment of MARL-based evacuation systems. The consistent superiority of PPO across all scenarios, combined with its remarkable generalization capabilities and stability under increasing complexity, establishes it as the preferred algorithmic choice for real-world evacuation applications where reliability and adaptability are paramount.

The reward function design, while effective in promoting basic coordination and safety behaviors, reveals limitations in guiding the emergence of complex multi-faceted coordination strategies. Future developments should consider incorporating hierarchical reward structures that provide intermediate objectives and milestone-based feedback to support more stable policy development in complex scenarios. Additionally, the integration of shared or collective reward components may

prove essential for promoting inter-agent cooperation in environments where task success depends on synchronized decision-making and communication.

The differential generalization capabilities observed across algorithms highlight the critical importance of algorithm selection in multi-agent systems where deployment conditions may vary significantly from training environments. The ability of PPO and A2C to maintain effective coordination across different agent populations without retraining represents a significant practical advantage, while DQN's requirement for scenario-specific hyperparameter tuning limits its applicability in dynamic real-world settings.

These findings collectively establish a foundation for developing robust, scalable, and adaptable evacuation systems capable of handling the complex coordination challenges inherent in real-world emergency scenarios while maintaining the safety and efficiency standards required for practical deployment.

5.3 Emergent Coordination Behaviors

This section presents a comprehensive analysis of emergent coordination behaviors observed in multi-agent evacuation scenarios, demonstrating how agents develop sophisticated coordination strategies spontaneously without explicit programming. The emergence of coordination represents one of the most fascinating and practically relevant aspects of MARL systems applied to emergency evacuation.

5.3.1 Implicit vs. Explicit Coordination

The experimental results reveal a fundamental distinction between implicit and explicit coordination mechanisms, each contributing uniquely to overall evacuation effectiveness. Implicit coordination emerges naturally through reinforcement learning, where agents develop cooperative behaviors as a result of optimizing individual rewards that collectively promote group objectives.

The most notable phenomenon of implicit coordination observed is the **spontaneous queue formation** at bottleneck points, particularly evident in Scenario 2 where agents must navigate through narrow corridors (Figure 5.7). This single-file organization is not explicitly programmed but emerges as an optimal solution learned by agents to minimize spatial conflicts and maximize flow through constrained passages. Behavioral analysis shows that agents learn to anticipate others' actions, adjusting their trajectories to avoid collisions and form orderly movement patterns.

Explicit coordination, on the other hand, manifests through communication mechanisms implemented in Scenarios 4 and 5. When agents can share information about meeting point locations, significant improvements in evacuation efficiency are observed. This communication capability allows agents to overcome partial observability limitations, facilitating collaborative discovery and sharing of critical evacuation route information.

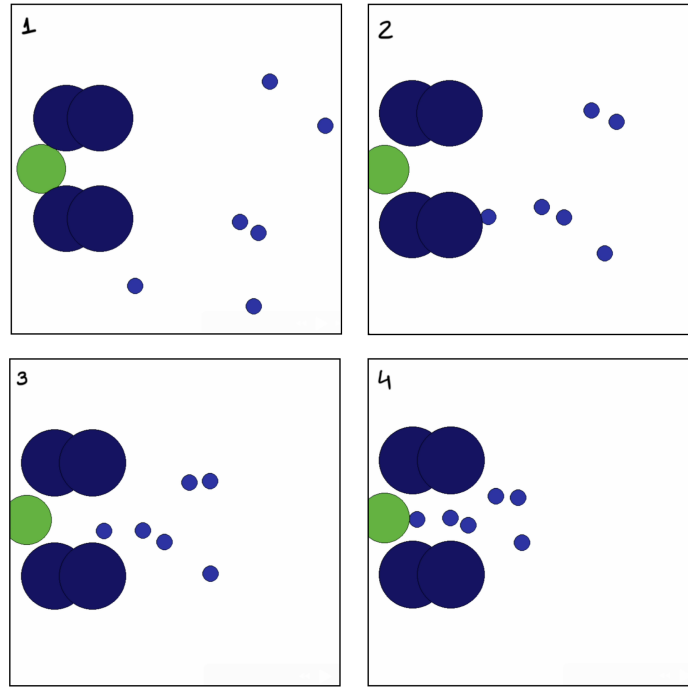


Figure 5.7: Emergent queue formation behavior in Scenario 2 (corridor navigation) showing 6 agents (blue circles) naturally organizing into a single-file formation when approaching the bottleneck created by wall constraints (dark blue big circles). The agents demonstrate spontaneous coordination without explicit queue-forming rules, with the leading agent positioned closest to the meeting point (green circle) and subsequent agents maintaining orderly spacing behind.

Comparative analysis between scenarios with and without communication reveals that while implicit coordination suffices for simple scenarios, explicit coordination becomes essential in complex environments with limited observability. The **Coordination Index** developed quantifies this difference, showing consistently higher values when both mechanisms operate synergistically.

5.3.2 Coordination Index: Methodology and Implementation

To quantitatively assess the degree of coordination among agents, a coordination index was developed and computed throughout the evaluation process. The metric is grounded in the spatial distribution and synchronization of agent movements, aiming to capture how closely agents align their trajectories and arrival times at the meeting point.

The coordination index calculation follows a systematic methodology designed to measure spatial clustering and synchronized behavior:

Data Collection Phase: For each evaluation episode, the final positions of all agents are recorded when they reach the meeting point or when the episode terminates. This provides a snapshot of the spatial distribution achieved by the agents at the conclusion of their evacuation attempt.

Centroid Calculation: The geometric centroid of all agent final positions is computed using standard coordinate averaging:

$$\text{Centroid} = \left(\frac{1}{N} \sum_{i=1}^N x_i, \frac{1}{N} \sum_{i=1}^N y_i \right) \quad (5.1)$$

where N is the number of agents and (x_i, y_i) represents the final position of agent i .

Distance Measurement: The Euclidean distance from each agent’s final position to the computed centroid is calculated:

$$d_i = \sqrt{(x_i - x_{\text{centroid}})^2 + (y_i - y_{\text{centroid}})^2} \quad (5.2)$$

Variance Computation: The variance of these distances serves as an indicator of spatial dispersion, where lower variance implies higher spatial coordination:

$$\text{Var}(d) = \frac{1}{N} \sum_{i=1}^N (d_i - \bar{d})^2 \quad (5.3)$$

where $\bar{d}$ is the mean distance to the centroid.

Index Formulation: The coordination index is defined as:

$$CI = \frac{1}{1 + \text{Var}(d)} \quad (5.4)$$

This formulation ensures that the index approaches 1 when agents are tightly clustered around the centroid (indicating high coordination) and decreases toward 0 as spatial dispersion increases (indicating poor coordination). The reciprocal relationship with variance provides an intuitive interpretation where higher coordination values correspond to better synchronized spatial arrangements.

This metric is particularly suitable for evacuation tasks as it rewards synchronized arrivals and penalizes scattered or uncoordinated behaviors. The index is robust to the number of agents and can be interpreted consistently across different scenarios, making it an effective tool for comparing coordination effectiveness across different algorithms and environmental conditions.

5.3.3 Scalability and Generalization

A particularly significant finding of this investigation concerns the generalization capability of learned policies regarding agent numbers. Specifically, when algorithms were trained with 6 agents, the resulting policies remained effective during evaluation even when agent numbers were increased. This robustness was observed for both PPO and A2C, which maintained high performance and coordination without requiring retraining or significant hyperparameter adjustments.

This generalization capability is highly desirable in practical applications, as it enables flexible deployment in environments where agent numbers may vary. The analysis suggests that algorithms successfully learn coordination principles rather than simply memorizing specific multi-agent configurations. Agents appear to internalize fundamental rules of spatial coordination, collision avoidance, and goal-directed movement that remain applicable regardless of the specific number of agents present in the environment.

However, this property was not shared by DQN. The DQN algorithm exhibited marked decline in efficiency when evaluated with larger agent populations than those used during training. This limitation is attributed to DQN’s heightened sensitivity to hyperparameters, which require careful tuning as agent populations increase. Without such adjustments, DQN struggles to maintain stable learning and effective coordination, leading to degraded performance in larger-scale scenarios.

Scalability analysis also reveals that emergent coordination behaviors remain stable even with larger agent populations. Queue formation, for example, adapts naturally to agent numbers, with longer queues forming automatically when more agents are present, while maintaining the same fundamental organizational structure.

This behavior is demonstrated in Figure 5.8, which shows a scenario in which 12 agents successfully maintain the formation of ordered queues despite the increase in population density. Remarkably, this coordination was achieved using a policy trained with only 6 agents, demonstrating the strong generalization capabilities of the MARL approach and its ability to scale emergent coordination behaviors to larger agent populations.

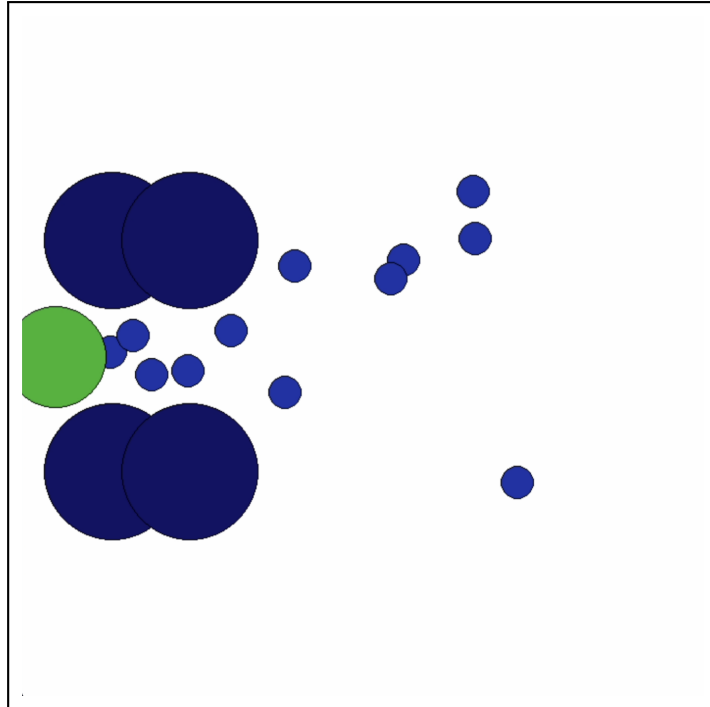


Figure 5.8: Queue formation scalability demonstration in Scenario 2 with 12 agents. The agents automatically organizing into stable queue structures, illustrating how emergent coordination behaviors remain consistent across different population sizes.

5.3.4 Empirical Results and Analysis

The coordination index results presented in Figure 5.9 correspond to the evaluation phase, displaying measurements across six distinct evaluation episodes. Each data point represents the coordination achieved by agents in a single evaluation episode, providing a comprehensive view of the spatial organization capabilities developed through training.

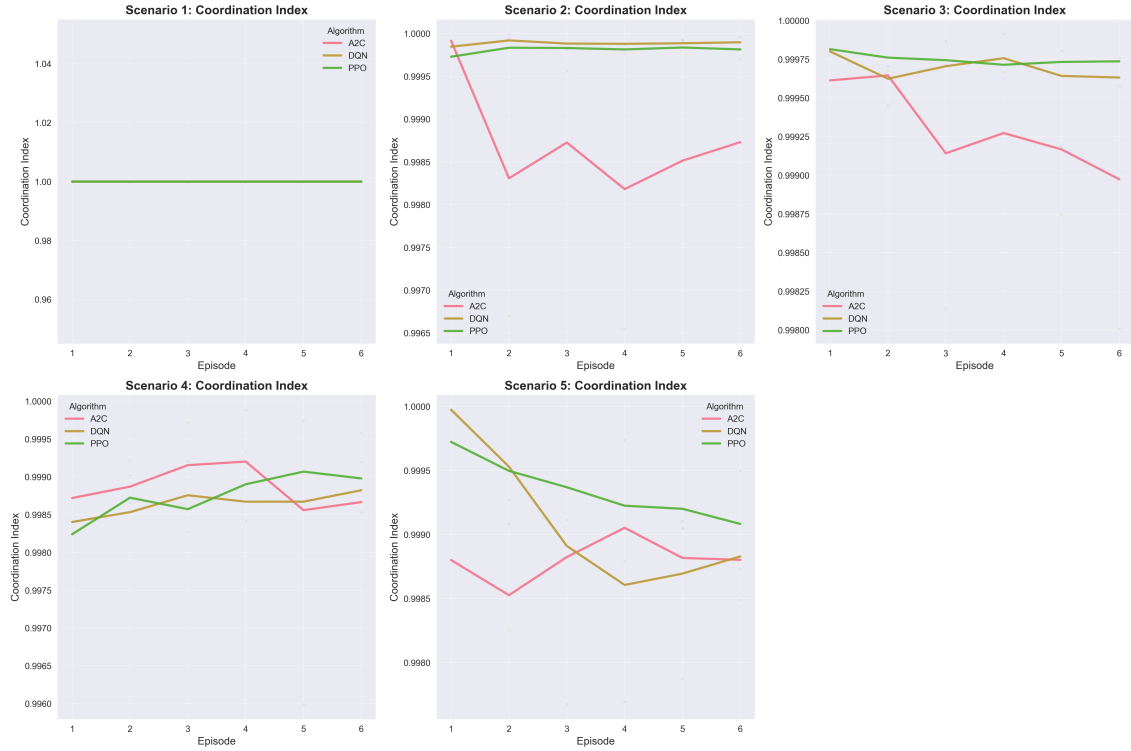


Figure 5.9: Coordination index per evaluation episode across all scenarios and algorithms. The index ranges from 0 to 1, where values approaching 1 indicate high spatial coordination (agents clustered near the centroid) and values near 0 indicate dispersion.

The empirical results demonstrate that across all algorithms and scenarios, agents maintain consistently high coordination throughout evaluation episodes. This indicates robust synchronized movement and spatial organization capabilities, with coordination index values typically exceeding 0.98 across all tested conditions. The consistency of these high values suggests that the learned coordination behaviors are stable and reliable, even when evaluated under varying initial conditions or stochastic environmental elements.

Among the algorithms, PPO and DQN generally achieve slightly higher coordination indices compared to A2C, suggesting marginal advantages in fostering tightly clustered, cooperative behaviors. However, the differences are subtle, and all methods demonstrate the ability to produce robust, decentralized coordination in the evaluation setting. The absence of significant drops or variability in the coordination index across episodes further underscores the stability and reliability of the learned policies.

The minor fluctuations observed between episodes likely reflect natural variations in agent starting positions or stochastic elements in the environment rather than systematic coordination failures. This stability across episodes demonstrates that the coordination behaviors learned during training are sufficiently robust to handle the variability inherent in real-world deployment scenarios.

5.3.5 Implications for Real-World Evacuation Systems

The emergent coordinated behaviors observed in this investigation have significant implications for real-world evacuation system design. The ability of agents to spontaneously develop effective coordination strategies suggests that MARL systems can surpass traditional rule-based approaches, particularly in unforeseen situations where adaptability is crucial.

The **transferability** of learned behaviors to different layouts and environmental configurations represents a significant practical advantage. Emergent coordination patterns, such as queue formation and predictive avoidance, are sufficiently general to be applicable to a variety of evacuation scenarios, from office buildings to public transportation stations.

The **scalability** demonstrated by PPO and A2C algorithms suggests that these systems can be implemented in large-scale evacuation scenarios without requiring extensive retraining. This property is particularly valuable for applications in critical infrastructure where evacuee numbers may vary significantly.

Finally, the **robustness** of emergent coordinated behaviors provides confidence in the reliability of these systems under real-world conditions, where perturbations and uncertainties are inevitable. The self-repair and adaptation capabilities of emergent coordinated systems represent a significant advantage over centralized systems that may fail completely when critical components are compromised.

The coordination index methodology developed in this work provides a quantitative framework for evaluating and comparing coordination effectiveness across different MARL approaches, establishing a foundation for future research in intelligent evacuation systems and distributed coordination mechanisms.

5.4 Environmental Complexity Analysis

This section presents a systematic analysis of how environmental complexity affects multi-agent coordination and learning dynamics across the five evacuation scenarios. The progressive increase in environmental complexity, from simple open spaces to constrained corridors with obstacles and partial observability, provides a natural experimental gradient that reveals the relative strengths and limitations of each algorithmic approach in handling distributed coordination challenges.

5.4.1 Impact of Spatial Constraints and Obstacles

The introduction of spatial constraints fundamentally alters the coordination dynamics required for successful evacuation. Scenarios 2, 3, and 5 incorporate obstacles and walls that create bottlenecks, force agents to coordinate their movements, and significantly increase the likelihood of congestion and collisions. This spatial complexity serves as a critical test of each algorithm's ability to learn sophisticated navigation strategies while maintaining collective coordination.

In Scenario 2, agents must traverse a narrow corridor bounded by walls, with obstacles strategically placed to create bottlenecks that force coordinated passage. The corridor design transforms the evacuation problem from simple goal-directed navigation to a complex coordination challenge where agents must learn to queue, yield, and synchronize their movements to avoid deadlocks. The experimental results reveal that this spatial constraint significantly impacts learning dynamics across all algorithms, with wider confidence intervals and increased training variance indicating the substantial coordination challenges introduced by bottleneck navigation.

Scenario 3 escalates the complexity by introducing additional obstacles alongside the corridor constraints, creating multiple navigation challenges that require agents to simultaneously manage obstacle avoidance, corridor navigation, and coordination with other agents. The combination of walls and scattered obstacles creates a complex spatial landscape where agents must develop multi-faceted navigation strategies that balance collision avoidance with efficient path planning. The learning curves for this scenario show multiple plateau regions, suggesting that agents require extended exploration to discover effective navigation strategies around multiple obstacles.

The collision analysis presented in Figure 5.10 demonstrates the progressive impact of spatial complexity on agent behavior. In Scenario 2, collision rates vary significantly across algorithms, with A2C consistently exhibiting the highest collision rates and DQN achieving the most conservative navigation. However, as complexity increases in Scenario 3, the performance gap between algorithms narrows, highlighting the escalating challenge imposed by denser environments and suggesting that the relative advantages of different approaches become less pronounced under increased spatial complexity.

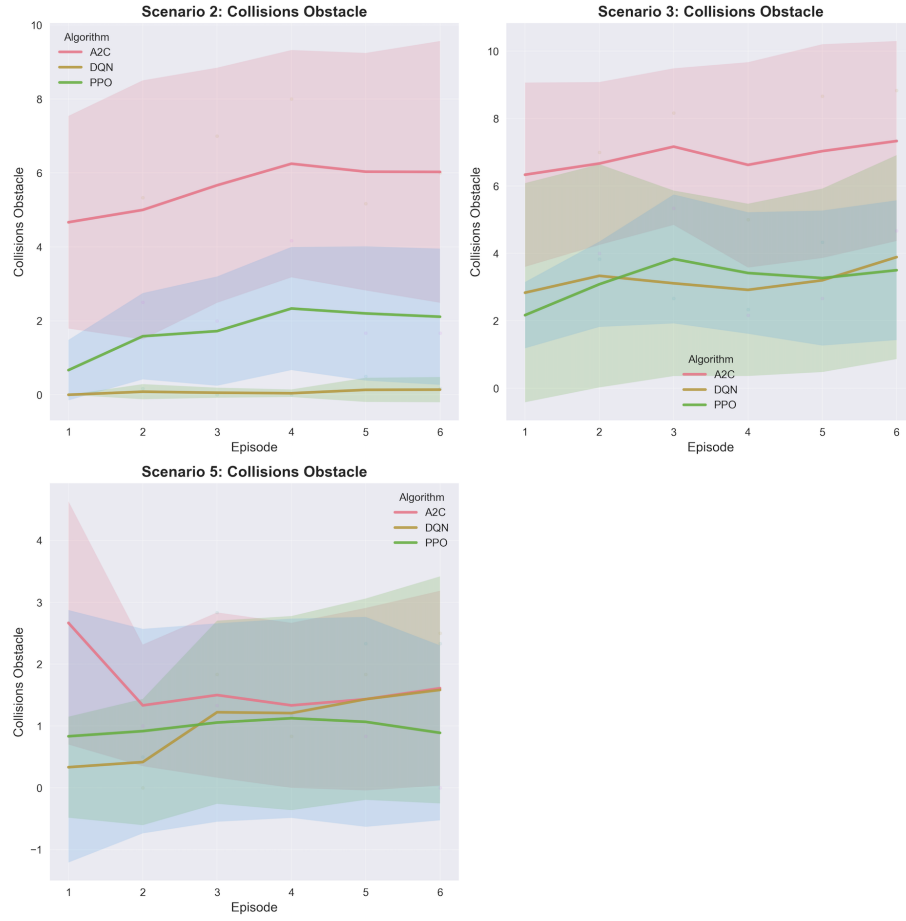


Figure 5.10: Number of collisions with obstacles per episode in Scenarios 2, 3, and 5 for all algorithms. Each panel shows the evolution of collisions throughout evaluation episodes, where lower values indicate better learning of collision prevention strategies.

5.4.2 Communication in Complex Spatial Environments

Scenario 5 represents the ultimate test of environmental complexity by combining spatial constraints with partial observability and communication requirements. In this scenario, agents must simultaneously master obstacle navigation, information discovery, and inter-agent communication. The results reveal a fascinating interaction between spatial complexity and communication effectiveness, where the ability to share information about meeting point locations becomes crucial for overcoming the combined challenges of limited observability and complex navigation.

Interestingly, the collision analysis shows that Scenario 5 exhibits lower collision rates compared to Scenarios 2 and 3, despite the additional complexity of partial observability. This counterintuitive result can be attributed to the agents' ability to share information, which fosters more coordinated and informed navigation decisions even in the presence of physical constraints. PPO demonstrates the most significant improvement in this scenario, achieving the lowest collision frequency and highlighting the synergistic effects of communication and spatial coordination.

The communication efficiency metric reveals that while information sharing remains highly effective in Scenario 5, there is a slight reduction compared to Scenario 4 due to the additional navigational challenges imposed by obstacles. The physical discovery of the meeting point may be delayed by obstacle navigation requirements, consequently affecting the initiation of information sharing. Nevertheless, the overall high efficiency values demonstrate the robustness of learned communication protocols under complex spatial constraints.

5.4.3 Algorithm-Specific Responses to Environmental Complexity

The experimental results reveal distinct algorithmic responses to increasing environmental complexity, with each algorithm demonstrating characteristic strengths and limitations as spatial and informational demands escalate.

Proximal Policy Optimization (PPO) emerges as the most robust algorithm across all complexity levels, maintaining stable performance trajectories even as environmental demands intensify. PPO's clipped surrogate objective proves particularly effective in complex multi-agent scenarios, preventing destructive policy updates that could destabilize learning when agents must simultaneously master multiple coordination mechanisms. The algorithm's consistent performance across scenarios demonstrates its ability to balance exploration and exploitation effectively, even when faced with the compound challenges of obstacle navigation, communication requirements, and partial observability.

Advantage Actor-Critic (A2C) shows competent performance in simpler environments but exhibits progressive decline as complexity increases. While A2C maintains relatively stable learning trajectories in initial scenarios, its performance becomes increasingly volatile as coordination demands intensify. The algorithm's difficulty in adapting to complex spatial environments is particularly evident in collision rates, where A2C consistently exhibits higher collision frequencies than other algorithms. This degradation pattern suggests that while actor-critic methods can effectively balance value-based and policy-based learning in moderate coordination scenarios, they may lack the stability mechanisms necessary to handle the complex interdependencies that emerge in highly constrained environments.

Deep Q-Networks (DQN) present the most concerning response to environmental complexity, with performance deteriorating rapidly as spatial and informational demands increase. The algorithm's fundamental reliance on experience replay and off-policy learning creates inherent challenges in complex multi-agent settings where the environment becomes highly non-stationary due to the concurrent learning of other agents navigating intricate spatial constraints. The most dramatic manifestation of this limitation occurs in Scenario 5, where DQN experiences catastrophic forgetting around the 500,000 training step mark, illustrating the algorithm's structural difficulty in maintaining previously learned coordination behaviors while adapting to new environmental challenges.

5.4.4 Reward Function Calibration Across Complexity Levels

The experimental investigation reveals that reward function effectiveness becomes increasingly dependent on environmental complexity, with different algorithms requiring distinct penalty structures to achieve optimal performance. The collision penalty calibration provides a clear example of this complexity-dependent tuning requirement.

A critical finding concerns the differential collision penalty values required across algorithms. A2C employs a collision penalty of 0.3, while both PPO and DQN use a lower penalty of 0.1. This difference emerged from empirical observations during preliminary experiments, where PPO with a collision penalty of 0.3 exhibited pathological behavior, remaining stationary in central regions and avoiding corridor navigation entirely (Figure 5.11). This risk-averse behavior, while reducing collisions, fundamentally impeded evacuation success by preventing agents from engaging with the spatial challenges necessary for goal achievement.

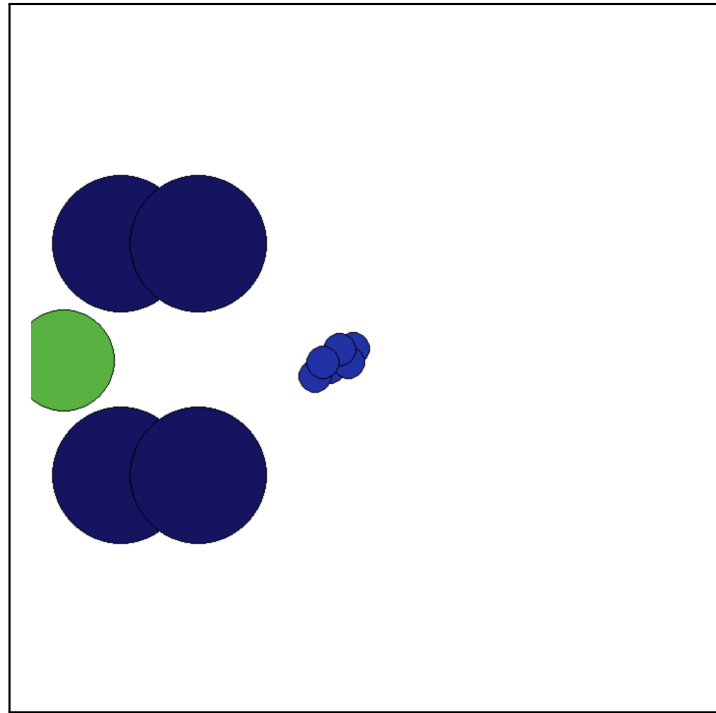


Figure 5.11: Spatial distribution of agents using PPO with a collision penalty of 0.3. Agents remain stationary in the center, avoiding the corridor.

This observation highlights the critical importance of algorithm-specific reward calibration in complex environments. The optimal balance between safety incentives and exploration encouragement varies significantly across learning paradigms, with value-based methods like DQN requiring different penalty structures than policy-based methods like PPO. The need for such calibration becomes more pronounced as environmental complexity increases, suggesting that reward function design must account not only for task objectives but also for the specific learning dynamics of the chosen algorithm.

5.4.5 Scalability Implications of Environmental Complexity

The analysis of environmental complexity reveals important implications for the scalability of MARL approaches to real-world evacuation scenarios. As complexity increases through the introduction of spatial constraints, partial observability, and communication requirements, the learning process becomes markedly slower and more fragile, with effective policy emergence often requiring extended training horizons that stretch throughout the entire training window.

The generalization capabilities observed across different agent populations provide encouraging evidence for scalability. PPO and A2C demonstrate remarkable robustness when evaluated with larger agent populations than those used during training, maintaining high performance and coordination without requiring retraining or significant hyperparameter adjustments. This generalization suggests that these algorithms successfully learn coordination principles rather than merely memorizing specific multi-agent configurations, indicating that the learned policies capture fundamental aspects of spatial coordination and collision avoidance that remain applicable regardless of population size.

However, the scalability challenges become more pronounced as environmental complexity increases. DQN's failure to demonstrate similar generalization capabilities, combined with its heightened sensitivity to hyperparameters, reveals critical limitations that extend beyond simple algorithmic performance to fundamental questions about scalability in complex multi-agent systems. The algorithm's marked decline in efficiency when evaluated with larger agent populations suggests that its learning process becomes overly specialized to specific environmental configurations, limiting its practical applicability in dynamic real-world settings.

5.4.6 Implications for Real-World Deployment

The systematic analysis of environmental complexity provides crucial insights for the practical deployment of MARL-based evacuation systems in real-world scenarios. The progressive increase in complexity from simple navigation to complex coordination with communication requirements demonstrates that while MARL approaches can handle sophisticated environmental challenges, the choice of algorithm and reward structure becomes increasingly critical as complexity escalates.

The consistent superiority of PPO across all complexity levels, combined with its remarkable generalization capabilities and stability under increasing environmental demands, establishes it as the preferred algorithmic choice for real-world evacuation applications where reliability and adaptability are paramount. The algorithm's ability to maintain stable performance trajectories even when faced with the compound challenges of obstacles, partial observability, and communication requirements suggests that it possesses the robustness necessary for deployment in unpredictable real-world environments.

The reward function analysis reveals that successful deployment requires careful calibration of incentive structures to match both the environmental complexity and the chosen learning algorithm. The pathological behaviors observed when reward structures are misaligned with algorithmic characteristics underscore the importance of comprehensive testing and validation before

real-world deployment.

Finally, the communication analysis demonstrates that explicit information sharing mechanisms can significantly enhance coordination effectiveness even in complex spatial environments. However, the slight reduction in communication efficiency observed in the most complex scenarios suggests that real-world deployment must account for the interaction between communication protocols and environmental constraints, ensuring that information sharing remains effective even when agents face multiple simultaneous challenges.

These findings collectively establish a foundation for developing robust, scalable, and adaptable evacuation systems capable of handling the complex coordination challenges inherent in real-world emergency scenarios while maintaining the safety and efficiency standards required for practical deployment.

5.5 Comprehensive Performance Evaluation

This section presents a comprehensive evaluation of multi-agent evacuation performance through a multidimensional metric framework that captures both the effectiveness and efficiency of learned coordination strategies. The evaluation employs four complementary metrics that collectively assess system reliability, temporal efficiency, spatial optimization, and safety performance across all experimental scenarios. This holistic approach enables rigorous comparison of algorithmic approaches while providing insights into the practical deployment requirements for real-world evacuation systems.

5.5.1 Evacuation Success and System Reliability

The evacuation success rate serves as the fundamental measure of system reliability, quantifying the proportion of agents that successfully reach designated meeting points within episode constraints. This metric directly reflects the system's ability to ensure comprehensive evacuation under varying environmental conditions and represents the most critical performance indicator for real-world deployment scenarios where incomplete evacuation constitutes system failure.

The empirical results demonstrate exceptional reliability across most experimental conditions, with success rates approaching perfect performance in scenarios with full observability and direct navigation paths (Figure 5.12). However, the introduction of partial observability and communication requirements in Scenarios 4 and 5 reveals important algorithmic differences in handling complex coordination challenges.

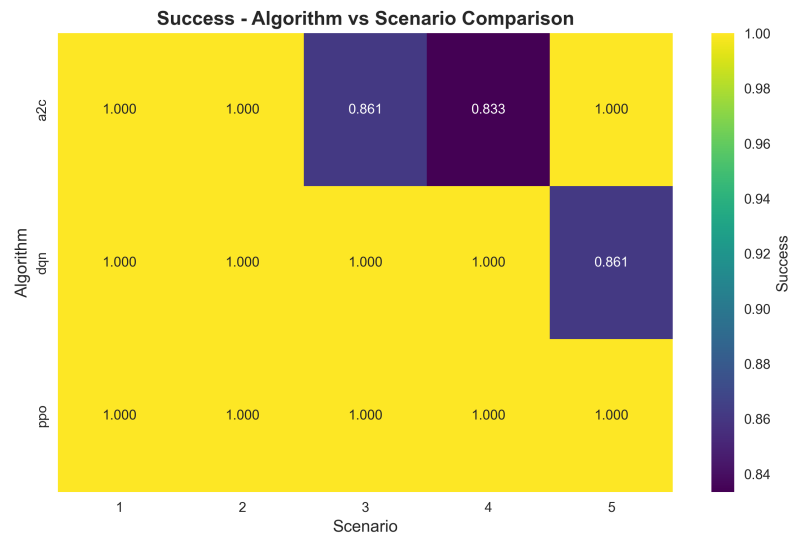


Figure 5.12: Evacuation success rates across all algorithms and scenarios, where darker colors indicate higher success rates. Perfect evacuation (1.0) appears in yellow, while partial failures show darker colors, like dark blue.

Proximal Policy Optimization maintains perfect evacuation success across all scenarios, demonstrating remarkable robustness even under the most challenging conditions involving partial observability, communication requirements, and obstacle navigation. This consistent performance underscores PPO's ability to develop stable coordination policies that remain effective across diverse environmental complexities and establishes it as the most reliable algorithmic choice for safety-critical evacuation applications.

In contrast, both Advantage Actor-Critic and Deep Q-Networks exhibit performance degradation in the most complex scenarios, with success rates declining to approximately 0.83 and 0.86 respectively in Scenarios 4 and 5. This degradation pattern suggests that these algorithms struggle with the compound challenges of partial observability, communication coordination, and spatial navigation, highlighting fundamental limitations in their ability to maintain learned coordination behaviors under increasing environmental complexity.

The differential performance across algorithms reveals critical insights about algorithmic suitability for evacuation applications. While A2C and DQN demonstrate adequate performance in simpler scenarios, their reduced reliability in complex environments raises concerns about their deployment in real-world evacuation systems where environmental conditions may be unpredictable and system failure carries severe consequences.

5.5.2 Temporal Efficiency and Evacuation Dynamics

Evacuation time represents a critical performance dimension that directly impacts survival outcomes in real emergency scenarios. The temporal analysis (Figure 5.13) reveals how different algorithms balance exploration, coordination, and goal-directed movement to achieve efficient collective evacuation across varying environmental complexities.

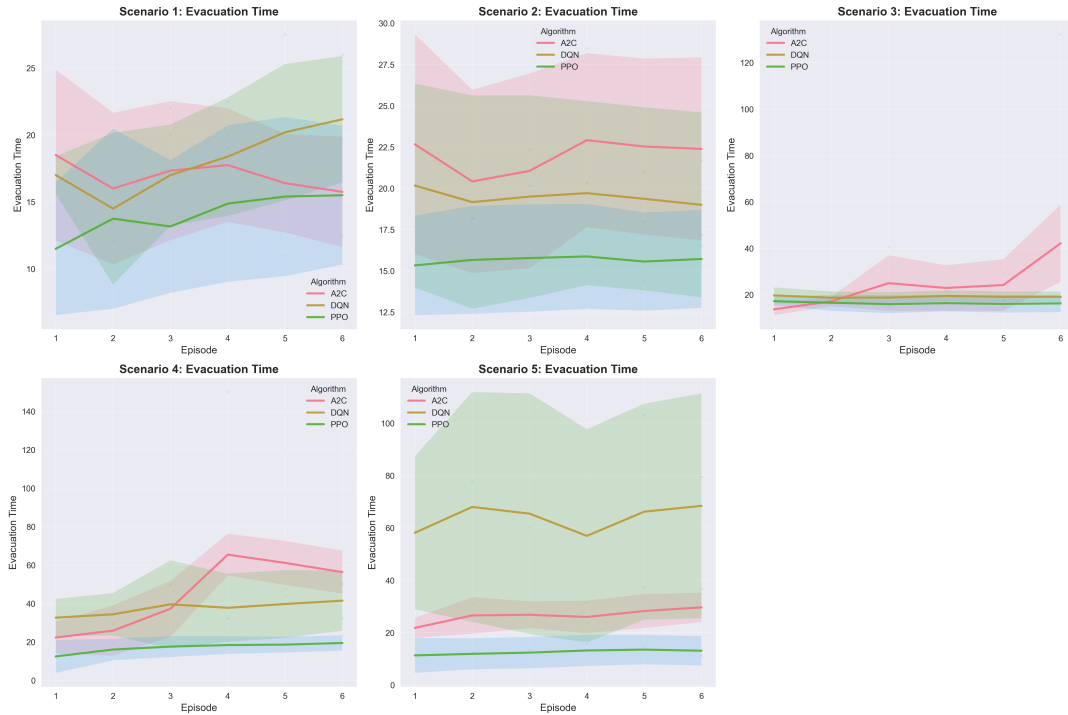


Figure 5.13: Mean evacuation time per episode across all scenarios and algorithms. Lower values indicate superior temporal efficiency, with shaded regions representing variability across evaluation episodes.

The temporal performance analysis reveals a clear algorithmic hierarchy that becomes increasingly pronounced as scenario complexity escalates. In the baseline scenarios characterized by open navigation and simple spatial constraints, all algorithms achieve comparable evacuation times, suggesting that basic goal-directed behavior can be learned effectively regardless of the underlying learning paradigm.

However, as environmental complexity increases through the introduction of obstacles, partial observability, and communication requirements, significant performance divergence emerges. PPO consistently achieves the lowest evacuation times across all scenarios, demonstrating superior ability to learn efficient navigation strategies while maintaining coordination effectiveness. This temporal efficiency stems from PPO’s stable policy updates that enable agents to develop coherent movement strategies without the oscillations and instabilities that characterize less robust learning approaches.

A2C exhibits progressively increasing evacuation times as scenario complexity escalates, with particularly pronounced delays in scenarios involving obstacles and partial observability. This pattern suggests that while A2C can learn basic coordination behaviors, it struggles to optimize temporal efficiency under complex environmental constraints where rapid adaptation and strategic planning become essential for effective evacuation.

DQN presents an intermediate performance profile, achieving reasonable evacuation times in simpler scenarios but showing increased variability and occasional delays in complex environments. The algorithm’s conservative navigation policies, while contributing to lower collision

rates, may impede rapid evacuation when aggressive coordination and risk-taking become necessary for optimal temporal performance.

5.5.3 Spatial Navigation Efficiency

Path efficiency quantifies the spatial optimization achieved by learned policies, measuring the ratio between actual traversed distances and optimal shortest paths to evacuation targets. This metric reveals how effectively agents learn to navigate complex spatial environments while avoiding unnecessary detours and maintaining coordination with other evacuees.

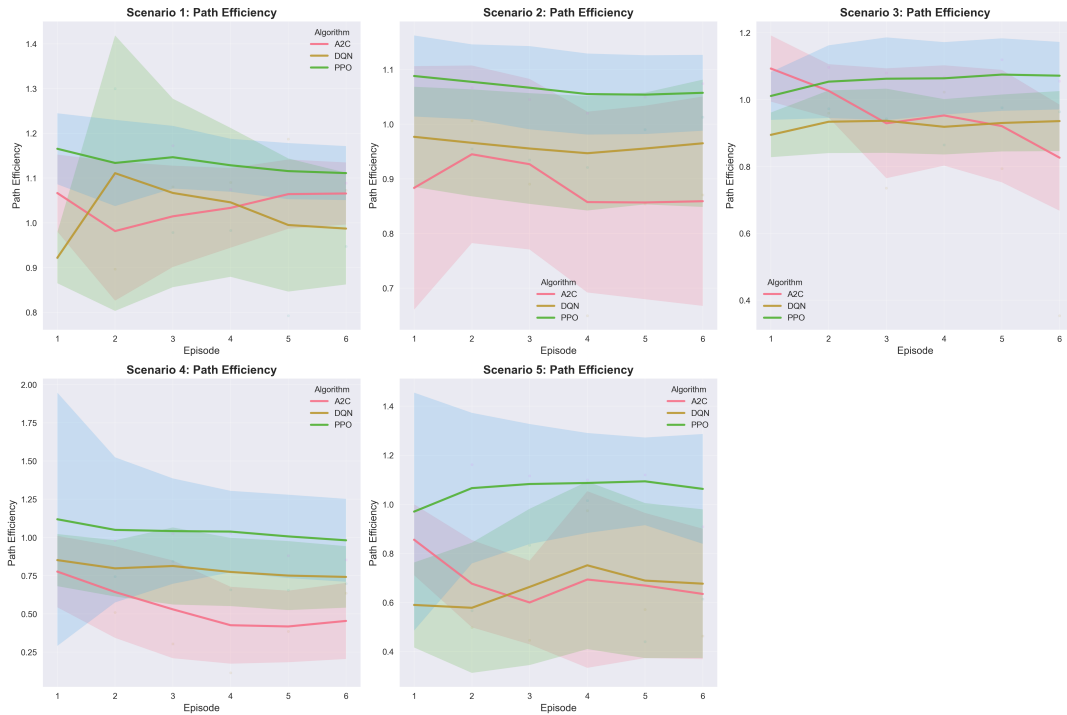


Figure 5.14: Path efficiency across all scenarios, where values near 1.0 indicate optimal navigation. Values slightly above 1.0 may occur due to reward structure design when agents approach meeting points.

The spatial efficiency analysis (Figure 5.14) reveals that PPO and DQN consistently achieve near-optimal path efficiency across all scenarios, with values clustering around 1.0 that indicate direct, efficient navigation strategies. This performance demonstrates that these algorithms successfully learn to balance individual path optimization with collective coordination requirements, avoiding the excessive detours and inefficient exploration patterns that can emerge in multi-agent learning environments.

A2C exhibits greater variability in path efficiency, particularly in scenarios with complex spatial constraints and partial observability. The algorithm's tendency toward longer, less efficient paths suggests difficulty in learning optimal spatial navigation strategies under coordination pressure, potentially reflecting instabilities in the learning process that prevent convergence to efficient movement policies.

Occasional path efficiency values exceeding 1.0 across all algorithms result from the specific reward structure implementation, where agents approaching meeting points within threshold distances experience velocity reduction and distance-based reward adjustments. This design choice ensures precise goal achievement and prevents overshooting behaviors, though it can produce path efficiency measurements slightly above theoretical optimum in cases where agents approach targets from suboptimal angles.

The spatial efficiency results provide important insights for real-world deployment, as efficient navigation directly impacts evacuation time and resource utilization. The superior performance of PPO and DQN in maintaining optimal spatial efficiency while coordinating with other agents suggests their suitability for deployment in environments where space utilization and movement efficiency are critical factors.

5.5.4 Safety Performance and Collision Analysis

Safety performance represents a fundamental requirement for evacuation systems, where collision avoidance and smooth traffic flow directly impact both individual and collective evacuation success. The collision analysis examines how different algorithms balance exploration, goal pursuit, and safety constraints across scenarios with varying spatial complexity and coordination demands.

The collision analysis reveals distinct algorithmic approaches to safety management that reflect underlying differences in learning paradigms and policy optimization strategies. DQN consistently achieves the lowest collision rates across scenarios with obstacles, demonstrating the effectiveness of its conservative navigation policies in maintaining safety margins while pursuing evacuation objectives. This safety-oriented behavior stems from DQN's value-based learning approach, which tends to avoid actions associated with negative outcomes and develops risk-averse policies that prioritize collision avoidance.

PPO maintains moderate collision rates while achieving superior overall performance across other metrics, suggesting an effective balance between safety consciousness and goal-directed behavior. The algorithm's ability to learn coordinated movement strategies that minimize conflicts while maintaining evacuation efficiency demonstrates the effectiveness of its policy gradient approach in handling the complex trade-offs inherent in multi-agent coordination tasks.

A2C exhibits the highest collision rates, particularly in scenarios with dense obstacle configurations, indicating difficulty in learning effective collision avoidance strategies under spatial constraints. This pattern suggests that A2C's learning dynamics may be less suited to environments where precise spatial coordination and safety management are critical for task success.

The differential collision performance across algorithms provides crucial insights for real-world deployment considerations. While aggressive navigation strategies may achieve faster evacuation times, the associated increase in collision risk could compromise safety in physical evacuation scenarios where collisions represent serious safety hazards rather than mere performance penalties.

5.5.5 Integrated Performance Assessment

The comprehensive evaluation across multiple performance dimensions reveals that algorithmic selection for evacuation applications requires careful consideration of the trade-offs between reliability, efficiency, and safety performance. No single algorithm optimizes all metrics simultaneously, necessitating deployment decisions based on specific operational priorities and environmental characteristics.

PPO emerges as the most balanced and robust choice for evacuation applications, achieving perfect reliability, superior temporal efficiency, optimal spatial navigation, and reasonable safety performance across all tested scenarios. This comprehensive performance profile, combined with its demonstrated stability under increasing environmental complexity, establishes PPO as the preferred algorithmic choice for real-world evacuation systems where consistent performance across diverse conditions is paramount.

DQN presents a compelling alternative for applications where safety performance takes precedence over temporal efficiency, achieving excellent collision avoidance while maintaining reasonable performance across other metrics. However, its susceptibility to catastrophic forgetting in complex scenarios and reduced reliability under partial observability limit its applicability to simpler evacuation environments.

A2C demonstrates adequate performance in basic scenarios but exhibits concerning degradation across multiple metrics as environmental complexity increases. This pattern suggests limited suitability for real-world evacuation applications where environmental conditions may be unpredictable and system robustness is essential.

The multidimensional evaluation framework developed in this work provides a foundation for rigorous assessment of evacuation system performance and establishes benchmarks for future research in intelligent evacuation technologies. The insights gained from this comprehensive analysis inform both algorithmic selection decisions and system design considerations for practical deployment in safety-critical evacuation scenarios.

5.6 Comparison with Traditional Approaches

This section presents a critical analysis of Multi-Agent Reinforcement Learning approaches compared to traditional evacuation modeling methods, drawing from the experimental evidence presented in previous sections and establishing a comprehensive framework for understanding the relative merits and limitations of each paradigm. The comparison is grounded in empirical observations from the five evacuation scenarios and provides insights into the practical deployment considerations for real-world evacuation systems.

5.6.1 Advantages of MARL

The experimental results demonstrate several fundamental advantages of Multi-Agent Reinforcement Learning approaches over traditional rule-based and physics-based evacuation models, par-

ticularly in scenarios requiring adaptive coordination and dynamic decision-making under uncertainty.

5.6.1.1 Adaptability vs. Fixed Rules

One of the most significant advantages observed in the experimental evaluation is MARL’s capacity for adaptive behavior in response to environmental changes and unexpected conditions. Traditional evacuation models, including Social Force Models and Agent-Based Models with pre-determined behavioral rules, operate under fixed decision-making frameworks that cannot evolve based on experience or environmental feedback. In contrast, the MARL agents demonstrated remarkable adaptability across the five experimental scenarios, with policies learned in simpler environments successfully transferring to more complex configurations without requiring manual recalibration.

The experimental evidence for this adaptability is particularly compelling in the progression from Scenario 1 to Scenario 5. While traditional approaches would require manual adjustment of parameters, rule modifications, or complete model redesign to handle the introduction of obstacles, partial observability, and communication requirements, the MARL agents adapted their strategies through the learning process itself. PPO, in particular, maintained consistent performance across all scenarios, demonstrating that learned policies can generalize to environmental variations that would challenge fixed-rule systems.

Furthermore, the generalization capability observed when agents trained with 6 agents successfully coordinated with larger populations illustrates a level of adaptability that is difficult to achieve with traditional approaches. Classical models typically require explicit programming for different population sizes and density configurations, whereas the MARL agents learned coordination principles that remained effective across varying agent numbers.

5.6.1.2 Emergent Coordination vs. Programmed Behavior

The experimental results reveal a fundamental distinction between the emergent coordination behaviors observed in MARL systems and the explicitly programmed coordination mechanisms typical of traditional approaches. The spontaneous queue formation observed in Scenario 2, where agents naturally organized into single-file formations at bottlenecks without explicit queue-forming rules, exemplifies the power of emergent coordination.

Traditional evacuation models typically implement coordination through predetermined behavioral rules or centralized optimization algorithms. For example, Social Force Models rely on attractive and repulsive forces with fixed parameters, while rule-based Agent-Based Models use if-then logic structures that must be manually designed for each coordination scenario. These approaches struggle to capture the nuanced, context-dependent coordination behaviors that emerge naturally in MARL systems.

The coordination index results presented in Section 5.5 demonstrate that MARL agents achieve consistently high levels of spatial coordination across all scenarios and algorithms, with coordination indices typically exceeding 0.98. This level of coordination emerges without explicit programming and adapts automatically to different environmental constraints. Traditional approaches would require separate coordination algorithms for each scenario type, increasing system complexity and reducing flexibility.

Moreover, the communication-enabled scenarios (4 and 5) showcase emergent information-sharing behaviors that would be challenging to implement in traditional systems. The high communication efficiency values observed indicate that agents learned not only when to share information but also how to integrate received information into their decision-making processes, creating sophisticated coordination protocols that emerged from the learning process rather than explicit design.

5.6.1.3 Performance in Dynamic Environments

The experimental evaluation demonstrates MARL's superior performance in dynamic environments where conditions change during execution or where multiple competing objectives must be balanced simultaneously. Traditional evacuation models typically assume static or slowly changing environments and struggle to adapt when faced with unexpected obstacles, blocked exits, or varying crowd densities.

The performance hierarchy observed across scenarios, with PPO consistently achieving superior results, followed by A2C and DQN, illustrates how different MARL algorithms handle dynamic complexity. Even the least robust algorithm (DQN) demonstrated periods of effective coordination before encountering stability issues in the most complex scenario. This suggests that MARL approaches, when properly implemented, can handle dynamic conditions that would overwhelm traditional fixed-rule systems.

The collision analysis presented in Section 5.4 provides quantitative evidence of MARL's effectiveness in dynamic environments. The ability of agents to learn collision avoidance strategies that adapt to different obstacle configurations and agent densities represents a significant advantage over traditional approaches, which typically rely on predetermined safety margins and fixed avoidance behaviors.

Additionally, the temporal efficiency results demonstrate that MARL agents learn to optimize evacuation times while maintaining safety constraints, achieving a dynamic balance that traditional approaches struggle to maintain across varying conditions. The consistently low evacuation times achieved by PPO and DQN across all scenarios indicate that learned policies can optimize multiple objectives simultaneously without requiring manual parameter tuning for each new environment.

5.6.2 Identified Limitations

Despite the demonstrated advantages, the experimental evaluation also reveals several significant limitations of MARL approaches that must be considered when evaluating their suitability for

real-world evacuation applications.

5.6.2.1 Computational Complexity

The most immediate limitation observed in the experimental framework concerns the substantial computational requirements for training effective MARL policies. The 700,000 training steps required across all scenarios represent a significant computational investment that traditional approaches do not require. Classical evacuation models, once calibrated, can be executed immediately without the extensive training phase that MARL demands.

The computational complexity extends beyond training to include the hyperparameter optimization process, particularly evident in the DQN implementation where scenario-specific parameter tuning was necessary to achieve reasonable performance. The detailed hyperparameter specifications provided in Appendix A illustrate the complexity of configuring MARL algorithms for optimal performance, requiring expertise and computational resources that may not be available in all deployment contexts.

Furthermore, the memory requirements for storing experience replay buffers (particularly for DQN) and the computational overhead of neural network inference during execution represent ongoing costs that traditional rule-based systems avoid. While modern hardware can accommodate these requirements, the computational burden may limit the scalability of MARL approaches to very large-scale evacuation scenarios or resource-constrained deployment environments.

The parallel training infrastructure required to achieve reasonable training times also represents a barrier to adoption, as traditional approaches can be implemented and tested on standard computing hardware without specialized machine learning infrastructure.

5.6.2.2 Extensive Training Requirements

The experimental results demonstrate that effective MARL policies require extensive training across diverse scenarios to achieve robust performance. The learning curves presented in Section 5.2.1 show that convergence to stable policies often requires hundreds of thousands of training steps, with some algorithms continuing to improve throughout the entire training window.

This extensive training requirement creates several practical challenges for real-world deployment. First, the need for comprehensive scenario coverage during training means that MARL systems must be exposed to a wide range of environmental conditions, agent populations, and obstacle configurations to develop robust policies. Traditional approaches, in contrast, can be deployed immediately once their parameters are calibrated for a specific environment.

Second, the training process requires careful monitoring and intervention, as evidenced by the catastrophic forgetting phenomenon observed in DQN during Scenario 5. This instability suggests that MARL training is not a “set-and-forget” process but requires ongoing supervision and potential intervention to maintain performance quality.

The domain expertise required to design appropriate reward functions, select suitable algorithms, and configure training parameters represents another barrier to adoption. Traditional approaches, while requiring domain knowledge for calibration, typically involve more intuitive parameter adjustments that can be understood and implemented by evacuation planning professionals without extensive machine learning expertise.

Additionally, the need for scenario-specific training means that MARL policies may not transfer effectively to environments significantly different from those encountered during training, requiring retraining or fine-tuning for new deployment contexts.

5.6.2.3 Hyperparameter Sensitivity (Especially DQN)

The experimental evaluation reveals significant sensitivity to hyperparameter configuration, particularly pronounced in the DQN algorithm but also evident in A2C and, to a lesser extent, PPO. This sensitivity represents a critical limitation for practical deployment, as small changes in algorithm configuration can dramatically impact performance.

The DQN results illustrate this challenge most clearly, with the algorithm requiring different hyperparameter configurations for different scenarios to achieve reasonable performance. The buffer size reductions, learning rate adjustments, and exploration parameter modifications detailed in Appendix A demonstrate the extent of tuning required to maintain stable learning across varying environmental complexities.

The catastrophic forgetting phenomenon observed in DQN during Scenario 5 represents an extreme manifestation of this sensitivity, where the algorithm's performance collapsed dramatically despite previous stable learning. This instability highlights the fragility of value-based approaches in complex multi-agent settings and suggests that hyperparameter sensitivity can lead to unpredictable failures in deployment scenarios.

Even the more robust algorithms showed sensitivity to reward function parameters, as evidenced by the pathological behavior observed when PPO was configured with higher collision penalties. This sensitivity to reward design parameters indicates that MARL approaches require careful calibration not only of algorithmic parameters but also of the incentive structures that guide learning.

The hyperparameter sensitivity challenge is compounded by the lack of theoretical guidance for parameter selection in multi-agent settings. While single-agent reinforcement learning provides some theoretical foundations for parameter choices, the non-stationary environment created by multiple learning agents makes parameter selection largely empirical, requiring extensive experimentation and validation.

Traditional approaches, while still requiring parameter calibration, typically involve parameters with clear physical or behavioral interpretations that can be set based on domain knowledge or empirical observations from evacuation drills and historical data. The abstract nature of many MARL hyperparameters makes them less intuitive to configure and more difficult to validate against real-world expectations.

5.6.3 Implications for Practical Deployment

The comparative analysis reveals that MARL and traditional approaches each offer distinct advantages and face unique challenges, suggesting that the optimal choice depends on specific deployment requirements, available resources, and operational constraints.

MARL approaches demonstrate clear superiority in scenarios requiring adaptive coordination, handling of unexpected conditions, and optimization across multiple competing objectives. The experimental evidence supports their deployment in complex, dynamic environments where traditional approaches would require extensive manual reconfiguration or fail to achieve adequate performance.

However, the computational requirements, training complexity, and hyperparameter sensitivity of MARL approaches may limit their applicability in resource-constrained environments or scenarios where immediate deployment is required without extensive development and testing phases.

The most promising deployment strategy may involve hybrid approaches that combine the adaptability of MARL with the reliability and interpretability of traditional methods, using MARL for complex coordination tasks while relying on traditional approaches for basic navigation and safety constraints. This combination could leverage the strengths of both paradigms while mitigating their respective limitations.

Future research should focus on developing more robust MARL algorithms with reduced hyperparameter sensitivity, more efficient training procedures, and better theoretical foundations for parameter selection in multi-agent settings. Additionally, the development of transfer learning techniques that enable MARL policies to generalize across different environments without extensive retraining would significantly enhance their practical applicability.

5.7 Summary

The comprehensive experimental evaluation presented in this chapter provides definitive answers to the three research questions that guided this investigation, while simultaneously demonstrating the practical viability of Multi-Agent Reinforcement Learning approaches for emergency evacuation scenarios. The empirical evidence establishes a clear foundation for understanding how MARL can be effectively applied to develop sophisticated evacuation strategies that surpass traditional approaches in adaptability, coordination effectiveness, and scalability.

RQ1: How can Multi-Agent Reinforcement Learning (MARL) be applied to develop effective decentralized evacuation strategies in dynamic and partially observable environments?

The investigation demonstrates that MARL can be effectively applied through a multi-layered approach that combines implicit and explicit coordination mechanisms. The experimental results reveal that agents spontaneously develop coordinated behaviors such as queue formation at bottlenecks in Scenario 2, without explicit programming of these strategies. This emergent coordination

stems from the optimization of individual reward functions that collectively promote group objectives, enabling agents to anticipate others' actions and adjust their trajectories to avoid collisions while forming orderly movement patterns. In environments with partial observability, such as Scenarios 4 and 5, inter-agent communication becomes essential, allowing agents to share critical information about meeting point locations and overcome the limitations imposed by restricted observation ranges.

The framework's effectiveness relies on a multi-component reward structure that integrates temporal penalties, collision avoidance incentives, goal-directed rewards, and scenario-specific bonuses to promote desired coordination behaviors. The algorithms demonstrate remarkable adaptability across varying environmental complexities, with PPO maintaining stable performance even when spatial constraints, communication requirements, and partial observability are simultaneously present. This adaptability manifests through policies that transfer effectively between scenarios of increasing complexity without requiring manual recalibration, suggesting that agents learn fundamental coordination principles rather than memorizing specific configurations. The decentralized nature of the approach ensures robustness against single points of failure while enabling scalable deployment across different agent populations and environmental configurations.

RQ2: Which MARL algorithms and reward shaping techniques most effectively minimize evacuation time and congestion in multi-agent evacuation scenarios?

The comparative analysis establishes a clear algorithmic hierarchy with Proximal Policy Optimization (PPO) emerging as the most robust and effective algorithm across all evacuation scenarios. PPO consistently achieves perfect evacuation success rates, superior temporal efficiency, and optimal spatial navigation while maintaining reasonable safety performance throughout all tested conditions. The algorithm's clipped surrogate objective function proves particularly well-suited to the non-stationary environment created by multiple simultaneously learning agents, preventing destructive policy updates that can destabilize coordination strategies. In contrast, Advantage Actor-Critic (A2C) demonstrates competent performance in simpler environments but exhibits progressive decline as complexity increases, with success rates declining to approximately 0.83 in Scenarios 4 and 5, while Deep Q-Networks (DQN) experiences catastrophic forgetting around the 500,000 training step mark in Scenario 5, illustrating structural limitations in handling complex multi-agent coordination tasks.

Regarding reward shaping techniques, the investigation reveals that algorithm-specific calibration is crucial for optimal performance, with A2C requiring a collision penalty of 0.3 while PPO and DQN achieve better results with a penalty of 0.1. The multi-component reward structure proves most effective when it balances temporal incentives, safety constraints, and goal-directed objectives without creating pathological behaviors such as excessive risk aversion. PPO with higher collision penalties exhibits counterproductive stationary behavior, avoiding necessary navigation through corridors, which demonstrates the critical importance of reward calibration matched to algorithmic characteristics. The optimal reward design successfully promotes emergent coordination behaviors including spontaneous queue formation and synchronized movement

patterns, while maintaining the flexibility necessary for agents to adapt their strategies across different environmental configurations and complexity levels.

RQ3: How do MARL-based evacuation strategies compare to traditional heuristic or rule-based methods in terms of adaptability, scalability, and coordination efficiency?

MARL-based evacuation strategies demonstrate significant advantages over traditional approaches across all evaluated dimensions, particularly in adaptability and coordination efficiency. The experimental evidence reveals that learned policies transfer effectively between scenarios of increasing complexity without requiring manual recalibration, contrasting sharply with traditional Social Force Models and rule-based Agent-Based Models that necessitate parameter adjustments or complete redesign when environmental conditions change. The spontaneous emergence of coordination behaviors such as queue formation and synchronized movement represents a fundamental advantage over traditional approaches that rely on explicitly programmed coordination mechanisms. The coordination index consistently exceeds 0.98 across all scenarios and algorithms, indicating robust spatial organization that emerges naturally rather than through predetermined behavioral rules, while the generalization capability observed when agents trained with 6 agents successfully coordinate with larger populations demonstrates learning of fundamental coordination principles rather than configuration-specific behaviors.

In terms of scalability, PPO and A2C maintain high performance when evaluated with agent populations larger than those used during training, suggesting that the learned coordination strategies capture essential principles of spatial coordination and collision avoidance that remain applicable regardless of population size. However, MARL approaches face significant limitations including substantial computational requirements for training, with 700,000 training steps representing a considerable investment compared to traditional models that can be executed immediately once calibrated. The hyperparameter sensitivity, particularly pronounced in DQN, and the extensive training requirements across diverse scenarios create barriers to deployment that traditional approaches avoid. Despite these limitations, the experimental evidence supports the deployment of MARL approaches, particularly PPO-based systems, in complex evacuation scenarios where environmental conditions are unpredictable and coordination demands exceed the capabilities of traditional rule-based systems, while suggesting that optimal deployment strategies likely involve hybrid approaches that leverage MARL's coordination capabilities for complex scenarios while maintaining traditional methods for basic navigation and safety constraints.

The synthesis of these findings establishes that MARL represents a paradigm shift in the development of evacuation strategies, offering unprecedented coordination capabilities and environmental adaptability at the cost of increased computational complexity and training requirements. The experimental evidence supports the deployment of MARL approaches, particularly PPO-based systems, in complex evacuation scenarios where environmental conditions are unpredictable and coordination demands exceed the capabilities of traditional rule-based systems, while suggesting that the optimal deployment strategy likely involves hybrid approaches that leverage

MARL's coordination capabilities for complex scenarios while maintaining traditional methods for basic navigation and safety constraints, combining the strengths of both paradigms while mitigating their respective limitations.

Chapter 6

Conclusions

This chapter synthesizes the principal contributions of the dissertation and outlines promising directions for future research in Multi-Agent Reinforcement Learning (MARL) applied to emergency evacuation scenarios. The first section 6.1 presents a comprehensive summary of the main advances achieved. The second section, 6.2, identifies key limitations of the current work and proposes a roadmap for advancing the field.

6.1 Main Contributions

This dissertation advances the field of Multi-Agent Reinforcement Learning (MARL) applied to emergency evacuation scenarios by integrating both theoretical and experimental insights, particularly those derived from the comprehensive analyses presented in Chapter 5. The work begins by introducing a robust MARL framework specifically tailored for evacuation contexts, enabling agents to learn coordinated strategies through decentralized decision-making. This framework effectively addresses the challenge of balancing individual agent objectives with the collective efficiency required for successful evacuations.

A central contribution of this research lies in the demonstration of how agents can develop both implicit and explicit coordination mechanisms. Through extensive experimentation, it was observed that agents are capable of achieving effective coordination without the need for explicit communication, relying instead on learned behavioral patterns and environmental cues. However, as the complexity of the environment increases, particularly in scenarios characterized by partial observability and the presence of obstacles, the integration of explicit communication mechanisms becomes essential. The results clearly show that enabling communication among agents leads to improved evacuation rates and reduced congestion, especially in the most challenging scenarios, thus highlighting the synergistic role of implicit and explicit coordination.

The dissertation also presents the development and validation of scalable training algorithms, namely PPO, DQN, and A2C, each evaluated across a spectrum of increasingly complex evacuation scenarios. The experimental findings reveal that PPO and DQN consistently outperform A2C in terms of robustness, efficiency, and adaptability. Notably, PPO maintains high performance and

stability even in the most demanding settings, while DQN, despite its strengths, exhibits some instability under high complexity, including phenomena such as catastrophic forgetting. These results underscore the importance of algorithm selection and the need for stability-inducing mechanisms in MARL applications.

A further significant contribution is the introduction of a multidimensional evaluation framework, which encompasses metrics such as evacuation rate, evacuation time, path efficiency, collision rates, coordination index, and communication efficiency. This comprehensive set of metrics enables a rigorous and nuanced comparison of MARL algorithms, providing quantitative evidence of the benefits conferred by learned coordination and communication strategies. The analysis demonstrates that as environmental complexity increases, the burden on coordination and communication mechanisms becomes more pronounced, necessitating sophisticated approaches to maintain high levels of performance.

The research also investigates the impact of environmental complexity, showing that the introduction of obstacles, bottlenecks, and partial observability significantly challenges agent learning and coordination. The findings highlight that the combination of emergent coordination and explicit communication is crucial for overcoming these challenges, as agents must not only navigate complex spatial constraints but also share critical information to adapt their strategies dynamically.

Insights into the generalization capabilities of learned evacuation policies are also provided, with the study identifying key factors that influence policy transferability across different layouts and scenarios. Guidelines are established for developing robust and adaptable evacuation strategies, emphasizing the importance of designing policies that can generalize beyond the specific conditions encountered during training.

Moreover, the dissertation offers a thorough comparison between MARL-based approaches and traditional rule-based strategies. The experimental validation demonstrates that MARL methods achieve significant improvements in evacuation times, reduce congestion, and enhance overall safety outcomes, thereby establishing their superiority in dynamic and uncertain environments.

Finally, the practical implications and limitations of the proposed MARL framework are discussed. While the simulation results are promising, it is acknowledged that the environment is simplified, operating in a two-dimensional space with homogeneous agents and without modeling psychosocial factors. Consequently, the direct transfer of learned policies to real-world settings requires further validation and integration with empirical data. Despite these limitations, the contributions of this work collectively advance the state of the art in MARL for emergency management, laying a solid foundation for the development of intelligent evacuation systems capable of enhancing public safety in real-world scenarios.

6.2 Future Work

The findings presented in this dissertation establish a solid foundation for Multi-Agent Reinforcement Learning applications in emergency evacuation scenarios. However, several promising research directions emerge from the current limitations and could significantly extend the practical

applicability and theoretical understanding of MARL-based evacuation systems.

The complexity of the dynamic environment represents a critical area for future investigation. Current scenarios assume static obstacles and fixed exit configurations. However, real evacuation scenarios often involve moving obstacles such as falling debris, spreading fires, or collapsing structures that dynamically alter available paths. Future work should investigate environments where agents must continuously adapt their evacuation strategies and develop more robust coordination mechanisms under rapidly changing conditions. Furthermore, real evacuation scenarios often involve agents moving in opposite directions, such as emergency responders entering while evacuees exit, or agents from different building sections converging. Investigating how MARL algorithms handle conflicting movement patterns and develop coordination protocols for bi-directional flow represents a critical research gap. Extending the current 2D framework to multistory buildings with stairs, elevators, and vertical movement constraints would significantly increase the realism of learned evacuation policies and practical applicability.

Heterogeneous agent populations offer another promising research direction. Future research should model agents with varying mobility constraints, including wheelchair users, elderly individuals, or injured persons with reduced movement speed. This heterogeneity would require developing inclusive evacuation strategies that account for vulnerable populations. Investigating scenarios with designated leaders, emergency responders, or guides could reveal how hierarchical coordination emerges in MARL systems and whether explicit role assignment improves evacuation efficiency. Incorporating psychological factors such as panic, stress-induced decision-making, and herding behavior would enhance the realism of agent interactions and test the robustness of learned coordination strategies under high-stress conditions.

Advanced communication and coordination mechanisms deserve to be explored further. While this work focused on implicit coordination, future research should explore explicit communication mechanisms where agents learn to share critical information about exit accessibility, hazard locations, or optimal routes. This could significantly improve coordination in scenarios with limited visibility or complex spatial layouts. Developing hierarchical coordination architectures where high-level coordinators manage groups of agents could address scalability challenges and improve performance in large-population scenarios. Investigating how MARL agents can effectively coordinate with human evacuees or emergency responders is crucial to real-world deployment.

Robustness and generalization remain significant challenges that require attention. Developing methods to transfer learned evacuation policies across different building layouts, architectural styles, and emergency types would significantly enhance the practical utility of MARL-based systems. Testing learned policies against worst-case scenarios, including equipment failures, blocked exits, or malicious agents, would validate the robustness of coordination mechanisms under extreme conditions. Conducting controlled experiments with human participants in evacuation drills or virtual reality environments would provide crucial validation of the coordination behaviors learned in simulation.

Scalability and computational efficiency improvements are essential for practical deployment. Investigating MARL approaches for evacuation scenarios with hundreds or thousands of agents

would require developing more efficient training algorithms and coordination mechanisms that scale beyond current limitations. Exploring distributed training architectures that can leverage edge computing resources during emergencies could enable real-time policy adaptation and deployment. Developing federated learning frameworks where multiple buildings or organizations can collaboratively train evacuation policies while preserving privacy and security constraints represents an important practical consideration.

Integration with Emergency Response Systems would bridge the gap between research and practice. Incorporating real-time data from IoT sensors, security cameras, and building management systems could enable dynamic policy adaptation based on actual emergency conditions. Developing interfaces between MARL evacuation systems and existing emergency response protocols could facilitate seamless integration with current emergency management practices. Investigating how MARL-based evacuation systems can meet safety regulations and certification requirements for deployment in real buildings represents a critical practical challenge.

Theoretical advances would strengthen the scientific foundation of MARL evacuation systems. Developing theoretical frameworks that provide convergence guarantees for MARL algorithms in evacuation scenarios would strengthen the reliability of learned policies. Advancing safe reinforcement learning techniques for evacuation scenarios, where policy failures can have life-threatening consequences, represents a crucial theoretical and practical challenge. Extending current approaches to handle multiple competing objectives, such as minimizing evacuation time while maximizing safety and ensuring equitable outcomes for all population groups, would enhance the practical applicability of these systems.

These research directions collectively represent a comprehensive roadmap for advancing MARL applications in emergency evacuation, potentially impacting public safety and emergency preparedness in increasingly complex urban environments. Integrating these advances could lead to intelligent evacuation systems capable of real-time adaptation, inclusive coordination, and robust performance under the most challenging emergency conditions. By pursuing these interconnected research paths, we can fully realize the potential of Multi-Agent Reinforcement Learning Approaches to Leverage Distributed Problem Solving in creating more effective, equitable, and resilient emergency response systems for our increasingly urbanized world.

References

- [1] Ibrahim H. Ahmed et al. “Deep Reinforcement Learning for Multi-Agent Interaction”. In: *AI Communications* 35.4 (Sept. 2022), pp. 357–368. ISSN: 0921-7126. DOI: [10.3233/AIC-220116](#).
- [2] Stefano V. Albrecht, Filippas Christianos, and Lukas Schäfer. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. Cambridge, Massachusetts: The MIT Press, 2024. ISBN: 978-0-262-04937-5.
- [3] João Emílio Almeida et al. “Serious Games for the Human Behaviour Analysis in Emergency Evacuation Scenarios”. In: *Cluster Computing* 20.1 (Mar. 2017), pp. 707–720. ISSN: 1573-7543. DOI: [10.1007/s10586-017-0765-z](#).
- [4] João Emilio Santos Carvalho de Almeida. “Serious Games as a Behaviour Elicitation Tool: Applications to Evacuation Scenarios”. PhD thesis. Porto, PT: Universidade do Porto, Faculdade de Engenharia, Jan. 2016.
- [5] Dario Amodei et al. *Concrete Problems in AI Safety*. July 2016. DOI: [10.48550/arXiv.1606.06565](#). arXiv: [1606.06565](#).
- [6] Kai Arulkumaran et al. “Deep Reinforcement Learning: A Brief Survey”. In: *IEEE Signal Processing Magazine* (2017). DOI: [10.1109/MSP.2017.2743240](#).
- [7] Kai Arulkumaran et al. “Deep Reinforcement Learning: A Brief Survey”. In: *IEEE Signal Processing Magazine* 34.6 (Nov. 2017), pp. 26–38. ISSN: 1558-0792. DOI: [10.1109/MSP.2017.2743240](#).
- [8] Matteo Bettini et al. *VMAS: A Vectorized Multi-Agent Simulator for Collective Robot Learning*. 2022. DOI: [10.48550/arXiv.1809.02627](#). arXiv: [2207.03530](#).
- [9] Mohcine Chraïbi et al. “Jamming Transitions in Force-Based Models for Pedestrian Dynamics”. In: *Physical Review E* 92.4 (Oct. 2015), p. 042809. DOI: [10.1103/PhysRevE.92.042809](#).
- [10] Petros Christodoulou. *Soft Actor-Critic for Discrete Action Settings*. 2019. DOI: [10.48550/arXiv.1910.07207](#). arXiv: [1910.07207](#).
- [11] Geng Cui, Libi Luo, and Xiaoping Shi. “A Data Driven Approach to Simulate Pedestrian Competitiveness Using the Social Force Model”. In: *Collective Dynamics* 6 (2021), pp. 1–15. DOI: [10.17815/CD.2021.118](#).
- [12] B. Fan et al. “Attention-Based Multiagent Graph Reinforcement Learning for Service Restoration”. In: *IEEE Transactions on Artificial Intelligence* 5.5 (2024), pp. 2163–2178. DOI: [10.1109/TAI.2023.3314395](#).
- [13] Jakob Foerster et al. “Counterfactual Multi-Agent Policy Gradients”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. Apr. 2018. DOI: [10.1609/aaai.v32i1.11794](#).

- [14] Jakob Foerster et al. *Counterfactual Multi-Agent Policy Gradients*. Dec. 2024. DOI: [10 . 48550/arXiv.1705.08926](https://doi.org/10.48550/arXiv.1705.08926).
- [15] Jakob N. Foerster et al. “Learning to Communicate with Deep Multi-Agent Reinforcement Learning”. In: *Advances in Neural Information Processing Systems 29 (NeurIPS 2016)*. Vol. 29. Curran Associates, Inc., 2016, pp. 2137–2145. ISBN: 9781510838819.
- [16] Neil De La Fuente and Daniel A. Vidal Guerra. *A Comparative Study of Deep Reinforcement Learning Models: DQN vs PPO vs A2C*. 2024. DOI: [10 . 48550/arXiv.2407 . 14151](https://doi.org/10.48550/arXiv.2407.14151). arXiv: [2407.14151](https://arxiv.org/abs/2407.14151).
- [17] Tuomas Haarnoja et al. “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. PMLR, July 2018, pp. 1861–1870.
- [18] Dirk Helbing. “Traffic and Related Self-Driven Many-Particle Systems”. In: *Reviews of Modern Physics* 73.4 (Dec. 2001), pp. 1067–1141. DOI: [10 . 1103/RevModPhys.73 . 1067](https://doi.org/10.1103/RevModPhys.73.1067).
- [19] Dirk Helbing, Illés Farkas, and Tamás Vicsek. “Simulating Dynamical Features of Escape Panic”. In: *Nature* 407.6803 (Sept. 2000), pp. 487–490. ISSN: 1476-4687. DOI: [10.1038/35035023](https://doi.org/10.1038/35035023).
- [20] Dirk Helbing and Anders Johansson. “Pedestrian, Crowd, and Evacuation Dynamics”. In: *Encyclopedia of Complexity and Systems Science*. Ed. by Robert A. Meyers. New York, NY: Springer New York, 2012, pp. 6476–6495. ISBN: 978-3-642-27737-5. DOI: [10.1007/978-3-642-27737-5_382-5](https://doi.org/10.1007/978-3-642-27737-5_382-5).
- [21] Dirk Helbing and Péter Molnár. “Social force model for pedestrian dynamics”. In: *Physical Review E* 51.5 (May 1995), pp. 4282–4286. DOI: [10.1103/PhysRevE.51.4282](https://doi.org/10.1103/PhysRevE.51.4282).
- [22] Peter Henderson et al. “Deep Reinforcement Learning That Matters”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. Apr. 2018. DOI: [10.1609/aaai . v32i1.11694](https://doi.org/10.1609/aaai.v32i1.11694).
- [23] Dom Huh and Prasant Mohapatra. *Multi-agent Reinforcement Learning: A Comprehensive Survey*. 2024. DOI: [10.48550/arXiv.2312.10256](https://doi.org/10.48550/arXiv.2312.10256). arXiv: [2312.10256](https://arxiv.org/abs/2312.10256).
- [24] Shariq Iqbal and Fei Sha. “Actor-Attention-Critic for Multi-Agent Reinforcement Learning”. In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, May 2019, pp. 2961–2970.
- [25] Jiechuan Jiang and Zongqing Lu. “Learning Attentional Communication for Multi-Agent Cooperation”. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS’18. Red Hook, NY, USA: Curran Associates Inc., Dec. 2018, pp. 7265–7275.
- [26] Jiechuan Jiang et al. “Graph Convolutional Reinforcement Learning”. In: *ICLR 2020: International Conference on Learning Representations*. Virtual Conference, Sept. 2019.
- [27] Arthur Juliani et al. *Unity: A General Platform for Intelligent Agents*. 2020. DOI: [10 . 48550/arXiv.1809.02627](https://doi.org/10.48550/arXiv.1809.02627). arXiv: [1809.02627](https://arxiv.org/abs/1809.02627).
- [28] Kirthevasan Kandasamy et al. “Tuning Hyperparameters without Grad Students: Scalable and Robust Bayesian Optimisation with Dragonfly”. In: *Journal of Machine Learning Research* 21.81 (2020), pp. 1–27. ISSN: 1533-7928.

- [29] Daewoo Kim et al. *Learning to Schedule Communication in Multi-agent Reinforcement Learning*. Feb. 2019. DOI: [10.48550/arXiv.1902.01554](#). arXiv: [1902.01554](#).
- [30] Woojun Kim, Jongeui Park, and Youngchul Sung. “Communication in Multi-Agent Reinforcement Learning: Intention Sharing”. In: *ICLR 2021: International Conference on Learning Representations*. Vienna, Austria, Oct. 2020.
- [31] James Kirkpatrick et al. “Overcoming catastrophic forgetting in neural networks”. In: *Proceedings of the National Academy of Sciences* 114.13 (Mar. 2017), pp. 3521–3526. ISSN: 1091-6490. DOI: [10.1073/pnas.1611835114](#).
- [32] Landon Kraemer and Bikramjit Banerjee. “Multi-agent reinforcement learning as a rehearsal for decentralized planning”. In: *Neurocomputing* (2016). ISSN: 0925-2312. DOI: [10.1016/j.neucom.2016.01.031](#).
- [33] Elon Langbeheim et al. “Reasoning about crowd evacuations as emergent phenomena when using participatory computational models”. In: *Frontiers in Education* 8 (2023), p. 1137828. DOI: [10.3389/educ.2023.1137828](#).
- [34] Jordan Langham-Lopez, Sebastian M. Schmon, and Patrick Cannon. *High Performance Simulation for Scalable Multi-Agent Reinforcement Learning*. 2022. DOI: [10.48550/arXiv.2207.03945](#). arXiv: [2207.03945](#).
- [35] Gene Li. *Agnostic Reinforcement Learning: Foundations and Algorithms*. 2025. DOI: [10.48550/arXiv.2506.01884](#). arXiv: [2506.01884](#).
- [36] Meng-Lin Li, Shaofei Chen, and Jing Chen. “Adaptive Learning: A New Decentralized Reinforcement Learning Approach for Cooperative Multiagent Systems”. In: *IEEE Access* 8 (2020), pp. 99404–99421. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2020.2997899](#).
- [37] Eric Liang et al. “RLlib Flow: Distributed Reinforcement Learning Is a Dataflow Problem”. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 5506–5517.
- [38] Eric Liang et al. “RLlib: Abstractions for Distributed Reinforcement Learning”. In: *Proceedings of the 35th International Conference on Machine Learning*. PMLR, July 2018, pp. 3053–3062.
- [39] Kezhong Liu et al. “A Survey of Crowd Evacuation on Passenger Ships: Recent Advances and Future Challenges”. In: *Ocean Engineering* 263 (Nov. 2022), p. 112403. ISSN: 0029-8018. DOI: [10.1016/j.oceaneng.2022.112403](#).
- [40] Ryan Lowe et al. “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments”. In: *Advances in Neural Information Processing Systems (NIPS 2017)*. Ed. by I. Guyon et al. Vol. 30. Long Beach, California, USA: Curran Associates, Inc., 2017, pp. 6382–6393. ISBN: 978-1-5108-6096-4.
- [41] Volodymyr Mnih et al. “Asynchronous Methods for Deep Reinforcement Learning”. In: *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, June 2016, pp. 1928–1937.
- [42] Volodymyr Mnih et al. “Human-Level Control through Deep Reinforcement Learning”. In: *Nature* 518.7540 (Feb. 2015), pp. 529–533. ISSN: 1476-4687. DOI: [10.1038/nature14236](#).
- [43] Kevin Murphy. *Reinforcement Learning: An Overview*. 2025. DOI: [10.48550/arXiv.2412.05265](#). arXiv: [2412.05265](#).

- [44] Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. SpringerBriefs in Intelligent Systems, Artificial Intelligence, Multiagent Systems, and Cognitive Robotics. Cham: Springer, 2016. ISBN: 978-3-319-28927-4 978-3-319-28929-8. DOI: [10.1007/978-3-319-28929-8](https://doi.org/10.1007/978-3-319-28929-8).
- [45] Frans A. Oliehoek and Christopher Amato. “The Decentralized POMDP Framework”. In: *A Concise Introduction to Decentralized POMDPs*. Ed. by Frans A. Oliehoek and Christopher Amato. Cham: Springer International Publishing, 2016, pp. 11–32. ISBN: 978-3-319-28929-8. DOI: [10.1007/978-3-319-28929-8_2](https://doi.org/10.1007/978-3-319-28929-8_2).
- [46] Theodor Panayotov and Ivo Emanuilov. *Adaptive Routing Protocols for Determining Optimal Paths in AI Multi-Agent Systems: A Priority- and Learning-Enhanced Approach*. Mar. 2025. DOI: [10.48550/arXiv.2503.07686](https://doi.org/10.48550/arXiv.2503.07686). arXiv: [2503.07686](https://arxiv.org/abs/2503.07686).
- [47] Hyoshin Park, Dahai Liu, and Sirish Namilae. *Multi-Agent Reinforcement Learning-Based Pedestrian Dynamics Models for Emergency Evacuation*. Final Report CATM-2022-R5-NCAT. Greensboro, NC: Center for Advanced Transportation Mobility, Transportation Institute, June 2022, p. 80. URL: <https://rosap.ntl.bts.gov/view/dot/64672> (visited on 06/25/2025).
- [48] Nuria Pelechano and Norman Badler. “Modeling Crowd and Trained Leader Behavior during Building Evacuation”. In: *IEEE Computer Graphics and Applications* 26.6 (Nov. 2006), pp. 80–86. ISSN: 0272-1716. DOI: [10.1109/MCG.2006.133](https://doi.org/10.1109/MCG.2006.133).
- [49] Xue Bin Peng et al. “Sim-to-Real Transfer of Robotic Control with Dynamics Randomization”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, Australia: IEEE Press, May 2018, pp. 1–8. DOI: [10.1109/ICRA.2018.8460528](https://doi.org/10.1109/ICRA.2018.8460528).
- [50] Rafael Pina et al. *Fully Independent Communication in Multi-Agent Reinforcement Learning*. Mar. 2024. DOI: [10.48550/arXiv.2401.15059](https://doi.org/10.48550/arXiv.2401.15059). arXiv: [2401.15059](https://arxiv.org/abs/2401.15059) [cs].
- [51] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. 1st. USA: John Wiley & Sons, Inc., 1994. ISBN: 0471619779.
- [52] Antonin Raffin et al. “Stable-Baselines3: Reliable Reinforcement Learning Implementations”. In: *Journal of Machine Learning Research* 22.268 (2021), pp. 1–8. ISSN: 1533-7928.
- [53] Tabish Rashid et al. “Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning”. In: *The Journal of Machine Learning Research* 21.1 (Jan. 2020), 178:7234–178:7284. ISSN: 1532-4435.
- [54] John Schulman et al. *Proximal Policy Optimization Algorithms*. Aug. 2017. DOI: [10.48550/arXiv.1707.06347](https://doi.org/10.48550/arXiv.1707.06347). arXiv: [1707.06347](https://arxiv.org/abs/1707.06347).
- [55] Nirajan Shiwakoti et al. “Animal Dynamics Based Approach for Modeling Pedestrian Crowd Egress under Panic Conditions”. In: *Procedia - Social and Behavioral Sciences*. Papers Selected for the 19th International Symposium on Transportation and Traffic Theory 17 (Jan. 2011), pp. 438–461. ISSN: 1877-0428. DOI: [10.1016/j.sbspro.2011.04.526](https://doi.org/10.1016/j.sbspro.2011.04.526).
- [56] I. M. Sticco et al. “Beyond the faster-is-slower effect”. In: *Physical Review E* 96.5 (Nov. 2017). DOI: [10.1103/physreve.96.052303](https://doi.org/10.1103/physreve.96.052303).
- [57] Peter Sunehag et al. “Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward”. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, July 2018, pp. 2085–2087.

- [58] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. 2nd. Adaptive Computation and Machine Learning Series. Cambridge, Massachusetts: The MIT Press, 2018. ISBN: 978-0-262-03924-6.
- [59] Ardi Tampuu et al. *Multiagent Cooperation and Competition with Deep Reinforcement Learning*. 2017. DOI: [10.48550/arXiv.1511.08779](https://doi.org/10.48550/arXiv.1511.08779). arXiv: [1511.08779](https://arxiv.org/abs/1511.08779).
- [60] Yunhao Tang and Shipra Agrawal. *Discretizing Continuous Action Space for On-Policy Optimization*. 2020. DOI: [10.48550/arXiv.1901.10500](https://doi.org/10.48550/arXiv.1901.10500). arXiv: [1901.10500](https://arxiv.org/abs/1901.10500).
- [61] J. K. Terry et al. “PettingZoo: Gym for Multi-Agent Reinforcement Learning”. In: *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 15032–15043. ISBN: 978-1-7138-4539-3.
- [62] Martijn Van Otterlo and Marco Wiering. “Reinforcement learning and markov decision processes”. In: *Adaptation, learning, and optimization* 12 (2012), pp. 3–42. DOI: [10.1007/978-3-642-27645-3_1](https://doi.org/10.1007/978-3-642-27645-3_1).
- [63] Jiawei Wang et al. *Multi-Task Multi-Agent Shared Layers Are Universal Cognition of Multi-Agent Coordination*. Dec. 2023. DOI: [10.48550/arXiv.2312.15674](https://doi.org/10.48550/arXiv.2312.15674). arXiv: [2312.15674](https://arxiv.org/abs/2312.15674) [cs].
- [64] Tonghan Wang et al. “ROMA: Multi-agent reinforcement learning with emergent roles”. In: *Proceedings of the 37th International Conference on Machine Learning* (2020), pp. 9876–9886.
- [65] Annie Wong et al. *Deep Multiagent Reinforcement Learning: Challenges and Directions*. 2022. DOI: [10.48550/arXiv.2106.15691](https://doi.org/10.48550/arXiv.2106.15691). arXiv: [2106.15691](https://arxiv.org/abs/2106.15691).
- [66] Dong Xu et al. “Simulating Multi-Exit Evacuation Using Deep Reinforcement Learning”. In: *Transactions in GIS* 25.3 (2021), pp. 1542–1564. ISSN: 1467-9671. DOI: [10.1111/tgis.12738](https://doi.org/10.1111/tgis.12738).
- [67] Yaodong Yang et al. “Mean Field Multi-Agent Reinforcement Learning”. In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. PMLR, July 2018, pp. 5571–5580.
- [68] Yupeng Yang et al. *Multi-Agent Reinforcement Learning-Based Evacuation Models under Emergency*. Tech. rep. CATM-2022-R3-ERAU. Greensboro, NC: Center for Advanced Transportation Mobility, Transportation Institute, Jan. 2022. URL: <https://rosap.ntl.bts.gov/view/dot/64673> (visited on 06/25/2025).
- [69] Lei Yuan et al. *A Survey of Progress on Cooperative Multi-agent Reinforcement Learning in Open Environment*. 2023. DOI: [10.48550/arXiv.2312.01058](https://doi.org/10.48550/arXiv.2312.01058). arXiv: [2312.01058](https://arxiv.org/abs/2312.01058).
- [70] H. Zhang, D. Li, and Y. He. “The Effect of Different Types of Internal Rewards in Distributed Multi-Agent Deep Reinforcement Learning”. In: *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2019, pp. 2890–2895. DOI: [10.1109/ROBIO49542.2019.8961464](https://doi.org/10.1109/ROBIO49542.2019.8961464).
- [71] Kaiqing Zhang et al. “Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents”. In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. PMLR, July 2018, pp. 5872–5881.
- [72] Ziyang Zhang, Lingye Tan, and Robert L. K. Tiong. “Evacuation Path Optimization Algorithm for Grassland Fires Based on SAR Imagery and Intelligent Optimization”. In: *Frontiers in Environmental Science* 13 (Mar. 2025). ISSN: 2296-665X. DOI: [10.3389/fenvs.2025.1522933](https://doi.org/10.3389/fenvs.2025.1522933).

- [73] Yujing Zhou et al. *Real-Time Deep Reinforcement Learning for Evacuation Under Emergencies*. Tech. rep. CATM-2024-R1-ERAU. Greensboro, NC: Center for Advanced Transportation Mobility, Transportation Institute, Jan. 2024. URL: <https://rosap.nhtl.bts.gov/view/dot/77393> (visited on 06/25/2025).
- [74] Changxi Zhu, Mehdi Dastani, and Shihan Wang. *A Survey of Multi-Agent Deep Reinforcement Learning with Communication*. 2024. DOI: [10.48550/arXiv.2203.08975](https://doi.org/10.48550/arXiv.2203.08975). arXiv: [2203.08975](https://arxiv.org/abs/2203.08975).

Appendix A

Algorithm Hyperparameters

This appendix provides a comprehensive overview of all hyperparameters used in the Multi-Agent Reinforcement Learning algorithms implemented for the emergency evacuation scenarios. All algorithms use the default hyperparameters from the Stable-Baselines3 library, with only specific parameters explicitly configured.

A.1 General Training Configuration

Table A.1: Common training parameters across all algorithms

Parameter	Value	Description
Total timesteps	700,000	Maximum training steps
Device	CPU	Processamento em CPU
Verbose level	1	Training progress output
Vectorized environments	4	Parallel environment instances
CPU cores	1	Single-threaded processing
Random seed	0	Reproducibility seed
Max cycles	150	Maximum episode length

A.2 Algorithm Implementation Details

All algorithms are implemented using the Stable-Baselines3 library with minimal configuration. The following table shows the exact implementation used in the experiments:

Table A.2: Algorithm implementation configuration

Algorithm	Implementation	Specified Parameters
PPO	<code>PPO(PPOPolicy, env, verbose=1, device="cpu")</code>	Policy, verbose, device
A2C	<code>A2C(A2CPolicy, env, verbose=1, device="cpu")</code>	Policy, verbose, device
DQN	<code>DQN(DQNPolicy, env, verbose=1, device="cpu")</code>	Policy, verbose, device

A.2.1 Proximal Policy Optimization (PPO)

PPO uses the default Stable-Baselines3 configuration with the following parameters:

Table A.3: PPO default hyperparameters (Stable-Baselines3)

Hyperparameter	Default Value	Source	Description
Policy network	MlpPolicy	Specified	Multi-layer perceptron
Learning rate	3×10^{-4}	Default	Adam optimizer learning rate
Number of steps	2048	Default	Rollout buffer size
Batch size	64	Default	Mini-batch size for updates
Number of epochs	10	Default	PPO update epochs
Clip range	0.2	Default	Policy clipping parameter
Clip range (value)	None	Default	Value function clipping
Entropy coefficient	0.0	Default	Exploration bonus
Value function coefficient	0.5	Default	VF loss scaling
Max gradient norm	0.5	Default	Gradient clipping threshold
GAE lambda	0.95	Default	Generalized Advantage Estimation
Discount factor (γ)	0.99	Default	Future reward discount
Device	CPU	Specified	CPU processing
Verbose	1	Specified	Training progress output

A.2.2 Advantage Actor-Critic (A2C)

A2C implementation uses the default Stable-Baselines3 parameters:

Table A.4: A2C default hyperparameters (Stable-Baselines3)

Hyperparameter	Default Value	Source	Description
Policy network	MlpPolicy	Specified	Multi-layer perceptron
Learning rate	7×10^{-4}	Default	RMSprop optimizer learning rate
Number of steps	5	Default	Rollout length
Entropy coefficient	0.01	Default	Exploration-exploitation balance
Value function coefficient	0.25	Default	Critic loss weight
Max gradient norm	0.5	Default	Gradient clipping
Discount factor (γ)	0.99	Default	Future reward weighting
RMSprop epsilon	1×10^{-5}	Default	Numerical stability
RMSprop alpha	0.99	Default	Moving average decay
Use RMSprop	True	Default	Optimizer selection
Device	CPU	Specified	CPU processing
Verbose	1	Specified	Training progress output

A.2.3 A2C Hyperparameters for Communication Scenarios (4 and 5)

For scenarios 4 and 5, which involve inter-agent communication, A2C required specific hyperparameter tuning to enable effective learning of communication protocols and positive listening behaviors. The following customized parameters were used:

Table A.5: A2C hyperparameters optimized for communication scenarios 4 and 5.

Parameter	Scenarios 4 & 5 Value	Rationale
Learning rate	1×10^{-3}	Enhanced adaptation to communication
Number of steps	10	Capture longer communication sequences
Discount factor (γ)	0.99	Unchanged
GAE lambda	0.95	Unchanged
Entropy coefficient	0.01	Maintain exploration incentive
Value function coefficient	0.5	Increased critic loss weight

A.2.4 Deep Q-Network (DQN)

DQN implementation follows the default Stable-Baselines3 configuration:

Table A.6: DQN default hyperparameters (Stable-Baselines3)

Hyperparameter	Default Value	Source	Description
Policy network	MlpPolicy	Specified	Q-network architecture
Learning rate	1×10^{-4}	Default	Adam optimizer rate
Buffer size	1,000,000	Default	Experience replay capacity
Learning starts	50,000	Default	Warm-up period
Batch size	32	Default	Training batch size
Target update interval	10,000	Default	Target network sync frequency
Train frequency	4	Default	Training steps per environment step
Gradient steps	1	Default	Optimizer steps per training
Exploration fraction	0.1	Default	ϵ -greedy exploration schedule
Exploration initial ϵ	1.0	Default	Initial exploration rate
Exploration final ϵ	0.05	Default	Final exploration rate
Tau	1.0	Default	Soft update coefficient
Discount factor (γ)	0.99	Default	Future reward discount
Device	CPU	Specified	CPU processing
Verbose	1	Specified	Training progress output

A.2.5 DQN Hyperparameters for Scenario1

Table A.7: DQN hyperparameters for Scenario 1 multi-agent coordination.

Parameter	Default Value	Scenario 1 Value
Learning Rate	1e-4	1e-4
Buffer Size	100,000	50,000
Learning Starts	50,000	1,000
Target Update Interval	10,000	500
Exploration Fraction	0.1	0.3
Exploration Initial Eps	1.0	1.0
Exploration Final Eps	0.05	0.05
Train Frequency	4	4
Gradient Steps	1	1
Batch Size	32	64
Tau (Soft Update)	1.0	0.005
Buffer Reset Interval	N/A	25,000

A.2.6 DQN Hyperparameters for Scenarios 2 to 5

Table A.8: DQN hyperparameters optimized for coordination scenarios 2 to 5.

Parameter	Default Value	Scenarios 2 & 3 Value	Rationale
Learning Rate	1×10^{-4}	1×10^{-4}	Maintained for stability
Buffer Size	1,000,000	50,000	Reduced to minimize non-stationarity
Learning Starts	50,000	500	Earlier learning for faster adaptation
Target Update Interval	10,000	200	More frequent updates for stability
Exploration Fraction	0.1	0.4	Extended exploration for coordination
Exploration Initial Eps	1.0	1.0	Full initial exploration maintained
Exploration Final Eps	0.05	0.02	Lower final exploration for precision
Train Frequency	4	2	More frequent training steps
Gradient Steps	1	2	Enhanced learning per update
Batch Size	32	128	Larger batches for stable gradients
Tau (Soft Update)	1.0	0.01	Soft updates for target network stability
Discount Factor (γ)	0.99	0.99	Standard future reward weighting

A.3 Network Architecture

All algorithms use the default MlpPolicy network architecture from Stable-Baselines3:

Table A.9: Default network architecture specifications

Component	Configuration	Notes
Input layer	52 neurons	Fixed observation space size
Hidden layer 1	64 neurons	ReLU activation (default)
Hidden layer 2	64 neurons	ReLU activation (default)
Policy head (PPO/A2C)	5 neurons	Discrete action space
Value head (PPO/A2C)	1 neuron	State value estimation
Q-head (DQN)	5 neurons	Q-value for each action
Activation function	ReLU	Default for hidden layers
Output activation	Softmax (policy) / Linear (value/Q)	Standard configuration

A.4 SuperSuit Wrapper Configuration

The multi-agent environment is processed through SuperSuit wrappers for compatibility with single-agent algorithms:

Table A.10: SuperSuit transformation pipeline

Transformation	Purpose	Configuration
<code>pad_observations_v0</code>	Normalize observation spaces	Automatic padding to fixed size
<code>pettingzoo_env_to_vec_env_v1</code>	Convert to vectorized format	Single environment instance
<code>concat_vec_envs_v1</code>	Parallel environment execution	4 environments, 1 CPU core
Base class	Stable-Baselines3 compatibility	<code>stable_baselines3</code>

Appendix B

Experiment Results

B.1 Scenario 1

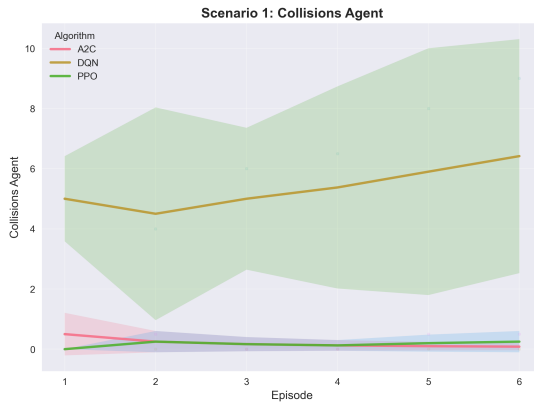


Figure B.1: Average number of collisions per agent across episodes in Scenario 1. PPO consistently outperforms other algorithms, maintaining near-zero collision rates. DQN exhibits a significant number of collisions, with increasing variance over time.

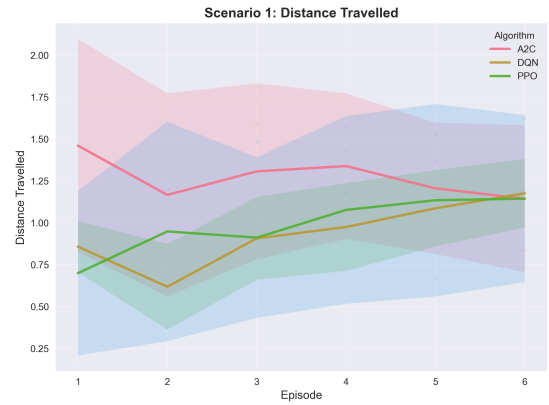


Figure B.2: Comparison of the average distance travelled by agents per episode. A2C leads in initial performance, but PPO and DQN gradually converge towards similar values by the final episodes.

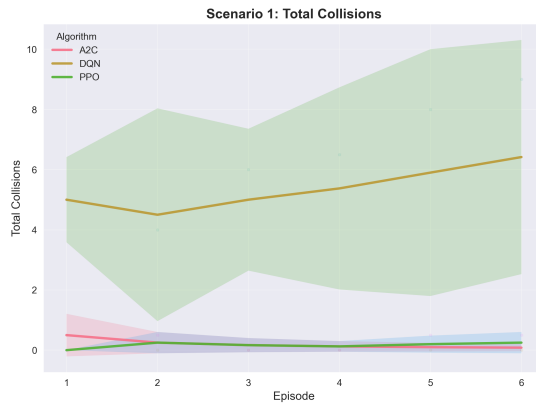


Figure B.3: Total number of collisions in the environment during each episode. The trend mirrors the per-agent analysis, with DQN yielding the highest collision counts. PPO and A2C maintain low collision frequencies.

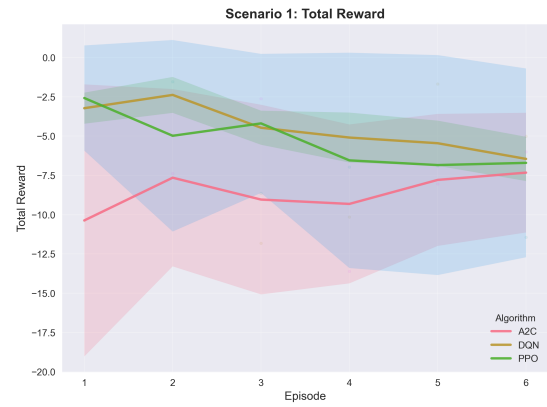


Figure B.4: Cumulative reward achieved per episode by each algorithm. PPO and DQN demonstrate relatively higher rewards compared to A2C, although all algorithms show a decreasing trend across episodes.

B.2 Scenario 2

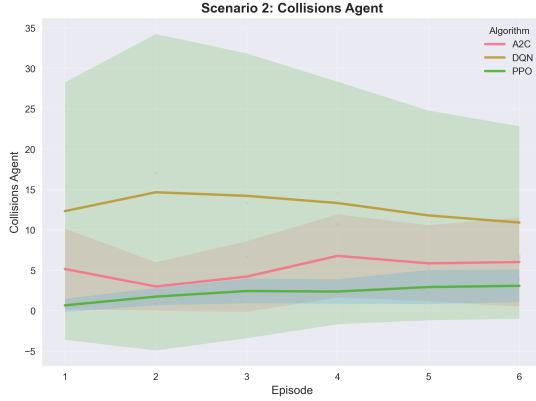


Figure B.5: Average number of collisions per agent throughout episodes in Scenario 2. DQN shows the highest collision rate, while PPO remains the most stable and exhibits the lowest values, suggesting better coordination among agents.

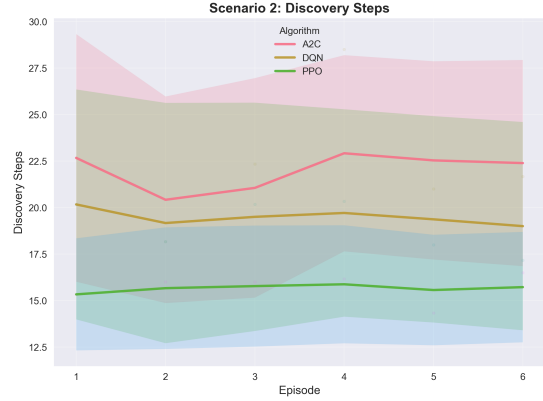


Figure B.6: Average number of steps required to discover the goal. A2C appears less efficient, requiring more steps on average. PPO shows improved efficiency, stabilizing around fewer discovery steps.

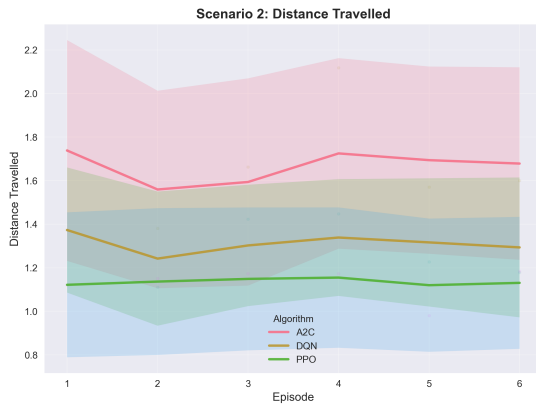


Figure B.7: Comparison of the average distance travelled by agents. A2C consistently covers longer distances, potentially reflecting more exploration, whereas PPO remains the most conservative across episodes.

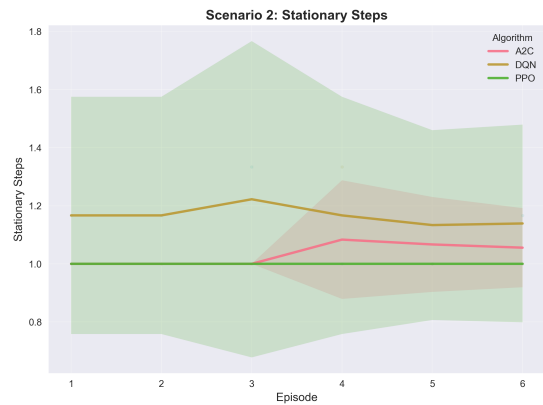


Figure B.8: Mean number of stationary steps per episode, indicating inactivity or indecision. All algorithms remain relatively stable, with PPO displaying minimal fluctuation, which may suggest consistent policy execution.

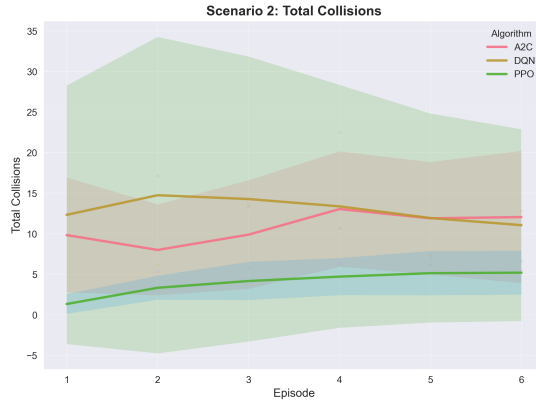


Figure B.9: Total collisions recorded in the environment across episodes. A similar trend to the per-agent collision analysis is observed, with DQN and A2C experiencing more frequent incidents compared to PPO.

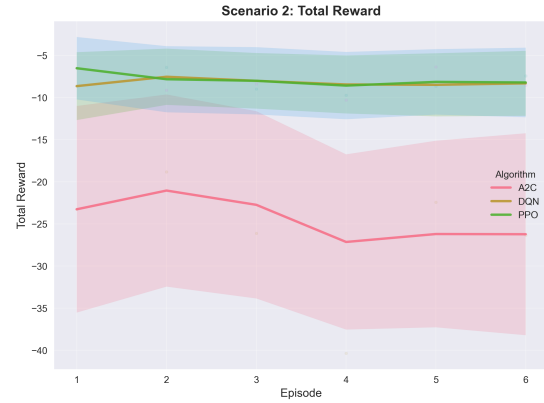


Figure B.10: Cumulative reward per episode for each algorithm. PPO maintains the highest average reward, while A2C receives consistently lower values, possibly due to excessive exploration or higher penalty accumulation.

B.3 Scenario 3

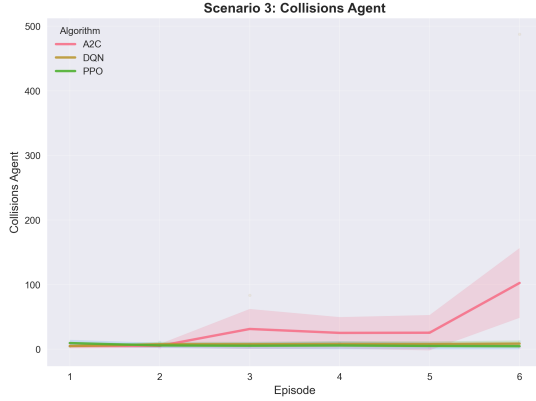


Figure B.11: Number of collisions per agent over time. A2C shows a significant spike in collision rate, whereas PPO and DQN maintain consistent and low collision frequencies throughout all episodes.

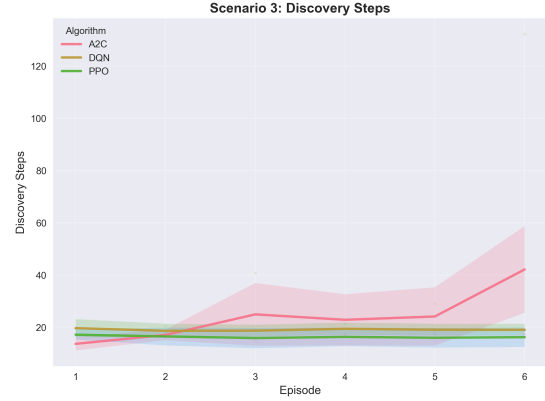


Figure B.12: Average number of steps required to discover the target in Scenario 3. While PPO and DQN remain stable, A2C shows a marked increase in discovery steps, particularly in later episodes, indicating performance degradation.

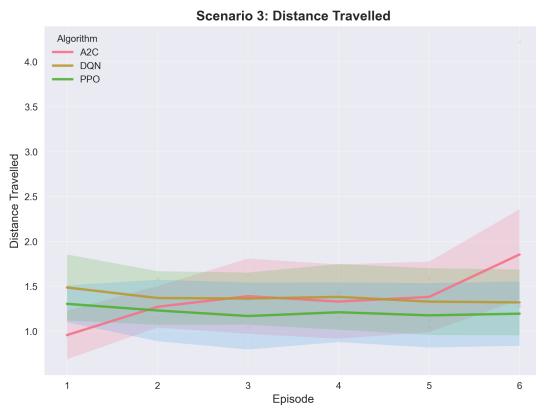


Figure B.13: Mean distance travelled by agents in each episode. A2C shows a steady increase, possibly due to inefficient exploration, while PPO demonstrates more conservative and consistent mobility patterns.

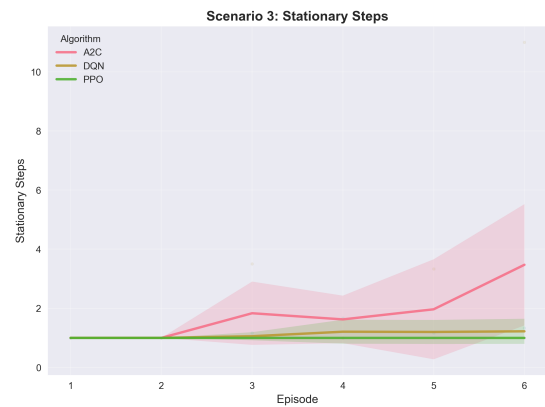


Figure B.14: Average number of stationary steps per episode, reflecting inactivity. PPO and DQN maintain stable behavior, while A2C increasingly idles over time, suggesting instability in decision-making.

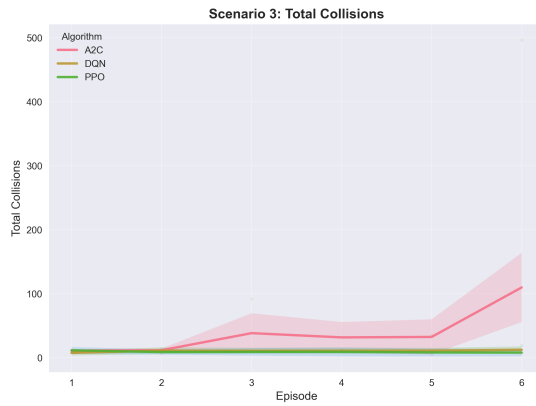


Figure B.15: Total collisions in the environment throughout episodes. A2C shows an exponential increase in collision count from episode 3 onward, while DQN and PPO demonstrate minimal and stable collision behavior.

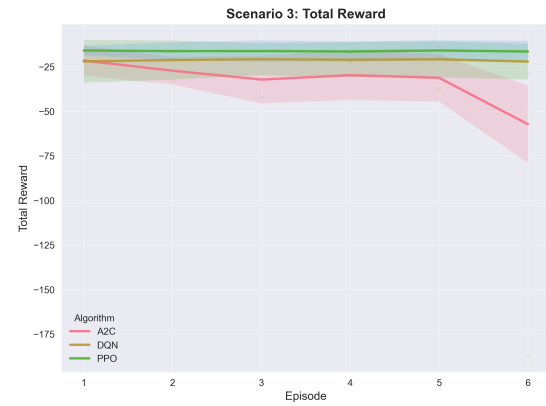


Figure B.16: Cumulative reward per episode for each learning algorithm. A2C experiences a sharp decline in total reward after episode 4, likely due to increased collisions and inefficiencies. PPO performs most consistently.

B.4 Scenario 4

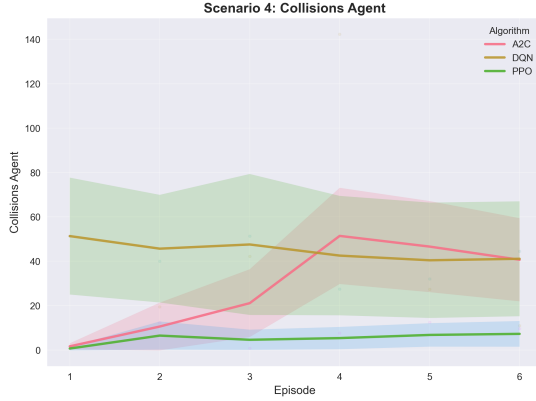


Figure B.17: Average number of collisions per agent in Scenario 4. A2C suffers a sharp increase after episode 3, indicating a collapse in behavior. PPO remains relatively stable, while DQN displays a slight downward trend.

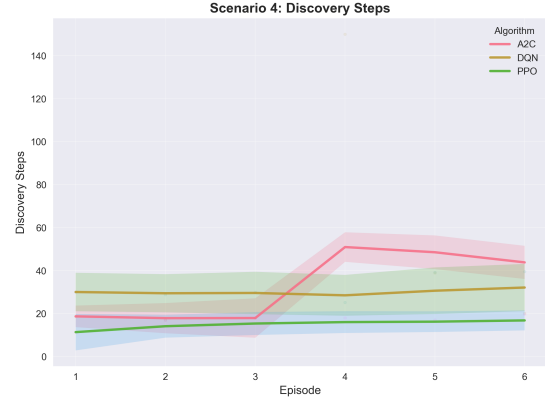


Figure B.18: Steps needed to discover the target. A2C experiences a significant jump from episode 4, consistent with the rising collisions and lower rewards, while PPO remains efficient throughout.

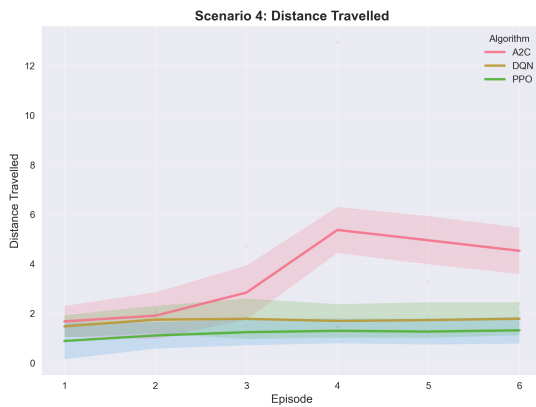


Figure B.19: Mean distance travelled by agents. A2C agents travel increasingly larger distances, likely inefficiently, while DQN and PPO maintain steady, low-mobility patterns.

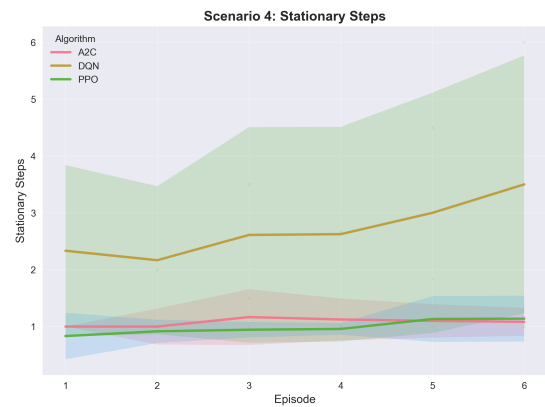


Figure B.20: Average number of inactive steps. DQN shows an upward trend in idle time, while PPO and A2C maintain near-constant low levels.

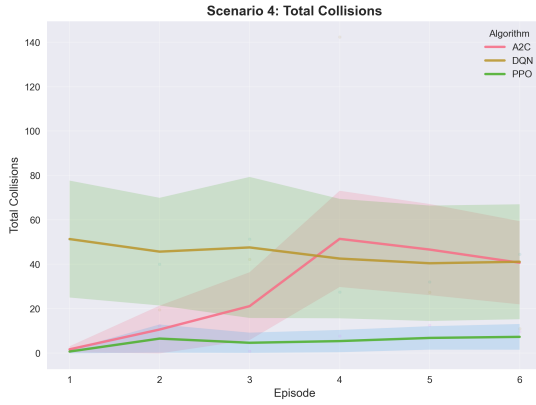


Figure B.21: Total number of collisions across all agents. A2C shows an explosive increase, whereas PPO maintains a low and stable collision rate. DQN improves slightly over time.



Figure B.22: Total reward obtained by agents. A2C's performance collapses drastically after episode 3, reaching values below -100. PPO and DQN remain robust, sustaining high reward values.

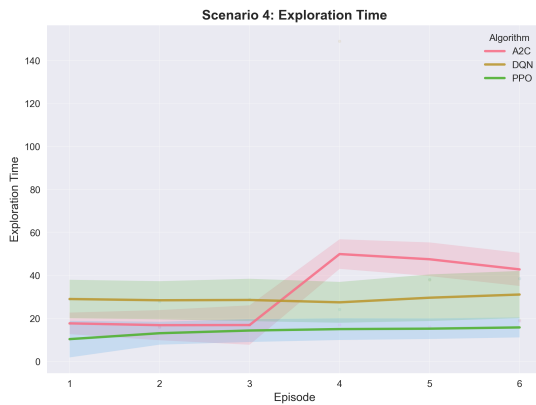


Figure B.23: Total time spent exploring per episode. A2C sees a large spike aligned with rising discovery steps and poor rewards, suggesting inefficient searching.

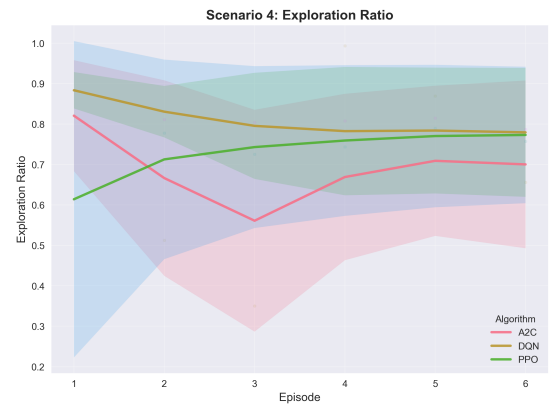


Figure B.24: Proportion of exploratory actions. PPO stabilizes above 0.75, suggesting balanced behavior. A2C drops below 0.6 in episode 3 and recovers slightly, whereas DQN shows a gradual decline.

B.5 Scenario 5

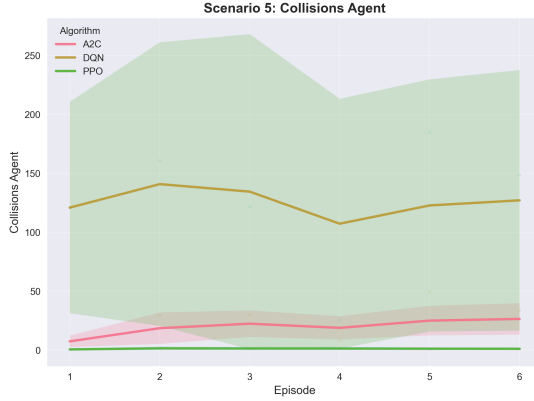


Figure B.25: Mean collisions per agent across episodes. DQN suffers from significantly higher collision rates, while PPO consistently maintains minimal collisions. A2C remains stable but suboptimal.

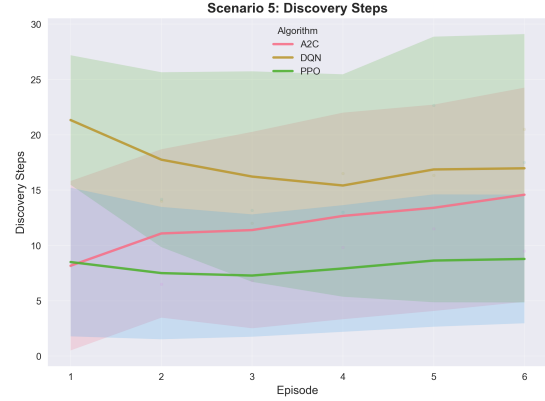


Figure B.26: Average number of steps required to discover a target in Scenario 5. DQN starts with better performance but converges to similar values as A2C by episode 6. PPO consistently shows the lowest discovery steps.

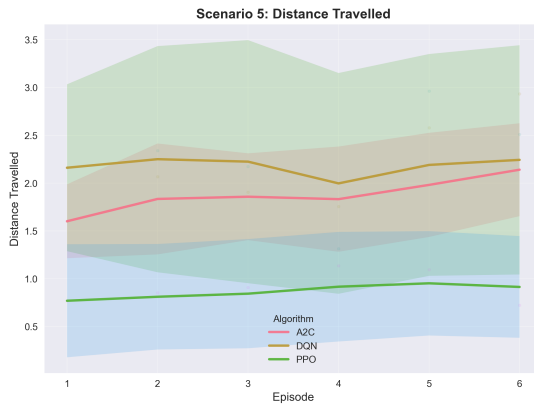


Figure B.27: Cumulative distance travelled by agents in Scenario 5. DQN and A2C consistently outperform PPO, suggesting more active exploration or longer trajectories.

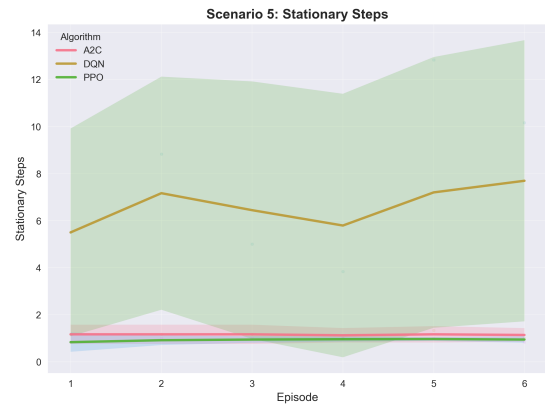


Figure B.28: Average number of stationary steps in Scenario 5. PPO exhibits minimal inactivity, while DQN increases stationary behavior over time. A2C remains stable.

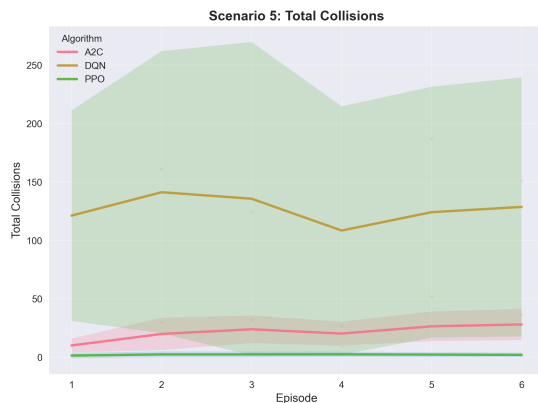


Figure B.29: Total number of collisions in the environment. DQN experiences a high and consistent number of collisions. PPO sustains minimal collisions across all episodes.

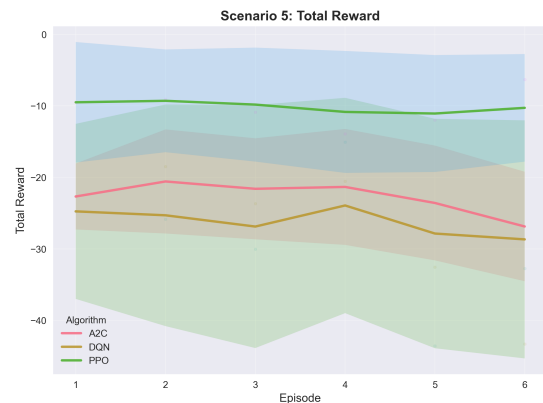


Figure B.30: Total reward received by agents across episodes. PPO significantly outperforms the others, showing a more stable and less negative reward trajectory.

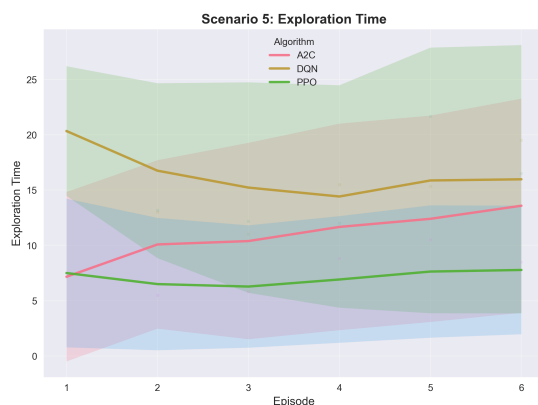


Figure B.31: Time spent exploring during each episode. PPO shows the lowest exploration time, while DQN and A2C increase over time, possibly due to less efficient target acquisition.

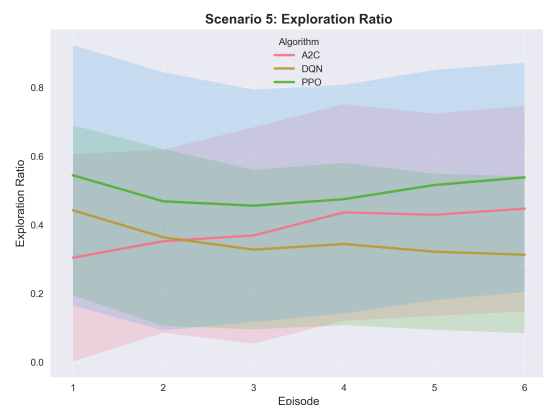


Figure B.32: Proportion of exploratory behavior over time. PPO maintains the highest exploration ratio, which gradually increases. A2C and DQN both start lower and plateau.