Titre: Title:	Integrated Real-Time Decision-Making in Smart Urban Freight Logistics: Modular and Adaptive Reinforcement Learning Approach
Auteur: Author:	Ali Shiri
Date:	2025
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Shiri, A. (2025). Integrated Real-Time Decision-Making in Smart Urban Freight Logistics: Modular and Adaptive Reinforcement Learning Approach [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/71111/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/71111/
Directeurs de recherche: Advisors:	Samira Keivanpour
Programme: Program:	Doctorat en génie industriel

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Integrated Real-Time Decision-Making in Smart Urban Freight Logistics: Modular and Adaptive Reinforcement Learning Approach

ALI SHIRI
Département de mathématiques et de génie industriel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie industriel

Novembre 2025

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Integrated Real-Time Decision-Making in Smart Urban Freight Logistics: Modular and Adaptive Reinforcement Learning Approach

présentée par **Ali SHIRI**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Martin TRÉPANIER, président
Samira KEIVANPOUR, membre et directrice de recherche
Jean-Marc FRAYRET, membre
Armin JABBARZADEH, membre externe

DEDICATION

To my kind and loving parents and siblings,
whose unwavering love, support, patience, and sacrifices paved the way for my life.

To all those who fought in silence;
who built and rebuilt with radiant hearts and never stopped.
They grew weary, but never bowed their heads.
and carried on with hope.

To those who lived with kindness,
who were wounded, yet chose to heal.
To all those whose selflessness
made a better world possible.

May this effort be an echo of faith in goodness,
and a small step toward building a brighter future

ACKNOWLEDGEMENTS

My heartfelt appreciation goes to my core family in Iran—my parents, brother, and sister—whose unwavering emotional support and constant encouragement were pivotal to the success of this project.

I am grateful to my dear friends, who stood by me throughout this journey. Their constant encouragement, late-night conversations, and belief in my abilities provided the strength I needed during the most challenging moments.

I gratefully acknowledges Asad Yarahmadi for his contribution in preparing the Montreal and Toronto datasets utilized in this study. Special thanks are also extended to Shiphual Logistics for their valuable collaboration, insights, and support, which greatly enriched the quality of this work.

This research was supported in part by Shiphual Logistics and Mitacs (Project IT30680), and the Natural Sciences and Engineering Research Council of Canada (NSERC).

I acknowledge the use of artificial intelligence tools for linguistic assistance and editing during the preparation of this manuscript, in accordance with the guidelines of the Ordre des ingénieurs du Québec (OIQ).

RÉSUMÉ

Cette thèse examine la conception et la mise en œuvre d'architectures d'apprentissage par renforcement (RL) visant à optimiser la prise de décision en temps réel dans les systèmes de logistique urbaine intelligente pour le transport de marchandises. La recherche est motivée par la complexité opérationnelle croissante des réseaux de fret urbain, due à la forte croissance du commerce électronique, à la variabilité dynamique de la demande et à la congestion urbaine persistante, qui mettent en évidence les limites des approches d'optimisation statiques et heuristiques traditionnelles.

L'objectif central de cette thèse est de développer des cadres évolutifs et fondés sur les données, permettant une prise de décision adaptative et coordonnée dans des environnements logistiques multi-agents, à l'échelle d'une ville et sous incertitude. À cette fin, quatre nouvelles architectures de RL sont proposées et évaluées au moyen de simulations multi-agents calibrées à partir de données réelles de logistique urbaine.

L'étude reconceptualise les opérations de fret comme un processus de décision markovien et propose quatre cadres RL de complexité croissante. Premièrement, un réseau neuronal profond à valeurs (DQN) centralisé traite des problèmes de jumelage discret de cargaisons. Deuxièmement, un modèle hiérarchique acteur-critique coordonne les décisions conjointes de jumelage et d'affectation. Troisièmement, un cadre hiérarchique imbriqué d'apprentissage par renforcement (NHRL) intègre la tarification dynamique avec les opérations de jumelage et d'affectation. Enfin, un système décentralisé d'apprentissage par renforcement multi-agents (MARL) permet une prise de décision autonome au niveau de chaque véhicule, avec des capacités dynamiques de changement de tâches.

Toutes les architectures intègrent un raisonnement spatial avancé grâce au système de maillage hexagonal H3 d'Uber, permettant un raisonnement spatial multi-résolution pour une granularité décisionnelle adaptative. Elles utilisent également des algorithmes modulaires de préfiltrage (PAMA, PADA, DEZE, ShipScan) qui transforment des problèmes de recherche combinatoire NP-difficiles en processus traitables en éliminant rapidement les options non réalisables ou à faible valeur. Les cadres sont évalués à l'aide de jeux de données synthétiques représentant les opérations de fret urbain à Montréal et Toronto, couvrant plus de 37 000 interactions véhicule-cargaison. L'évaluation des performances porte sur les récompenses cumulées, l'optimisation des revenus, les taux de réussite de jumelage, le temps d'inactivité des véhicules, le temps d'attente des cargaisons et l'efficacité des distances de prise en charge. L'approche proposée montre des améliorations significatives par rapport aux

modèles de référence, notamment jusqu'à 26% d'augmentation des taux de réussite de jumelage, 9% d'amélioration de l'utilisation de la flotte, 70% de réduction du temps d'inactivité des véhicules, 50% de diminution du temps d'attente des expéditeurs et 13% de réduction du kilométrage à vide.

Le modèle DQN établit l'efficacité fondamentale du RL à base de valeurs pour les tâches de jumelage discret, tout en démontrant que le maillage H3 et le préfiltrage réduisent considérablement la charge computationnelle. Le cadre HRL développe cette base en coordonnant des agents spécialisés de jumelage et d'affectation via un hub centralisé, ce qui permet d'améliorer les taux de réussite, les récompenses et de réduire le temps d'inactivité, le kilométrage à vide et le coût computationnel. L'architecture NHRL va plus loin en intégrant des mécanismes de tarification dynamique avec un raisonnement spatial multi-résolution, augmentant les revenus et les taux de jumelage. Le système MARL représente l'approche la plus avancée, permettant une prise de décision totalement décentralisée et contextuelle au niveau de chaque véhicule, atteignant des taux de jumelage et de consolidation plus élevés. L'analyse comparative des algorithmes de RL (PPO, TRPO, DDPG) valide la supériorité en performance et en robustesse de l'optimisation par politiques proximales (PPO) dans des environnements de fret dynamiques.

La thèse se conclut par l'examen des considérations pratiques liées à la mise en œuvre dans des plateformes logistiques commerciales, en soulignant le potentiel transformateur des architectures RL pour moderniser les opérations de fret urbain grâce à des capacités d'optimisation intelligentes, évolutives et en temps réel. La recherche apporte à la fois des cadres théoriques et des méthodologies pratiques pour les systèmes logistiques urbains de nouvelle génération.

ABSTRACT

This dissertation investigates the design and implementation of reinforcement learning (RL) architectures to optimize real-time decision-making in smart urban freight logistics systems. The research is motivated by the increasing operational complexity of urban freight networks, driven by rapid e-commerce growth, dynamic demand patterns, and persistent urban congestion, which collectively expose the limitations of traditional static and heuristic optimization approaches.

The central objective of the dissertation is to develop scalable, data-driven frameworks that enable adaptive and coordinated decision-making under uncertainty in multi-agent, city-scale logistics environments. To this end, four novel RL-based architectures are proposed and evaluated through agent-based simulations calibrated with real-world urban freight data.

The study reconceptualizes freight operations as a Markov Decision Process and develops four progressively sophisticated RL frameworks. First, a centralized Deep Q-Network (DQN) addresses discrete shipment matching problems. Second, a hierarchical actor-critic model coordinates joint matching and dispatching decisions. Third, a nested hierarchical reinforcement learning (NHRL) framework integrates dynamic pricing with matching and dispatching operations. Finally, a decentralized multi-agent reinforcement learning (MARL) system enables autonomous vehicle-level decision-making with dynamic task switching capabilities.

All architectures incorporate advanced spatial reasoning through Uber's H3 hexagonal indexing system, which enables multi-resolution spatial reasoning for adaptive decision granularity, and employ modular pre-filtering algorithms (PAMA, PADA, DEZE, ShipScan) that transform NP-hard combinatorial search problems into tractable processes by rapidly eliminating infeasible or low-value options. The frameworks are evaluated using synthetic datasets representing urban freight operations in Montréal and Toronto, encompassing over 37,000 vehicle-shipment interactions. Performance evaluation focuses on cumulative rewards, revenue optimization, match success rates, vehicle idle time, shipment waiting time, and pickup distance efficiency. The proposed approach demonstrates significant improvements over baseline models, including up to a 26% increase in match success rates, a 9% boost in fleet utilization, a 70% reduction in vehicle idle time, a 50% decrease in shipper waiting time, and a 13% reduction in empty mileage.

The DQN model establishes the foundational effectiveness of value-based RL for discrete matching tasks while demonstrating how H3 spatial indexing and pre-filtering substantially reduce computational overhead. The HRL framework advances this foundation by coordi-

nating specialized matching and dispatching agents through a centralized coordination hub, achieving improved match rates, reward, and reduced vehicle idle time, Empty Mileage, and time Computation Cost. The NHRL architecture extends capabilities further by integrating dynamic pricing mechanisms with multi-resolution spatial reasoning, increasing revenue and match rate. The MARL system represents the most sophisticated approach, enabling fully decentralized, context-aware decision-making at individual vehicle levels, attaining a higher match rate and consolidation. Comparative analysis of reinforcement learning algorithms (PPO, TRPO, DDPG) validates the superior performance and robustness of Proximal Policy Optimization in dynamic freight environments.

The dissertation concludes by examining practical implementation considerations for commercial logistics platforms, highlighting the transformative potential of RL-based architectures in modernizing urban freight operations through intelligent, scalable, and real-time optimization capabilities. The research contributes both theoretical frameworks and practical methodologies for next-generation urban logistics systems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

CPM	Context Prioritization Module
Ctx1	Context Scoring Strategy 1 (Capacity-Based)
Ctx2	Context Scoring Strategy 2 (Capacity + Time-Based)
DDPG	Deep Deterministic Policy Gradient
DEZE	Dispatch Eligibility Zone Evaluator
DF	Deep Freight
DL	Deep Learning
DQL	Deep Q-Learning
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
GAE	Generalized Advantage Estimation
GHU	Geospatial Hexagonal Unit
GHU6	Geospatial Hexagonal Unit at Resolution 6
GHU7	Geospatial Hexagonal Unit at Resolution 7
H3	Hierarchical Hexagonal Spatial Index
HA	Heuristic Algorithm
HRL	Hierarchical Reinforcement Learning
LTL	Less-than-Truckload
MARL	Multi-Agent Reinforcement Learning
MDP	Markov Decision Process
MSE	Mean Squared Error
NHRL	Nested Hierarchical Reinforcement Learning
OD	Origin-Destination
PADA	Pre-filtering Algorithm for Dispatching Agent
PAMA	Pre-filtering Algorithm for Matching Agent
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
SAC	Soft Actor-Critic
ShipScan	Shipment Scanning Algorithm
TRPO	Trust Region Policy Optimization
TSR	Traveling Salesman Routing

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

1.1 Background and Motivation

Urban freight logistics is undergoing a rapid and complex transformation, driven by the rapid expansion and changing customer expectations, and increasing pressure on urban infrastructure. The rise of smart freight platforms has fundamentally changed how goods move through cities, considering multiple stakeholders perspectives, including carriers (vehicle operators), shippers (businesses requiring delivery services), and platform operators (technology providers facilitating marketplace coordination).

These smart freight platforms serve as intermediaries that connect supply and demand in real-time, operating through several core components: dynamic pricing modules that adjust rates based on market conditions, matching algorithms that assign shipments to available vehicles, dispatching systems that determine vehicle routes and schedules, and customer interfaces that provide transparency and tracking capabilities. Unlike traditional freight brokerages that rely on manual coordination and static contracts, these platforms leverage real-time data streams to optimize operations continuously across thousands of simultaneous transactions.

However, the coordination of these platform components presents significant technical and operational challenges. The platforms must simultaneously manage dynamic pricing decisions (determining optimal rates for different zones and time periods), shipment-to-vehicle matching (assigning incoming requests to available vehicle), and vehicle dispatching and routing (optimizing movement patterns across urban networks). Each of these decisions is interdependent and influenced by real-time operational constraints, while the complexity is compounded by the high-dimensional, spatio-temporal nature of urban freight environments where thousands of vehicles and shipment requests interact continuously across diverse geographic zones.

The urgency of addressing these coordination challenges is highlighted by global projections. The World Economic Forum's "Future of the Last Mile Ecosystem" report [1] projects that without significant intervention, the number of delivery vehicles in major cities is expected to increase by approximately 36% by 2030. This surge will contribute to a 32% rise in delivery-related traffic emissions and add an estimated 11 minutes to average daily commute times. These trends not only threaten urban sustainability but also highlight the growing operational burden on freight systems. In parallel, McKinsey & Company's analysis of last-

mile delivery economics [2] estimates that final-mile delivery costs account for up to 53% of total shipping expenses, with inefficient routing and dispatching contributing to 20–30% inflation in operational costs.

Urban freight systems must balance multiple, often conflicting objectives across different stakeholder groups. Carriers prioritize reducing empty mileage and idle time to maximize vehicle utilization and operational profitability. Shippers demand fast and reliable service with transparent and competitive pricing. Smart platforms seek to maximize throughput and revenue while maintaining service levels that satisfy both carriers and shippers, ensuring long-term marketplace sustainability. These objectives are tightly coupled and must be addressed simultaneously to achieve system-wide efficiency that benefits all stakeholders through increased platform revenue, improved carrier efficiency, and enhanced service quality for shippers.

Limitations of Current Optimization Approaches

Decision-making in freight logistics has traditionally relied on classic optimization methods, heuristics/metaheuristics, and hybrid frameworks that combine both.

Linear programming and mixed-integer formulations provide optimal solutions for well-defined problems but encounter computational tractability issues when scaled with thousands of variables and real-time constraints. The joint optimization of pricing (continuous decision space), matching (combinatorial assignment), and dispatching (spatial-temporal routing) results in mixed-integer nonlinear programming formulations that become computationally inefficient for real-time deployment [3].

While traditional heuristics such as nearest-neighbor matching, greedy assignment, and rule-based algorithms offer computational efficiency and scalability, they suffer from several critical limitations in real-time urban freight contexts. First, they require manual parameter tuning for different operational conditions and cannot automatically adapt when system dynamics change-for example, when demand patterns shift noticeably or new urban zones emerge. Second, they lack learning capabilities, meaning they cannot improve performance based on historical experience or recognize emerging patterns in shipment requests and vehicle availability.

In real-world operations, it is common to see hybrid architectures where traditional optimization solves the static or tactical layer (e.g., daily route planning), and reactive rule-based or heuristic modules handle real-time adjustments. While this approach mitigates some scalability issues, it still struggles to capture the feedback loops and interdependencies among

pricing, matching, dispatching, and routing decisions in volatile markets.

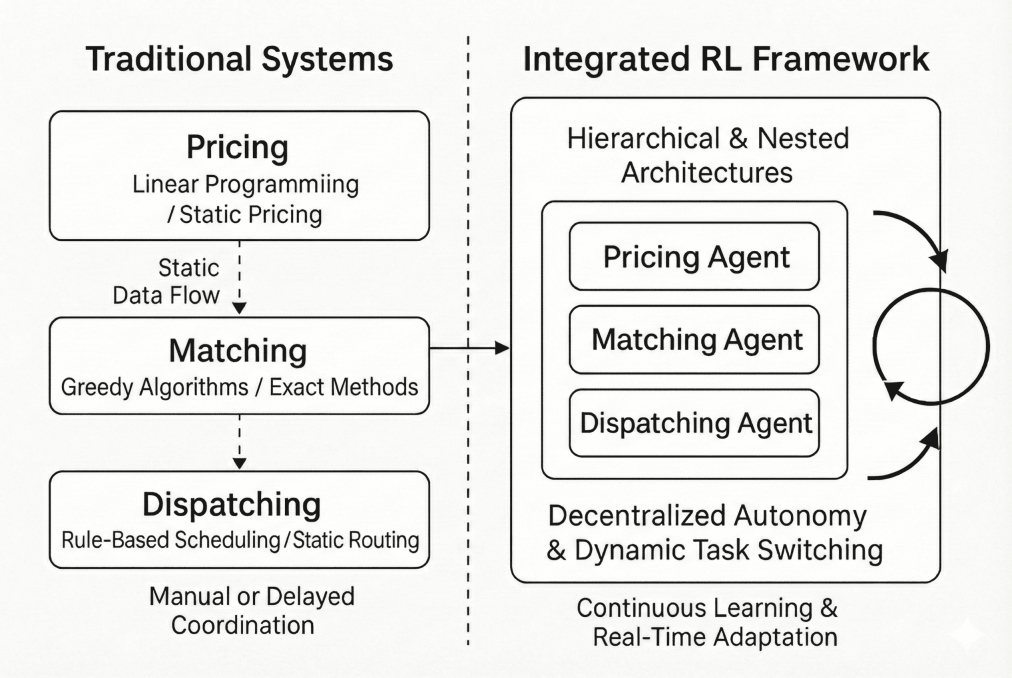

Figure 1.1 Comparison of Traditional Sequential Freight Decision-Making and Proposed Integrated RL Framework

To manage the complexity of large-scale optimization problems, current practice often relies on temporal or spatial decomposition strategies. While these approaches help make the problem more tractable, they frequently overlook critical interdependencies between decisions. For example, aggressive pricing in high-demand zones may attract vehicles for certain segments but can simultaneously suppress demand from more price-sensitive customers, potentially leading to uneven load distribution. Similarly, greedy matching algorithms may optimize short-term assignment rates but ignore long-term dispatching consequences, resulting in geographic imbalances that increase repositioning costs and reduce overall system efficiency. Figure 1.1 illustrates this contrast between traditional sequential decision-making and the proposed integrated reinforcement learning approach.

Integrated Optimization

A unified and scalable framework that jointly optimizes all three core components—pricing, matching, and dispatching—in real-time, particularly through adaptive learning architectures, remains a significant challenge in smart urban freight logistics. The transition toward integrated, learning-based solutions addresses several key technical requirements. By

reconceptualizing freight operations as sequential decision processes, reinforcement learning frameworks can capture the temporal dependencies inherent in logistics operations and adapt continuously to changing conditions. This superiority has been demonstrated in recent studies, including Turan et al. [4], who developed an RL framework for fleet management, and Chen et al. [5], who applied RL to improve logistics distribution.

This research proposes a structured progression of RL-based architectures, ranging from centralized single-agent models to decentralized multi-agent systems. Each architecture is designed to address specific operational tasks—such as shipment matching, dispatching, and dynamic pricing—while maintaining scalability and adaptability. The use of Hierarchical Reinforcement Learning (which decomposes complex decision problems into hierarchical sub-tasks) and Nested Hierarchical Reinforcement Learning (which further structures decision-making through multiple nested layers) enables the decomposition of multiple decision processes, improving solution quality and computational efficiency. Modular pre-filtering algorithms—which reduce computational complexity by narrowing the action space—are introduced to support real-time feasibility. The final architecture employs Multi-Agent Reinforcement Learning (where multiple autonomous agents learn and coordinate their actions in shared environments) to support vehicle-level autonomy, allowing agents to switch dynamically between tasks based on contextual cues.

In summary, the motivation for this thesis stems from the need to develop integrated, adaptive decision-making frameworks that can coordinate pricing, matching, and dispatching decisions in real-time urban freight platforms. Through modular reinforcement learning architectures, this research aims to overcome the coordination limitations of current fragmented approaches and demonstrate measurable improvements in system efficiency, operational costs, and service quality for all platform stakeholders.

1.2 Research Objectives and Questions

The overarching goal of this thesis is to develop an integrated decision-support system for freight optimization. To achieve this, we define three specific research objectives:

- **Objective 1:** To develop a scalable, multi-agent framework capable of jointly optimizing pricing, matching, and dispatching in dynamic urban environments.
- **Objective 2:** To evaluate the efficacy of Reinforcement Learning (RL) methodologies—specifically Hierarchical and Multi-Agent architectures—in handling the high-dimensional state spaces of urban logistics compared to traditional static approaches.

- **Objective 3:** To design decentralized mechanisms that enable vehicle-level autonomy and dynamic task-switching, thereby overcoming the computational bottlenecks of centralized control systems.

To fulfill these objectives, the research addresses the following three core research questions:

- **RQ1: How can integrated, adaptive optimization frameworks overcome the fundamental limitations of traditional fragmented approaches in coordinating real-time operational decisions within dynamic urban freight environments?.** This question addresses the core logistics challenge of joint optimization by capturing the interdependencies between pricing strategies, matching algorithms, and dispatching-routing policies. The investigation encompasses what coordination mechanisms are necessary to achieve superior performance compared to sequential decision-making, how adaptation should occur in response to changing demand patterns and network conditions, and what quantifiable performance improvements can be achieved across stakeholder objectives.
- **RQ2: What model innovations enable scalable, multi-agent optimization of interdependent logistics tasks across different spatial and temporal scales while maintaining real-time operational feasibility?** This question focuses on the design principles and structural choices required to achieve computational tractability and responsiveness in city-scale freight logistics. It examines how hierarchical decomposition, nested decision layers, and spatial abstractions can reduce computational complexity without sacrificing solution quality. The investigation considers coordination mechanisms among agents, the role of centralized versus decentralized control, and the trade-offs between global optimality and local autonomy in dynamic urban environments.
- **RQ3: What strategies and system-level innovations can ensure scalability, robustness, and coordination when deploying RL-based frameworks for city-scale freight logistics, particularly under conditions of high uncertainty and heterogeneous agent behaviors?** This question addresses the practical challenges of implementing RL-driven architectures in real-world freight platforms. It explores methods for stabilizing learning in dynamic environments, ensuring policy generalization across diverse demand patterns and network topologies, and achieving effective coordination among heterogeneous agents with different capabilities and objectives. The inquiry includes robustness to data variability, adaptability to operational disruptions, and the integration of interpretability mechanisms to support operational trust.

These research questions guide the investigation toward fundamental logistics improvements while acknowledging that advanced optimization techniques, including RL, represent the most promising path toward addressing these challenges given the dynamic, multi-agent, and high-dimensional nature of urban freight logistics.

1.3 Core Research Contributions

This thesis makes five principal contributions to the field of smart urban freight logistics and RL-based optimization:

1. **Modular Reinforcement Learning Architectures for Urban Freight Tasks.**
   A structured sequence of RL-based frameworks is proposed—ranging from single-agent value-based models to decentralized multi-agent systems—to address core operational tasks such as shipment matching, dispatching, and dynamic pricing. Each architecture is designed to match the scale and complexity of the corresponding task, offering a flexible and adaptive approach to urban freight optimization. This modular approach enables practitioners to implement specific components based on their operational requirements and technical constraints.

2. **Hierarchical and Nested Control for Multi-Task Optimization.**
   The thesis introduces hierarchical and nested reinforcement learning architectures to jointly optimize interdependent decisions across pricing, matching, and dispatching. These architectures enable temporal and spatial decomposition of decision processes, improving scalability and solution quality in dynamic environments. This contribution demonstrates how complex logistics problems can be decomposed into manageable subproblems while maintaining coordination across decision layers.

3. **Scalable Spatial Intelligence via H3 Indexing and Pre-Filtering.**
   A key innovation is the integration of Uber's H3 hexagonal spatial indexing for geographic abstraction and cross-zone coordination. Combined with task-specific prefiltering mechanisms, this approach enhances spatial reasoning, reduces computational complexity, and supports efficient real-time decision-making at city scale.

4. **Dynamic Multi-Agent Coordination with Context-Aware Task Switching.**
   A decentralized multi-agent RL framework is developed, enabling vehicle-level agents to autonomously switch between tasks such as matching, routing, and repositioning based on contextual cues. This design supports scalable coordination, system robustness, and interpretability in large-scale freight systems with heterogeneous agent behaviors. This

contribution enables adaptive resource allocation and improves system resilience under varying operational conditions.

5. **Realistic, Large-Scale Simulation Environment for Urban Freight Evaluation.**
   The thesis delivers a custom-built agent-based simulation platform calibrated with real-world data from Montréal and Toronto. The platform captures operational constraints such as supply and demand variability and enables rigorous evaluation of RL-based architectures across multiple performance dimensions and urban settings. This simulation environment provides a validated testbed for future research and practical deployment validation.

Collectively, these contributions advance the state of the art in urban freight logistics by demonstrating that integrated, learning-based decision architectures can outperform traditional, siloed methods. They also establish a practical and methodological foundation for future deployment in real-world logistics systems.

1.4 Thesis Roadmap and Model Progression

The research follows a structured four-stage progression that mirrors the complexity of real-world deployment challenges, with each stage building upon previous model innovations:

Stage 1 (Chapter 4, Article 1): Foundation Building – Establishes the core technologies and validates fundamental concepts through a single-task architecture for real-time shipment–vehicle matching. This stage demonstrates the feasibility of value-based reinforcement learning combined with H3 spatial indexing and pre-filtering for computational efficiency.

Stage 2 (Chapter 5, Article 2): Hierarchical Integration – Introduces multi-task coordination by jointly optimizing matching and dispatching within a hierarchical reinforcement learning framework. This stage leverages H3 spatial abstraction and modular pre-filtering to improve scalability and responsiveness. The transition to Actor-Critic methods enables handling of large action spaces and provides more stable learning convergence.

Stage 3 (Chapter 6, Article 3): Comprehensive Optimization – Achieves full integration of pricing, matching, and dispatching decisions through a nested hierarchical architecture. This stage incorporates multi-resolution spatial reasoning and dynamic pricing mechanisms to enhance revenue and service quality under real-time constraints. The implementation of Proximal Policy Optimization supports complex multi-objective optimization.

Stage 4 (Chapter 7, Article 4): Agent Autonomy – Develops a decentralized multi-agent

system enabling vehicle-level autonomy with dynamic task switching among matching, routing, and repositioning. This stage ensures scalability, robustness, adaptability, and resource enhancement for city-scale freight operations. The MARL framework enables emergent coordination behaviors and system-wide optimization through local agent interactions.

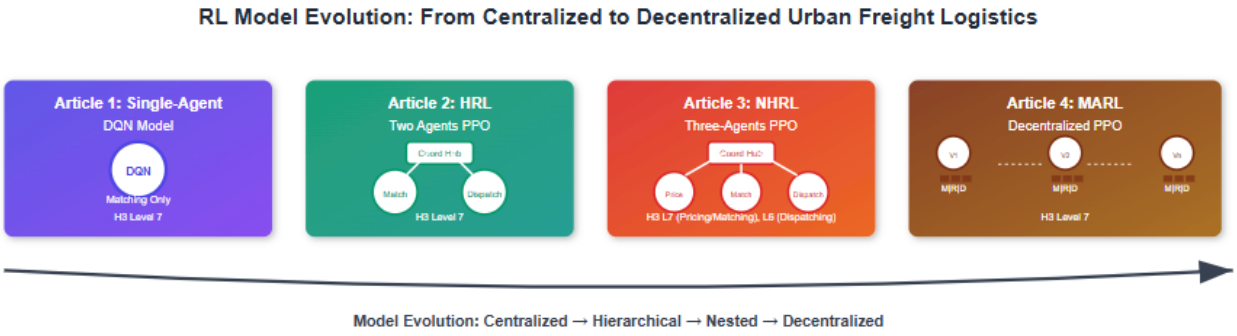

Figure 1.2 RL Model Evolution: From Centralized to Decentralized Urban Freight Logistics.

This systematic progression ensures that each model innovation is rigorously validated before introducing additional complexity, while maintaining a clear trajectory toward operational scalability and real-world applicability. Figure 1.2 illustrates the model evolution across the four articles, highlighting the transition from centralized decision-making to fully decentralized, context-aware systems.

1.5 Methodological Approach

This dissertation adopts a Design Science Research methodology, integrating theoretical algorithm development with empirical validation through large-scale simulation. The research progresses iteratively, moving from problem conceptualization to algorithmic design, simulation modeling, and performance benchmarking. While Chapter 3 provides the comprehensive technical details regarding agent designs, mathematical formulations, datasets, and the simulation environment, this section briefly highlights the core methodological pillar of the work: the transition from static optimization to adaptive, agent-based decision-making using Reinforcement Learning. This approach allows for the capture of temporal dependencies and dynamic interactions inherent in urban freight logistics, which traditional methods struggle to address.

1.6 Dissertation Structure

This dissertation is organized into nine chapters:

- **Chapter 1 – Introduction:** Presents the background, research motivation, problem statement, objectives, and contributions.
- **Chapter 2 – Literature Review:** Provides a comprehensive review of classical optimization, machine learning, and RL approaches in freight logistics, along with a gap analysis and positioning of the proposed research.
- **Chapter 3 – Research Methodology:** Details the problem formulations, RL algorithmic design, system architectures, simulation environments, and evaluation framework.
- **Chapters 4–7 – Research Articles:** Each chapter presents one of the four core studies, following the model evolution from single-agent foundations through hierarchical coordination to fully distributed multi-agent systems. Each study includes methodology, experimental design, results analysis, and discussion.
- **Chapter 8 – General Discussion:** Synthesizes findings across articles, highlights methodological innovations, and discusses theoretical and practical implications.
- **Chapter 9 – Conclusion:** Summarizes the research outcomes, outlines limitations, and proposes directions for future research.

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction and Chapter Objectives

Urban freight logistics faces a critical transformation challenge: how can fragmented, reactive systems evolve into integrated, adaptive frameworks capable of coordinating complex operational decisions in real-time? The exponential growth of e-commerce and evolving consumer expectations have exposed fundamental weaknesses in traditional optimization approaches, creating an urgent need for methodological innovation that can address the scale, dynamism, and complexity of modern urban freight networks.

This chapter examines the methodological landscape that has emerged in response to these challenges, tracing the evolution from classical optimization techniques to contemporary reinforcement learning frameworks. The investigation is driven by the recognition that urban freight optimization is not merely a technical problem requiring better algorithms, but a fundamental coordination challenge that demands new approaches to multi-stakeholder decision-making under uncertainty.

The literature review is structured to support the thesis's core mission of developing integrated optimization frameworks for urban freight logistics. Three methodological research questions guide this investigation:

RQ1 How can integrated, adaptive optimization frameworks overcome the fundamental limitations of traditional fragmented approaches in coordinating real-time operational decisions within dynamic urban freight environments?.

RQ2 What model innovations enable scalable, multi-agent optimization of interdependent logistics tasks across different spatial and temporal scales while maintaining real-time operational feasibility?.

RQ3 What strategies and system-level innovations can ensure scalability, robustness, and coordination when deploying RL-based frameworks for city-scale freight logistics, particularly under conditions of high uncertainty and heterogeneous agent behaviors?.

It is important to note that these questions position Reinforcement Learning as the central methodological approach. Consequently, this review primarily investigates the applicability, strengths, and limitations of RL paradigms in solving logistics decision problems, rather than providing an exhaustive survey of all possible operations research techniques.

These methodological inquiries directly inform the thesis's problem-centric research agenda. RQ1 establishes whether current approaches can overcome the fundamental limitations of fragmented decision-making that plague urban freight systems. RQ2 reveals the architectural innovations needed to achieve scalable coordination across spatial and temporal scales. RQ3 identifies the strategic foundations required for robust deployment of integrated frameworks in real-world environments characterized by high uncertainty and heterogeneous stakeholder behaviors.

Through this structured examination, the chapter builds toward a clear understanding of why traditional approaches fail to address the coordination challenges inherent in urban freight logistics, and how emerging methodologies can provide the foundation for next-generation integrated optimization frameworks. The analysis establishes both the theoretical necessity and practical feasibility of the modular reinforcement learning architectures proposed in subsequent chapters.

2.2 Foundational Concepts and Problem Context

2.2.1 Urban Freight Transportation Systems

Urban freight transportation systems constitute complex networks of interconnected stakeholders, including freight carriers, shippers, and digital logistics platforms, all operating within the constraints of urban infrastructure and regulatory frameworks [6]. These systems are characterized by several distinctive features: high demand volatility driven by consumer behavior patterns, spatial and temporal heterogeneity reflecting urban geography and dynamics, and the imperative for real-time coordination among multiple decision-makers with potentially conflicting objectives.

The operational architecture of urban freight systems typically encompasses multiple decision-making layers, ranging from tactical pricing and resource allocation to operational real-time dispatching [7]. This hierarchical structure creates interdependencies that demand optimization approaches capable of both learning and coordinated decision-making across temporal and spatial scales.

Critical challenges in urban freight systems include: travel distance and operational costs that vary significantly across urban networks; fluctuating demand patterns requiring dynamic resource allocation, vehicle idle time, shipment waiting time and capacity management; service quality constraints, freight handling requirements, and customer satisfaction metrics; and the necessity for multi-stakeholder coordination among entities with divergent operational objectives and business models [8].

The complexity is further amplified by the interdependence of decisions across different operational dimensions. Pricing strategies directly influence demand patterns and customer behavior, which in turn affect optimal matching algorithms and dispatching decisions. Similarly, vehicle positioning and routing decisions impact service quality and operational costs, creating feedback loops that traditional optimization approaches struggle to capture effectively.

2.2.2 Optimization Objectives in Freight Logistics

The optimization landscape in urban freight logistics encompasses multiple interconnected objectives that must be balanced to achieve system-wide efficiency. This multi-objective nature represents one of the fundamental challenges in developing effective optimization frameworks for urban freight operations.

Dynamic pricing strategies serve simultaneously as revenue optimization tools and demand management mechanisms, requiring continuous adjustment based on real-time supply-demand conditions, and market dynamics [9]. Effective pricing strategies must consider multiple factors including customer price sensitivity and elasticity, operational costs and capacity constraints, service quality differentiation, and long-term customer relationship management. Recent research has demonstrated the effectiveness of RL-based pricing strategies in both centralized platform architectures and decentralized market structures for improving revenue generation and demand-supply alignment [10, 11].

The matching process represents a critical operational decision that directly impacts service quality, resource utilization, and overall system efficiency. This optimization challenge involves not only identifying feasible assignments but optimizing across multiple criteria including: travel distance and time efficiency, shipment waiting times and service level agreements, vehicle idle time and utilization rates, vehicle capacity constraints and consolidation opportunities, service priority levels and customer preferences, and special handling requirements for specific freight types.

The complexity increases significantly when considering consolidated shipments, multiple pickup and delivery points, time windows, and dynamic re-optimization as new requests arrive. Learning-based approaches, particularly DQN and Actor-Critic methods, have demonstrated effectiveness in handling these multi-criteria matching problems in both ride-sharing and freight systems [12–14].

Dispatching efficiency encompasses both the timely execution of matched assignments and the proactive positioning of resources to meet anticipated demand. This includes real-time

rerouting based on new service requests, strategic repositioning of idle vehicles to areas of expected high demand, and coordination of fleet-wide operations to optimize system-level performance metrics [14].

RL-based dispatching methodologies have shown particular promise in handling individual vehicle decision-making while optimizing fleet-wide performance [15, 16]. These approaches demonstrate superior performance compared to traditional rule-based dispatching systems, particularly in dynamic environments with uncertain demand patterns.

2.2.3 Multi-objective and Real-time Optimization

Multi-objective optimization in freight logistics involves balancing competing goals including cost minimization, service quality maximization, resource utilization optimization, environmental impact reduction, and stakeholder satisfaction [17]. The challenge lies not only in defining appropriate trade-offs among these objectives but also in adapting these trade-offs dynamically as system conditions evolve and business priorities shift. For example, minimizing operational costs by consolidating shipments may increase delivery times, potentially compromising service quality. Conversely, prioritizing on-time delivery may require dispatching underutilized vehicles, increasing costs. RL frameworks must learn to balance such trade-offs dynamically.

Real-time operational constraints impose additional complexity by requiring decisions to be made within tight time windows while only partial information about future system states is available. This necessitates optimization approaches that can handle uncertainty, adapt to changing conditions, and learn from historical patterns while remaining computationally tractable for real-time implementation [17].

The integration of learning-based models with real-time feedback mechanisms and spatio-temporal state representations has emerged as a promising approach for addressing these challenges. These models can capture complex patterns in historical data while adapting to new conditions, providing a foundation for effective real-time decision-making in dynamic urban freight environments [18].

2.3 Classical and Traditional Optimization Approaches

2.3.1 Exact Methods and Computational Limitations

Classical exact optimization methods, including linear programming, mixed integer linear programming, and dynamic programming, have formed the mathematical foundation of

freight logistics optimization for several decades. These approaches provide theoretical optimality guarantees for well-defined problem instances and have been successfully applied to various logistics problems including facility location, network design, and vehicle routing [19–22].

However, the applicability of exact methods to real-time urban freight optimization is severely constrained by several fundamental limitations. The computational complexity of these methods typically grows exponentially with problem size, making them unsuitable for large-scale urban networks with hundreds of vehicles and thousands of shipment requests. For instance, the vehicle routing problem with time windows, a core component of freight optimization, is NP-hard, and exact solutions become computationally intractable for realistic problem sizes [23, 24].

The static nature of classical methods presents another critical limitation. These approaches require complete problem specification at the time of optimization and cannot adapt to dynamic changes in system conditions such as demand-supply fluctuations, or market condition. In urban freight environments characterized by real-time operational dynamics, this inflexibility renders classical approaches impractical for operational decision-making [25]. The integration of different freight task decisions into a unified optimization framework using classical methods results in computationally intractable formulations for realistic problem sizes [26].

2.3.2 Heuristic and Metaheuristic Methods

Heuristic and metaheuristic approaches, including genetic algorithms, tabu search, simulated annealing, and ant colony optimization, emerged as practical alternatives to exact methods by trading optimality guarantees for computational efficiency and scalability [27]. These methods can handle larger problem instances and accommodate more complex objective functions, making them more suitable for urban freight applications than exact methods.

Genetic algorithms have been extensively applied to vehicle routing problems with time windows and capacity constraints, demonstrating good performance in finding near-optimal solutions within reasonable computational times [28]. The evolutionary approach allows for exploration of diverse solution spaces and can handle multi-objective optimization through techniques such as Pareto-based selection and non-dominated sorting.

Tabu search has shown particular effectiveness in dynamic vehicle routing scenarios by maintaining search memory and avoiding cycles in solution space [29]. The adaptive memory structures in tabu search enable the algorithm to escape local optima and explore promis-

ing regions of the solution space, making it well-suited for complex logistics optimization problems.

Despite these advantages, heuristic and metaheuristic approaches suffer from significant limitations when applied to real-time urban freight optimization. They typically require problem-specific tuning of parameters and operators, limiting their generalizability across different operational scenarios and problem instances. More critically, these methods lack learning capabilities and cannot improve their performance based on historical experience or adapt their solution strategies to changing environmental conditions [30].

2.4 Learning-Based Approaches

2.4.1 Supervised and Unsupervised Learning

Traditional machine learning approaches have found extensive application in freight logistics, primarily for predictive modeling, pattern recognition, and data-driven decision support. Supervised learning methods, including linear and nonlinear regression models, decision trees, random forests, and neural networks, have been employed to forecast demand patterns, predict delivery times, estimate pricing elasticity, and classify shipment characteristics [31, 32].

These supervised approaches excel at capturing complex nonlinear relationships in historical data and can provide reasonably accurate predictions for short-term operational planning. For example, neural network models have been successfully applied to predict delivery time windows based on historical performance data, traffic patterns, and shipment characteristics [33, 34]. Similarly, regression models have been used to estimate demand elasticity for dynamic pricing applications, enabling more responsive pricing strategies.

Unsupervised learning techniques, particularly clustering algorithms such as k-means, hierarchical clustering, and DBSCAN, have been applied to customer segmentation, vehicle grouping, geographic zone creation, and demand pattern identification [35, 36]. These approaches help identify underlying structure in logistics data and inform strategic decision-making regarding service design, resource allocation, and market segmentation. However, the clustering methods often produce irregular partitions and require manual tuning. In contrast, this research employs H3 indexing as a more scalable and consistent alternative, enabling multi-resolution spatial reasoning and seamless integration with RL frameworks.

Despite their utility for predictive and analytical tasks, traditional machine learning methods have fundamental limitations for adaptive real-time decision-making in dynamic environments [37]. They typically require extensive labeled training data and perform poorly when

faced with novel situations not represented in historical datasets. More importantly, they lack the sequential decision-making capabilities needed for dynamic optimization problems where current actions influence future system states and opportunities.

2.4.2 Reinforcement Learning

Reinforcement Learning has emerged as a paradigm particularly well-suited for sequential decision-making problems in dynamic, uncertain environments. The fundamental strength of RL lies in its ability to learn optimal policies through direct interaction with the environment, without requiring explicit models of system dynamics or extensive labeled datasets [17].

In the context of urban freight logistics, RL frameworks must address unique challenges including multi-stakeholder coordination, real-time decision-making under uncertainty, and the inherent trade-offs between immediate operational efficiency and long-term system performance.The freight logistics optimization problem can be formally characterized as a Markov Decision Process, defined by the tuple (S, A, P, R, γ), where S represents the state space encompassing vehicle locations, demand patterns, and resource availability; A denotes the action space including matching decisions, pricing strategies, and resource allocation choices; P defines the state transition probabilities capturing the stochastic nature of urban transportation dynamics; R represents the reward function encoding multiple operational objectives; and γ is the discount factor balancing immediate versus future rewards. This formalism enables the agent to learn which actions maximize long-term system performance based on rewards like delivery efficiency or operational cost.

This formalism enables RL agents to learn policies that maximize long-term system performance based on rewards such as pricing, matching, routing, and dispatching efficiency, operational cost optimization, and service quality metrics [5].

The inherent complexity of urban freight systems manifests through several key characteristics: high-dimensional state spaces incorporating spatial-temporal correlations and multi-modal data; multi-objective optimization requirements balancing efficiency, cost, service quality, and sustainability; dynamic and uncertain demand patterns exhibiting both predictable and stochastic components; multi-agent interactions among competing and cooperating stakeholders; and real-time operational constraints requiring rapid decision-making capabilities within tight time windows [38, 39].

2.4.3 Deep Q-Networks and Value-Based Methods

Deep Q-Networks represent a seminal advancement in applying deep learning to reinforcement learning, particularly relevant for freight logistics applications involving discrete decision spaces [40, 41]. The DQN architecture addresses the function approximation challenges inherent in large state spaces through the integration of experience replay mechanisms and target network stabilization techniques.

The experience replay buffer D stores transitions (s_t, a_t, r_t, s_{t+1}) that are sampled uniformly during training, enabling the algorithm to break temporal correlations that can destabilize learning in sequential decision-making environments. This mechanism is particularly valuable in freight logistics applications where operational patterns exhibit strong temporal dependencies and cyclical behaviors [41].

In freight logistics applications, DQN demonstrates particular efficacy in handling high-dimensional state representations that incorporate spatial data structures, vehicle condition parameters, demand pattern distributions, and time-series features. The experience replay mechanism enhances learning efficiency by leveraging the temporal correlations commonly observed in freight operations, where similar operational conditions tend to cluster temporally [41].

However, DQN is mostly limited to discrete action spaces, which can be restrictive for many freight logistics applications requiring continuous control variables. Extensions such as Rainbow DQN and Dueling DQN have addressed some of these limitations but still face challenges in handling truly continuous action spaces [41].

2.4.4 Actor-Critic Methods and Policy Gradient Approaches

Actor-Critic methods combine the advantages of value-based and policy-based approaches by maintaining separate networks for policy representation (actor) and value function approximation (critic). This architecture addresses the high variance issues inherent in pure policy gradient methods while maintaining the capability to handle continuous action spaces essential for many freight logistics applications [12, 42].

The actor network learns a policy $\pi(a|s)$ that directly maps states to actions, while the critic network learns a value function $V(s)$ or Q-function $Q(s, a)$ that estimates the expected return from each state or state-action pair. This dual-network architecture enables more stable learning compared to pure policy gradient methods while maintaining the flexibility to handle both discrete and continuous action spaces [42].

Proximal Policy Optimization represents a significant advancement in policy gradient meth-

ods, providing stable learning through clipped objective functions that prevent large policy updates [42]. PPO has demonstrated superior performance in many logistics applications due to its robust convergence properties and computational efficiency, making it suitable for real-time applications where training time is limited.

Trust Region Policy Optimization provides stronger theoretical convergence guarantees through the enforcement of KL-divergence constraints on policy updates [43]. While TRPO offers superior theoretical properties, its increased computational complexity often makes PPO more practical for real-time freight logistics applications where computational resources and response time constraints are critical considerations.

2.4.5 Continuous Control and Deterministic Policy Gradients

Deep Deterministic Policy Gradient extends the DQN framework to continuous action spaces through the integration of actor-critic architecture with deterministic policy gradients [44]. DDPG is particularly suitable for freight logistics applications involving continuous control variables such as dynamic pricing, and resource allocation with continuous parameters.

The evolution of RL applications in freight logistics demonstrates a clear progression from simple single-agent approaches to sophisticated multi-agent frameworks capable of handling the complexity and scale of urban freight operations. The choice of specific RL methodologies depends critically on the balance between performance requirements, computational constraints, and operational risk tolerance.

2.5 Hierarchical and Multi-Agent Reinforcement Learning

Table 2.1 provides an overview of the RL algorithms discussed, serving as a foundation for the hierarchical and multi-agent architectures explored in subsequent sections.

2.5.1 Hierarchical Reinforcement Learning

Hierarchical Reinforcement Learning has emerged as a crucial advancement for managing the complexity and inherent in optimization problems. The fundamental rationale for HRL lies in its ability to decompose complex decision-making problems into manageable subproblems organized in a hierarchical structure, enabling more efficient learning and better generalization across different operational scenarios [45].

The hierarchical decomposition typically features high-level agents responsible for strategic decisions such as regional pricing policies, capacity allocation, and long-term resource

Table 2.1 Summary of Reinforcement Learning Algorithms for Freight Applications

Algorithm	**Key Characteristics**	**Advantages**	**Limitations**
Model-Free On-Policy			
Actor-Critic(PPO) [42]	Policy gradient with clipping	Stable learning, handles continuous actions	Sample inefficient, requires direct interaction
Actor-Critic(TRPO) [43]	Trust region constraints	Theoretical guarantees	Computationally expensive
Model-Free Off-Policy			
Actor-Critic(DDPG) [44]	Deterministic policy gradient	Handles continuous control	Sensitive to hyperparameters
DQN [41]	Deep Q-learning	Proven stability, experience replay	Computation cost

planning, while low-level agents handle tactical operations including individual vehicle dispatching, route optimization, and real-time adjustments [45]. This decomposition allows the system to adapt to different temporal scales of decision-making, with strategic decisions updated less frequently than operational ones.

Critical analysis of existing HRL studies reveals several key advantages: improved sample efficiency through task decomposition and knowledge transfer, better transferability of learned policies across different scenarios and problem instances, enhanced interpretability of decision-making processes through hierarchical structure, and reduced computational complexity compared to flat RL approaches for large-scale problems [45].

However, limitations include the challenge of defining appropriate hierarchical structures for specific problem domains, the potential for suboptimal coordination between hierarchical levels, and the difficulty of learning effective high-level policies when low-level policies are still evolving [45].

2.5.2 Multi-Agent Reinforcement Learning

Multi-agent reinforcement learning frameworks address the inherently distributed nature of urban freight systems where multiple autonomous decision-makers must coordinate their actions to achieve system-wide efficiency [11]. MARL approaches recognize that urban freight optimization involves multiple autonomous agents that must learn to coordinate while potentially having conflicting local objectives and incomplete information about global system

state.

Recent research in transportation platforms has implemented MARL [13,16,46]. For example, a decentralized MARL framework allows individual vehicles to learn local policies while using a centralized value network for cooperative training, enabling scalable coordination without requiring centralized control during execution [47].

Centralized Training with Decentralized Execution has emerged as a particularly effective paradigm for urban freight applications [46, 48]. This approach allows agents to benefit from global information and coordination during training while maintaining autonomous decision-making capabilities during execution. CTDE addresses the scalability challenges of centralized control while avoiding the coordination failures common in fully decentralized systems.

The CTDE framework is particularly well-suited for freight logistics applications because it allows for: efficient coordination during training using global system information, scalable execution with decentralized agent decision-making, robustness to communication failures during operational deployment, and adaptation to varying network topologies and agent populations [49].

Studies have demonstrated that MARL frameworks can effectively handle joint optimization of pricing and dispatching in ride-sourcing platforms, which share many operational characteristics with freight systems [50]. These approaches demonstrate superior performance in revenue generation, service quality metrics, and resource utilization compared to rule-based or single-agent alternatives.

2.6 Spatial Partitioning and Geospatial Analysis

2.6.1 Spatial Partitioning Techniques

Spatial partitioning represents a fundamental component of urban freight optimization, providing the geographic framework for organizing and coordinating logistics operations across urban areas. Traditional clustering methods, including k-means clustering, hierarchical clustering, and DBSCAN, have been widely used to partition urban areas based on demand density, geographic proximity, and operational efficiency considerations [4, 51].

K-means clustering offers computational efficiency and produces balanced partitions, making it suitable for resource allocation applications where equal workload distribution is desired. However, k-means struggles with irregular geographic shapes and varying demand densities typical of urban environments, often producing partitions that do not align well with natural

geographic boundaries [52].

DBSCAN addresses some limitations of k-means by identifying clusters of arbitrary shapes and effectively handling outliers, which is important in urban environments with irregular demand patterns. However, DBSCAN requires careful parameter tuning and may produce unbalanced partitions that are unsuitable for equitable resource allocation across service regions [53].

Hierarchical clustering methods provide multi-scale partitioning capabilities that can be useful for hierarchical optimization frameworks. However, these methods typically have high computational complexity and may not produce partitions that are well-suited for real-time operational decision-making [54].

The H3 Hexagonal Hierarchical Spatial Index represents a significant advancement in spatial partitioning for urban applications, offering several advantages over traditional clustering-based methods . H3 provides a multi-resolution hexagonal grid system that addresses many limitations of traditional spatial partitioning approaches.

Key advantages of H3 include: uniform geographic coverage without gaps or overlaps, ensuring complete spatial coverage; hierarchical structure enabling multi-scale analysis and optimization; consistent neighbor relationships that facilitate spatial optimization algorithms; efficient indexing and query capabilities for real-time applications; and standardized spatial resolution that enables consistent performance across different geographic regions.

Comparative analysis reveals that H3's hexagonal structure provides more uniform distance relationships compared to square grids, reducing edge effects and improving the accuracy of spatial optimization algorithms. The hierarchical nature of H3 enables seamless integration with hierarchical RL frameworks, where different decision-making levels can operate at different spatial resolutions [55].

The multi-resolution capabilities of H3 are particularly valuable for freight logistics applications where different types of decisions require different spatial granularity. For example, strategic planning decisions might use coarse-resolution H3 cells covering large urban areas, while operational dispatching decisions might use fine-resolution cells covering individual neighborhoods.

2.6.2 Integration with RL Frameworks

The integration of spatial partitioning methods with RL frameworks has shown significant potential for improving the effectiveness of urban freight optimization. Spatial partitioning provides the state space structure that RL algorithms use to organize and generalize learning

across different geographic regions.

Several studies have demonstrated successful integration of spatial partitioning with RL for transportation applications [12, 14, 18]. For example, in [56], a spatio-temporal deep RL model jointly learns pricing and dispatching policies using grid-based city representations enhanced with demand forecasts.

Graph-based representations that encode spatial relationships among partitioned regions have been particularly effective, enabling RL agents to leverage geographic structure for improved decision-making [57, 58].

However, research gaps remain in this integration, particularly limited research has explored the integration of H3 spatial partitioning with RL frameworks, representing a significant opportunity for methodological innovation.

2.7 Technical Limitations and Existing Research Gaps

2.7.1 Scalability Challenges

Systematic analysis of existing literature reveals several critical limitations that hinder the practical application of current optimization methods to urban freight transportation. Scalability remains a fundamental challenge, as most existing RL approaches have been demonstrated on relatively small problem instances or simplified scenarios that do not reflect the complexity of real-world urban freight networks [12, 59].

The scalability challenge manifests in several dimensions: computational complexity that grows exponentially with the number of agents, vehicles, or spatial regions; real-world urban freight networks involve thousands of vehicles and tens of thousands of daily shipment requests. The gap between research prototypes and deployment-ready systems remains substantial, limiting the practical impact of academic advances [60].

2.7.2 Multi-Objective Integration Limitations

Current approaches typically optimize individual objectives or pairs of objectives separately, lacking integrated frameworks that can simultaneously optimize multiple freight operations decisions while accounting for their interdependencies [61, 62]. For instance, optimizing pricing strategies without considering their impact on matching efficiency and dispatching costs can lead to suboptimal system-wide performance.

The lack of integrated multi-objective optimization is particularly problematic in freight logistics where these decisions are inherently coupled. Pricing decisions directly influence

demand patterns, which affect optimal matching strategies, which in turn impact dispatching efficiency and resource utilization [63]. Independent optimization of these components can lead to conflicting objectives and suboptimal overall system performance.

This limitation is exacerbated by the difficulty of defining appropriate reward functions that capture the complex trade-offs among multiple objectives. Most existing approaches does not use combinations of objectives, which may not capture the relationships and dynamic trade-offs that characterize real-world multiple freight operations [64].

To consolidate insights across the reviewed literature, Table 2.2 provides a comparative analysis, highlighting the current state of research and development in the online transportation sector. The table is crucial for identifying both the advancements and the persisting challenges within the freight domain. By mapping studies across key capabilities—such as real-time matching, dynamic pricing, consolidation, and freight-specific focus—it offers a clear overview of where research efforts have been concentrated and where significant opportunities for future exploration remain.

This comparative analysis reveals several key insights: very few studies address joint optimization of multiple objectives simultaneously; real-time pricing capabilities are particularly underdeveloped; freight-specific considerations are addressed in only a subset of studies; and consolidation handling, which is critical for freight efficiency, is rarely integrated with RL approaches.

2.7.3 Identification of Research Gaps

The systematic review reveals several critical gaps in the existing literature that justify the need for new research approaches:

- **Gap 1: Limited Joint Optimization Frameworks** — Most existing studies focus on individual optimization problems rather than integrated frameworks that can simultaneously optimize multiple objectives while accounting for their interdependencies.
- **Gap 2: Insufficient Real-Time Capabilities** — Current approaches often lack the efficiency and responsiveness required for real-time operational decision making in dynamic urban freight environments.
- **Gap 3: Scalability Limitations** — Traditional optimization methods struggle to scale in the face of large, real-world urban freight networks involving thousands of vehicles and dynamic shipment requests. Their computational complexity grows rapidly

Table 2.2 Reference Comparison Table

Reference	**Matching**	**Real-Time Matching**	**Pricing**	**Real-Time Pricing**	**Joint Matching-Pricing**	**Consolidation**	**Freight**
[40]	✓	✓	✓	×	×	×	×
[10]	✓	✓	×	×	×	×	×
[11]	✓	✓	×	×	×	×	×
[65]	✓	×	×	×	×	×	×
[59]	✓	✓	×	×	×	×	×
[66]	✓	✓	×	×	×	×	×
[67]	✓	✓	✓	×	✓	✓	×
[12]	✓	✓	×	×	×	×	×
[13]	×	×	×	×	×	×	×
[15]	✓	✓	×	×	×	×	×
[16]	✓	✓	×	×	×	×	×
[47]	✓	✓	×	×	×	×	×
[68]	✓	✓	×	×	×	×	×
[56]	✓	✓	×	×	×	×	×
[69]	✓	✓	×	×	×	✓	✓
[70]	×	×	×	×	×	×	×
[48]	✓	✓	×	×	×	✓	✓
[46]	✓	×	×	×	×	×	✓
[18]	×	×	✓	×	×	×	✓
[71]	×	×	✓	×	×	×	✓
[50]	×	×	✓	✓	×	×	×
[49]	×	×	✓	✓	×	×	×
[72]	×	×	✓	✓	×	×	×
[73]	×	×	×	×	×	×	×
[74]	×	×	✓	×	×	×	×
[4]	×	×	✓	✓	×	×	×
[75]	×	×	✓	✓	×	×	×
[76]	✓	×	✓	×	×	×	✓
[57]	×	×	×	✓	×	×	×
[77]	×	×	✓	✓	×	×	×
[78]	×	×	✓	✓	×	×	×

with problem size, rendering them impractical for real-time decision-making at urban scale.

- **Gap 4: Lack of Comparative Evaluation of RL Algorithms** — Existing freight logistics studies rarely compare the capabilities of different RL algorithms such as DQN, PPO, TRPO, and DDPG under consistent conditions. This absence of benchmarking obscures the strengths and limitations of value-based vs. actor-critic methods for handling continuous actions, real-time constraints, and dynamic environments. A systematic, domain-specific comparison is essential to guide algorithm selection and ensure practical deployment.
- **Gap 5: Inadequate Spatial Modeling** — Current spatial partitioning methods often use clustering that does not capture the dynamic nature of urban freight operations.
- **Gap 6: Insufficient Multi-Agent Coordination** — While some multi-agent approaches exist, they typically assume fixed agent roles and do not address the dynamic task-switching requirements of real-world freight operations.

2.8 Justification and Positioning of the Proposed Research

The review of existing literature on optimization and reinforcement learning in urban freight logistics reveals several key gaps that motivate the need for a more integrated, scalable, and adaptive decision-making framework. While prior work has made considerable progress in specific areas—such as dynamic pricing [65], ride-matching [12], or vehicle dispatching [16]—very few studies have addressed the joint optimization of multiple components in a unified framework suitable for freight operations.

A major limitation across many RL-based studies is the focus on single-agent systems or fixed task structures. In contrast, real-world freight platforms operate in environments where agents must frequently switch between different tasks based on the current operational context. This dynamic task-switching is rarely modeled in traditional RL architectures and is typically handled through hardcoded business rules or independent modules, leading to inefficiencies.

Additionally, many models rely on cluster-based spatial partitioning or simplified state representations. Tools like H3 hexagonal grids may improve upon earlier approaches, combining hierarchical spatial partitioning with RL architectures in a way that addresses the scalability and adaptability requirements of real-world freight systems.

This dissertation advances the state of the art by proposing several key innovations that address the identified research gaps:

- Demonstrating the superiority of ReLU over Tanh activation functions in our models;
- Developing a Hierarchical Reinforcement Learning framework that integrates matching and dispatching tasks within a spatially hierarchical structure;
- Establishing the advantages of Actor-Critic methods over Deep Q-Networks (DQN) for continuous and complex decision-making;
- Demonstrating the superiority of H3 spatial indexing over clustering-based methods in capturing geographic granularity and ensuring consistent spatial resolution;
- Introducing a Nested Hierarchical Reinforcement Learning framework that jointly optimizes pricing, matching, and dispatching under spatial hierarchy;
- Validating the superiority of Proximal Policy Optimization over Trust Region Policy Optimization and Deep Deterministic Policy Gradient in the studied freight logistics scenarios;
- Proposing a Context-Aware Structure-Switching MARL model that enables per-vehicle autonomy and adaptive task switching among matching, routing, and dispatching;
- Introducing a decentralized Prioritization Module that enables each vehicle to compute dynamic context-aware scores based on capacity, load ratio, and service time, facilitating autonomous task prioritization and emergent coordination without centralized control.
- Designing real-time-capable architectures through modular pre-filtering algorithms and scalable system design;
- Implementing a multi-resolution spatial framework based on H3 indexing to enhance geographic adaptability and precision.

These innovations address core limitations of existing methods—namely, poor scalability, lack of joint decision-making, weak adaptability, and computational impracticality in real-time freight systems. They position this research at the intersection of operational optimization, artificial intelligence, and smart transportation. To provide a structured overview of how these innovations are distributed across the four core articles of this dissertation, figure 2.1 presents a mapping table that links each contribution to its corresponding article.

Innovation	Article 1 (DQN + H3)	Article 2 (HRL + Actor-Critic)	Article 3 (NHRL + PPO)	Article 4 (MARL + Context Switching)
Logistics Problem Scope	Single-objective matching at a local scale	Joint matching and dispatching	Multi-objective optimization (pricing, matching, dispatching)	Large-scale coordination via decentralized vehicle autonomy
H3 Indexing	✓	✓	✓	✓
Modular Pre-filtering Algorithms	✓	✓	✓	✓
ReLU vs. Tanh Activation	✓			
Hierarchical RL for Matching & Dispatching		✓		
Actor-Critic vs. DQN		✓		
H3 Indexing vs. Clustering		✓		
NHRL for Pricing-Matching-Dispatching			✓	
PPO vs. TRPO vs. DDPG			✓	
Multi-resolution Spatial Framework			✓	
Context-Aware Structure-Switching MARL				✓
Decentralized Prioritization Module				✓

Figure 2.1 Mapping of Key Innovations to Thesis Articles. Each checkmark indicates the article in which the corresponding innovation is introduced or evaluated.

2.9 Summary and Research Positioning

This literature review reveals a clear progression from classical optimization methods through heuristic approaches to modern learning-based frameworks, with particular emphasis on the evolution toward hierarchical and multi-agent reinforcement learning systems. Classical methods, while providing optimality guarantees, prove inadequate for the scale and dynamic nature of urban freight optimization. Heuristic approaches offer improved computational efficiency but lack learning capabilities and adaptability to changing operational conditions.

The emergence of RL has provided new opportunities for adaptive optimization in dynamic environments, with hierarchical and multi-agent extensions addressing scalability and coor-

dination challenges. However, significant gaps remain in terms of integrated multi-objective optimization, real-time adaptability, and scalable implementation for urban-scale networks.

The identified research gaps justify the need for hierarchical RL frameworks that can simultaneously optimize multiple interconnected decisions while adapting to changing conditions and scaling to realistic urban freight networks. The integration of advanced spatial partitioning methods with sophisticated multi-agent coordination mechanisms represents a promising direction for addressing these challenges.

Key insights from this review include the importance of hierarchical decomposition for managing complexity, the effectiveness of multi-agent coordination for distributed decision-making, the potential of spatial partitioning for scalable urban optimization, and the critical need for integrated approaches that recognize the interdependencies among multiple decisions.

This literature synthesis establishes the theoretical foundation for developing next-generation RL architectures that can simultaneously address scalability, adaptability, and integration challenges in urban freight logistics. The identified gaps directly inform the design of the four progressive RL frameworks presented in subsequent chapters, each building systematically upon the theoretical insights and methodological advances reviewed in this chapter.

The comparative analysis presented in Table 2.2 consolidates these insights, revealing that while individual components have received significant research attention, integrated frameworks capable of joint optimization remain largely unexplored, representing a critical opportunity for methodological advancement.

CHAPTER 3 RESEARCH METHODOLOGY

3.1 Overview of Methodological Approach

This research adopts a design science research methodology, focusing on the creation and rigorous evaluation of computational artifacts to address complex organizational problems in urban logistics. The methodology follows an iterative process, progressing from problem conceptualization and architectural design to simulation-based validation and performance benchmarking.

Reinforcement Learning was selected as the core algorithmic paradigm because the decision-making problems in freight logistics—specifically pricing, matching, and dispatching—are inherently stochastic and sequential. Unlike static optimization methods, which struggle with real-time uncertainty, RL agents can learn adaptive policies that account for the long-term consequences of immediate actions, mathematically modeled as Markov Decision Processes. To manage the computational complexity of city-scale operations, we integrated Uber's H3 hierarchical spatial indexing. This geospatial abstraction transforms continuous urban coordinates into discrete, manageable hexagonal units, enabling multi-resolution reasoning that balances local precision with global scalability.

To synthesize the connections between the data inputs, algorithmic processing, and decision outputs across the dissertation's four contributions, Figure 3.1 illustrates the global decision flow. The process initiates with raw Shipment and Vehicle Data, which is processed through H3 Spatial Indexing and domain-specific Pre-filtering modules (PAMA, ShipScan, PADA, DEZE) to reduce state-action dimensionality. These spatially filtered states are subsequently ingested by the RL Agents (utilizing DQN and Actor-Critic architectures), which execute specific operational tasks to generate optimized Logistics Decisions.

3.2 Problem Formulation

The core objective of this research is to develop RL approaches capable of optimizing interdependent decision-making processes in urban freight logistics: dynamic pricing, shipment-vehicle matching, vehicle dispatching, and routing. These processes are inherently coupled and must be addressed jointly to ensure system-wide efficiency, responsiveness, and profitability.

To formalize the problem, each decision process is modeled as a Markov Decision Process,

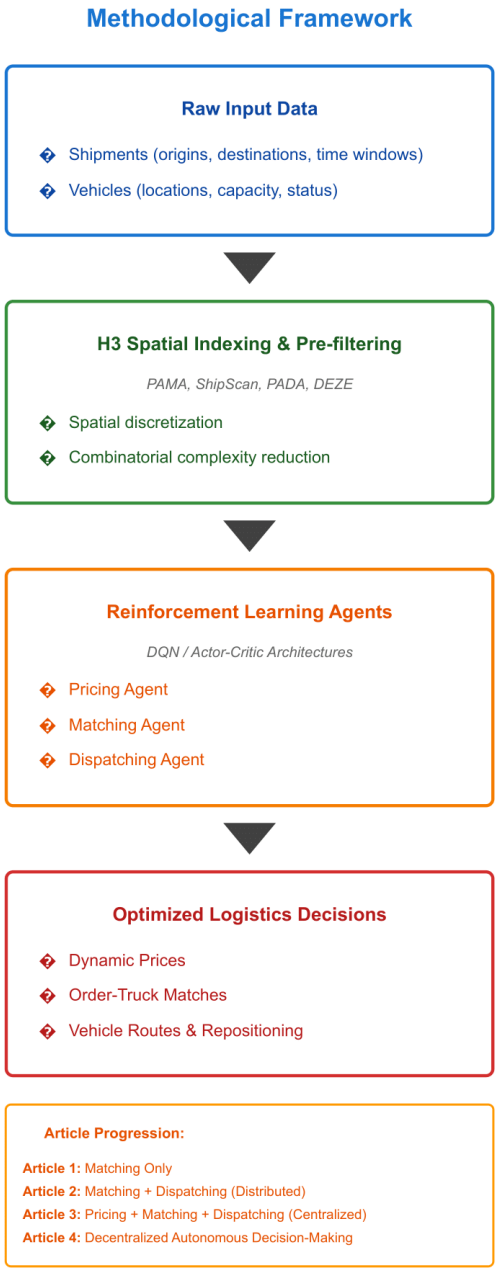

Figure 3.1 Global Synthesis of the Methodological Framework: From Data Input to Operational Decisions.

defined by the tuple:

$$(S, A, P, R, \gamma) \tag{3.1}$$

Where:

- **S** is the state space, representing the current configuration of the freight system, including vehicle locations, shipment requests, time windows, and freight environment conditions.
- **A** is the action space, encompassing decisions such as price adjustments, vehicle-to-shipment assignments, repositioning of idle vehicles, and routing.
- **P** is the state transition function, capturing the probabilistic evolution of the system in response to actions taken under uncertainty.
- **R** is the reward function, quantifying the desirability of outcomes based on operational objectives such as revenue, service quality, and resource utilization.
- $\boldsymbol{\gamma}$ is the discount factor, balancing immediate and future rewards.

Each RL agent is trained to learn a policy $\pi(a \mid s)$ that maximizes the expected cumulative reward over time. The freight environment is dynamic and stochastic, with shipment requests arriving continuously and vehicle availability fluctuating due to service constraints and geographic dispersion.

To manage the complexity of the urban freight network, the system is spatially partitioned using the H3 hierarchical hexagonal indexing system, which enables multi-resolution decision-making, allowing for scalable coordination across geographic zones.

This formalization provides the foundation for designing RL architectures that are both scalable and adaptable, capable of learning optimal policies in complex, real-time freight environments.

3.3 Reinforcement Learning Algorithms

This section presents the RL algorithms employed in the dissertation to address the dynamic and multi-objective nature of urban freight logistics. The selection of algorithms is guided by the characteristics of each decision-making task and the need for scalability, stability, and adaptability in real-time environments.

3.3.1 Value-Based Methods

Value-based methods estimate the expected return of actions and derive policies by selecting actions that maximize this value. In this research, DQN are used in Article 1 to optimize shipment-vehicle matching. DQN is particularly effective for discrete action spaces and enables stable learning through mechanisms such as experience replay and target network updates. The agent learns a Q-function $Q(s, a)$ that approximates the expected cumulative reward for taking action a in state s, and selects actions using an ϵ-greedy strategy to balance exploration and exploitation.

3.3.2 Actor-Critic Methods

Actor-Critic methods combine the strengths of value-based and policy-based approaches. The actor learns a policy $\pi(a|s)$ that maps states to actions, while the critic estimates the value function $V(s)$ or the action-value function $Q(s, a)$. This dual-network architecture improves learning stability and supports continuous action spaces as well. In Article 2, Article 3, Article 4, Actor-Critic methods are employed to jointly optimize matching and dispatching tasks. Specifically, Proximal Policy Optimization is adopted due to its robust convergence properties and computational efficiency. PPO uses a clipped objective function to prevent large policy updates, ensuring stable learning in dynamic environments.

3.3.3 Hierarchical Reinforcement Learning

In Article 2, a Hierarchical Reinforcement Learning framework is introduced to manage the complexity of joint decision-making. HRL decomposes the problem into high-level and low-level agents, where the low-level agent coordinates dispatching across zones, and the high-level agent handles matching shipments to vehicles. This structure improves sample efficiency and enables modular policy learning.

3.3.4 Nested Hierarchical Reinforcement Learning

Article 3 introduces a Nested Hierarchical Reinforcement Learning framework to jointly optimize pricing, matching, and dispatching decisions in urban freight logistics. The NHRL architecture features three coordinated RL agents—each responsible for one of the core decision tasks—operating across multiple spatial resolutions using the H3 indexing system. Pricing and matching agents operate at a finer resolution (H3 level 7), while the dispatching agent operates at a coarser resolution (H3 level 6), enabling scalable coordination across geographic zones.

The nested structure allows for modular learning, where each agent focuses on its specific task while interacting through a centralized coordination hub. This design improves sample efficiency, supports multi-resolution reasoning, and enables system-wide optimization. PPO is used as the learning algorithm for all agents due to its stability and suitability for high-dimensional, complex decision-making problems.

3.3.5 Multi-Agent Reinforcement Learning

Article 4 introduces a Multi-Agent Reinforcement Learning framework to enable vehicle-level autonomy and dynamic task switching. Each vehicle operates as an independent agent capable of switching between matching, routing, and dispatching tasks based on its operational context. The MARL architecture supports decentralized decision-making, lightweight peer-to-peer communication emergent coordination through shared policies and local prioritization mechanisms. This approach enhances scalability and responsiveness in large-scale freight networks.

3.3.6 Algorithm Selection Justification

The choice of RL algorithms is based on empirical performance, computational feasibility, and alignment with the operational characteristics of urban freight logistics. PPO is selected as the primary learning algorithm for HRL, NHRL, and MARL frameworks due to its balance between stability and sample efficiency. DQN is used for discrete matching tasks where action spaces are well-defined and limited. The HRL, NHRL, and MARL paradigms ensure that the proposed architectures can adapt to varying levels of complexity, modularity, and decentralization across different articles.

3.4 System Architecture

Each article in this dissertation introduces a distinct system architecture tailored to its specific decision-making scope and reinforcement learning paradigm. This section summarizes the architectural designs employed across the four articles, highlighting their structural components, agent configurations, and coordination mechanisms.

3.4.1 Article 1: Single-Agent Architecture for Matching

Article 1 presents a centralized, single-agent architecture based on DQN for real-time shipment-vehicle matching. This foundational approach establishes the core technologies and validates

fundamental concepts in freight optimization. The system includes:

- A single RL agent trained to select optimal vehicle matches for incoming shipment requests.
- A pre-filtering algorithm that reduces the candidate vehicle pool based on H3 spatial partitioning (resolution level 7), capacity, quality service, and service time.
- A structured state space encoding shipment and vehicle attributes, and a discrete action space representing match decisions.
- No coordination hub is required, as decisions are made centrally by the agent.

This architecture demonstrates the feasibility of RL-based matching but is constrained to single-task optimization without multi-objective decision-making.

3.4.2 Article 2: Dual-Agent Hierarchical Architecture

Article 2 introduces a hierarchical reinforcement learning framework that addresses the limitation of single-task focus by enabling joint optimization of matching and dispatching. The transition from single-agent to dual-agent HRL was necessary to move beyond simple matching and coordinate dispatching decisions that optimize fleet positioning for future demand. The system features:

- A matching agent responsible for pairing shipments with vehicles.
- A dispatching agent that repositions idle vehicles to high-demand zones.
- Both agents operate under a centralized coordination hub that maintains system state and synchronizes decisions.
- The architecture uses H3 spatial indexing at resolution level 7 for matching and dispatching.
- Pre-filtering modules (PAMA and PADA) are used to reduce action space complexity.

While this approach successfully integrates matching and dispatching, it lacks dynamic pricing capabilities and operates at a single spatial resolution, limiting its ability to handle strategic and operational decisions simultaneously.

3.4.3 Article 3: Nested Hierarchical Architecture

Article 3 proposes a Nested Hierarchical Reinforcement Learning framework that addresses the pricing limitation by jointly optimizing pricing, matching, and dispatching. The move to three-agent NHRL was essential to capture the interdependencies between pricing strategies and operational efficiency, as pricing decisions directly influence demand patterns and resource utilization:

- Three specialized agents: pricing, matching, and dispatching, each trained independently using PPO.
- A centralized coordination hub manages shared state information and synchronizes agent actions.
- Pricing and matching agents operate at H3 resolution level 7; dispatching operates at resolution level 6.
- Modular pre-filtering algorithms (PAMA and DEZE) are integrated to reduce computational overhead.
- The architecture supports multi-resolution spatial reasoning and system-wide coordination.

Although this framework achieves comprehensive optimization, it introduces increased complexity in coordination and training overhead. The reliance on a centralized hub, while beneficial for synchronization, may become a bottleneck under high-frequency decision-making scenarios.

3.4.4 Article 4: Decentralized Multi-Agent Architecture

Article 4 introduces a fully decentralized Multi-Agent Reinforcement Learning framework that enables scalable learning and decision-making across heterogeneous agent types by vehicle-level autonomy. The transition to decentralized MARL was necessary to achieve real-time adaptability, reduce centralized coordination overhead, and support interpretable context-aware task switching. This design empowers each vehicle to autonomously select and execute operational tasks—such as matching, routing, or dispatching—based on local conditions, thereby enhancing system responsiveness, scalability, and robustness in dynamic urban freight environments.:

- Each vehicle is modeled as an independent agent capable of switching between three internal decision structures: matching, routing, and dispatching.
- No centralized coordination hub is used; agents operate autonomously based on local context.
- Task-switching is governed by interpretable context variables such as load ratio, shipment count, and idle time.
- Pre-filtering modules (ShipScan and PADA) are used locally by each agent to constrain decision space.
- Shared policy networks are employed across agents to ensure scalability and training efficiency.

Each architecture is designed to reflect the operational realities and decision-making requirements of urban freight logistics, ranging from centralized optimization to fully distributed autonomy.

3.5 Agent Design and Learning Structures

This section details the design of reinforcement learning agents and their learning structures across the four articles. Each article introduces a distinct agent configuration tailored to its operational scope—ranging from centralized single-agent models to decentralized multi-agent frameworks with dynamic task switching.

3.5.1 Article 1: Single-Agent Matching Agent

Article 1 employs a centralized DQN agent for shipment-vehicle matching. The agent operates over a discrete action space and is trained using experience replay and target network updates. The state space encodes shipment and vehicle attributes, while the action space represents candidate vehicle selections. The agent learns a Q-function $Q(s, a)$ to estimate expected rewards and selects actions using an ϵ-greedy strategy. The learning structure is straightforward, with a single neural network trained to optimize matching efficiency.

3.5.2 Article 2: Hierarchical Reinforcement Learning

Article 2 introduces a two-Agents HRL framework with two actor-critic agents:

- **Matching Agent:** Operates at H3 resolution level 7, selecting optimal vehicle-shipment pairings using a pre-filtered candidate pool.
- **Dispatching Agent:** Operates at H3 resolution level 6, repositioning idle vehicles to high-demand zones.

Each agent uses PPO algorithm for stable learning. The coordination hub synchronizes agent actions and updates the environment state. The agents are trained independently with task-specific state spaces, reward functions, and action definitions.

3.5.3 Article 3: Nested Hierarchical Reinforcement Learning

Article 3 presents a three-agents NHRL framework for joint optimization of pricing, matching, and dispatching:

- **Pricing Agent:** Adjusts vehicle prices dynamically based on local supply-demand conditions.
- **Matching Agent:** Assigns shipments to vehicles to maximize revenue and service quality.
- **Dispatching Agent:** Repositions idle vehicles across broader zones to anticipate demand.

Each agent operates at a distinct spatial resolution and is trained using PPO. The coordination hub manages shared state information and ensures sequential decision-making. Modular pre-filtering algorithms (PAMA and DEZE) reduce action space complexity. The agents interact indirectly through shared environment updates.

3.5.4 Article 4: Decentralized Multi-Agent Reinforcement Learning

Article 4 introduces a fully decentralized MARL framework where each vehicle(via its embedded on-board software agent) is modeled as an independent agent capable of switching between three internal decision structures:

- **Matching Structure:** Activated when the vehicle is empty or underutilized.
- **TSR Structure:** Activated when the vehicle carries multiple shipments, optimizing delivery sequence.

- **Dispatching Structure:** Activated when the vehicle is idle beyond a threshold, repositioning to high-demand zones.

Each structure is trained using PPO with shared policy networks across the fleet. Vehicles dynamically switch between structures based on interpretable context variables such as load ratio, shipment count, and idle time. Experience buffers are maintained separately for each structure, and training is performed using structure-specific mini-batches. The shared-policy architecture ensures scalability and generalization, while pre-filtering modules (ShipScan and PADA) constrain decision spaces for real-time feasibility.

3.5.5 Reward Shaping Strategy

Across all architectures, the reward functions were designed using a multi-objective shaping approach. The primary objective (revenue or match success) acts as the dominant signal, while secondary operational metrics (distance, idle time, service quality) are integrated as weighted penalty terms. This composite structure ensures that agents optimize for global system performance without compromising local operational feasibility. The weights assigned to these components were calibrated iteratively to prevent any single objective from destabilizing the learning process, ensuring a balanced policy that aligns vehicle-level actions with platform-level goals.

3.6 Simulation Environment and Datasets

To evaluate the performance of the proposed reinforcement learning frameworks, a simulation environment was developed to replicate realistic urban freight operations. The simulation integrates spatial, temporal, and operational dynamics using synthetic datasets that reflect real-world logistics conditions.

3.6.1 Simulation Environment

The simulation environment models urban freight logistics using H3 hexagonal spatial partitioning at multiple resolution levels. Vehicles and shipment requests are distributed across the city grid, and decisions are made at each time step based on agent policies. The environment assumes that vehicles can traverse approximately one hexagon at resolution level 6 per time step, corresponding to an effective travel distance of approximately 7.45 kilometers. This spatial-temporal mapping enables the integration of geographic and time-based constraints into the decision-making process.

For Articles 1, 2, and 3, the simulation adheres to two fundamental assignment principles:

- **Vehicle Assignment:** Each vehicle may be assigned to at most one shipment or remain idle.
- **Shipment Assignment:** Each shipment request may be assigned to at most one vehicle or remain unassigned.

In contrast, in Article 4, vehicles are allowed to consolidate multiple shipments, enabling one-to-many matching. This reflects real-world less-than-truckload operations and supports more flexible and efficient resource utilization. The simulation environment for Article 4 includes:

- **Consolidated Matching:** Vehicles can accept multiple shipments based on capacity and service constraints.
- **Dynamic Task Switching:** Vehicles autonomously switch between matching, routing, and dispatching tasks based on operational context.

3.6.2 Datasets

The simulation relies on two primary datasets:

- **Origin-Destination (O-D) Data:** This dataset includes details of shipping requests, such as geocodes for pickup and drop-off locations, H3 hexagon IDs, volume type, availability, quality scores, and timestamps. Geocodes are indexed by latitude and longitude. Volume and capacity are categorized differently across articles: in Articles 1 and 3, they are grouped into three main types—light, medium, and heavy—while in Articles 2 and 4, a more granular classification is used, with nine subtypes: light (1, 2, 3), medium (1, 2, 3), and heavy (1, 2, 3), allowing for finer control over matching and dispatching decisions, distributed across 100 discrete timestamps (T1 to T100).
- **Truck Data:** This dataset contains vehicle geolocation at the time of availability, vehicle capacity (light, medium, heavy), quality scores, and service time windows. Vehicles are spatially distributed across the H3 grid and dynamically updated throughout the simulation.

The utilization of two distinct urban contexts—Montréal and Toronto—enables comprehensive evaluation that demonstrates the generalizability and robustness of the proposed models across different urban geographies, population densities, and network topologies. This multi-city approach represents a significant research contribution as it validates that the RL frameworks can adapt to varying urban scales and spatial configurations, enhancing their practical applicability for real-world deployment across diverse metropolitan areas.

These datasets were synthetically generated using domain-informed distributions and spatial density functions to reflect realistic urban freight patterns. They support the training and evaluation of RL agents under diverse operational scenarios.

3.7 Evaluation Metrics and Experimental Setup

To assess the performance of the proposed RL frameworks, a comprehensive set of evaluation metrics and experimental configurations were employed. These metrics capture operational efficiency, service quality, and system responsiveness across different agent architectures and urban freight scenarios.

3.7.1 Evaluation Metrics

The following metrics were used to evaluate agent performance across all articles:

- **Cumulative Reward:** Measures the total reward accumulated by the agent over the simulation horizon, reflecting overall policy effectiveness.
- **Revenue:** Total monetary value generated through successful shipment-vehicle matches, influenced by dynamic pricing strategies.
- **Successful Matches:** Number of shipment requests successfully paired with real vehicles, indicating system throughput.
- **Fictitious Matches:** Number of matches made to placeholders due to insufficient availability, used to assess supply-demand alignment.
- **Vehicle Idle Time:** Average duration vehicles remain unassigned, reflecting fleet utilization efficiency.
- **Shipment Waiting Time:** Average time shipments wait before being matched, indicating service responsiveness.

- **Pickup Distance:** Average distance traveled by vehicles to reach shipment origins, used to evaluate spatial efficiency and sustainability.
- **Training Time:** Total time required to train the agent under different neural network configurations.

These metrics were consistently applied across all articles to ensure comparability and to highlight the strengths and limitations of each architectural approach.

3.7.2 Baseline Models and Comparative Framework

To rigorously evaluate the proposed RL architectures, each article incorporates baseline models and configuration variants tailored to its methodological scope. These baselines serve as reference points for assessing improvements in performance metrics. Statistical significance of the results was validated through Analysis of Variance (ANOVA) testing, ensuring the robustness and reliability of the proposed models.

Article 1 includes the following baseline comparisons:

- Different activation functions (ReLU vs. Tanh).
- Varying neuron counts per layer (64 to 1024).

Key performance metrics—such as reward per match, distance per match, earliest service time, and training duration—are used to identify optimal network configurations.

Article 2 evaluates the HRL framework against:

- Multiple actor-critic configurations with varying neuron sizes for both matching and dispatching agents.
- A centralized DQN agent using H3 partitioning.
- An actor-critic model with clustering-based spatial zoning.

Performance is assessed using training time, average reward, successful matches, and empty mileage.

Article 3 evaluates the HRL framework against:

- Fixed-price matching (1-agent baseline).

- Joint pricing-matching (2-agent baseline).
- Full NHRL (3-agent model with dispatching).
- Algorithmic comparisons across PPO, TRPO, and DDPG.

Experiments span two urban contexts—Montréal and Toronto—and assess metrics such as cumulative reward, revenue, successful and fictitious matches, idle time, and shipment waiting time.

Article 4 evaluates the Full MARL framework against five progressively enhanced system variants

1. **Baseline Global**: A centralized RL agent with monolithic matching logic.
2. **Baseline MARL**: Decentralized matching agents without task switching or prioritization.
3. **LTL + TSR**: Reactive switching between matching and routing based on load state.
4. **Ctx1**: capacity and size-based prioritization scoring.
5. **Ctx2**: Enhances Ctx1 with time-service urgency scoring.

Performance was evaluated using successful and fictitious matches, idle time, and shipment waiting time, pickup distance, used trucks, and utilization gain.

Across all articles, baseline comparisons are conducted using consistent metrics to ensure comparability and to highlight the strengths and limitations of each architectural innovation. These evaluations are detailed in Chapters 4 through 7.

3.7.3 Experimental Setup

Experiments were conducted using simulated urban freight environments for two Canadian cities: Montréal and Toronto. The datasets included:

- **Montréal:** 9,866 shipment requests and 11,397 vehicles.
- **Toronto:** 17,351 shipment requests and 20,000 vehicles.

The simulation was executed on a workstation equipped with an Intel Core i7 processor, 64 GB RAM, and an NVIDIA RTX 3060 Ti GPU.

3.7.4 Hyperparameter Tuning Strategy

The strategy for hyperparameter selection evolved alongside the model complexity. For the initial value-based models (Article 1), we employed a systematic grid search to identify optimal neuron counts and activation functions. However, for the more complex Actor-Critic and PPO architectures (Articles 2, 3, and 4), the high-dimensional state space rendered exhaustive grid search computationally prohibitive. Consequently, we adopted an empirical tuning approach. This involved starting with established baselines from transportation literature and iteratively refining learning rates (alpha) and clipping parameters (epsilon) to balance training stability with convergence speed. In Article 4, specifically, context thresholds for task switching were tuned manually based on observing fleet utilization rates during preliminary simulation runs.

3.8 Summary of Methodology and Contributions

This chapter presented the methodological foundation of the dissertation, detailing the reinforcement learning architectures, agent designs, simulation environments, and evaluation strategies employed across four research articles. The methodologies were tailored to address the challenges of real-time decision-making in urban freight logistics, including dynamic pricing, shipment-vehicle matching, and vehicle dispatching.

The research introduced a progression of RL-based frameworks, beginning with a centralized DQN agent for matching (Article 1), followed by a two-agents hierarchical actor-critic model for matching and dispatching (Article 2), a nested hierarchical reinforcement learning framework for joint pricing, matching, and dispatching (Article 3), and culminating in a decentralized multi-agent reinforcement learning architecture with dynamic task switching (Article 4).

Each framework was evaluated using a simulation environment built on H3 spatial partitioning and synthetic datasets for Montréal and Toronto. Evaluation metrics included cumulative reward, revenue, successful matches, vehicle idle time, shipment waiting time, and pickup distance. The experiments demonstrated the scalability, adaptability, and operational efficiency of the proposed models.

Table 3.1 provides a comparative overview of the RL architectures, algorithms, decision tasks, spatial resolutions, and datasets used across the four articles. This tabular summary highlights the evolution of methodological complexity and geographic scalability.

Table 3.1 Comparison of RL Architectures Across Dissertation Articles

Article	Algorithm	Architecture	Decision Tasks	Spatial Resolution (H3)	Dataset
Article 1	DQN	Centralized Single-Agent	Matching	Level 7	Montréal
Article 2	Actor-Critic (PPO)	HRL (2 Global Agents)	Matching, Dispatching	Level 7 (both tasks)	Montréal
Article 3	Actor-Critic (PPO)	NHRL (3 Global Agents)	Pricing, Matching, Dispatching	Level 7 (Pricing/Matching), Level 6 (Dispatching)	Toronto Montréal
Article 4	Actor-Critic (PPO)	MARL (Decentralized)	Matching, Routing, Dispatching	Level 7 Level 7	Toronto, Montréal

Key methodological contributions include:

- **Architectural Innovation:** Development of modular RL architectures ranging from centralized to fully decentralized systems, enabling scalable and context-aware decision-making.
- **Hierarchical and Nested Structures:** Introduction of HRL and NHRL frameworks that decompose complex logistics problems into tractable sub-tasks across spatial and temporal scales.
- **Dynamic Task Switching:** Implementation of vehicle-level autonomy through interpretable context-aware switching among matching, routing, and dispatching tasks.
- **Spatial Pre-filtering Modules:** Design of PAMA, PADA, DEZE, and ShipScan to reduce action space complexity and enable real-time decision-making.
- **Empirical Validation:** Demonstration of significant improvements in reward, revenue, match success, and resource utilization across multiple urban freight scenarios.

These contributions collectively advance the state of the art in smart freight logistics by offering scalable, adaptive, and interpretable RL solutions for real-time operations.

CHAPTER 4 ARTICLE 1: REAL-TIME RL-BASED MATCHING WITH H3 GEOHASH PARTITIONING IN SMART FREIGHT PLATFORM

Ali Shiri[1], **Asad YarAhmadi**[1], **Samira Keivanpour**[1], **Amina Lamghari**[2]

[1] *Department of Mathematics and Industrial Engineering, Polytechnique Montreal, Canada*
[2] *Department of Management, Université du Québec à Trois-Rivières, Canada*

2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)
Submitted on May 12, 2024
Published on October 7, 2024
DOI: 10.1109/VTC2024-Fall63153.2024.10757474

Abstract

This research presents a novel Deep Q-Learning (DQL) framework designed for efficient real-time matching of shipments and vehicles in the freight transportation sector. The framework utilizes the H3 geospatial indexing system for accurate positioning and employs a pre-filtering mechanism to streamline the matching process. When evaluated on a simulated model of Montreal's transportation network, the framework demonstrates promising results in generating matches that reduce travel distance and prioritize timely service. Through extensive experimentation, a configuration utilizing ReLU activation was identified as particularly efficient, even under limited computational resources. This research contributes to the development of advanced, real-time matching algorithms in logistics and showcases the potential of integrating reinforcement learning with geospatial analysis to address complex transportation challenges. These findings offer valuable insights for freight companies seeking to improve their matching processes, potentially leading to cost reductions and enhanced service quality.

Keywords: Real-time Matching, Freight Transportation, Reinforcement Learning, Deep Q-Learning, Learning-based algorithms, Smart Transportation

4.1 Introduction

4.1.1 Problem Context

The transportation of goods by commercial vehicles is fundamental to economic growth, with truck transportation accounting for 77% of domestic freight movement in Canada in 2020 [79]. However, traditional freight matching methods are often labor-intensive and inefficient, considering the ability to respond to real-time changes in demand and supply. This often leads to a low service rate, a time-consuming process for all parties involved, and ultimately, increased costs and delays.

To overcome these limitations, real-time data from GPS and IoT devices, integrating it with advanced learning models can be a solution. This approach enables the creation of a dynamic and responsive system that continuously optimizes matching decisions, leading to improved efficiency for smart freight platforms. In the context of smart freight platforms, mathematical optimization is widely used for real-time matching of shipments and carriers, considering factors like cost, distance, and delivery time windows.

Learning-based approaches, on the other hand, are increasingly employed to predict demand, optimize pricing strategies, and personalize recommendations for shippers and carriers, to enhance the efficiency and effectiveness of freight transportation systems. Mathematical optimization aims to determine the optimal solution by maximizing or minimizing an objective function subject to a set of constraints. Exact methods guarantee an optimal solution but may exhibit high computational complexity, especially for large-scale or complex problems [80]. Heuristic methods [81], and metaheuristics, which are higher-level heuristics guiding the search process of other heuristics, are also employed. The choice of the most suitable optimization methodology often depends on the trade-off between solution quality, computational cost, and the specific characteristics of the optimization problem at hand. Alternatively, learning-based approaches use historical data to improve decision-making, employing techniques like machine learning [82], and reinforcement learning [83]. These methods can adapt to dynamic environments and learn from experience, making them suitable for complex or uncertain problems.

4.1.2 Learning-based Matching Models

Research endeavors are conducted to enhance matching efficiency, exploring two distinct domains: ride-sharing and freight operations. In ride-sharing, Gao et al. [65] integrate batched matching models with data-driven proactive guidance strategies to maximize the matching rate and minimize the total idle driving costs through machine learning. Haliem et al. [40]

propose a scalable framework for dynamically matching and routing vehicles based on real-time demand. This approach aims to minimize unnecessary travel distance and idle time for maximum vehicle capacity utilization.

Numerous research efforts have been focused on reducing the time for the ride-sharing matching context. Wang et al. [10] integrate deep reinforcement learning and integer linear programming to present a multi-stage sequential decision-making model that minimizes transfer times of accepted requests, dead times, and maximizes the incentives for serving requests. Singh et al. [59] Singh et al. propose a distributed reinforcement learning model to optimize ride-sharing. By anticipating future demand, they aim to reduce waiting and idle times for both passengers and drivers through a multi-hop ride-sharing approach. Manchella et al. [67] introduce an approach to integrate a demand-aware insertion-based route planning algorithm for both goods and passengers that can scale up to the maximum capacity of each vehicle and minimize the additional travel time incurred by orders due to participating in a shared vehicle. Li et al. [16] develop a deep learning algorithm to minimize the total travel cost of all travelers under the worst-case travel time scenario. Haliem et al. [84], Qin et al. [56] and, Manchella et al. [69] propose deep learning models to minimize time as well.

In the freight transportation domain, Chen et al. [48] introduce Deepfreight, a combined learning and optimization approach for efficient multi-transfer freight delivery. Their model aims to improve reliability and efficiency by integrating a deep learning algorithm with a mathematical optimization model. Shu et al. [18] develop static and dynamic compensation strategies to increase platform service scale while minimizing costs. They also focus on improving matching efficiency and optimizing the process for unmatched orders, Tian et al. [71] introduce a method that evaluates the degree of compatibility between vehicles and cargo, involving two primary components: the attribute matching degree and the environmental influence degree. Guo et al. [46] suggest a rolling horizon approach to optimize shipment-to-service matching decisions over time, aiming to minimize overall costs. Research efforts have been conducted to enhance various operational aspects and optimize the logistical processes to achieve greater efficiency.

4.1.3 Research Gaps & Contributions

While learning models have been widely explored in various domains, their application in real-time decision-making for freight transportation, particularly using reinforcement learning techniques, remains relatively under-explored. Furthermore, there is a gap in research addressing the dual challenge of minimizing both the distance between cargo and vehicles and the earliest service time of vehicles to shipments .

This paper proposes a novel approach to address these challenges by utilizing Deep Q-Network (DQN) with key variables such as location coordinates and earliest service time. DQN presents a promising approach for real-time freight transportation matching due to its inherent ability to learn and adapt within dynamic environments. Unlike traditional rule-based methods, DQL continuously refines its decision-making capabilities through ongoing interaction with the environment. This data-driven learning process allows the algorithm to optimize matching decisions in response to real-time feedback and changing conditions, ultimately resulting in improved efficiency and responsiveness within the complex and often unpredictable freight transportation landscape.

The integration of the H3 partitioning system allows for efficient spatial representation, offering a standardized and balanced approach compared to traditional clustering methods. The uniform nature of H3 hexagons ensures a more accurate and equitable representation of geospatial areas, contributing to more effective decision-making.

Prefiltering mechanism is designed to select suitable candidates to match a shipment based on the H3 partitioning system and earliest service time, resulting in a reduction of the computational workload and making the state space more meaningful to the RL-agent.

This approach introduces a novel action space, allowing for more diverse and informed decision-making in the matching process. Lastly, our research contributes a novel reward structure and reward function.

4.1.4 Outline

The paper is structured as follows: Section II presents an overview of the Deep Q-learning model. Section III describes the proposed methodology, model, and dataset. Section IV presents the results and efficiency of the algorithm through numerical simulations. The paper is concluded in Section V.

4.2 Deep Q-Learning

The DQN's innovative dual neural network setup—a main network and a target network—enables accurate Q-value estimations, driving its widespread adoption. The networks share the same architecture but utilize distinct weight values. For every K step, the weights of the main network' are duplicated onto the target network to improve the stability and efficiency of the learning process. The main network calculates the Q-values for the current state, representing the expected rewards for all possible actions, while the target network calculates the Q-values for the next state, denoted by Q(s', a').

DQN aims to determine the optimal Q function, as shown in Equation (4.1). This function helps in making decisions that maximize the cumulative rewards over multiple steps, all while adhering to a specific policy π. The discount factor γ plays a crucial role, helping to balance the importance of immediate rewards against those gained in the long run.

$$Q^*(s,a) = \max_{\pi} \mathbb{E}\Big[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots \Big| \, s_t = s, a_t = a, \pi\Big] \tag{4.1}$$

The Bellman equation, a fundamental concept introduced by Bellman in 1966 [85], is used to repeatedly update the Q-values. In Equation (4.2), the parameter α represents the learning rate, which determines the extent of change in the Q-values at each learning step.

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha\left[R(s,a) + \gamma \cdot \max_{a'} Q(s',a')\right] \tag{4.2}$$

Finally, the loss function for this process is given in Equation (4.3), where θ_i and θ_i^- denote the Q-value parameters in the current and future states. Rather than calculating the entire summation, stochastic gradient descent is commonly used to update the prediction network's parameters. These parameters are then transferred to the target network's parameters every K time step.

$$\nabla_{\theta_i} L(\theta_i) = \mathbb{E}_{K_{[s,a,r,s']}}\Big[(r + \gamma \max_{a'} Q(s',a';\theta_i^-) - Q(s,a;\theta_i))\nabla_{\theta_i} Q(s,a;\theta_i)\Big] \tag{4.3}$$

4.3 Methodology

Our approach aims to exploit the strengths of DQN and Hierarchical Hexagonal Hierarchies (H3), geospatial indexing system developed by Uber Technologies, Inc. H3's ability to partition the Earth's surface into a grid of hexagons with varying resolution levels proves particularly valuable in this context.

Unlike clustering methods that may produce irregularly shaped or sized areas, the H3 system employs hexagons of uniform area, ensuring a precise and balanced representation of geographical locations. This consistency is crucial for effective delivery management, as it allows for accurate distance calculations and fair workload distribution. Moreover, H3's hierarchical structure enables multi-scale analysis: each hexagon can be subdivided into seven smaller hexagons at a finer resolution. This allows for seamless transitions between detailed views of specific areas and broader overviews of delivery networks, enhancing flexibility and eliminating the need for constant adjustment of clustering parameters.

We tailor various aspects of DQN, including the state space, action space, reward structure, and reward function to suit the requirements of the DQN approach and meet our specific needs. In the process of constructing the state space, we also develop a pre-filtering algorithm to create a pool of candidate vehicles for matching. These candidates represent potential pairings of shipment with vehicles.

Our proposed DQN approach scales seamlessly to manage both massive datasets and expansive geographic areas. This scalability is achieved through a dual-strategy framework: a hierarchical geospatial indexing system (H3) and an efficient pre-filtering search algorithm. H3 effectively handles large spaces by offering multiple levels of granularity, allowing dynamic adjustment of geospatial data resolution, and its hexagonal tiling provides uniform spatial data representation. Additionally, the pre-filtering mitigates the dimensionality by offering a pool of vehicle candidates, which reduces the computational load on the DQN in even large-scale datasets. Notably, pre-filtering transforms the problem from an NP-hard challenge to a solvable one by structuring the data in a way that simplifies the computational complexity, especially in large areas. This ensures that our system can rapidly make accurate matching decisions for numerous vehicles and shipment requests. Such scalability is essential for real-world applications where managing large volumes of data and widely dispersed locations is a common challenge.

4.3.1 Pre-filtering Algorithm

The main objective of the pre-filtering algorithm is to identify the best vehicles for matching and to create the state space. This involves, considering factors such as H3 resolutions, vehicle capacity, and quality scores. Capacity refers to the maximum load a vehicle can accommodate or the quantity required by a shipment. The quality score represents the minimum service desirability for each shipment request and the corresponding desirability of each vehicle. In our approach, we establish a pool of N candidates to construct the state space for our RL-agent within the DQN framework. This pool is formed as follows:

1. The algorithm first assesses the H3 tag of shipment at resolution level 7, as well as the capacity and quality score required for the specific demand. It then searches for vehicles within that hexagonal region and evaluates how many of them meet these criteria. If an exact match of N vehicles is found, they are directly added to the pool of candidates.

2. If more than N vehicles are identified, N of them are randomly selected for inclusion in the pool of candidates. However, if fewer than N suitable vehicles are found, the algorithm expands its search to the parent hexagon at resolution level 6, offering a

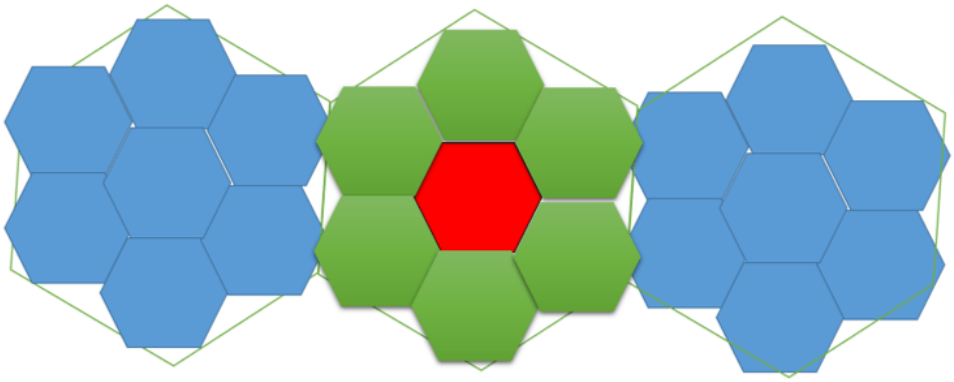

Figure 4.1 Hexagon Representation of Shipment Origin at Resolution Level 7(red Hexagon) and Parent Hexagon at Resolution Level 6

larger area for exploration.

3. In this larger area, the algorithm seeks M trucks in the parent hexagon to address the lack of required vehicles to fill the pool, while considering the corresponding H3 tag, capacity, and quality score requirements. Figure 4.1 illustrates the hexagon representing the shipment origin at resolution level 7, along with the parent hexagon at resolution level 6.

4. If a sufficient number of new vehicles are found in the larger area, these are integrated with the initial candidates to complete the pool. Otherwise, if the search identifies more vehicles than needed, M of them are randomly chosen. In cases where the second-stage search still doesn't provide enough candidates, the algorithm creates fictitious candidates to ensure the pool reaches the required size.

The developed algorithm is summarized in figure 4.2.

4.3.2 State Space

The state space is a multidimensional vector that contains information about the pick-up location for cargo, the locations of candidate vehicles (selected using the pre-filtering algorithm), and the time required to service the demand. specifically, it is $(2 + 3 * N)$-element vector: two elements represent the latitude and longitude of the pickup location, and N, sets of three elements each, correspond to a candidate vehicle. For each vehicle, the first two elements represent the latitude and longitude of its location, while the third element indicates the service time required to pick up the cargo. Figure 4.3 illustrates the state space.

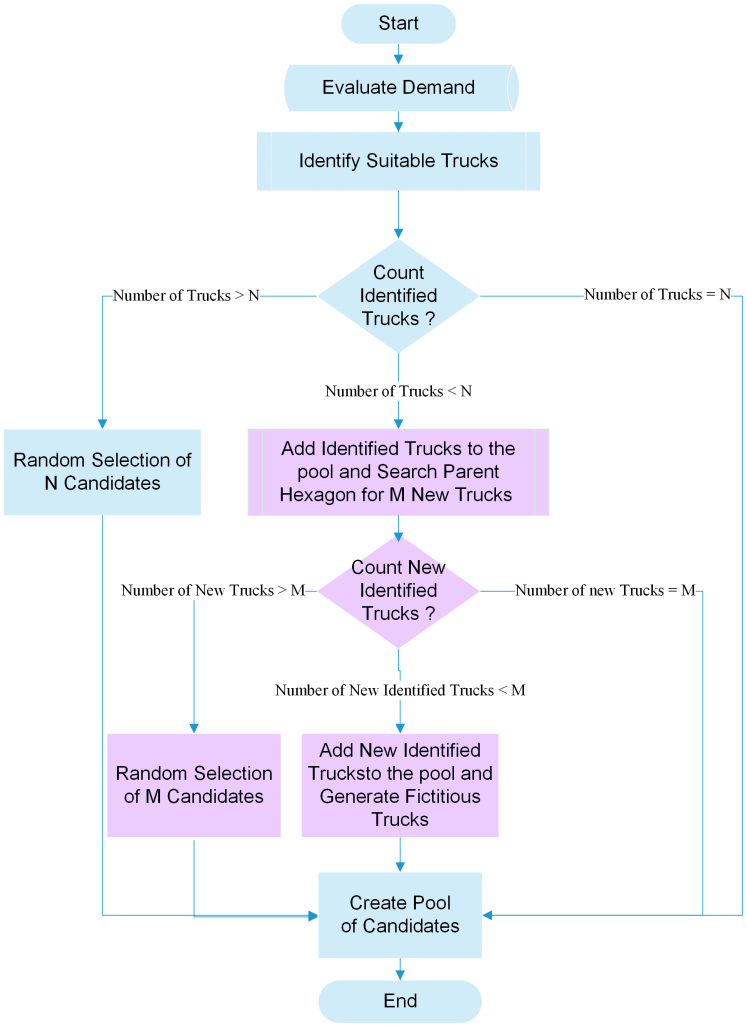

Figure 4.2 Pre-filtering Algorithm Flowchart.

4.3.3 Action Space

The action space in DQN serves as a structured framework guiding decision-making. It's represented as a vector with $(1 + N)$ elements, where each element corresponds to a distinct action. The first element allows not assigning a shipment to any candidate truck, useful when the pool is saturated with fictitious candidates. The remaining N elements represent matching the shipment with one of the N candidate vehicles. This empowers our RL agent to strategically allocate shipments, optimizing efficiency by avoiding pointless actions.

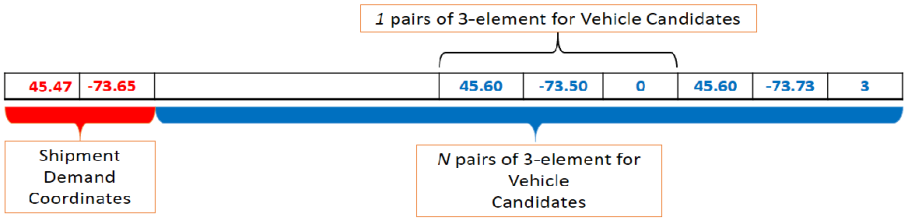

Figure 4.3 Visualization of the State Space

4.3.4 Reward Structure & Reward function

The reward structure is a system that uses rewards and punishments to guide the RL-agent's decision-making process. When the RL-agent makes a correct decision, it receives a reward, encouraging it to keep making those good decisions. Conversely, when it makes an incorrect decision, it might get a penalty, discouraging it from making it again.

This system helps our RL-agent learn how to match shipments with trucks in the best way. The core of our reward system lies in the following function:

$$\text{Reward function} = -\sum_{r \in R} \sum_{d \in D} [(d_{rd} + T_{rd} \cdot S) \cdot X_{rd}] \tag{4.4}$$

In equation (4.4), let R and D represent the set of shipment requests and the set of trucks, respectively. The decision variable X_{rd} is equal to 1 if a match occurs, and 0 otherwise. The term d_{rd} represents the actual distance between the shipment r and the vehicle d, with S representing the diameter of the hexagon in resolution 6. However, the reward function doesn't focus solely on physical distance. It also accounts for the time factor by introducing T_{rd}, which quantifies the difference in timestamps between the shipment request and the vehicle availability to pick up. To give time a comparable weight in the decision-making process, the reward function converts it into a distance metric by multiplying it with S, which is a factor that allows the model to prioritize vehicles not only based on physical distance but also on their ability to provide service earlier. For example, if demand is on timestamp T_1, and a vehicle is available on timestamp T_4, it may still provide acceptable service based on time, score, and capacity. In practical terms, the equation can be simplified to $d_{rd} + 3S$. This approach provides a trade-off between distance and time, allowing the model to consider both factors when matching shipments with vehicles.

Note that the negative sign in the reward function is a critical element in reinforcement learning, ensuring that the optimization process aligns to maximize the reward function while minimizing distance and time factors. We have four branches for reward, each serving

a specific purpose:

1. *Correct "Match" Action*: If the RL-agent accurately matches a shipment with a suitable vehicle, it receives a reward. This reward reinforces the importance of precise matching, ultimately leading to more efficient logistics.

2. *Wrong "Match" Action*: If the RL-agent matches a shipment with a fictitious vehicle, it incurs a penalty of P. The penalty discourages actions that lead to unfeasible matching.

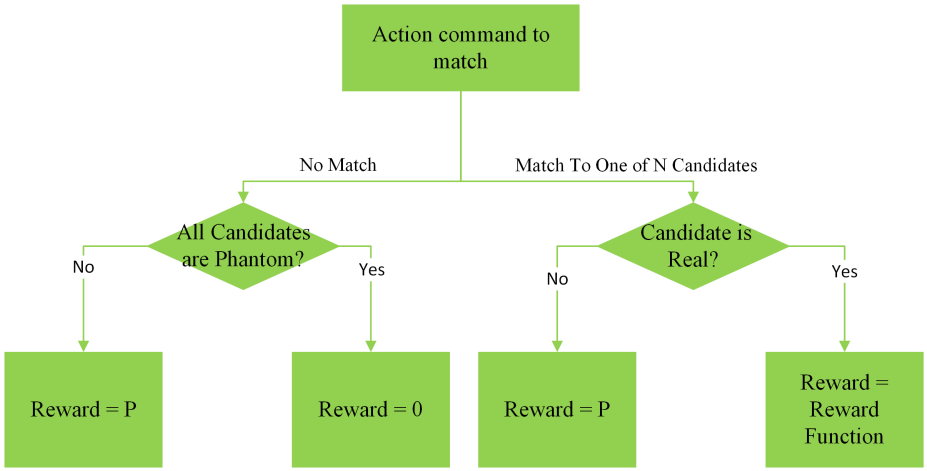

Figure 4.4 Reward Structure

3. *Correct "No Match" Action*: If the RL-agent chooses not to match a shipment with any vehicles when they are all fictitious, it neither gains a reward nor incurs a penalty. This reflects the RL-agent's capacity to discern situations where a suitable match is unattainable.

4. *Wrong "No Match" Action*: If the RL agent fails to match a shipment with any vehicle, despite the availability of suitable candidates, it incurs a penalty of P. This penalty discourages inaction when feasible matches exist, promoting the efficient utilization of available resources.

The reward structure, through these mechanisms, guides the RL-agent to make informed decisions about matching shipments with vehicles effectively. It reinforces the importance of accurate matches while discouraging actions that may lead to inefficiencies or mismatches. Figure 4.4 demonstrates the reward structure.

4.3.5 Environment

The environment in our model operates under certain assumptions related to the movement of vehicles through hexagonal grids. Specifically, we assume that vehicles can traverse only one hexagon at a resolution of 6 during each timestamp.

To provide context, it's important to note that the average edge length of hexagons at Resolution 7 is approximately 1.41 kilometers. However, at Resolution 6, this edge length increases to around 3.73 kilometers. Additionally, the diameter of each hexagon is approximately two times larger than the edge length of the hexagon. Consequently, vehicles can effectively cover a distance of approximately 7.45 kilometers within a single timestamp.

This significant assumption allows our model to convert time into distance. Integrating temporal and spatial aspects into the decision-making process is crucial for efficient freight logistics.

In addition to the aforementioned assumption, several other key considerations influence the dynamics of our model:

- *Vehicle Matching Constraint*: Each vehicle in the system can either be matched with one shipment request or remain unmatched. This constraint reflects the one-to-one nature of the matching process, where each vehicle is assigned to at most one shipment request at a timestamp.
- *Shipment Matching Constraint*: Similarly, each shipment request can either be matched with one vehicle or remain unmatched. This reflects the one-to-one nature of the matching process from the perspective of shipments.
- *Exclusive Time Availability*: Vehicles are available for matching during specific timestamps based on their contracts. If a vehicle is not matched in its designated time slot, it becomes unavailable for future timestamps due to contractual obligations.
- *Time Windows for Shipments*: Each shipment has specified earliest and latest time windows during which it must be serviced. This temporal constraint aligns with real-world logistics operations and enhances the model's realism and practicality.
- *Reconsideration of Vehicles*: After dropping off cargo, the vehicle becomes available for reconsideration in the candidate pool. This feature allows for a continuous flow of vehicles and fosters adaptability in the matching process.
- *Temporal Discretization for Transportation Operations*: In addition to considering

travel time, the model accounts for Z discrete timestamps dedicated to the execution of pick-up and drop-off activities for each match and delivery.

These additional considerations provide a comprehensive and detailed representation of the dynamic nature of the freight logistics problem and significantly contribute to the model's ability to address real-world challenges effectively.

4.3.6 Dataset

Our study requires two datasets: Origin-Destination (O-D) data and truck data. The O-D data encompasses details of shipping requests, including geocodes for both shippers' and drop-off locations, hexagon ID in the H3 indexing system, volume type, availability, quality scores, and timestamp. The geocode columns are indexed by longitude and latitude. The Volume and capacity are categorized into three types: light, medium, and heavy, across a hundred different timestamps, denoted as T1 to T100. The dataset also includes the trucks' geolocation at the time of availability, Vehicle capacity is categorized into three types: light, medium, and heavy, timestamp, and quality scores. Like the O-D data, timestamps here also span 100 timeslots.

Research on trucking data sources like the Federal Highway Administration (FHWA), Canada's government Open Data, and Ontario Open Data showed two main limitations: the data is highly aggregated, providing only the number of trips per city area over a period, and lacks detailed information on shippers' and carriers' locations, shipment types, and other specifics. To address this, we first identified factors influencing truck trip generation, such as population density and the presence of shopping centers and industrial areas. We then collected and input this data into QGIS, using a density function to calculate factor densities. Based on these densities, we randomly generated and spatially distributed different numbers of trips in the Montreal metropolitan area for the case study, resulting in $11,397$ unique vehicles and $9,866$ unique shipment requests. The geographical density of the dataset for vehicle, origin, and destination of shipment request is demonstrated in figure 4.5 for the H3 partitioning system on resolution level 7.

4.4 Simulation Results

In this section, the simulation results for the real-time matching model based on the generated dataset are presented and analyzed. All the experimental simulations are run on a computer with 4 cores Intel Core i7, 64 GB RAM, and 8 GB RAM VGA. The DQN-based framework is constructed by PyTorch in Python.

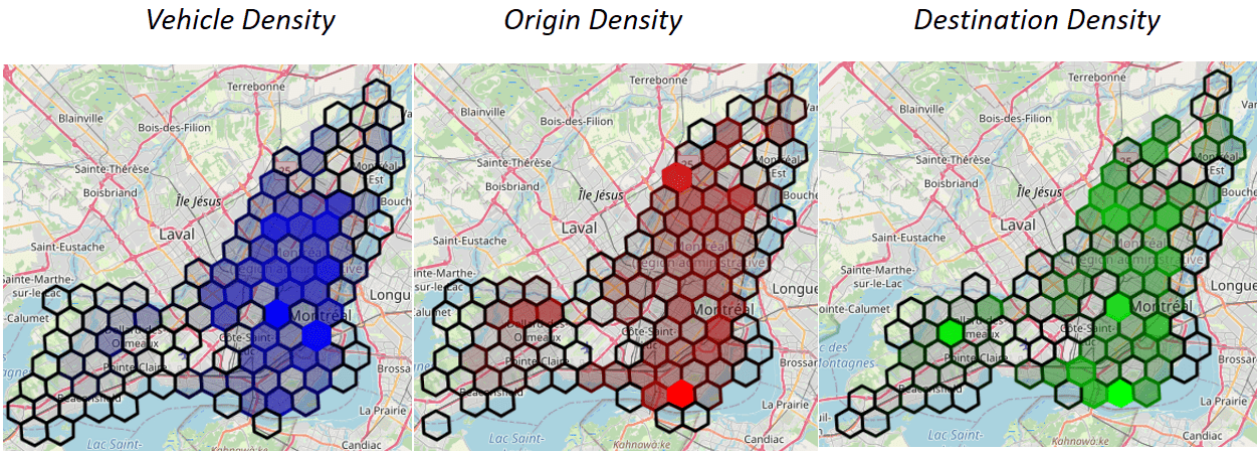

Figure 4.5 Geographical Density for the H3 partitioning system on resolution level 7

4.4.1 Experiment Setup and Hyperparameter Configuration

We considered various combinations of activation functions and the number of neurons per layer to evaluate the performance of networks with 2 hidden layers. We employed Deep Q-learning with key hyperparameters: an exploration rate starting at 1.0 and decreasing by 0.5% per episode, a discount factor of 0.99, a learning rate of 0.0001, and a soft update parameter (TAU) of 0.001. The agent updated its Q-network every 4 time steps. These settings were crucial for efficient learning and achieving optimal results in our systematic exploration of over 2,000 neural network architectures for a specific task. The softmax activation function was used in the output layer of the DQN.

We set N equal to 10 in our model, so the dimension of the state space is 32, and the dimension of the action space is 11. we set Z equal to 10, representing discrete timestamps dedicated to the execution of pick-up and drop-off activities for each match and delivery. P is imposed as a -1000. This penalty, approximately 135 times larger than the diameter of a resolution 6 hexagon, is designed to strongly discourage such infeasible matches or inaction when feasible matches exist, thereby promoting efficient and accurate decision-making by the agent. Our analysis aims to uncover our application's suitable neural network configuration dataset normalization.

4.4.2 Performance Evaluation and Comparative Analysis

As illustrated in figures 4.6 and 4.7, all models demonstrate increasing matching and reward scores over episodes. Table 4.1 presents a comprehensive evaluation of the DQL model's performance under various configurations. The evaluation metrics include reward per match,

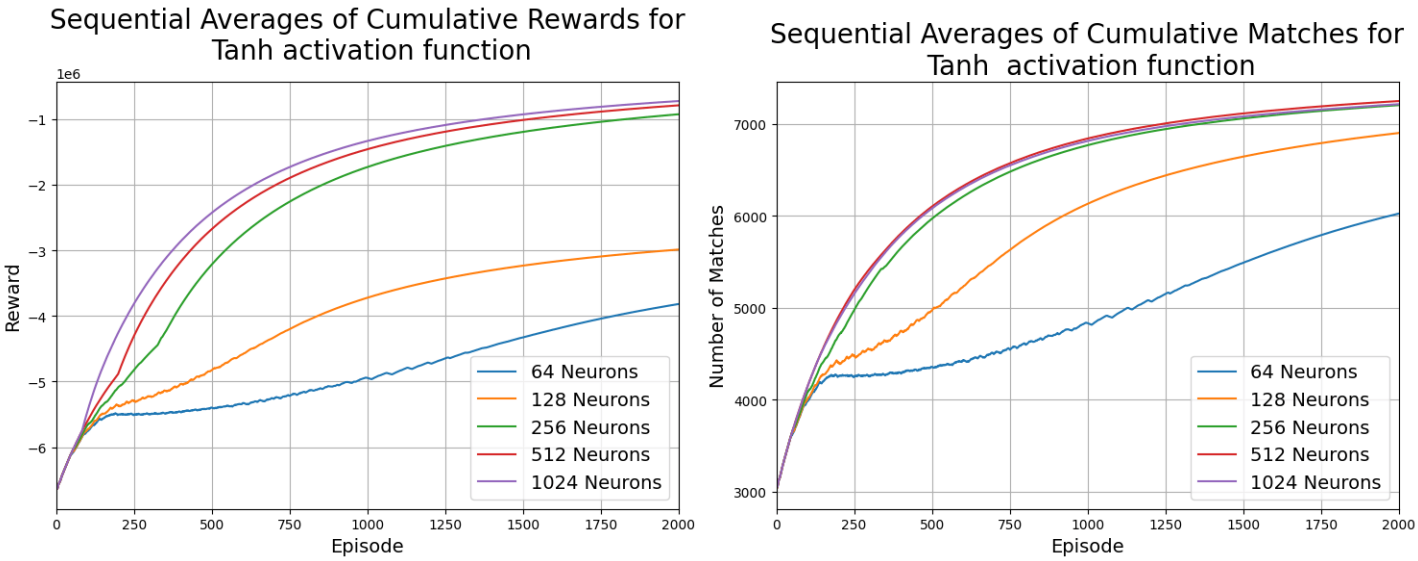

Figure 4.6 Sequential Averages of Cumulative Rewards(Left) and Sequential Averages of Cumulative Matches(Right) for Tanh Activation

distance per match, average earliest time of service(ETS) per match, and training time.

These performance metrics highlight several key benefits for real-life freight transportation. The reward per match generally improves as the number of neurons increases, with the optimal performance observed using 1024 neurons and ReLU activation, achieving a reward of -98.20. This indicates that matching agent generates higher rewards, translating to more profitable operations. Additionally, the distance per match decreases with an increase in neurons for both activation functions. The minimal distance of 6.15 km achieved with 1024 neurons and ReLU activation suggests that vehicles travel shorter distances, reducing fuel consumption and operational costs.

The model's scalability is enhanced by its performance and the design elements such as pre-filtering and the H3 grid system, which streamline computations and ensure the model can handle large-scale operations efficiently. The earliest time per match across all configurations is less than one timestamp, meaning matches occur within the same timestamp as the shippers' request. This ensures prompt matching and timely deliveries, which are crucial for maintaining high customer satisfaction and operational efficiency in freight transportation scenarios. The training time remains relatively stable across different neuron configurations, indicating that increasing model complexity does not significantly impact computational efficiency. This stability facilitates seamless integration into existing freight transportation platforms. Its adaptability is also evident, as the model can be fine-tuned to suit different operational scenarios and varying real-world conditions, enhancing its practical utility.

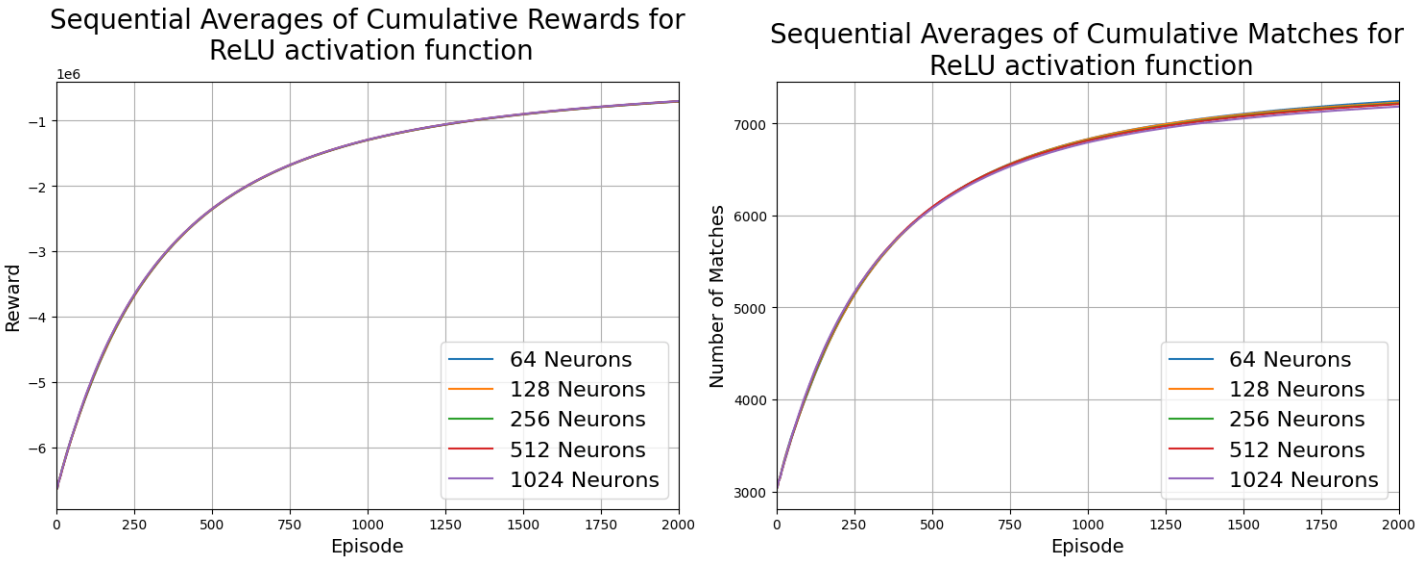

Figure 4.7 Sequential Averages of Cumulative Rewards per Episode(Left) and Sequential Averages of Cumulative Matches(Right) for ReLU Activation

In summary, the ReLU activation function, coupled with 1024 neurons per layer, effectively balances model complexity with computational efficiency, making it the most suitable choice for real-time matching tasks in our case study. These insights demonstrate that adopting this neural network configuration can significantly enhance the efficiency and profitability of freight transportation operations, making it a valuable strategy for freight transportation applications. The model's ability to scale and adapt, supported by advanced design features, ensures it can meet the demands of large-scale and diverse freight transportation environments, highlighting its potential for widespread integration and impact.

4.5 Conclusion

This research introduces a novel approach to real-time matching in freight transportation using DQN. By integrating the H3 geospatial indexing system—a standardized spatial framework—the efficiency of the matching process is enhanced. Additionally, a pre-filtering mechanism, which considers both distance and earliest time of service, reduces computational overhead and focuses the model on relevant matches. A case study in Montreal validated the model's effectiveness and capability in reducing operational costs and improving minimizing travel distance and match times. This translates to lower operational costs and improved delivery performance.

The contributions of this research have substantial practical implications for the freight trans-

Table 4.1 Performance Metrics for Different Activation Functions and Neuron Configurations per Layer

Neurons per Layer	***Reward per Match***	***Distance per Match (Kilometers)***	***ETS per Match (Timestamp)***	***Training Time (Hour)***
Tanh Activation Function				
64	-634.04	7.75	0.1474	64.63
128	-433.04	7.55	0.1260	64.84
256	-128.60	7.42	0.1357	64.91
512	-109.54	7.24	0.1402	65.16
1024	-100.02	6.66	0.1546	65.26
ReLU Activation Function				
64	-98.54	7.55	0.1317	64.15
128	-98.51	7.01	0.1536	65.16
256	-98.44	6.83	0.1612	65.18
512	-98.27	6.69	0.1658	65.18
1024	**-98.20**	**6.15**	**0.1789**	**65.19**

portation industry. Firstly, it addresses the existing gap in the application of reinforcement learning, in the context of real-time freight matching. Secondly, the proposed framework has implications for industrial applications, including reduced operational costs, improved delivery times, and increased overall efficiency. By adopting this approach, freight companies can achieve dynamic logistics operations, better handle fluctuating demand, and enhance overall customer satisfaction. Future research can enhance this approach by incorporating consolidation strategies and considering uncertainty parameters such as weather and traffic conditions. Furthermore, integrating pricing strategies into the learning process could provide additional insights and optimization opportunities for real-time freight transportation matching.

Acknowledgment

This work is supported by Shiphual Logistics and Mitacs through the Mitacs Accelerate program IT30680.

CHAPTER 5 ARTICLE 2: REAL-TIME MATCHING AND DISPATCHING FOR URBAN FREIGHT TRANSPORTATION: A HIERARCHICAL REINFORCEMENT LEARNING THROUGH ACTOR-CRITIC AND H3 SPATIAL PARTITIONING

Ali Shiri, **Asad YarAhmadi**, **Samira Keivanpour**,

IEEE Transactions on Intelligent Transportation Systems
Submitted on September 17, 2024
First Revision Submitted on April 22, 2025
Second Revision Submitted on July 30, 2025
Published on September 04, 2025
DOI: 10.1109/TITS.2025.3601536

Abstract

Real-time freight matching and dispatching present complex challenges due to the dynamic nature of supply and demand, high-dimensional decision spaces, and the need for rapid response under operational constraints. In this paper, we propose a novel Hierarchical Reinforcement Learning (HRL) framework that jointly optimizes matching and dispatching processes in freight transportation, offering enhanced responsiveness, modularity, and coordination under real-time constraints. To mitigate computational demands without sacrificing matching accuracy, we present two efficient pre-filtering algorithms, PAMA (Pre-filtering Algorithm for Matching Agent) and PADA (Pre-filtering Algorithm for Dispatching Agent), which enhance the H3 hexagonal geospatial partitioning system. A GIS-based simulation of freight flow in Montreal provides realistic validation. Experimental results show that our framework improves successful match rates by 1.73%, reduces vehicle idle time by 5.7%, and maintains low empty mileage. Compared to state-of-the-art baselines including Deep Q-Network and clustering-based methods, our HRL model achieves superior reward efficiency, scalability, and adaptability in dynamic freight environments. These findings underscore the potential of HRL to enhance the operational efficiency of smart freight platforms.

Keywords: Hierarchical Reinforcement Learning, Freight Transportation, Actor-Critic, Matching, Dispatching, Logistics.

5.1 Introduction

The international freight forwarding market was valued at $209.14 billion in 2023, which is critical in facilitating global trade and supply chains [86]. Online freight sharing platforms have revolutionized this industry by optimizing the matching process between careers and shippers which has led to a reduction in GHG emissions, increasing pairing carriers to shippers, and decreasing empty mileage.

Existing research in reinforcement learning(RL) for freight-sharing systems has predominantly focused on Deep Q-Network(DQN), which presents challenges related to computational complexity. While Actor-Critic and Hierarchical Reinforcement Learning(HRL) offer promising avenues for more complex decision-making, their application in this field remains limited. Our primary contribution lies in the novel development and application of an HRL framework featuring a unique dual-agent Actor-Critic architecture specifically tailored for the simultaneous optimization of freight matching and dispatching.

HRL presents an appropriate approach to address the challenges of freight sharing. By simplifying the problem into smaller, more tractable components, it has the potential to reduce learning time and computational resource requirements. To the best of the authors' knowledge, no prior work comprehensively integrates freight matching and dispatching via coordinated Actor-Critic HRL.

In addition, previous research in ride and freight-sharing optimization has primarily focused on either matching or dispatching in isolation. Although some studies have attempted to integrate matching and dispatching, they often rely on heuristic approaches that lack generalizability in dynamic, real-world environments [84].

Our work aims to bridge this gap by proposing a novel HRL framework that mainly addresses the interaction between matching and dispatching. By leveraging the strengths of HRL and incorporating innovative pre-filtering algorithms, we aim to develop a more efficient and adaptive solution for real-time freight transportation optimization. The main contributions of this study can be summarized as follows:

- **HRL Framework for Urban Freight Transportation:** We propose a novel HRL framework that optimizes real-time freight matching and dispatching, improving operational efficiency and addressing the limitations of isolated optimization approaches.
- **Hierarchical Hexagonal Hierarchy(H3) Based Geospatial Pre-filtering:** We design two spatial pre-filtering algorithms that leverage the H3 partitioning to significantly reduce search space and computation time, enabling scalable deployment in

urban freight environments.

- **Actor-Critic Network within HRL:** We employ Actor-Critic networks to ensure stable learning and smooth policy updates, offering better adaptability and convergence than value-based methods like DQN in dynamic freight environments.
- **GIS-Based Urban Freight Simulation in Montreal:** We construct a realistic freight simulation environment using demographic, geographic, and freight flow data from Montreal, providing a robust testbed to evaluate and validate our proposed framework.
- **Quantitative Performance Gains over Baselines:** Our HRL framework outperforms DQN and clustering-based methods, and improving match rate by 1.73%, reducing idle time by 5.7%, and lowering empty mileage compared to the Actor-Critic matching baseline.

This paper is structured as follows: Section II reviews related work in freight transportation optimization. Section III details the proposed HRL framework. Section IV presents a case study, while Section V analyzes the results. Finally, Section VI summarizes findings and outlines future research directions.

5.2 Literature Review

Early online transportation platforms primarily relied on greedy algorithms, emphasizing immediate goals such as minimizing passenger wait times by assigning the nearest available driver. However, this approach often overlooked critical aspects like demand forecasting, leading to suboptimal driver distribution and increased "deadheading"—unproductive cruising without passengers—which in turn prolonged overall ETAs [87, 88].

Recognizing these limitations, the field has progressively shifted toward learning-based strategies, particularly reinforcement learning (RL) and deep learning [89]. These methods integrate real-time and historical data—such as GPS and traffic conditions—and support proactive dispatching, enabling more adaptive and effective matching [90]. RL, grounded in the Markov Decision Process framework, enables agents (drivers or platforms) to learn optimal actions that maximize long-term rewards through environment interaction, without a prespecified model [91]. This shift frames the field as a complex, agent-based system rather than a static optimization problem.

In agent modeling, two dominant paradigms have emerged. First, multi-agent systems view individual drivers as agents, whose experiences collectively inform centralized matching poli-

cies, simplifying system-level reward structures [92]. Second, centralized-agent approaches treat the platform as a unified agent, simplifying large action spaces but requiring tight coordination to maintain high-quality matches [93, 94]. Additionally, system performance is sensitive to key parameters like the matching window and radius, both of which directly affect wait times and match success rates [95].

While early matching strategies struggled with spatial-temporal imbalances, modern research incorporates proactive dispatching to reposition idle vehicles based on anticipated demand. This improves future match outcomes and complements real-time matching systems.

For individual driver repositioning, common reward metrics include trip fare, net profit, idle cruising distance, and the ratio of trip mileage to idle cruising mileage [96–99]. Action spaces range from fine-grained road vectors to more scalable grid systems (e.g., hexagonal partitions). State representations typically use spatiotemporal features in low-dimensional formats, often with tabular or deep learning-based value functions. Algorithms span from model-free methods like Q-learning and DQN to hybrid and Monte Carlo approaches [99–102].

Regarding RL paradigms for matching and dispatching, DQN has been widely used due to its ability to handle high-dimensional inputs. However, it introduces significant computational overhead due to its reliance on experience replay and network updates [103, 104]. Actor-Critic methods offer improved learning stability and better performance in continuous action spaces by learning both the policy and value function simultaneously, though they are computationally heavier [105, 106].

HRL introduces a hierarchical structure to decompose complex decision-making into subtasks. This can enhance convergence and learning efficiency, making it particularly effective for dynamic transportation systems requiring coordination between dispatching and matching [105]. Despite these advantages, existing freight systems often treat matching and dispatching as separate tasks, resulting in inefficiencies. While recent studies explore joint optimization using RL, none have fully integrated HRL with geospatial indexing systems like H3 for scalable multi-resolution adaptation. This gap presents a unique opportunity to develop an anticipatory, unified framework that dynamically responds to spatial-temporal demand patterns across urban freight networks. The key distinctions between our HRL-based model and prior works are summarized in Table 5.1.

5.3 Methodology

This study introduces an innovative hierarchical reinforcement learning (HRL) framework designed to optimize real-time freight matching and dispatching through actor-critic net-

Table 5.1 Freight Matching Optimization Models Comparison

Ref	Key Challenges	Metrics	Method	Matching Criteria	Optimization Goals
[69]	Dynamic Fleet Management Multi-hop Routing	Fleet Utilization, Vehicle Profits Idle Driving Time, Number of Hops, Request Acceptance Rate, Occupancy Rate	DRL	ETA Capacity	Minimize: Wait Time, Mismatch, Travel time, Vehicles
[48]	Multi-transfer Freight Delivery	Delivery Rate, Fuel Consumption Number of Unfinished Packages, Average Driving Time	MILP+ DF	ETA Truck Capacity Arrival Time	Maximize: Served Requests Minimize: Fuel Consumption, Driving Time
[46]	Dynamic Shipment Matching Multi-modal Coordination	Total Cost, Computation Time Utilization of Services, Delay Time Carbon Emissions	HA	Truck Capacity Time Window Carbon Emissions	Minimize: Matching Cost, Total Costs
[18]	Compensation Strategy Stochastic Arrivals	Order Acceptance Scale, Compensation Costs	-	Departure Time Truck Capacity Pick-up Distance	Maximize: Orders Accepted Minimize: Compensation Costs
[71]	Dynamic Matching Complexity Algorithm Efficiency	Matching Success Rate Computational Efficiency	DL	Environmental Impacts Vehicle-Cargo Matching	Matching Efficiency
Our Model	**Dynamic Matching-Dispatching Algorithm Efficiency**	**Average Reward, Average Match, Waiting Time, Reward Per Match Time Computation Cost**	**HRL + H3**	**Geo Proximity Truck Capacity Satisfaction Score**	**Maximize: Reward, Matches Minimize: Empty Mileage, Vehicle Idle Time, Capacity Differences**

DRL: Deep Reinforcement Learning, HA: Heuristic Algorithm, DL: Deep Learning, DF: Deep Freight, HRL: Hierarchical Reinforcement Learning

works. The framework ensures an adaptive, data-driven response by aligning shipments with vehicles based on contextual considerations. It employs an actor-critic algorithm that integrates policy-based (actor) and value-based (critic) reinforcement learning methods: the actor network, parameterized by θ, learns a policy $\pi_\theta(a|s)$ to maximize expected cumulative rewards, while the critic network, parameterized by w, approximates the value function $V_\pi(s)$ to guide policy updates. To enhance training stability and convergence, we adopt the Proximal Policy Optimization (PPO) algorithm, which introduces a clipped surrogate objective to prevent excessively large policy updates. The PPO loss function is defined as:

$$L_{\text{PPO}}(\theta) = \mathbb{E}_t \left[\min\Big(r_t(\theta) A_{\pi_\theta}(s_t, a_t), \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon) A_{\pi_\theta}(s_t, a_t) \Big) \right] \tag{5.1}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio between new and old policies, $A_\pi(s_t, a_t)$ is the advantage function, and ϵ clips large policy updates.

The critic minimizes the temporal difference (TD) error:

$$L_{\text{critic}}(w) = \frac{1}{2} \left(r_t + \gamma V_\pi(s_{t+1}) - V_\pi(s_t) \right)^2 \tag{5.2}$$

PPO ensures a balance between exploration and exploitation, leading to reliable convergence in dynamic freight environments.

The framework utilizes an H3 geospatial zoning system to efficiently manage spatial data. H3 provides uniform, multi-resolution hexagonal partitions, reducing computational complexity. The matching agent selects optimal vehicle-cargo pairings using a Pre-filtering Algorithm for

the Matching Agent (PAMA), which narrows search areas using H3 partitions. This targeted search minimizes response time and computational overhead.

The dispatching agent dynamically assigns unassigned vehicles to high-demand Geospatial Hexagonal Units (GHUs) to enhance coordination. Vehicles are repositioned hierarchically, considering real-time updates to avoid unnecessary travel and maximize efficiency.

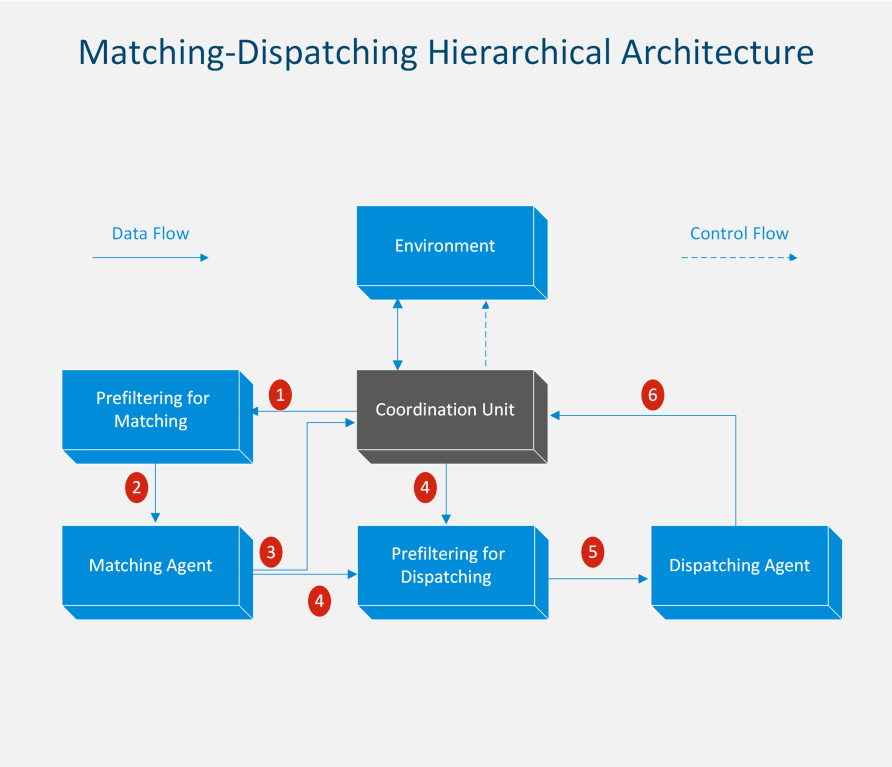

Figure 5.1 Matching-Dispatching Hierarchical Framework Design and Component Interaction

By continuously refining policies based on real-time data, this hierarchical learning approach improves freight logistics and enhances service quality.

5.3.1 Model Architecture

Figure 5.1 illustrates the architecture of our proposed hierarchical framework and the interaction steps between its components. The coordination hub is responsible for maintaining and updating the information at each time step according to the dispatching and matching decisions made, as detailed in algorithm 1. The framework incorporates several internal components to facilitate the matching and dispatching processes. Specifically, PAMA generates a pool of the N most suitable vehicle candidates for matching. The matching agent optimizes vehicle-shipment pairings using RL. Pre-filtering Algorithm for the Dispatching Agent

(PADA) identifies eligible GHUs for relocating idle vehicles by analyzing spatiotemporal patterns of supply and demand within GHUs, and the dispatching agent proactively positions idle vehicles.

Algorithm 1 Hierarchical Framework for Matching and Dispatching

Input: Shipment requests, Vehicle attributes
Output: Matched vehicles, Dispatched vehicles
procedure HIERARCHICAL FRAMEWORK

    for $t \in T$ **do**
        Fetch Shippers' requests at time step t, $\mathcal{R}$
        for $r \in \mathcal{R}$ **do**
            Create Matching pool Based on demand r features
            if The pool contains real vehicle **then**
                Perform Matching demand r to selected vehicle
            else
                Increment r
            end if
        end for
        Update the environment information
        Fetch idle vehicles at time step t, $\acute{\mathcal{V}}$
        for $\acute{v} \in \acute{\mathcal{V}}$ **do**
            Obtain eligible H3 GHUs to dispatch vehicle $\acute{v}$
            if is(eligible H3) **then**
                Perform Dispatching the vehicle $\acute{v}$ to the selected nearest eligible GHUs
            else
                Increment $\acute{v}$
            end if
        end for
    end for
end procedure

At each time step, the framework receives shipment requests and vehicle attribute information. Based on each shipment's preferences and features, a matching pool of suitable vehicle candidates is created. The matching agent then uses an actor-critic network to select the optimal vehicle from this pool. Following the matching decision, the coordination hub updates all relevant system information in real time.

To dispatch idle vehicles, the PADA evaluates idle vehicles' service time, attributes, and predicted demand in GHUs. It then identifies nearby eligible GHUs—defined as spatial cells that are reachable by the vehicle within its remaining service time and are projected to have excess demand during the vehicle's arrival window. These GHUs are marked as viable

repositioning targets. This predictive capability ensures vehicles are proactively moved to areas where they are most likely to be matched efficiently.

The dispatching agent uses an actor-critic network to dispatch each idle vehicle to the nearest eligible GHU. Once dispatched, the coordination hub updates real-time geospatial and vehicle assignment data, preventing repeated dispatches to the same high-demand GHUs and enabling the matching agent to anticipate future vehicle availability. This synergy between continuous spatiotemporal updates and coordinated agent actions enables optimal resource allocation, balancing short-term responsiveness with long-term efficiency.

5.3.2 Coordination Hub

The coordination hub is the central component of the HRL framefowk, integrating matching and dispatching agents through comprehensive information management. It maintains real-time data on vehicle attributes like location, capacity, quality score, and service time, enabling precise vehicle suitability assessments and efficient shipment request processing. To facilitate the interaction between these agents, a central coordination hub is essential; it serves as the connecting interface that synchronizes the actions of the matching and dispatching agents described later. Continuously updating the operational information and tracking vehicle assignments, the hub ensures that both agents work with the most current information.

5.3.3 Pre-filtering Algorithm for Matching Agent

The PAMA aims to identify the most suitable vehicle for each shipment request based on spatial proximity, service time, capacity, and quality score. Capacity reflects either the vehicle's load limit or the shipment's demand, while the quality score captures the service expectations from both shipment and driver perspectives. To keep the search space manageable, the pre-filtering algorithm is applied to eliminate suboptimal candidates early. This significantly reduces the action space, enhances scalability, and ensures real-time decision-making remains tractable in large-scale environments.

In our approach, PAMA as outlined in Algorithm 2 and 3 generates a pool of N candidate vehicles for the matching agent. If there are insufficient real vehicles that meet the shipment's criteria, PAMA introduces fictitious vehicles as placeholders to maintain a consistent pool size. This ensures a stable input structure for the matching agent. The pool is constructed as follows:

1. The algorithm first assesses the H3 tag of a shipment's GHU at resolution level H(level 7 in our implementation), along with the required capacity and quality score for the

Algorithm 2 PAMA Part 1

Input: ShipmentRequirements, VehiclesInfo, ResolutionLevel H, RequiredCandidates N.
Output: Pool of candidate vehicles.
procedure VEHICLE SELECTION ALGORITHM

    Assess shipment's origin and requirements.
    Search for vehicles within the hexagonal region.
    Evaluate how many vehicles meet the criteria, and z_{p_r} gets its number.
    if $z_{p_r} > N$ **then**
        $p_r \leftarrow$ select_random_vehicles(Vehicles, N)
    else if $z_{p_r} == N$ **then**
        $p_r \leftarrow$ Vehicles
    else
        Expand search to the surrounding ring.
        $m_{p_r} \leftarrow$ count_suitable_vehicles(ring_vehicles)
        if $m_{p_r} > (N - z_{p_r})$ **then**
            $p_r \leftarrow p_r +$ select_random_vehicles(Vehicles, $N - z_{p_r}$)
        else if $m_{p_r} == (N - z_{p_r})$ **then**
            $p_r \leftarrow p_r + ring_vehicles$
        else
            $p_r \leftarrow p_r + ring_vehicles$
            Proceed to Vehicle Selection Algorithm Part 2
        end if
    end if
end procedure

shipment. It then searches for all available vehicles within this initial GHU that satisfy these criteria.

- If more than N vehicles are identified ($z_{p_r} > N$), the algorithm randomly selects N vehicles to form the candidate pool.(GO to END)
- If exactly N vehicles meet the criteria ($z_{p_r} = N$), they are directly added to the pool of candidates. (GO to END)
- If ($z_{p_r} < N$) are found in the initial GHU They are directly added, and the algorithm expands its search to the two surrounding rings, increasing the potential pool of candidates. (GO to STEP2)

2. In the surrounding rings, the algorithm seeks required m_{p_r} vehicles to complete the candidate pool, while ensuring they meet service time, capacity, and quality score requirements. Figure 5.2 illustrates the initial GHU representing the shipment origin,

Algorithm 3 PAMA Part 2

1: **if** W_{p_r}==0 **then**
2: Vehicles ← Quality(GHUs, *quality_req*)
3: g_{p_r} ← Count_Suitable_Vehicles(Vehicles)
4: **if** g_{p_r}==N - z_{p_r} - m_{p_r} **then**
5: $p_r \leftarrow p_r + Vehicles$
6: **else if** g_{p_r} >N - z_{p_r} - m_{p_r} **then**
7: $p_r \leftarrow p_r +$ Select_Random(Vehicles, $N - z_{p_r} - m_{p_r}$)
8: **else**
9: $p_r \leftarrow p_r + Vehicles$
10: Vehicle ← Capacity(GHUs, *quality_req*)
11: k_{p_r} ← Count_Suitable_Vehicles(Vehicle)
12: **if** k_{p_r} == N - z_{p_r} - m_{p_r} - g_{p_r} **then**
13: $p_r \leftarrow p_r + Vehicles$
14: **else if** k_{p_r} >N - z_{p_r} - m_{p_r} - g_{p_r} **then**
15: $p_r \leftarrow p_r +$ Select_Random(Vehicles, $N - z_{p_r} - m_{p_r} - g_{p_r}$)
16: **else**
17: $p_r \leftarrow p_r + Vehicles$
18: $p_r =$ Add_Fictitious_Candidates(p_r , $N - z_{p_r} - m_{p_r} - g_{p_r} - k_{p_r}$)
19: **end if**
20: **end if**
21: **else if** W_{p_r} >0 **then**
22: Expand_SearchIn_Hexagon($VehiclesInfo$, *origin*, H)
23: **if** g_{p_r} >N - z_{p_r} - m_{p_r} **then**
24: $p_r \leftarrow p_r +$ Select_Random(Vehicles, $N - z_{p_r} - m_{p_r}$)
25: **else if** g_{p_r} == N - z_{p_r} - m_{p_r} **then**
26: $p_r \leftarrow p_r + Vehicles$
27: **else**
28: $p_r \leftarrow p_r + Vehicles$
29: Vehicles ← Quality(GHUs, *quality_req*)
30: l_{p_r} ← Count_Suitable_Vehicles(Vehicles.
31: **if** l_{p_r} >N - z_{p_r} - m_{p_r} - g_{p_r} **then**
32: $p_r \leftarrow p_r +$ Select_Random_Vehicles(Vehicles, $N - z_{p_r} - m_{p_r} - g_{p_r}$)
33: **else if** l_{p_r} == N - z_{p_r} - m_{p_r} - g_{p_r} **then**
34: $p_r \leftarrow p_r + Vehicles$
35: **else**
36: $p_r \leftarrow p_r + Vehicles$
37: $o_{p_r} \leftarrow \{v \in vehicles \mid v.\text{capacity} = capacity\_req + 1\}$
38: **if** o_{p_r} >N - z_{p_r} - m_{p_r} - g_{p_r} - l_{p_r} **then**
39: $p_r \leftarrow p_r +$ Select_Random(Vehicles, $N - z_{p_r} - m_{p_r} - g_{p_r} - l_{p_r}$)
40: **else if** o_{p_r} == N - z_{p_r} - m_{p_r} - g_{p_r} - l_{p_r} **then**
41: $p_r \leftarrow p_r + Vehicles$
42: **else**
43: $p_r \leftarrow p_r + Vehicles$
44: Add_Fictitious_Candidates(p_r)
45: **end if**
46: **end if**
47: **end if**
48: **end if**

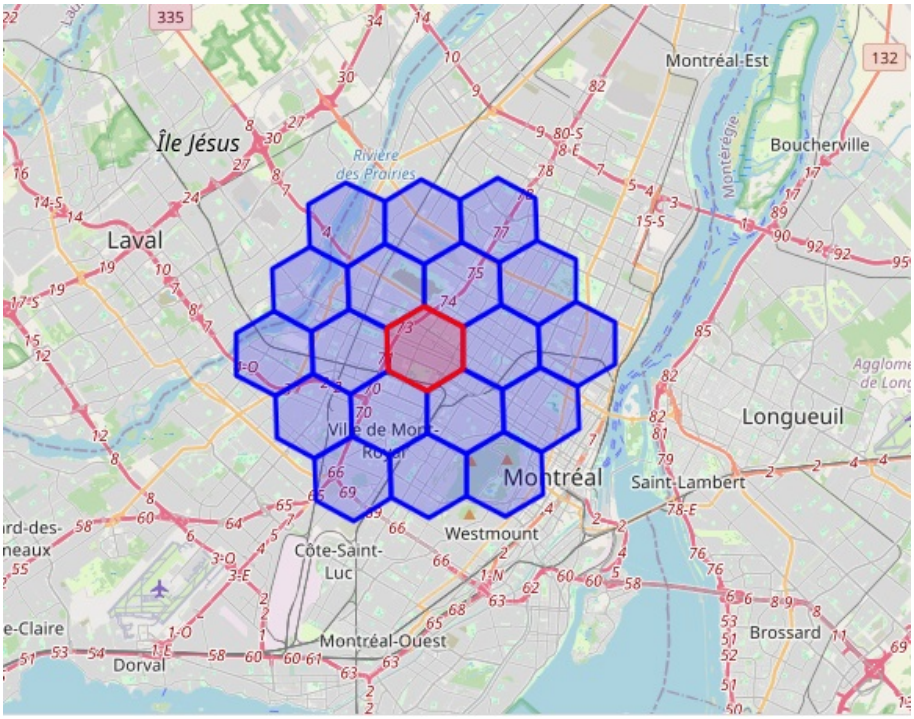

Figure 5.2 Initial GHU(red) and its two surrounding rings(blue)

along with the surrounding rings.

- If the search identifies ($m_{p_r} > N - z_{p_r}$) vehicles, $N - z_{p_r}$ of them are randomly chosen to finalize the pool. (GO to END)
- If ($m_{p_r} = N - z_{p_r}$) are found in the surroundings, all found vehicles are added to complete the pool. (GO to END)
- If ($m_{p_r} < N - z_{p_r}$) vehicles are found, the pool includes all m_{p_r} vehicles and expands the search criteria based on the shipment's required time of service. (GO to STEP 3 or 5)

3. If a shipment requires immediate pickup, the algorithm seeks g_{p_r} available vehicles with a quality score reduced by one unit within the initial H3 GHU and its surrounding hexagonal ring to identify sufficient vehicles for the pool.

- If ($g_{p_r} > N - z_{p_r} - m_{p_r}$) , a random selection of g_{p_r} vehicles is made. (GO to END)
- If ($g_{p_r} = N - z_{p_r} - m_{p_r}$) are found in the surrounding ring, they are combined with the initial candidates to complete the pool. (GO to END)
- If ($g_{p_r} < N - z_{p_r} - m_{p_r}$) within the initial GHU and its surrounding GHUs, they are combined with the initial candidates to complete the pool. (GO to STEP 4)

4. The algorithm searches for vehicles with one unit higher capacity (k_{p_r}) within the same area.
    - If ($k_{p_r} = N - z_{p_r} - m_{p_r} - g_{p_r}$) are found, they are added to the existing pool. (GO to END)
    - If ($k_{p_r} > N - z_{p_r} - m_{p_r} - g_{p_r}$) , a random selection of k_{p_r} vehicles is made. (GO to END)
    - If fewer than k_{p_r} suitable vehicles are available,they are added, and the pool creates fictitious candidates to meet the required size. (GO to END)
5. If the shipment can wait for pickup within the next tolerable time steps ($W_{p_r} > 0$), the pool expands its search in the initial GHU and its surrounding ring to search for g_{p_r} vehicles that meet the exact requirements.
    - If ($g_{p_r} > N - z_{p_r} - m_{p_r}$), a random selection of g_{p_r} vehicles is made. (GO to END)
    - If ($g_i = N - z_{p_r} - m_{p_r}$) are found in the surrounding ring, they are combined with the initial candidates to complete the pool. (GO to END)
    - If ($g_{p_r} < N - z_{p_r} - m_{p_r}$), they are added. (GO to Step 6)
6. The pool searches for required vehicles with one unit lower quality score (l_{p_r}) within the initial H3 GHU and its surrounding hexagon ring within the next tolerable time steps.
    - If ($l_{p_r} = N - z_{p_r} - m_{p_r} - g_{p_r}$), they are added to the existing pool.(GO to END)
    - If ($l_{p_r} > N - z_{p_r} - m_{p_r} - g_{p_r}$), a random selection of l_{p_r} vehicles is made. (GO to END)
    - If (l_{p_r} <N$-z_{p_r} - m_{p_r} - g_{p_r}$), they are added. (GO to Step7)
7. The algorithm searches for vehicles with one unit higher capacity (o_{p_r}) within the initial H3 GHU and its surrounding hexagon ring.
    - ($o_{p_r} = N$ - z_{p_r} - m_{p_r} - g_{p_r} - l_{p_r}), they are added to the existing pool. (GO to END)
    - If (o_{p_r} >N - z_{p_r} - m_{p_r} - g_{p_r} - l_{p_r}), a random selection of o_{p_r} vehicles is made. (GO to END)
    - If ($o_{p_r} < N - z_{p_r} - m_{p_r} - g_{p_r} - l_{p_r}$), they are added, and the pool creates fictitious candidates to ensure the pool reaches the required size. (GO to END)

Once the pool is finalized, PAMA sends its details to the matching agent. If the pool contains only fictitious vehicles, PAMA excludes it from matching.

5.3.4 Matching Agent

The upper-level agent in the HRL framework is the matching agent, which pairs shipments with suitable vehicles by selecting from the pool generated by PAMA. PAMA evaluates multiple factors—including proximity to the shipment location, available capacity, vehicle type, quality score, and service time—to ensure only the most relevant options are presented. The matching agent then balances these factors to optimize resource utilization while meeting diverse logistical requirements, though trade-offs are inevitable: prioritizing proximity may reduce capacity efficiency, matching capacity could compromise quality scores, and focusing on quality scores might increase empty mileage to pick up.

To effectively manage these trade-offs, the agent employs an RL approach guided by a reward function, which assigns numerical values to potential matches. By continuously refining its decision-making based on cumulative rewards, the agent enhances pairing efficiency and ensures optimal real-time freight matching.

Matching Reward Function

The matching agent uses a reward-based system to evaluate and select optimal pairings, maximizing expected rewards. Its reward function assesses multiple key factors to guide these decisions, ensuring efficient freight matches. It encourages shorter empty mileage to enhance operational efficiency and minimizes idle and waiting times for both vehicles and shipments, promoting faster matches in available time. Additionally, it prioritizes capacity alignment, rewarding vehicles that closely match shipment sizes while discouraging the use of significantly oversized options. The function also accounts for quality score parity, slightly rewarding vehicles that exceed shipper expectations while penalizing those that fall short.

Mathematically, this is formalized in equation (5.3) as $F(X_{rv})$, which incorporates several key components: (1) d_{rv}: distance between shipment location and vehicle to pick up, (2) T_v: idle time for the vehicle, (3) T_r: waiting time for shipment, (4) D: the diameter of the GHU at H3 resolution 6, (5) W_v^d: weight assigned to distance per kilometer for each vehicle type, (6) W_{rv}^q: weight of quality scores parity, (7) W_{rv}^c: weight of capacity parity, reflecting the difference between the shipment's size requirement and the vehicle's actual capacity. (8) The binary decision variable X_{rv} is equal to 1 if a match occurs with a real vehicle and 0 otherwise. (9) U: a loss amount applied when an unserved shipment switches to another platform, discouraging matches with fictitious vehicles

$$F(X_{rv}) \quad = \quad -\sum_{r\in\mathcal{R}}\sum_{v\in\mathcal{V}}\Big[(d_{rv}+(T_v+T_r)\,D)\,W_v^d \;+\; W_{rv}^q \;+\; W_{rv}^c\Big]X_{rv} \;+\; U\,(1-X_{rv}) \quad (5.3)$$

The negative sign ensures optimization minimizes undesirable factors while effectively maximizing the overall reward.

Matching Environment

In our simulation, we make certain assumptions regarding vehicle movement across GHUs. We assume that vehicles can move across a single GHU at a H3 resolution 6—equivalent to three GHUs at a H3 resolution 7—within each time step. This translates to an effective travel mileage of approximately 7.45 kilometers per time step. This spatial-temporal mapping is crucial for integrating both distance and time into our logistics optimization framework.

The simulation's dynamics are further defined by two fundamental assignment principles that reflect real-world operational constraints:

- *Vehicle Assignment*: Each vehicle may be assigned to at most one shipment or remain idle.
- *Shipment Assignment*: Each shipment request may be assigned to at most one vehicle or remain unassigned.

State Space of Matching Agent

The state space is represented as a multidimensional vector that encapsulates all relevant information needed for decision-making. This vector combines two key components: (1) the shipment's details r, which include pick-up coordinates, cargo size, shipment waiting time, and satisfaction score; and (2) the candidate vehicle pool P_r, comprising each vehicle's current location, capacity, service time, and quality score. These components are merged into a single state vector $s_r = (r,\ P_r)$. The combined state vector provides the matching agent with all the necessary information to make optimal decisions.

Action Space of Matching Agent

The action space represents the set of strategic choices available to the matching agent for vehicle-shipment assignments. For a given shipment r at time step t, the agent selects an action a_r from N discrete actions, where each action corresponds to matching the shipment

with one of the N candidate vehicles in the pool. The selected vehicle is directed to the shipment's location to fulfill the request.

To ensure robust learning, the agent uses an epsilon-greedy strategy to balance exploitation of known actions and exploration of new ones. This promotes broader search space coverage while maintaining efficiency in real-time freight matching. After matching, unassigned vehicles are repositioned by the dispatching agent to improve future match opportunities.

Algorithm 4 PADA

```
Input: Shipper requests, Vehicle attributes, Timestamp, GHUs' tag and index
Output: Eligible GHUs, Non eligible GHUs
for idle vehicle do
    Fetch idle vehicle's attributes
    Calculate Remaining service time.
    Determine All reachable hexagonal rings on the remaining time.
    for each GHU within the hexagonal rings do
        if GHU within the hexagonal rings then
            Calculate Arrival time.
            Estimate Demands on the vehicle based on arrival time.
            Estimate Amount of the vehicle type based on arrival time.
            Calculate Excess demand
            if Has excess Demand then
                Assign 1
            else
                Assign −1
            end if
        else
            Assign −1
        end if
    end for
end for
```

5.3.5 Pre-filtering Algorithm for Dispatching Agent

The PADA, outlined in algorithm 4 is designed to identify the eligible GHUs to relocate idle vehicles. It assesses projected demand imbalances while considering vehicle capacity constraints and quality score compatibility with shipment requirements. The algorithm follows these steps:

1. *Initial Evaluation*: Each idle vehicle is assessed based on its current H3 tag, along with its capacity, quality score, and remaining service time.

2. *Spatial Search Expansion*: The search expands to surrounding hexagonal rings to identify feasible relocation options. It is assumed that a vehicle can traverse up to three GHUs per time step in H3 resolution level 7.

3. *Demand-Supply Analysis*: The algorithm evaluates spatiotemporal supply and demand within the identified GHUs, estimating future demand and supply based on the vehicle's expected arrival time.

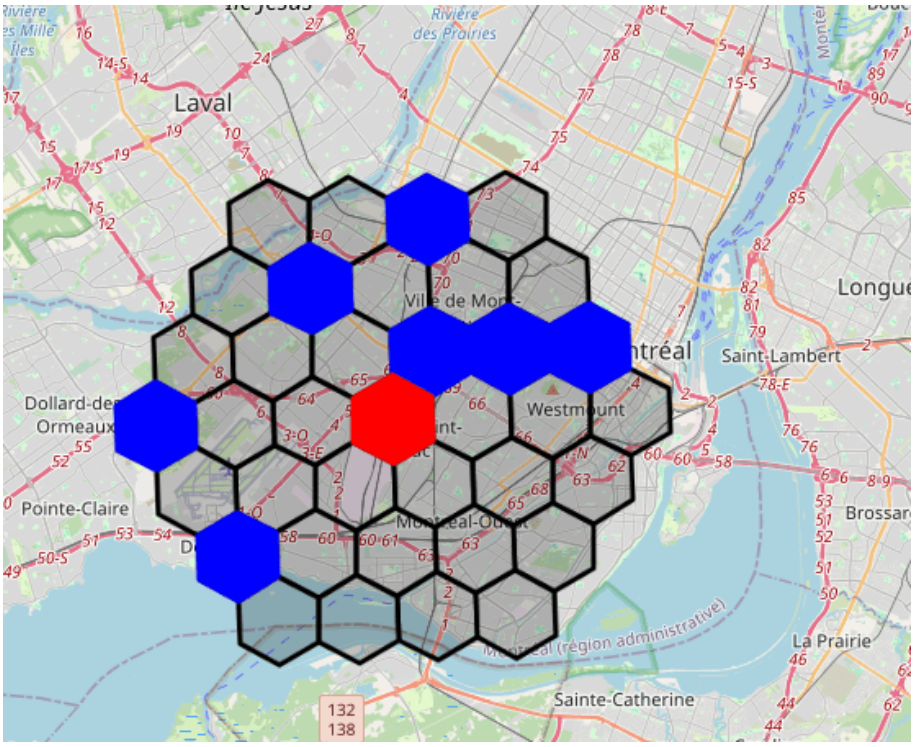

Figure 5.3 Initial GHU(red) and Surrounding Rings(black) with Identified Surplus Demand(blue)

4. *GHU Eligibility Assignment*: GHUs with projected surplus demand are marked as eligible for relocation with a value of 1, while those without surplus demand or outside the selected GHUs are assigned a value of -1.

Figure 5.3 illustrates the initial GHU(red), surrounding hexagonal rings available for relocation(black), and eligible GHUs for vehicle dispatch(blue).

5.3.6 Dispatching Agent

The dispatching agent optimizes vehicle deployment by directing idle vehicles to high-demand GHUs. This agent enhances operational efficiency through intelligent resource allocation,

minimizing operational delays while balancing supply-demand dynamics.

State Space of Dispatching Agent

The state space of the dispatching agent is represented as a comprehensive vector $s_v = (E_{\acute{v}}, (latH_{\acute{h}}, lonH_{\acute{h}}), (latV_{\acute{v}}, lonV_{\acute{v}}))$ that integrates all critical information necessary for optimal dispatching decisions. This vector allows the agent to make informed choices for efficient vehicle relocation and demand fulfillment. It consists of the following components:

- *Hexagon Eligibility Vector ($E_{\acute{v}}$)*: A binary vector generated by algorithm 4, indicating which GHUs are eligible for vehicle dispatch.
- *Hexagon Central Points Vector ($latH_{\acute{h}}, lonH_{\acute{h}}$)*: It stores the latitude and longitude of the central points of all GHUs, providing spatial references for distance calculations.
- *Vehicle Positions Vector ($latV_{\acute{v}}, lonV_{\acute{v}}$)*: Records the latitude and longitude of an idle vehicle, crucial for evaluating remaining service time, identifying reachable GHUs, and computing distances.

Action Space of Dispatching Agent

: The action space consists of a set of Q discrete actions, where each action corresponds to dispatching the idle vehicle to one of the Q GHUs.

The action space, denoted as $A_{\acute{v}}$, is defined as $A_{\acute{v}} = \{1, 2, ..., Q\}$ where each action $a_{\acute{v}}\ \varepsilon\ A_{\acute{v}}$ represents the index of the chosen GHU. Upon selecting an action, the agent directs the idle vehicle to the central point of the corresponding GHU.

To balance exploration and exploitation, the dispatching agent employs the ϵ-greedy strategy.

Dispatching Reward Function

The reward function encourages efficient dispatching by minimizing the distance between idle vehicles and eligible GHUs, thus promoting efficient dispatching. The function is formulated as:

$$G(Y_{\acute{v}\acute{h}}) = -\sum_{\acute{v}\in\acute{\mathcal{V}}}\sum_{\acute{h}\in\acute{\mathcal{H}}} \acute{d}_{\acute{v}\acute{h}} Y_{\acute{v}\acute{h}} + \acute{U}\,(1 - Y_{\acute{v}\acute{h}}) \tag{5.4}$$

In the equation (5.4), $\acute{v}$ represents each idle vehicle, and $\acute{h}$ represents each eligible GHU. The term $\acute{d}_{\acute{v}\acute{h}}$ denotes the distance between vehicle $\acute{v}$ and GHU $\acute{h}$. $Y_{\acute{v}\acute{h}}$ is a binary decision variable that is 1 if idle vehicle $\acute{v}$ is dispatched to eligible GHU $\acute{h}$ and 0 otherwise. The penalty $\acute{U}$

is applied for any noneligible relocation, ensuring that the agent is discouraged from making such decisions. The negative sign ensures the agent maximizes rewards by minimizing total dispatch distances, thereby enhancing fleet efficiency.

5.3.7 Modularity and Generalizability of PAMA and PADA

The proposed pre-filtering algorithms—PAMA and PADA—are designed as modular components that operate independently of the RL backbone. Their modular design allows seamless integration into various RL frameworks, including DQN, policy-gradient methods, and multi-agent systems. By decoupling spatial preprocessing from policy learning, these modules reduce the search space efficiently and enhance scalability, making the overall framework adaptable to diverse real-time freight and logistics environments.

5.3.8 Computational Complexity Analysis

The proposed HRL system achieves high computational efficiency by combining spatial prefiltering and Actor-only neural inference during deployment.

The PAMA confines candidate search to a bounded H3 neighborhood (initial GHU plus two rings), resulting in constant-time complexity $\mathcal{O}(1)$ per shipment. Similarly, the PADA restricts evaluations to GHUs within a mobility range, incurring $\mathcal{O}(1)$ complexity per idle vehicle.

During training, both matching and dispatching agents employ Actor and Critic neural networks composed of L hidden layers. The matching agent's Actor and Critic networks contain $M_a^{(m)}$ and $M_c^{(m)}$ neurons per layer, respectively, with combined forward-pass complexity $\mathcal{O}\Big(L \cdot (M_a^{(m)})^2 + L \cdot (M_c^{(m)})^2\Big)$. The dispatching agent uses analogous networks with $M_a^{(d)}$ and $M_c^{(d)}$ neurons sizes per hidden layer, leading to training complexity $\mathcal{O}\Big(L{\cdot}(M_a^{(d)})^2{+}L{\cdot}(M_c^{(d)})^2\Big)$.

At deployment, only Actor networks are active, reducing per-inference complexity to $\mathcal{O}\Big(L \cdot (M_a^{(m)})^2\Big)$ for each shipment and $\mathcal{O}\Big(L \cdot (M_a^{(d)})^2\Big)$ for each idle vehicle. These bounded and scalable structures ensure real-time responsiveness and efficient scalability in large-scale urban freight operations.

5.4 Case study

To perform a matching process, two datasets are required: Origin-Destination (OD) data and trucking data. The OD data includes characteristics of shipping requests such as the geolocation of the origin and destination, commodity type, and timing, while the trucking

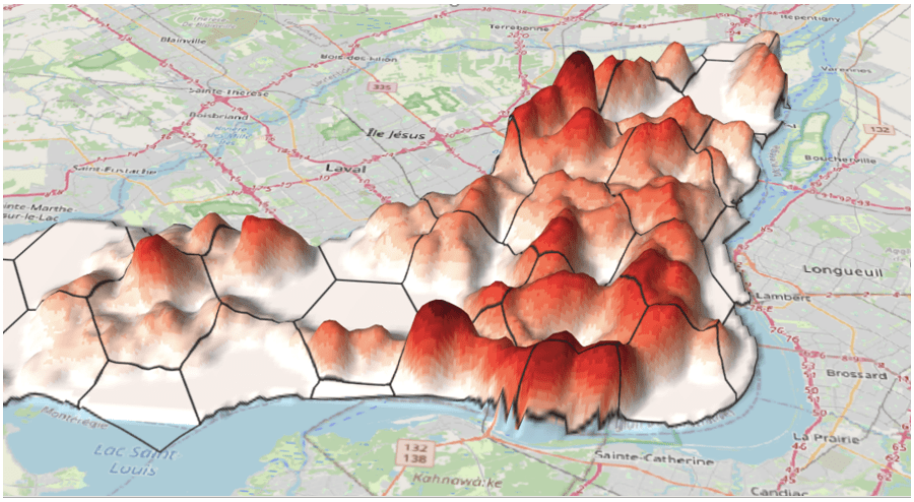

Figure 5.4 Distribution of simulated trips in Montreal

data describes carrier attributes such as capacity and availability. The following methodology was developed to construct both datasets.

The OD data construction involves three main steps:

- Identifying the key factors that influence the generation of shipping requests within an urban environment.
- Developing a method to approximate the geographical locations of shipment origin and destination points.
- Structuring the OD data and defining relevant characteristics of each shipment.

Urban freight demand models provide a basis for understanding freight movement patterns and their structural components, as explored in our previous work on freight matching in Montreal [107]. Based on Caspersen [108] and the Federal Highway Administration [109], a typical urban freight network comprises factories, distribution centers, shopping centers, and end consumers.

Industrial areas serve as hubs for distributing goods to commercial and residential areas, while warehouses may act as both shipment origins and reverse logistics nodes. Residential areas and shopping centers can both initiate and receive shipments. These elements influence trip generation, and their local density correlates with shipping activity. As such, areas with higher concentrations of these features are more likely to generate or attract freight demand.

To estimate these probabilities, we collected spatial data on population, industrial zones, and shopping centers across Montreal. Using QGIS, we generated and reclassified raster density

layers into 11 levels and merged them to produce a composite density surface. This surface was then converted into point layers representing shipping origins and destinations, randomly assigned within the urban network. Simulation results showed that the majority of requests originated from denser areas like downtown Montreal, while zones such as Sonneville and Montreal East produced fewer. Figure 5.4 illustrates the spatial distribution of generated trips throughout the island of Montreal.

Since origin and destination layers were created separately, OD pairings were formed by linking points within 11 defined zones. High-density zones produced more origin-destination pairs, while low-density regions contributed fewer. This zoning method enabled realistic linkage of trip endpoints while preserving variability across spatial regions.

To enrich the OD dataset, we incorporated additional shipment attributes based on forms and reports from the Transportation Association of Canada (TAC) and Transport Canada. Following the work of Lightstone, Belony, and Cappuccilli [110], we included data fields such as request date, shipment size and weight, and commodity category (e.g., wood, grain).

For the trucking dataset, we assumed trucks were distributed in proportion to regional freight demand. Each simulated truck was assigned a type, service availability, and cargo profile, ensuring consistency with the heterogeneous nature of urban freight systems. This method enables the simulation to reflect realistic spatial and logistical variability throughout the city.

5.5 Results

This section presents and analyzes the simulation results for the real-time matching and the hierarchical matching-dispatching framework using the dataset. All experiments were conducted on a system equipped with an Intel Core i7 processor, 64 GB RAM, and an 8 GB GPU. The actor-critic approach was implemented using PyTorch in Python.

5.5.1 Hyperparameter Selection and Network Archicture

The actor and critic networks in our hierarchical reinforcement learning framework are designed with two hidden layers, leveraging the Rectified Linear Unit (ReLU) activation to balance nonlinearity and computational efficiency [107]. A softmax function is employed in the output layer of the actor network.

The policy is updated every 64 timestep, ensuring frequent updates to the learning process. The policy was updated for 20 epochs in each PPO update, allowing the network to learn

Table 5.2 Time Computation Cost of Training for Different Neuron Configurations of Matching and Matching-Dispatching Agents

	Matching Agent				Dispatching Agent			
Critic/Actor Neurons	**128**	**256**	**512**	**1024**	**128**	**256**	**512**	**1024**
128	9h10	9h08	9h14	9h14	17h11	17h12	17h10	17h14
256	9h08	9h10	9h11	9h10	17h09	17h13	17h13	17h11
512	9h11	9h13	9h11	9h15	17h12	17h15	17h11	17h13
1024	9h13	9h16	9h11	9h16	17h10	17h16	17h12	17h11

Table 5.3 Average Reward of Matching Agent for Different Neuron Configurations of Agents

	Matching Agent				Dispatching Agent			
Critic/Actor Neurons	**128**	**256**	**512**	**1024**	**128**	**256**	**512**	**1024**
128	-179255	-184352	-163151	-181196	-189924	-171830	-188209	-182565
256	-199693	-246368	-180099	-197511	-166753	-194896	-186055	-200369
512	-183628	-811575	-181255	-225378	-174954	-247984	-172045	-187450
1024	-684375	-178434	-211954	-406534	-188222	-183905	-218360	-241631

effectively from each batch of experiences. The clip parameter for PPO was set to 0.2, controlling the range within which updates are made to stabilize learning. A discount factor of 0.99 was chosen to balance immediate and future rewards, emphasizing long-term benefits. The learning rates for the actor and critic networks were set to 0.00005 and 0.0001, respectively. These hyperparameters were meticulously selected to ensure robust learning while maintaining computational efficiency. We carefully consider these hyperparameter values to achieve a balance between learning speed, stability, and performance.

In our model, we set the candidate vehicle pool size parameter $N = 10$. The penalty parameters U and $\acute{U}$ are both set to 1000. This penalty value - approximately 135 times larger than the diameter of a resolution-6 GHU - strongly discourages two undesirable actions: matching with fictitious vehicles and dispatching infeasible relocations.

5.5.2 Matching Performance Evaluation

This subsection evaluates the performance of various actor-critic network configurations for the matching agent based on four key metrics: cumulative reward, number of successful matches, average empty mileage, and training time. The goal is to identify the optimal

Table 5.4 Average Successful Matches of Matching Agent for Different Neuron Configurations of Agents

	Matching Agent				Dispatching Agent			
Critic/Actor Neurons	**128**	**256**	**512**	**1024**	**128**	**256**	**512**	**1024**
128	3705	3713	3693	3726	3743	3751	3730	3753
256	3716	3642	3688	3698	3757	3708	3755	3716
512	3693	3265	3720	3672	3717	3686	3761	3726
1024	3339	3686	3699	3521	3732	3741	3721	3713

Table 5.5 Average Empty Mileage for Different Neuron Configurations of Agents

	Matching Agent				Dispatching Agent			
Critic/Actor Neurons	**128**	**256**	**512**	**1024**	**128**	**256**	**512**	**1024**
128	7.63	7.62	7.48	7.63	7.55	7.50	7.53	7.58
256	7.70	7.57	7.65	7.60	7.56	7.41	7.44	7.57
512	7.57	7.57	7.69	7.64	7.46	7.58	7.53	7.57
1024	7.71	7.62	7.66	7.55	7.58	7.53	7.54	7.55

configuration that maximizes overall reward while maintaining high match efficiency.

Table 5.2 summarizes training times across configurations, showing consistency with an average duration around 9 hours, confirming minimal impact of network size on computational cost. Table 5.3 presents cumulative reward values, and Table 5.4 shows the average number of successful matches achieved under each configuration. The newly added Table 5.5 reports the average empty mileage, offering insights into vehicle movement efficiency during pickup.

To ensure the statistical reliability of our findings, we conducted two separate ANOVA tests using reward and match count values across training episodes. Both tests yielded p-value less than 0.001 with 3,199 degrees of freedom, indicating statistically significant differences between configurations.

A higher reward per match translates to increased profitability and reduced operational costs for the freight platform. Among the tested configurations, A512-C128 (Actor: 512, Critic: 128) achieved the highest cumulative reward (-163151) and 3,693 successful matches. This results in a reward per match of approximately -44 (computed as total reward divided by match count), the most efficient among all matching-only setups. Furthermore, it recorded an average empty mileage of 7.48 km, the shortest among all matching-only setups, suggesting

improved spatial alignment between vehicles and shipment locations.

While A1024-C128 achieved the highest match count (3,726), it incurred slightly longer average mileage and reduced reward efficiency. These observations highlight a critical trade-off: network architectures with larger Actor sizes can capture more matches but may introduce inefficiencies such as longer detours or mismatches.

These results demonstrate the importance of balancing Actor-Critic network sizes for optimal performance. A moderately sized Actor network appears especially effective in exploring the action space and learning optimal policies. The reward function—designed to consider empty mileage to pick up, shipment waiting time, vehicle idle time, capacity alignment, and satisfaction scores—allowed the agent to develop robust matching strategies in dynamic freight environments.

The results also validate our framework as a solid foundation for further real-time freight optimization tasks, offering a replicable methodology for evaluating network architectures under HRL settings.

5.5.3 Matching-Dispatching Performance Evaluation

This subsection evaluates the performance of the matching agent when integrated with various configurations of the dispatching agent. The aim is to assess the added value of the dispatching component in improving overall matching effectiveness.

The matching-only configuration (without dispatching) serves as the baseline for comparison. As in the previous section, ANOVA tests confirmed that differences in cumulative reward and match count across configurations were statistically significant (p-value ≈ 0).

Training times for all configurations remained relatively stable (Table 5.2), further confirming that architectural changes in the dispatching agent do not substantially impact training efficiency. This consistency provides flexibility in network selection based on performance goals rather than computational constraints.

The optimal configuration for the integrated system was the A512-C128 matching agent paired with the A128-C256 dispatching agent. This combination achieved 3,757 successful matches, maintained a reward per match of -44. It also maintains an average empty mileage of 7.56 km, which is only marginally higher than the best matching-only setup (7.48 km). This slight increase is acceptable given the added complexity of coordination and is offset by gains in match count and reduced idle time.

Although the configuration with A512-C512 dispatching agent yields a slightly higher match count (3,761), it is associated with slightly worse reward efficiency. Hence, the A512-C128

Figure 5.5 Comparison of reward and successful matches across training episodes for matching-only agent vs. integrated matching-dispatching agents.

matching + A128-C256 dispatching setup is preferred due to its optimal balance of matching performance and operational efficiency.

The integration of the dispatching agent led to a notable improvement in operational responsiveness, with the average vehicle idle time decreasing from 0.157 to 0.148 of timestamps (a 5.7% reduction). Reduced idle time translates to boosts in vehicle utilization, allowing more time in revenue-generating activities and less in waiting. This improves efficiency, enabling freight companies to move more goods with the same fleet and lower operational costs.

As visualized in figure 5.5, the integrated approach outperforms the matching-only setup across all key metrics—reward accumulation, match count, and spatial efficiency. The dispatching agent's role in preemptively repositioning vehicles near future demand zones contributed directly to the observed 1.73% increase in successful match rate and maintained average empty mileage to pick up, despite increased matching complexity.

These results demonstrate that dispatching enhances system responsiveness and spatial balance without compromising service quality or efficiency. The inclusion of empty mileage as an evaluation metric further affirms the practical benefits of hierarchical coordination, supporting real-world logistics optimization goals.

5.5.4 Comparison with State-of-the-Art Approaches

To assess the effectiveness of the proposed HRL framework, we compare it against two baselines: (1) Actor-Critic with kmeans clustering-based spatial zoning (AC+Clustering), and (2) Deep Q-Network with H3 partitioning (DQN+H3). Table 5.6 summarizes performance across both matching and matching-dispatching scenarios.

To enhance this comparison, we introduce *fictitious matches* cases where no real vehicle meets shipment criteria, requiring a placeholder. Fewer fictitious matches reflect better supply-demand alignment and system robustness, especially under constrained availability. This metric is newly included in Table 5.6.

In the matching scenario, our proposed Actor-Critic + H3 model outperforms both baselines in terms of reward per match (-44), successful match count (3693), and empty mileage (7.48 km). Notably, it significantly reduces reliance on fictitious matches, achieving only 17 fictitious matches compared to 19 in the clustering-based model and 230 in the DQN-based model. These results confirm the model's capacity to make reliable and high-quality assignments in a real-time environment.

The benefits become more evident in the integrated matching-dispatching scenario. Actor-Critic + H3 achieves the highest match rate (3757) with the lowest reward penalty per match (-44). It maintains a low fictitious match count (17), while DQN + H3 shows substantial degradation in all metrics, including a much higher fictitious match rate (253), a lower number of matches (3176), and increased empty mileage (8.7 km). Despite the added complexity of dispatching, the Actor-Critic + H3 model maintains consistent training time (17h09), whereas DQN + H3 requires over 20 hours.

These results offer two key insights: the Actor-Critic method consistently outperforms DQN, exhibiting superior convergence stability, more effective reward optimization, and enhanced policy performance in dynamic logistics environments. Simultaneously, the H3 geospatial partitioning system demonstrates clear advantages over clustering-based zoning, offering a more precise, uniform, and scalable spatial representation that improves vehicle positioning accuracy and enhances the effectiveness of real-time dispatching and matching decisions.

Table 5.6 Comparison of Reinforcement Learning Models: Actor-Critic vs. DQN and H3 vs. Clustering

Performance Metric	**AC+H3**	**AC+Clustering**	**DQN+H3**
Matching Scenario			
Reward per Match	−44	−49	−251
Successful Matches	3693	3407	3202
Empty Mileage (KM)	7.48	7.5	8.6
Fictitious Matches	17.0	19.0	230.0
Time Computation Cost	9h14	9h13	9h45
Matching-Dispatching Scenario			
Reward per Match	−44	−49	−294
Successful Matches	3757	3421	3176
Empty Mileage (KM)	7.56	7.6	8.7
Fictitious Matches	17.0	19.0	253.0
Time Computation Cost	17h09	17h10	20h02

5.5.5 discussion

The results highlight a fundamental trade-off between reward maximization and match success rates across different network configurations. Specifically, larger Actor networks—particularly in the matching agent—demonstrate superior performance in fulfilling more customer requests but often at the expense of marginally lower cumulative rewards. Conversely, smaller Actor networks prioritize reward optimization but may compromise on the number of successful matches. This balance reflects the inherent tension between operational efficiency and service quality in freight logistics.

For real-world applications, this trade-off carries significant practical implications. Higher match efficiency directly enhances customer satisfaction by ensuring timely pairings between shipments and carriers, which can improve retention and market competitiveness. However, this benefit may incur higher operational costs or suboptimal resource allocation, potentially reducing short-term profitability. Transportation firms must therefore strategically select configurations based on their priorities: configurations emphasizing reward optimization may suit cost-sensitive operations, while those favoring match efficiency may benefit service-oriented platforms focused on customer experience.

A key strength of our HRL framework is its ability to address the trade-off between efficiency and reward through coordinated dispatching. By repositioning vehicles based on predicted demand and spatial trends, the dispatching agent improves matching while main-

taining reward-per-match by reducing idle time and unnecessary travel. This coordination is crucial in complex logistics settings with dynamic demand, where separate optimization of matching and dispatching is insufficient. The framework's scalability—shown by consistent computational costs (Table 5.2)—and adaptability across conditions highlight its suitability for large-scale freight networks.

5.6 Conclusion

This study presented a novel HRL framework designed for the joint optimization of real-time freight matching and dispatching. The framework's architecture employs a hierarchical structure of Actor-Critic agents, enabling coordinated decision-making for both matching shipments to vehicles and proactively repositioning idle vehicles. A key contribution is the integration of the H3 geospatial partitioning system alongside novel pre-filtering algorithms (PAMA and PADA), which enhances computational efficiency and improves the quality of real-time matching decisions.

Experimental results, obtained through realistic simulations of freight operations in Montreal, demonstrate that the proposed HRL framework outperforms baseline models. Specifically, the joint HRL model achieves a 1.73% improvement in the successful match rate, a 5.7% reduction in vehicle idle time, and maintains low empty mileage. These quantitative improvements highlight the framework's potential to enhance the efficiency of real-world freight transportation.

Our findings reveal a trade-off between maximizing reward and increasing match rates, a challenge commonly encountered in dynamic logistics. The HRL structure mitigates this by facilitating coordination between matching and dispatching, enabling adaptive policies that balance short-term efficiency with long-term system performance.

Future work will focus on extending the framework to incorporate additional real-world complexities, such as dynamic pricing, multi-agent interactions, and real-time traffic conditions, further enhancing the robustness and applicability of smart freight systems in increasingly complex urban environments.

Acknowledgment

This work is supported by Shiphual Logistics and Mitacs through the Mitacs Accelerate program IT30680.

CHAPTER 6 ARTICLE 3: NESTED HIERARCHICAL REINFORCEMENT LEARNING FOR REAL-TIME JOINT PRICING, MATCHING, AND DISPATCHING IN URBAN FREIGHT TRANSPORTATION

Ali Shiri, **Samira Keivanpour**,

International Journal of Production Economics
Submitted on January 29, 2025
First Revision submitted on May 21, 2025
Second Revision submitted on August 20, 2025

Abstract

This paper presents a novel Nested Hierarchical Reinforcement Learning (NHRL) framework for jointly optimizing real-time pricing, shipment matching, and vehicle dispatching in smart urban freight transportation. The proposed model features three coordinated reinforcement learning agents operating at multiple spatial resolutions using H3 geospatial indexing system. A pricing agent dynamically adjusts rates in fine-grained zones based on local supply-demand conditions; a matching agent pairs shipments and vehicles to maximize revenue and service quality; and a dispatching agent relocates idle vehicles across broader areas to anticipate demand hotspots. Modular pre-filtering algorithms (PAMA and DEZE) enable scalable action space reduction. Extensive simulation studies in Montréal and Toronto demonstrate that the NHRL framework outperforms fixed and two-agent baselines across key metrics such as reward, revenue, idle and waiting times, and match accuracy. The results confirm the model's generalizability and robustness for dynamic freight environments, offering practical insights for logistics platforms seeking operational efficiency and profitability.

Keywords: Hierarchical Reinforcement Learning, Smart Freight Transportation, Multi-Resolution Spatial Optimization, Urban Logistics Optimization, Multi-Level Decision-Making, Logistic.

6.1 Introduction

The global freight transportation sector is undergoing a significant transformation driven by the rise of smart technologies and platform-based logistics systems. These smart freight platforms are redefining traditional freight operations by enabling real-time decision-making, improving service quality, and enhancing vehicle utilization by reducing idle time, increasing

match success rates, and ensuring more consistent deployment of available fleet resources across zones. By leveraging data-driven coordination and dynamic control over pricing, matching, and dispatching, these platforms reduce inefficiencies in fleet operations. This leads to better alignment between vehicle availability and shipment demand, minimizes idle capacity, balances supply across regions, and increases the overall effectiveness of resource deployment. According to [111] ([111]), the global digital freight brokerage industry, valued at USD 4.2 billion in 2023, is projected to grow at a compound annual growth rate (CAGR) exceeding 5% through 2032. Similarly, [112] ([112]) estimate that the digital freight forwarding market will reach USD 92.37 billion by 2029, nearly tripling its 2024 valuation. This surge reflects the growing need for intelligent, scalable, and operationally efficient freight management solutions. Urban freight inefficiencies, including underutilized assets, suboptimal dispatching, and congestion-related delays, have been shown to contribute up to 20–30% of total operational costs in last-mile logistics [113]. Moreover, such delays are estimated to impose over $100 billion in annual economic losses across OECD countries [114].

Despite this momentum, traditional freight optimization methods face substantial challenges in adapting to dynamic operational conditions. Techniques such as linear programming, mixed-integer formulations, and heuristic approaches like genetic algorithms have demonstrated effectiveness under static or predictable conditions. However, they are often inadequate when applied to high-frequency, real-time decisions in spatially distributed freight systems. These limitations result in persistent inefficiencies, including pricing mismatches, delivery delays, underutilized vehicles, and uneven service distribution across regions.

RL has emerged as a powerful tool for addressing dynamic decision-making in transportation [115]. It has been applied to optimize individual components such as pricing [116], dispatching [66], and shipment matching [107]. However, few studies have tackled their joint optimization. For instance, [117] coordinated pricing and matching but omitted dispatching, whereas [118] integrated only matching and dispatching. In real-world freight operations, particularly in digital freight platforms like Shiphaul, these tasks are inherently coupled: pricing affects demand, which influences match availability and dispatch priorities. Treating them in isolation undermines system responsiveness, supply-demand balance, and overall platform profitability.

If pricing, matching, and dispatching are handled independently rather than jointly optimized, cascading operational failures can occur. For example, raising prices in a high-demand zone without proactive dispatching may leave the area undersupplied, resulting in delayed shipments and shipper dissatisfaction. Similarly, matching decisions made without considering dynamic pricing updates may prioritize lower-revenue assignments, reducing platform

profitability. Dispatching without awareness of ongoing pricing and matching strategies can also cause idle vehicles to relocate inefficiently. Therefore, integrating these interdependent processes is critical to maintaining system-wide balance, responsiveness, and revenue optimization in dynamic freight environments.

We aim to develop a unified decision-making framework for smart freight platforms that addresses the following research question: *How can pricing, matching, and dispatching decisions be jointly optimized in real time to improve system-wide efficiency, responsiveness, and revenue in dynamic freight transportation environments?*

This paper contributes by proposing a *Nested Hierarchical Reinforcement Learning* framework that jointly optimizes pricing, matching, and dispatching decisions across multiple spatial resolutions. Unlike prior works that address these components in isolation or partial integration, our model coordinates real-time decision-making using specialized agents, Uber's H3 geospatial hierarchy, and modular pre-filtering to ensure scalability and responsiveness in urban freight environments.

To support multi-resolution spatial reasoning, we adopt Uber's H3 geospatial indexing system. H3 divides the Earth's surface into uniform hexagonal cells across multiple resolution levels. In particular, resolution levels refer to H3's geospatial indexing system, where resolution level 6 covers approximately 36.13 km^2 areas, and resolution level 7 covers approximately 5.16 km^2 areas. In our framework, pricing and matching are conducted at H3 resolution 7, enabling fine-grained fare adaptation and localized shipper-vehicle pairing based on supply-demand conditions and service attributes. Dispatching operates at a coarser resolution 6, supporting scalable repositioning of idle vehicles across wider geographic zones.

The hierarchical structure of H3 allows seamless information aggregation and coordination across levels. Each level-6 hexagon contains exactly seven level-7 hexagons, enabling consistent decision flows between agents.

Key Contributions

- **Unified NHRL Framework:** We introduce a novel NHRL-based architecture that integrates pricing, matching, and dispatching into a unified, multi-level reinforcement learning framework. The nested structure enables coordination across different spatial and temporal scales while reflecting the natural dependencies between these components and maintaining computational traceability.
- **Dynamic Pricing Strategy:** A pricing agent adjusts vehicle rates in real-time based on localized supply-demand imbalances at H3 resolution level 7. This fine-grained

spatial resolution facilitates precise revenue maximization while maintaining service balance across different freight zones.

- **Revenue-Centric Matching:** The matching agent pairs shipments with appropriate vehicles by maximizing revenue under constraints such as quality of service, travel distance, vehicle capacity, and customer preferences at H3 resolution level 7. This approach ensures both platform profitability and stakeholder satisfaction.
- **Proactive Dispatching Mechanism:** A dispatching agent relocates idle vehicles to high-demand areas using spatiotemporal demand forecasting and H3-based geospatial zoning at resolution level 6. This coarser resolution balances computational efficiency with effective vehicle repositioning, enhancing utilization and system responsiveness.
- **Multi-Resolution Coordination:** By leveraging H3's multi-resolution spatial structure, we coordinate decision-making hierarchically across broader and finer geographic levels, a feature largely absent in prior smart freight platforms.
- **Comprehensive Evaluation:** We conduct extensive simulations across two major Canadian urban freight networks—Montréal and Toronto—to validate the scalability and robustness of the proposed NHRL framework. Through a series of controlled experiments, we compare three configurations: fixed-price matching (1-agent), pricing-matching (2-agent), and pricing-matching-dispatching (3-agent). We analyze cumulative rewards, successful matches, vehicle idle time, shipment waiting time, match to fictitious vehicle, and revenue. Additionally, we benchmark PPO against TRPO and DDPG, and perform a two-way ANOVA sensitivity analysis on penalty parameters to confirm stability. This multi-city, multi-metric evaluation demonstrates the generalizability and operational strength of our framework.

The proposed framework not only advances the state of integrated decision-making in logistics but also offers practical benefits for urban freight platforms seeking to enhance real-time efficiency. Unlike existing frameworks that independently optimize platform components or operate on a single spatial resolution, our model introduces a nested hierarchical coordination across multiple spatial levels with modular pre-filtering, enabling scalable decision-making in dynamic freight environments.

This paper is organized as follows: Section 2 reviews the related literature on reinforcement learning applications in freight logistics. Section 3 outlines the problem context and modeling assumptions. Section 4 introduces the reinforcement learning foundations and justifies the algorithmic choices. Section 5 details the proposed NHRL methodology, including agent

design and spatial coordination. Section 6 describes the case study setup and data simulation process. Section 7 presents the experimental results and performance evaluation. Finally, Section 8 concludes the paper and outlines directions for future research.

6.2 Literature Review

Reinforcement Learning methods typically fall into three primary categories: value-based, policy-based, and actor-critic approaches. Value-based methods (e.g., Q-learning [119], DQN) estimate the expected return of state-action pairs and derive deterministic policies by maximizing this estimate. These methods are effective in discrete action spaces but encounter challenges when extended to continuous domains such as dynamic pricing. Policy-based methods (e.g., REINFORCE [96]) directly optimize the policy using gradient ascent. Although better suited for continuous action spaces, they often suffer from high variance in gradient estimates. Actor-critic methods (e.g., PPO [42], DDPG [44]) combine the strengths of both: an actor proposes actions while a critic evaluates them, enabling stable and efficient learning even in complex, high-dimensional environments.

Recent developments include MARL and HRL; MARL enables simultaneous learning among multiple agents, making it suitable for decentralized logistics systems [120, 121], while HRL introduces a hierarchy of agents operating at different levels, improving scalability and coordination. Despite these advancements, most current logistics RL applications focus on isolated components—either pricing, matching, or dispatching— or dual integration rather than exploiting the full capabilities of integrated, multi-resolution hierarchical architectures.

Smart freight transportation systems have evolved through various methodological approaches aimed at improving efficiency, profitability, and operational responsiveness. This section examines the literature on three key operational components—pricing, matching, and dispatching—with particular focus on RL applications and traditional optimization methods.

6.2.1 Pricing

Pricing mechanisms in transportation platforms significantly influence demand patterns, revenue generation, and stakeholder equity. Both traditional optimization and RL-based approaches have been developed to address these challenges.

Early pricing models prioritized fairness and demand management through conventional optimization techniques. [122] proposed integrated assignment models that jointly optimize pricing and delivery routing in freight O2O platforms. [88] developed fairness-oriented linear assignment models for both ride-hailing and ride-pooling services. [123] introduced a two-stage

stochastic pricing framework that balances passenger welfare with supply efficiency. [124] applied machine learning models including random forests and K-means clustering for fare prediction and allocation in urban settings. [125] explored subscription pricing models combining fixed discounts with real-time adjustments to stabilize market dynamics. While these approaches provided reliable solutions, their limited adaptability to rapidly changing conditions prompted exploration of more responsive alternatives.

RL has enabled more adaptive and scalable pricing strategies. [126] employed cooperative DDPG algorithms to model heterogeneous stakeholder behaviors, simultaneously maximizing platform revenue and service quality. [120] developed a MARL framework that effectively balanced efficiency with income equity. [49] introduced a spatio-temporal pricing model using PPO, achieving significant profitability improvements through dynamic coordination across time and space. [127] combined DQN with K-Means clustering to create a dynamic region-division pricing strategy that adapts to real-time supply-demand fluctuations. [128] applied Deep State–Action–Reward–State–Action (DSARSA) networks to optimize long-term revenue and service metrics. These methods excel in continuous adaptation to non-stationary demand patterns and complex relationships between price, and demand elasticity.

6.2.2 Matching

Effective matching between shipments and vehicles is essential for balancing supply and demand while optimizing system-wide objectives. The literature shows a clear evolution from simple heuristics to sophisticated learning-based approaches.

Early resource-sharing platforms primarily relied on greedy algorithms for immediate matching, focusing on minimizing wait times for the next available request [87,88]. [129] introduced linear assignment models to ensure fairness in vehicle-traveler pairings. [130] developed the Single Request Batch Assignment with Travel Time (SRBAT) strategy to balance computational efficiency with response times in large networks. [131] proposed ElasticShare, a greedy algorithm-based solution using dummy orders and vehicles to simulate future market conditions.

With the rise of real-time data analysis, RL-based matching methods have demonstrated superior performance in dynamic environments. [118] implemented a Hierarchical RL framework with PPO to optimize shipment-vehicle pairings within hexagonal spatial grids. [126] utilized cooperative DDPG to model driver behavior heterogeneity, improving response rates in multi-agent settings. [121] developed a Goal-Reaching Collaboration (GRC) algorithm using MARL to incentivize drivers toward high-revenue regions. [89] introduced a sustainable freight-matching framework optimizing vehicle-shipper pairings while minimizing costs and

emissions. [107] also implemented a Deep Q-Learning framework with H3 geospatial indexing and pre-filtering mechanisms, significantly reducing travel distances and improving system responsiveness. These approaches account for future state value through temporal difference learning and adapt to spatiotemporal patterns through experience.

6.2.3 Dispatching

Dispatching strategies are crucial for minimizing service delays and optimizing fleet operations, particularly in dynamic urban environments. Both traditional optimization and RL methods have been applied to address these challenges.

Conventional dispatching models have focused on adapting to uncertain demand and optimizing operational costs. [132] employed dynamic programming to create adaptable models that optimize speed and cost in real-time. [133] developed a bi-objective model for battery electric truck dispatching, focusing on energy and labor cost optimization. [134] proposed a Multi-Graph Hierarchical Multi-Head Attention-DDPG algorithm for fragmented markets, facilitating third-party platform integration through cooperative strategies.

RL approaches to dispatching have leveraged spatiotemporal information for enhanced predictive capabilities. [118] utilized HRL agents for real-time vehicle dispatching based on predictive demand modeling. [121] introduced a MARL-based dispatching strategy evaluating city states and profitability. [120] proposed JDRCL, a MARL framework combining dispatching with driver repositioning using primal-dual iterative training to enhance equity and efficiency. Recent advancements have extended action spaces to neighboring cells or road network abstractions, enabling more granular control over vehicle positioning [100, 135].

6.2.4 Synthesis, Technical Limitations, and Research Gaps

Despite growing adoption of RL and data-driven approaches in dynamic freight and ride-sharing systems, several recurring limitations can be identified across the reviewed studies. Most works rely on single-layer architectures with either grid-based [11] or zone-based spatial modeling [66], which limits scalability and responsiveness to heterogeneous regional characteristics. These approaches often fail to balance computational efficiency with fine-grained regional decision-making.

Temporal dynamics are typically modeled over short decision horizons, making them insufficient for capturing long-term effects such as future demand waves, price surges, or supply imbalances [15]. Although a few studies incorporate batched decision-making or long-horizon estimators, such as offline value-based dispatch [15], continuous adaptive learning over ex-

tended time horizons remains largely underexplored.

Furthermore, while some efforts utilize distributed or multi-agent coordination [11, 59, 136], coordination overhead often scales poorly, especially under dense demand scenarios. Global communication requirements or greedy myopic heuristics restrict effectiveness in real-world systems. Additionally, only a few frameworks such as PassGoodPool [136] have explored joint pooling of passengers and goods. However, despite its effectiveness in multi-entity coordination, PassGoodPool does not adopt a nested hierarchical reinforcement learning structure. It primarily relies on static policy rules and does not support dynamic coordination across multi-resolution spatial layers or inter-agent abstraction, limiting its scalability and generalization in large-scale urban freight environments.

The DARM+DPRS approach [84] proposes a dual-layer framework for pricing-aware dispatching and routing. While it tackles joint optimization to some extent, its methodology is primarily based on rule-based or linear reward heuristics, rather than using RL agents operating across nested or hierarchical spatial resolutions. This restricts the framework's ability to adapt in dynamic environments or scale to multi-agent systems with interleaved decision scopes.

Passenger-goods co-transportation and transfer-based freight scenarios are seldom modeled jointly [136]. When considered, these models do not generalize well to evolving operational contexts, such as varying cargo constraints. Also, reward signals in most studies are static and domain-specific (e.g., idle time, delay penalties) [10, 66], which limits transferability to new geographies or policy constraints.

Crucially, hierarchical decision architectures that can decompose joint tasks across multiple spatial and temporal resolutions are lacking in the literature. This inhibits efficient coordination across central (global) and local (vehicle-level) objectives. Moreover, few models explicitly integrate upstream demand forecasting with downstream tactical actions such as repositioning or delay-aware dispatching [137].

These gaps motivate our proposed NHRL framework, which addresses coordination challenges across spatial granularity and temporal scope, supports scalable vehicle-level decision-making, and enables joint pricing, matching, and dispatching in real-time.

6.2.5 Justification for the Proposed Framework

Our selection of PPO is justified by its empirical robustness and training stability. PPO outperforms DDPG and TRPO in our experiments by constraining the policy update magnitude with a clipped objective, thereby preventing performance collapse in dynamic freight envi-

ronments [42]. Its balance of stability, simplicity, and sample efficiency makes it well-suited for multi-agent architectures.

The proposed NHRL framework addresses these technical gaps through architectural innovations. First, by assigning specialized agents to pricing, matching, and dispatching, the framework reduces state-action complexity per agent and accelerates convergence. Second, using H3-based spatial hierarchy, the model enables fine-grained control (resolution 7) for local decisions and coarse-grained control (resolution 6) for broader anticipatory actions. This multi-resolution abstraction supports structured spatial decomposition, policy generalizability, and scalability.

The NHRL framework enables agents to communicate implicitly via a shared coordination hub, ensuring coherent inter-agent dynamics. Our approach also integrates modular prefiltering (PAMA, DEZE) to reduce the effective action space without compromising flexibility, facilitating training and real-time feasibility.

In summary, our proposed NHRL framework builds upon and extends the current literature by jointly optimizing three interdependent tasks across multiple spatial resolutions using stable PPO-based agents, overcoming key limitations in action space design, reward shaping, agent coordination, and spatial generalization. The following Problem Context and Modeling Assumptions section outlines the operational setting and foundational assumptions that guide the design and integration of the proposed NHRL framework.

6.3 Problem Context and Modeling Assumptions

Building upon the limitations of existing approaches identified in the literature review, this section formalizes the joint optimization problem addressed by our NHRL framework. While prior research often treats pricing, matching, and dispatching as distinct optimization problems, we argue that their integration is essential for overcoming the inefficiencies observed in real-world smart urban freight platforms. Our NHRL framework is designed to jointly optimize these interdependent processes, enhancing platform responsiveness, vehicle utilization, and revenue performance in dynamic urban freight environments.

6.3.1 Problem Description

Modern urban freight platforms must operate under highly dynamic and uncertain conditions, making real-time decision-making particularly challenging. A key difficulty lies in the unpredictability of demand—shipment requests can appear at any time, with varying locations, load sizes, quality requirements, and time constraints. Simultaneously, vehicle

availability fluctuates across time and space due to constraints such as driver working hours, vehicle capacity, and operational zones. This combination often results in spatial imbalances, where some regions suffer from surplus vehicle supply while others face shortages, leading to prolonged idle times and lost revenue opportunities.

Furthermore, the three decision processes—pricing, matching, and dispatching—are tightly coupled. A price adjustment directly influences the volume and spatial distribution of demand, which in turn affects the pool of feasible shipment-vehicle matches. These matches then determine how idle vehicles should be repositioned to maintain spatial balance and service continuity. Conventional optimization techniques, such as linear and heuristic approaches, often lack the flexibility and responsiveness required for real-time, high-frequency decisions in dynamic and spatially distributed freight networks.

This leads to several inefficiencies: pricing strategies are often static or rule-based, failing to reflect real-time demand elasticity; matching algorithms may be greedy or short-sighted, overlooking longer-term system impacts; and dispatching decisions are typically reactive, relying on post-hoc repositioning rather than anticipating future demand hotspots. As a result, platforms experience mismatches, lower match rates, increased operational costs, and dissatisfied stakeholders.

To illustrate the interdependence of these decisions, consider a scenario in which a regional distribution center experiences a sudden influx of shipment requests due to an unexpected promotion by a major e-commerce retailer. This localized demand spike in a suburban industrial zone requires an immediate increase in vehicle availability to avoid shipment delays. If the platform cannot adjust prices swiftly to attract vehicles toward the affected area, the dispatching agent will lack the necessary incentives to relocate idle vehicles there, resulting in spatial mismatch. Without sufficient vehicle presence, matching rates decline, shipment waiting time increases, and the platform incurs revenue losses. This cascading effect highlights the need for a joint optimization strategy that simultaneously adjusts prices, matches shipments with compatible vehicles, and dispatches vehicles in anticipation of shifting demand.

6.3.2 System Overview and Stakeholders

The smart freight transportation platform serves as an intermediary that coordinates the interactions among three key stakeholder groups: shippers, vehicles, and the platform operator itself. Shippers are entities that generate freight demand by submitting shipment requests. Each request typically includes details such as pickup and drop-off locations, cargo size, minimum quality requirements, and pickup time constraints. Vehicles, on the other hand, are assets with diverse attributes—such as capacity, service quality, and geographic availability

and working time constraints—responsible for fulfilling these shipment tasks. The platform acts as the decision-making engine, dynamically adjusting prices, assigning shipment requests to vehicles, and dispatching idle vehicles in response to forecasted supply-demand patterns.

The platform utilizes a hierarchical spatial indexing system to efficiently manage geographic decision-making at different resolutions. This approach helps balance precision and computational efficiency across the platform's operations. The specific implementation of this spatial indexing system is detailed in the Methodology section.

6.3.3 Key Decision Variables

Our framework models freight operations as a MDP with three interdependent decision processes. Each of the following decisions is governed by a dedicated RL agent:

- **Pricing Decision**: Dynamic prices assigned to vehicle types in specific geographic areas based on local supply-demand imbalances.
- **Matching Decision**: Binary decisions indicating whether specific shipment requests are paired with available vehicles.
- **Dispatching Decision**: Decisions to reposition idle vehicles to selected geographic areas to anticipate future demand.

The detailed mathematical formulations of these decision variables and their corresponding state spaces, action spaces, and reward functions are presented in the Methodology section.

6.3.4 Constraints and Assumptions

The system operates under the following constraints and domain-specific assumptions derived from real-world freight operations:

- **Capacity Constraint**: A vehicle can only be assigned to a shipment if it has sufficient cargo space to accommodate the shipment's load, ensuring that capacity limitations are respected during the matching process.
- **Service Time Constraint**: Vehicles must have sufficient available service time to pick up an assigned shipment cargo.
- **One-to-One Matching**: Each vehicle can be matched with at most one shipment per time step, and each shipment can be assigned to at most one vehicle.

- **Repositioning Limits**: Dispatched vehicles must be within the feasible movement range defined by their current location and remaining service time, reflecting physical movement capabilities.
- **Price Elasticity**: Demand in each geographic unit is modeled using a uniform cumulative distribution function, where the probability of acceptance linearly decreases with price increase.

6.3.5 Performance Metrics and Objectives

The NHRL framework jointly optimizes three interconnected objectives across the pricing, matching, and dispatching modules, addressing key performance metrics that directly impact platform viability and stakeholder satisfaction:

- **Revenue Maximization**: Enhance platform profit by dynamically adjusting prices and securing high-value shipment-vehicle matches. Performance is measured through total revenue and cumulative reward.
- **System Efficiency**: Increase the number of successful matches while reducing inefficiencies such as fictitious vehicle assignments (unmet demand), vehicle idle time per match, shipment waiting time per match, and mileage to pick up.

Each agent in the NHRL framework is guided by a specific reward function tailored to its operational role and designed to promote coordination with other agents. These reward functions are constructed to balance short-term objectives with long-term system performance and are described in detail in the Methodology section. In addition, we conduct a two-way ANOVA sensitivity analysis to evaluate the robustness of our framework to changes in policy parameters α and β, assessing their effect on key performance metrics. We perform a comparative analysis of learning algorithms—PPO, TRPO, and DDPG—evaluating their performance in term of overall reward outcomes.

6.4 Reinforcement Learning Method

This section introduces the reinforcement learning concepts and algorithms that form the foundation of our NHRL framework. We first present key RL principles, then explain our selection of the PPO algorithm, and finally connect these concepts to our freight logistics application.

6.4.1 Reinforcement Learning Fundamentals

Reinforcement learning addresses sequential decision-making problems where an agent learns optimal behavior through interaction with its environment. In the context of freight logistics, RL provides a natural framework for handling the dynamic, stochastic nature of pricing, matching, and dispatching decisions. The core components of an RL system include:

- **State space** ($\mathcal{S}$): The set of possible situations the agent may encounter.
- **Action space** ($\mathcal{A}$): The set of possible decisions the agent can make.
- **Policy** (π): A mapping from states to actions that determines the agent's behavior.
- **Reward function** (r): A signal indicating the immediate benefit of taking an action in a given state.

A policy π is formally defined as:

$$\pi_\theta(a|s) = \pi : S \times A \to [0, 1], \tag{6.1}$$

which represents a probability distribution over the state-action space. Given a state s, the policy returns the probability of taking action a, and samples an action according to this distribution.

The objective in RL is to find the optimal policy π^* that maximizes the discounted cumulative expected rewards J_π:

$$J_{\pi^*} = \max_\pi J_\pi = \max_\pi \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(t) \right], \tag{6.2}$$

where $\gamma \in (0, 1]$ is the discount factor. Determining how much the agent values future rewards compared to immediate ones. In our freight logistics context, this represents the long-term revenue and efficiency trade-offs that the platform must navigate.

Q-value function $Q_\pi(s, a)$ represents the expected return when taking action a in state s and following policy π thereafter:

$$Q_\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(t) \mid s(0) = s, a(0) = a \right]. \tag{6.3}$$

State value function $V_\pi(s)$ represents the expected return when starting in state s and following policy π::

$$V_\pi(s) = \mathbb{E}_{a(0),\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(t) \mid s(0) = s \right] \tag{6.4}$$

Advantage function $A_\pi(s, a)$ measures the relative benefit of taking action a in state s compared to the average performance in that state:

$$A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s) \tag{6.5}$$

The advantage function is particularly useful in our context as it helps each agent (pricing, matching, and dispatching) identify actions that perform better than the current policy's average behavior.

6.4.2 Reinforcement Learning Approaches

Reinforcement learning algorithms can be categorized into three main types, each with different characteristics relevant to our freight logistics application:

Critic-only methods (such as Q-learning [119] and SARSA [138]) maintain a value function and derive deterministic policies by selecting actions that maximize the estimated value:

$$a^* = \arg\max_a Q_\pi(s, a), \quad \pi(a^*|s) \leftarrow 1 \tag{6.6}$$

These methods work well for discrete action spaces but struggle with the continuous decision spaces in pricing problems.

Actor-only methods (or policy gradient methods) such as REINFORCE [96], directly optimize the policy parameters θ via gradient ascent:

$$\theta(t+1) = \theta(t) + \alpha \nabla_\theta \mathbb{E}_{\pi_{\theta(t)}} \left[\sum_\tau \gamma^\tau r(\tau) \right] \tag{6.7}$$

The advantage of these methods is their ability to handle continuous action spaces, which is essential for our pricing agent that must set prices from a continuous range.

Actor-critic methods combine both approaches by maintaining both a policy (actor) and a value function (critic):

$$\theta(t+1) = \theta(t) + \alpha \nabla_\theta \mathbb{E}_{\pi_{\theta(t)}} \left[Q_{\pi_{\theta(t)}}(s, a) \right] \tag{6.8}$$

Actor-critic methods can produce actions in continuous spaces while reducing the high vari-

ance typically associated with pure policy gradient methods, making them particularly suitable for the complex decision-making required in freight logistics.

6.4.3 Selection of Proximal Policy Optimization

In designing our NHRL framework, we adopt PPO as the primary RL algorithm due to its empirical robustness and practical advantages in complex, high-dimensional environments such as urban freight logistics. PPO is effective in balancing the stability and performance of policy updates. PPO introduces a clipped surrogate objective function that restricts the extent of policy change at each iteration. This constraint helps mitigate the risk of performance collapse due to overly aggressive updates and thereby contributes to training stability—an essential requirement when learning in stochastic and continuously evolving environments.

Furthermore, PPO exhibits strong sample efficiency by enabling multiple epochs of optimization over a fixed batch of data, which is especially beneficial in simulation-based freight settings where generating labeled data can be computationally expensive. Another advantage of PPO is its relative insensitivity to hyperparameter tuning compared to alternatives like TRPO or DDPG, making it more practical for real-world deployment. Collectively, these properties make PPO a highly suitable choice for supporting stable, efficient, and scalable learning in our nested hierarchical architecture.

While alternative RL algorithms such as DDPG, and TRPO have been applied in various logistics applications, they present significant limitations for our use case. DDPG, while capable of handling continuous action spaces, is known for its sensitivity to hyperparameters and training instability. TRPO provides stability via constrained updates but is computationally intensive. These limitations make them less practical for the joint, multi-resolution, and high-dimensional structure of our freight logistics environment. In contrast, PPO strikes an effective balance between performance, stability, and sample efficiency, making it the most suitable choice for our nested hierarchical architecture.

6.4.4 Application to Freight Logistics

The application of RL principles within our NHRL framework addresses several critical limitations observed in conventional freight optimization approaches. Central to our design is the use of a hierarchical agent structure, which reflects the natural decomposition of operational tasks in urban freight logistics. By assigning distinct agents to pricing, matching, and dispatching decisions—each operating at different temporal and spatial granularities—the

framework allows for more focused learning and improved convergence rates. This hierarchical decomposition not only reduces the dimensionality of each agent's action and state space but also enhances coordination by enabling the flow of information across decision layers.

Additionally, the nested structure allows for inter-agent dependencies to be explicitly modeled. For instance, pricing decisions at the fine-grained level directly influence the availability and distribution of demand, which in turn affects the matching process. Similarly, the outcomes of the matching process inform the dispatching agent's decisions about where to pre-position idle vehicles in anticipation of future demand. By capturing these interdependencies, the NHRL framework enables integrated, context-aware decision-making that adapts to real-time conditions and maximizes system-wide objectives such as revenue, efficiency, and service equity.

6.5 Methodology: NHRL Framework

This section formalizes our approach to the joint optimization problem described in the previous section. We present a comprehensive NHRL framework that integrates pricing, matching, and dispatching for freight transportation. We begin with an overview of the framework architecture, followed by a detailed explanation of the H3 spatial partitioning system. We then describe each agent's design, including state spaces, action spaces, and reward functions, providing the mathematical formulations that were introduced conceptually earlier. Finally, we explain the coordination mechanisms that enable effective integration of these components.

6.5.1 Framework Overview

Building on the challenges identified in the Problem Context section, we propose a NHRL framework to jointly optimize pricing, matching, and dispatching decisions in dynamic urban freight transportation systems. The framework comprises three interdependent RL agents: pricing, matching, and dispatching. Each operates at a distinct spatial scale, yet they collectively interact with a centralized environment. This nested architecture addresses the limitations of traditional approaches that treat these processes in isolation.

The pricing agent dynamically adjusts vehicle prices across localized regions by analyzing real-time supply-demand elasticity. The matching agent pairs shipment requests with available vehicles in real time to optimize platform revenue, ensure service quality, and enhance vehicle utilization and revenue maximization. Meanwhile, the dispatching agent proactively relocates idle vehicles to zones with anticipated future extra demand, thereby improving

spatial resource availability and reducing delays.

This hierarchical decomposition not only reflects practical planning realities, but also enhances learning performance. By breaking down the complex joint optimization problem into specialized sub-tasks, the NHRL framework reduces the state-action space each agent must explore, allowing for more targeted policy learning. Coordination among agents is managed through a centralized Coordination Hub. It aggregates real-time environmental data—including shipment requests, vehicle attributes, and zone-level statistics—and ensures that decisions are executed in the correct sequence. The hub serves as a shared interface that facilitates the synchronization of agent actions and environmental updates.

The proposed framework addresses several core challenges in smart freight logistics. It effectively manages inter-agent dependencies, wherein, as detailed in Section 3 pricing decisions directly influence demand patterns that, in turn, affect matching feasibility and dispatching priorities. It accounts for spatiotemporal dynamics by adapting decisions to rapidly evolving shipment volumes, service constraints, and vehicle availability across time and space. Additionally, its scalable architecture, based on agent modularity and hierarchical design, enables seamless extension to larger and more complex urban freight networks.

6.5.2 H3 Spatial Indexing and Multi-Resolution Structure

To manage spatial decision-making across varying granularities, the NHRL framework utilizes the Hierarchical Hexagonal indexing system developed by Uber. H3 divides the Earth's surface into uniformly shaped hexagonal cells across multiple resolution levels, providing consistent spatial representation and enabling hierarchical decision-making through parent-child zone relationships.

The framework adopts a dual-resolution strategy to balance precision and computational efficiency. GHU7 is used for pricing decisions, offering fine-grained spatial granularity. This allows the pricing agent to respond sensitively to localized supply-demand fluctuations. In contrast, GHU6 is employed for dispatching, This coarser granularity reduces the action space, making dispatch decisions more scalable and tractable. Importantly, each GHU6 zone nests exactly seven GHU7 zones, facilitating seamless information aggregation between agents operating at different spatial levels.

Compared to conventional zoning methods such as K-Means clustering, H3 offers several advantages [118]. Its uniform hexagonal shape avoids edge distortion and ensures balanced spatial coverage. The hierarchical nesting supports multi-resolution learning, and its global spatial indexing simplifies integration and scalability. Through H3, the NHRL framework

achieves resolution-aware optimization—enabling localized pricing and broader vehicle repositioning—while maintaining structural coherence across agents.

6.5.3 System Architecture and Data Flow

The system architecture of the NHRL framework is depicted in Figure 6.1. At its core lies a coordination hub that manages information exchange between agents and maintains a consistent representation of the environment state. The coordination hub serves as a central repository for all relevant data, including vehicle attributes (location, capacity, quality score, remaining service time), shipment requests (origin, destination, size, quality requirements, pickup time constraint), and pricing information for each GHU7 and vehicle type. The interaction between system components follows a sequential flow as outlined in Algorithm 5. At each time step:

1. The coordination hub provides the pricing agent with current supply-demand information for each GHU7 and vehicle type.
2. The pricing agent adjusts prices based on local market conditions, and the coordination hub is updated with new pricing information.
3. The PAMA selects suitable candidate vehicles for each shipment request based on capacity, quality, proximity, and time constraints.
4. The matching agent assigns shipments to vehicles from the candidate pool, optimizing for revenue based on considerations.
5. After matching, the coordination hub updates vehicle and shipment status information.
6. The DEZE identifies eligible GHU6s for vehicle repositioning based on predicted demand and revenue potential.
7. The dispatching agent relocates idle vehicles to selected GHU6s, optimizing for future matching opportunities.
8. The coordination hub updates all relevant information for the next time step.

This sequential processing ensures that each agent has access to the most current information when making decisions. The temporal dependencies between agents are carefully managed to maintain system coherence while allowing each agent to focus on its specific optimization objective.

Algorithm 5 Nested Hierarchical Framework for Pricing, Matching, and Dispatching

Input: Shipper requests, Vehicle attributes
Output: Matched vehicles, Dispatched vehicles
procedure HIERARCHICAL FRAMEWORK

    for $t \in T$ **do**
        for $h \in H$ **do**
            for $k \in K$ **do**
                Fetch Number of Shippers' requests for Vehicle type k for GHU7 h at time step t
                Fetch Number of available Vehicle type k at time step t for GHU7 h
                Calculate Price for vehicle type k in GHU7 h
            end for
        end for
        Update the Coordination Hub information
        Fetch Shippers' requests at time step t, $\mathcal{R}$
        for $r \in \mathcal{R}$ **do**
            Create Matching pool Based on demand r features
            if The pool contains real vehicle **then**
                Perform Matching demand r to selected vehicle
            else
                Increment r
            end if
        end for
        Update the environment and Coordination Hub information
        Fetch unassigned vehicles at time step t, $\acute{\mathcal{V}}$
        for $\acute{v} \in \acute{\mathcal{V}}$ **do**
            Obtain eligible H3 GHU6s to dispatch vehicle $\acute{v}$
            if is(eligible H3) **then**
                Perform Dispatching the vehicle $\acute{v}$ to the selected nearest eligible GHU6s
            else
                Increment $\acute{v}$
            end if
        end for
    end for
end procedure

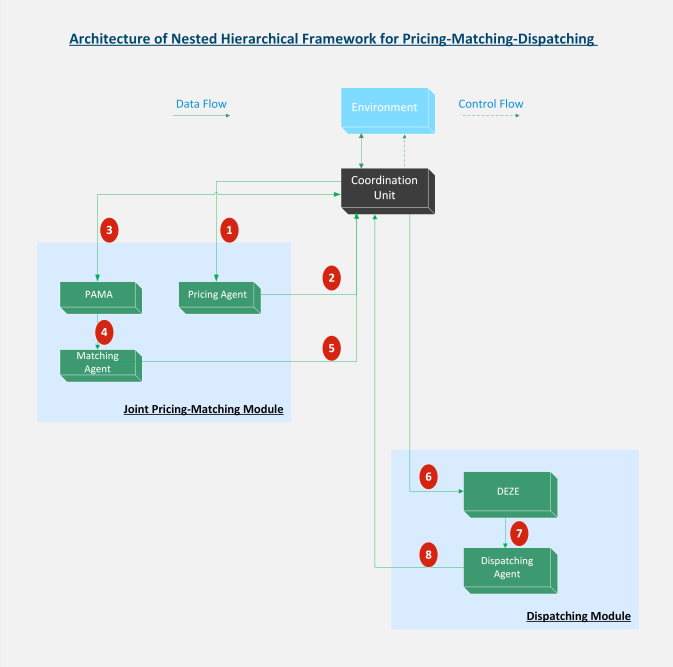

Figure 6.1 Architecture of Nested Hierarchical Framework and Component Interaction For Pricing-Matching-Dispatching

6.5.4 Pricing Agent

The pricing agent operates at the resolution 7 and is responsible for dynamically adjusting vehicle rates based on local supply-demand conditions. The pricing process is formulated as a MDP with the following components:

State Space

The state space for the pricing agent in each GHU7 h per vehicle type k is represented by a five-dimensional vector:

$$s_{hk}^{F} = (H_h, W_{hk}, Z_{hk}, V_k, P_k) \tag{6.9}$$

Here, H_h denotes the unique identifier for GHU7 h, W_{hk} represents the current number of demand requests in GHU7 h for vehicle type k, and Z_{hk} indicates the number of available vehicles of type k within the same GHU7. Additionally, the variable V_k denotes the vehicle type identifier for category k, allowing for customized pricing strategies aligned with the unique characteristics of each vehicle category. Lastly, P_k denotes the baseline price associated with vehicle type k.

This state s_{hk}^F representation captures the essential information needed to make informed pricing decisions, including the current market conditions (supply-demand ratio) and baseline pricing

Action Space

The action space for the pricing agent consists of continuous price adjustments relative to the maximum allowable price $P_k^{\max}$ for each vehicle type. For each GHU7 h and vehicle type k, the agent outputs parameters defining a truncated normal distribution:

$$a_{hk}^F = (\mu_{hk}, \sigma_{hk}) \tag{6.10}$$

where μ_{hk} represents the mean and σ_{hk} the variance of the distribution. The final price is determined by sampling from this distribution, constrained to the interval $[0, 1]$, and multiplying by $P_k^{\max}$:

$$P_{hk} = P_k^{\max} \cdot \text{sample}\left(\text{TruncatedNormal}(\mu_{hk}, \sigma_{hk}, 0, 1)\right) \tag{6.11}$$

This approach allows the agent to explore the price space efficiently while maintaining reasonable bounds on price adjustments.

Reward Function

The reward function for the pricing agent is designed to maximize expected revenue across all GHU7s and vehicle types. The reward $F(P_{hk})$ is formulated as:

$$F(P_{hk}) = \sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{K}} P_{hk} M_{hk} \tag{6.12}$$

where M_{hk} represents the induced demand in GHU7 h for vehicle type k, modeled as a function of price using a cumulative distribution function:

$$M_{hk} = \lambda_{hk}(1 - \text{CDF}(P_{hk})) \tag{6.13}$$

Here, λ_{hk} represents the baseline demand rate in GHU7 h for vehicle type k. This formulation captures the elasticity of demand with respect to price, with higher prices generally reducing demand but potentially increasing revenue up to an optimal point.

To ensure operational feasibility, we impose a supply constraint such that the induced demand

M_{hk} does not exceed the number of available vehicles Z_{hk} in each GHU7. This is formalized as:

$$M_{hk} \leq Z_{hk} \tag{6.14}$$

This constraint ensures that the pricing agent does not induce more demand than can be feasibly served, thereby integrating real-time supply-side dynamics into the decision-making process. By bounding demand with local supply, the agent avoids unrealistic over-promising in high-demand, low-supply scenarios.

This reward structure not only maximizes localized revenue but also promotes coordination across agents. By modulating demand within the limits of vehicle availability, the pricing agent ensures the matching agent receives a balanced and serviceable set of requests, thereby improving match feasibility and platform efficiency. Additionally, elevated prices in undersupplied zones serve as implicit signals for the dispatching agent, highlighting potential vehicle shortages. This enables proactive repositioning strategies, enhancing systemic responsiveness and spatial equilibrium across the platform.

6.5.5 Matching Agent

The matching agent operates at the resolution 7 and is responsible for pairing shipments with suitable vehicles to maximize revenue while satisfying service constraints. Before matching decisions are made, a pre-filtering algorithm PAMA reduces the potential matching space to improve computational efficiency.

Pre-filtering Algorithm for Matching Agent

PAMA plays a critical role in reducing computational complexity by narrowing the pool of candidate vehicles for each shipment request. Instead of evaluating all available vehicles, PAMA selects a subset of the top N candidates based on four key criteria: *capacity compatibility*, ensuring the vehicle can accommodate the shipment load; *quality alignment*, requiring the vehicle's service rating to meet or exceed the shipment's minimum standard; *proximity*, prioritizing vehicles closer to the pickup location; and *availability*, filtering for vehicles that are idle or will become available within the shipment's required time window.

The algorithm evaluates all vehicles and then selects the top N vehicles as the candidate pool, which is passed to the matching agent for final assignment decisions. This targeted approach significantly reduces the dimensionality of the action space, allowing the matching agent to learn more efficiently and make faster, higher-quality pairing decisions.

As outlined in Algorithm 6, PAMA follows a sequential workflow:

1. *Initial Search within the Region:* The algorithm first evaluates each shipment's GHU7 tag and considers the capacity and quality score required for the request. It searches for available vehicles within this initial GHU7 that match the criteria. If more than N candidates are found, N vehicles are randomly selected. If exactly N vehicles are available, all are directly selected. If fewer than N vehicles are found, all are added to the pool, and the search expands to surrounding rings.

2. *Expansion to Surrounding Rings:* The search extends to the two adjacent hexagonal rings (as illustrated in Figure 6.2) to find additional vehicles meeting the service time, capacity, and quality requirements. If the needed number of candidates is found, they are randomly selected or directly added. Otherwise, if candidates are still insufficient, the algorithm adjusts its criteria based on urgency.

3. *Urgency Handling — Immediate Pickup:* If the shipment requires immediate pickup, the algorithm relaxes the quality requirement by one unit and searches again within the initial GHU and surrounding rings. Vehicles found are either randomly selected or directly added depending on the number retrieved. If still insufficient, the algorithm next adjusts capacity.

4. *Further Relaxation — Capacity Adjustment:* When necessary, the algorithm searches for vehicles with one unit higher capacity. If enough are found, they are selected; otherwise, fictitious candidates are introduced to fill any remaining gaps, ensuring the pool size reaches N.

5. *Time Tolerance Handling:* If the shipper can tolerate additional pickup time, the algorithm re-expands its search, first prioritizing vehicles with exact requirements. If the search remains insufficient, it relaxes the quality score downward by one unit, followed by an upward adjustment of capacity.

6. *Fallback and Fictitious Candidates:* At any stage where the required number of candidates cannot be fulfilled with real vehicles after all relaxations, fictitious vehicles are created to complete the pool. If all candidates are fictitious, the pool is invalidated; otherwise, it is forwarded to the matching agent for further processing.

This structured approach ensures that the pool creation mechanism balances the trade-off between shipment requirements and vehicle availability, while maintaining operational continuity through adaptive candidate relaxation strategies.

Algorithm 6 PAMA

```
1: Input: Shipment requirements, vehicle database, required size N
2: Output: Candidate pool
3: procedure CreatePool(Shipment, Vehicles, N)
4:     Search origin GHU for exact matches
5:     if Enough vehicles then
6:         Randomly select N candidates; return
7:     end if
8:     Add found vehicles to pool
9:     Expand search to surrounding rings
10:    if Pool reaches N then
11:        Finalize pool; return
12:    end if
13:    if Immediate pickup required then
14:        Relax quality (one level lower); search
15:        if Pool reaches N then
16:            Finalize pool; return
17:        end if
18:        Relax capacity (one level higher); search
19:        if Pool reaches N then
20:            Finalize pool; return
21:        end if
22:        Add fictitious candidates; return
23:    else
24:        if Waiting is tolerable then
25:            Relax quality; search
26:            if Pool reaches N then
27:                Finalize pool; return
28:            end if
29:            Relax capacity; search
30:            if Pool reaches N then
31:                Finalize pool; return
32:            end if
33:            Add fictitious candidates; return
34:        end if
35:    end if
36: end procedure
```

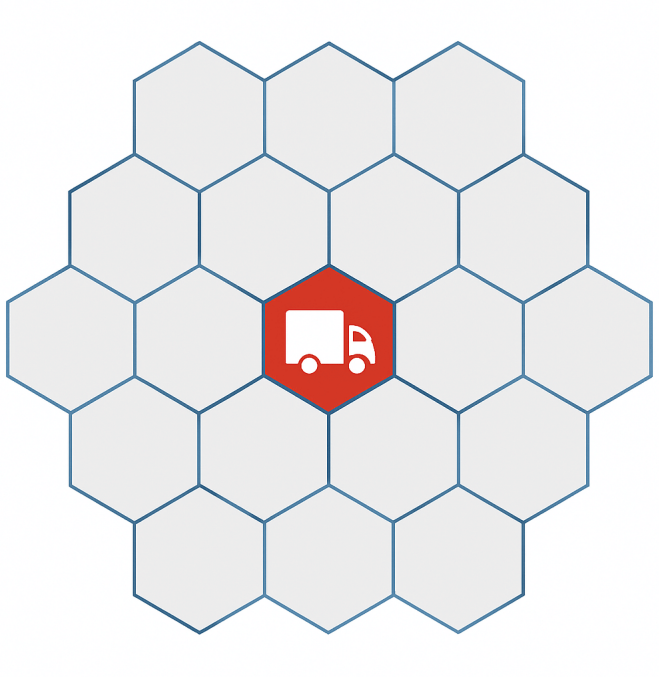

Figure 6.2 Origin hexagon along with its first and second ring of neighboring hexagons

State Space

The state space for the matching agent consists of information about both the shipment request and the pool of candidate vehicles. For each shipment request r, the state is represented as:

$$s_r^G = (O_r, L_r) \tag{6.15}$$

where O_r includes shipment attributes such as pickup and drop-off locations, cargo size, quality requirements, and pricing information. L_r represents the pool of candidate vehicles, including their current locations, capacities, quality scores, waiting times, and cost parameters.

This comprehensive state representation enables the matching agent to evaluate the potential benefit of each possible match while considering both immediate and longer-term implications.

Action Space

The action space for the matching agent consists of discrete choices among the N candidate vehicles identified by PAMA. For each shipment request r, the agent selects one action from:

$$a_r^G \in \{1, 2, \ldots, N\} \tag{6.16}$$

where actions 1 through N correspond to selecting one of the candidate vehicles.

Reward Function

The reward function for the matching agent is designed to optimize the matching process by considering several key factors. The primary objective is to maximize overall revenue while ensuring optimal quality and capacity matches and minimizing empty mileage, shipment request waiting time, and vehicle idle time. The reward function, denoted as $G(X_{rv})$, is formulated as follows:

$$G(X_{rv}) = \sum_{r\in\mathcal{R}} \sum_{v\in\mathcal{V}} \left[\left(d_r^t P_{rv}^h (1 - \alpha I(q_v < q_r) - \beta I(c_v > c_r) \right) - (T_r + T_v)\, C_{rv}^w - C_v^e d_{rv}^l - f_v \right] X_{rv} - U\,(1 - X_{rv}) \tag{6.17}$$

In this equation, $G(X_{rv})$ represents the overall reward function for the matching agent, capturing the total reward derived from the matching process. The sets $\mathcal{R}$ and $\mathcal{V}$ denote the shipment requests and available vehicles, respectively, where each shipment request r initiates a transportation task, and each vehicle v can be selected to serve it. The variable d_r^t indicates the distance to reach the drop-off location of the request r, while P_{rv}^h denotes the price charged per unit distance from the origin GHU7 h. The penalty rates α and β are applied when the vehicle's quality q_v is lower than the required quality q_r and when the vehicle's capacity c_v exceeds the required capacity c_r, respectively. The indicator functions $I(q_v < q_r)$ and $I(c_v > c_r)$ evaluate these conditions, returning 1 or 0 accordingly.

The variables T_v and T_r represent the idle time and waiting times for the vehicle and shipment request, respectively, while C_{rv}^w captures the cost associated with these times. The cost of traveling per unit mileage to pick up cargo r for vehicle v is indicated by C_v^e, and d_{rv}^l denotes the empty mileage required for vehicle v to travel to the pickup location. A fixed cost associated with the transportation service for vehicle v is represented by f_v. The decision variable X_{rv} indicates whether a valid match is made between shipment request r and vehicle v (equal to 1 for a match, and 0 otherwise). Lastly, U signifies a penalty incurred when a request is unserved, thereby incentivizing the matching process to avoid pairing with fictitious vehicles. This design inherently addresses the concern of unrealistic assignments

involving fictitious vehicles, as such pairings are heavily penalized and avoided unless no real alternatives exist.

This comprehensive reward function ensures that the matching agent considers all relevant factors when making pairing decisions, promoting efficient and effective matches that balance immediate revenue with longer-term system performance.

The reward function $G(X_{rv})$ not only optimizes local match decisions but also supports inter-agent coordination in the NHRL framework. Specifically, the pricing agent's output (P^h_{rv}) directly influences the revenue term in the matching reward, ensuring that the matching agent aligns with dynamic pricing policies when selecting vehicles. This encourages the selection of high-revenue matches that reflect real-time market conditions.

6.5.6 Dispatching Agent

The dispatching agent operates at the resolution 6 and is responsible for repositioning idle vehicles to anticipate future extra demand. Like the matching agent, the dispatching process is preceded by a pre-filtering DEZE that identifies promising relocation targets.

Dispatch Eligibility Zone Evaluator

The pre-filtering algorithm for dispatching identifies high-potential GHU6s for vehicle repositioning based on predicted demand patterns. For each idle vehicle $\hat{v}$, DEZE operates as follows:

1. *Initial Evaluation of Vehicle Regions:* The algorithm begins by evaluating the H3 spatial tag of each idle vehicle at a resolution level 6, selected to balance spatial detail with computational efficiency. Each vehicle's capacity, quality score, and remaining service time are taken into account to determine its suitability for relocation to high-demand areas.

2. *Search Expansion to Surrounding Hexagonal Rings:* To identify viable relocation options, the algorithm examines adjacent hexagonal rings within each vehicle's reachable range, determined by its remaining service time. For this study, we assume that vehicles can traverse up to one GHU6 per each time step. By expanding the search area to multiple rings, the algorithm increases the probability of locating high-demand zones within reach of the vehicles.

3. *Combined Demand-Supply Analysis and Feature Adjustment:* The algorithm integrates spatiotemporal demand-supply forecasting with quality and capacity alignment. It

initially seeks GHU6s with exact quality and capacity matches and progressively relaxes the criteria to include zones where the required quality is one level higher or the capacity requirement is one level lower than the vehicle's specifications. This flexible search design allows vehicles to serve near-optimal demand locations when exact matches are unavailable, enhancing relocation accuracy and operational efficiency.

4. *GHU6 Scoring and Selection:* Each GHU6 is then assigned a score based on anticipated demand levels; GHU6s with surplus demand receive a score of 1, GHU6s without sufficient demand are assigned a score of -1, and GHU6s outside the selected area also receive a score of -1(as example, illustrated in Figure 6.3). Only GHU6s with adequate demand scores are prioritized for relocation, optimizing match rates while reducing unnecessary dispatches.

By scoring and prioritizing GHU6s, DEZE enables the dispatching agent to make more targeted, efficient repositioning decisions, ultimately improving overall system responsiveness and vehicle utilization.

State Space

The state space for the dispatching agent includes information about both the idle vehicle and potential destination GHU6s. For each idle vehicle $\hat{v}$, the state is represented as:

$$s_{\hat{v}}^{B} = (E_{\hat{v}}, REV_{\hat{v}}, (latH_{\acute{h}}, lonH_{\acute{h}}), (latV_{\hat{v}}, lonV_{\hat{v}})) \tag{6.18}$$

where:

- *Hexagon Eligibility Vector ($E_{\hat{v}}$)*: DEZE generates a vector that indicates the eligibility of each GHU6. Each entry in the vector determines if a GHU6 is fit for vehicle dispatch, considering the expected future demand.
- *GHU6 Revenue Vector ($REV_{\hat{v}}$)*: Represents the expected revenue in each GHU6, allowing the agent to prioritize high-revenue areas within feasible distances.
- *GHU6 Central Points Vector ($latH_{\acute{h}}, lonH_{\acute{h}}$)*: Provides the latitude and longitude of the central points for each GHU6. These coordinates provide spatial references necessary for calculating distances and determining feasible dispatch routes.
- *Vehicle Positions Vector ($latV_{\hat{v}}, lonV_{\hat{v}}$)*: Specifies the current geographic location of each unassigned vehicle. These coordinates are crucial for evaluating the remaining service time and determining which GHUs each vehicle can reach.

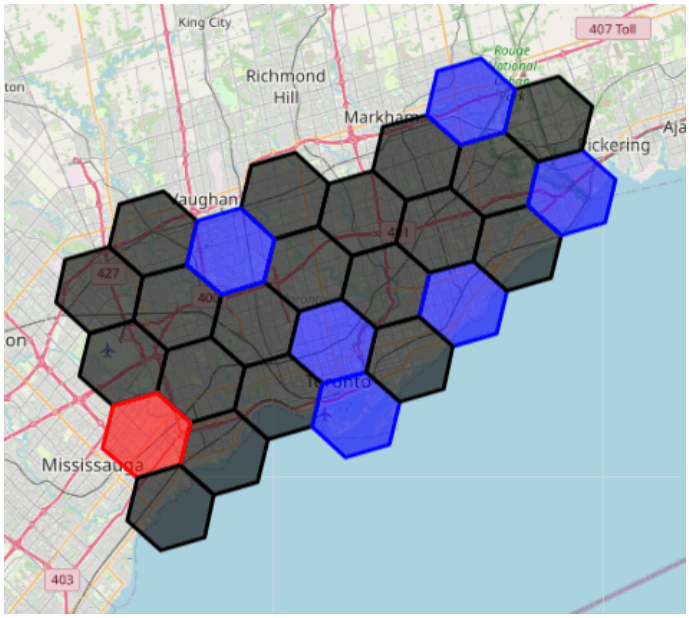

Figure 6.3 Origin Hexagon(red) and Eligible(blue) and Non-eligible Zones(black)

This state representation enables the dispatching agent to assess the spatial and economic trade-offs involved in repositioning vehicles, balancing relocation costs against anticipated future revenue.

Action Space

The action space for the dispatching agent consists of discrete choices among the eligible GHU6s identified by DEZE. For each idle vehicle $\hat{v}$, the agent selects one action from:

$$a_{\hat{v}}^{B} \in \{1, 2, \ldots, M\} \tag{6.19}$$

where actions 1 through M correspond to repositioning the vehicle to one of the M GHU6s.

Reward Function

The reward function evaluates the expected revenue from dispatching a vehicle to a selected demand-dense GHU6, while also accounting for vehicle movement costs. To ensure rational dispatching, the function penalizes excessive travel distances and rewards efficient repositioning to high-revenue zones. This trade-off between relocation cost and future revenue encourages the agent to prioritize nearby, high-revenue GHU6s rather than distant, less efficient options. The reward function is mathematically defined as:

$$B(Y_{\acute{v}\acute{h}}) = -\sum_{\acute{v}\in\acute{V}}\sum_{\acute{h}\in\acute{H}}\left(g_{\acute{v}\acute{h}} - d^{\acute{l}}_{\acute{v}\acute{h}}C^{\acute{l}}_{\acute{v}\acute{h}}\right)Y_{\acute{v}\acute{h}} + \acute{U}\left(1 - Y_{\acute{v}\acute{h}}\right) \tag{6.20}$$

In the equation (6.20), for each idle vehicle $\acute{v}$ and GHU6 $\acute{h}$, let $g_{\acute{v}\acute{h}}$ represent the expected revenue, $d^{\acute{l}}_{\acute{v}\acute{h}}$ denote the distance from $\acute{v}$ to the center of $\acute{h}$, and $C^{\acute{l}}_{\acute{v}\acute{h}}$ signify the cost per kilometer. The binary decision variable $Y_{\acute{v}\acute{h}}$ equals 1 if $\acute{v}$ is dispatched to $\acute{h}$ and 0 otherwise. A loss penalty $\acute{U}$ is imposed for unfeasible assignments, discouraging non-eligible dispatch choices.

This reward structure encourages the dispatching agent to balance immediate relocation costs against potential future revenue, promoting more efficient utilization of idle vehicles across the freight network.

By aligning relocation incentives with expected revenue and feasibility constraints, the dispatching agent's reward function complements the objectives of the pricing and matching agents. High-revenue zones, shaped by the pricing agent's elastic pricing, naturally attract more vehicle repositioning. In this way, the dispatching agent reinforces spatiotemporal coordination across all layers of decision-making, contributing to a stable and adaptive freight logistics ecosystem. Rather than communicating directly, each agent adapts its behavior based on shared environment, thereby achieving decentralized coordination through observed outcomes.

6.5.7 Modularity and Generalizability of PAMA and DEZE

The proposed pre-filtering algorithms—PAMA and DEZE—are designed with a modular and generalizable architecture to ensure broad applicability across diverse RL frameworks and freight logistics environments.

These modules function as independent spatial preprocessing layers, operating prior to decision-making and significantly reducing the size of the action space each agent must explore. This modularity enables seamless integration of PAMA and DEZE with various RL algorithms—including DQNs, Policy Gradient methods, and MARL—without requiring structural modifications to the learning backbone.

By decoupling computationally intensive spatial filtering from policy optimization, the algorithms enhance both scalability and training efficiency, particularly in large-scale urban freight systems. PAMA dynamically narrows the candidate vehicle pool for shipment requests based on real-time availability, proximity, and service compatibility. Similarly, DEZE identifies feasible and high-revenue dispatching zones using hierarchical spatial analysis com-

bined with flexible attribute thresholds.

Importantly, this design also improves the generalizability of the overall framework. The pre-filtering logic can be adapted for a variety of logistics scenarios—such as ride-hailing, public transportation, and delivery routing—where real-time resource assignment and repositioning are critical. Furthermore, the algorithms are resolution-agnostic, capable of operating under different levels of geographic granularity depending on the specific application needs.

In summary, the modular architecture of PAMA and DEZE enhances the versatility, scalability, and adaptability of the NHRL framework, positioning it as a robust and extensible solution for real-time decision-making in dynamic transportation and logistics ecosystems.

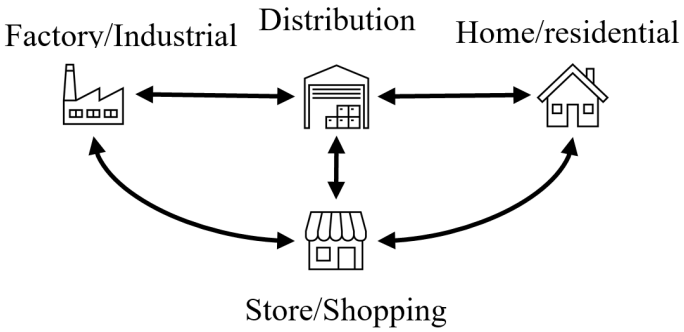

Figure 6.4 Trip-based structure of urban freight components in simulation model

6.6 Case study

To evaluate the effectiveness of our NHRL framework, we constructed a large-scale simulation environment reflecting realistic urban freight operations in two Canadian cities: Montréal and Toronto. Due to the lack of open-access operational data from freight platforms, the datasets for OD requests and trucking fleets were synthetically generated based on domain-specific distributions, geographic density data, and transportation modeling best practices. This section details our data generation process and assumptions to ensure alignment with realistic operational scenarios.

The simulation for the nested pricing, matching, and dispatching model relies on two main datasets: OD data and trucking data. The OD data contains attributes of shipping requests, such as origin and destination locations and commodity types, while the trucking data describes vehicle features, including capacity and driver availability.

6.6.1 OD Data Construction

The OD data construction follows three primary steps:

- **Identification of Shipping Request Factors**: Factors influencing the generation of shipping requests include population density, the presence of industrial areas, shopping centers, and distribution centers. These elements reflect areas with high economic activity and demand for freight services.
- **Geolocation Approximation**: Approximate locations for shipping origins and destinations were established by analyzing spatial data for population, industrial zones, and shopping centers in a city. Using QGIS, density layers were created for these features.
- **Data Structuring**: The structure and characterization of OD data are developed to represent the spatial distribution of requests effectively.

Urban freight demand models serve as the foundation, focusing on key urban freight components: factories, distribution centers, shopping centers, and end consumers—as illustrated in Figure 6.4, based on [108] and [109]. In a city's urban freight network, industrial areas act as hubs for transporting goods to distributions, shopping centers, and residential areas. The density of these components correlates with trip generation likelihood, allowing estimation of shipping request probabilities across different zones.

To achieve accurate spatial representation, the city's population, industrial areas, and shopping centers were mapped and analyzed using QGIS. The density layers for these components were combined into a single spatial representation, where each zone's density value determines the probability of trip generation. This probability directly affects the number of origins and destinations assigned to a zone. High-density areas, such as downtown, generate more requests due to concentrated economic and residential activity, while low-density suburban areas generate fewer requests. The OD dataset links origin and destination points within these zones, with higher-density areas hosting a greater number of trips.

6.6.2 Additional OD Data Attributes

To enhance the OD data, additional characteristics were incorporated—such as request dates, shipment size and weight, and commodity types—based on forms and reports from the Transportation Association of Canada (TAC) and Transport Canada, and informed by the work of [110].

6.6.3 Trucking Data Simulation

The initial spatial distribution of trucks is modeled to match the demand density, ensuring a realistic allocation of vehicles. Zones with higher concentrations of shipping requests, such as industrial and commercial hubs, are assigned a proportionally higher number of trucks. Each simulated truck is characterized by its type and cargo, reflecting the diverse logistical dynamics of the city. By aligning truck availability with demand density, the simulation achieves a realistic representation of urban logistics and operational conditions.

Although operational data was synthetically generated, it was calibrated against real-world statistics. These include urban freight activity levels, vehicle fleet compositions, and structures based on Transport Canada reports and TAC guidelines, ensuring realistic operational assumptions.

The geographical density of shipment origins, destinations, and available vehicles for both Montréal and Toronto is visualized in Figures 6.5 and 6.6, using the H3 partitioning system at resolution levels 6 and 7. Each figure presents the spatial distribution of freight activity across urban zones, illustrating variations in demand and supply. The density layers reflect origin and destination request frequencies as well as vehicle availability, thereby highlighting regions with high shipping activity and logistic capacity.

6.7 Results and Analysis

6.7.1 Experimental Setup

To validate the effectiveness of the proposed NHRL framework, we conducted comprehensive simulation experiments using two datasets representing urban freight operations in two Canadian cities: Montréal and Toronto. The experimental dataset includes a total of 9,866 shipments and 11,397 vehicles in Montréal, and 17,351 shipments and 20,000 vehicles in Toronto. These represent realistic operational scales for urban freight systems and provide a rigorous basis for evaluating the scalability and performance of the proposed NHRL framework.

To facilitate a clear and incremental performance assessment, we adopted a tiered evaluation approach consistent with recent literature [84, 136]. The simulations evaluated three configurations: (1) a baseline Fixed-Price Matching model, (2) a Joint Pricing-Matching model (two-agent setup), and (3) the full Pricing-Matching-Dispatching NHRL model (three-agent setup).

The fixed-price matching model (1-agent) serves as a foundational benchmark, reflecting

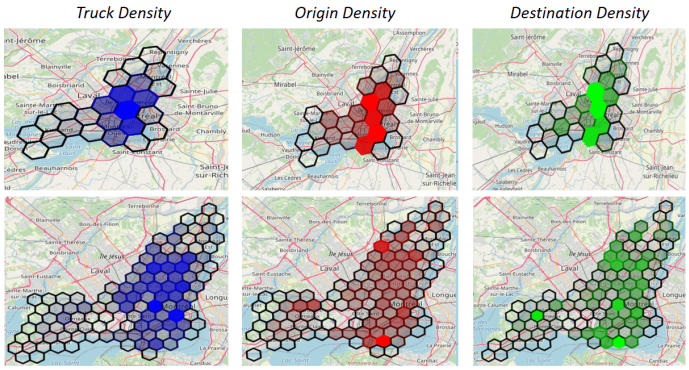

Figure 6.5 Geographical density of shipment origins, destinations, and available vehicles in Montréal using H3 partitioning at resolution level 6 (top) and resolution level 7 (bottom).

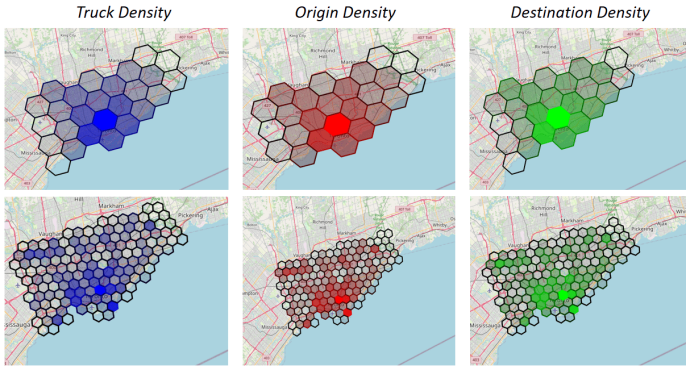

Figure 6.6 Geographical density of shipment origins, destinations, and available vehicles in Toronto using H3 partitioning at resolution level 6 (top) and resolution level 7 (bottom).

traditional freight platforms with static pricing and myopic assignment logic—comparable to early configurations in existing RL-based studies. The joint pricing-matching model (2-agent) incorporates dynamic pricing via RL while omitting dispatching, mirroring mid-level integrations where pricing and assignment are optimized jointly without vehicle repositioning. Finally, our full NHRL model (3-agent) introduces a dispatching agent for proactive vehicle relocation using H3-based forecasts, enabling coordinated multi-resolution decision-making. This stepwise setup allows for isolating the incremental contributions of each decision layer and reflects a realistic, modular deployment strategy for smart freight platforms.

All simulations were executed on a workstation with an Intel Core i7 processor, 64GB RAM, and an NVIDIA RTX 3060 Ti GPU with 8GB VRAM. The actor-critic networks were implemented using PyTorch in Python.

6.7.2 Model Configuration

Network Architecture and Hyperparameters

The hyperparameters for our actor-critic network were carefully selected based on empirical testing in PPO-based reinforcement learning for transportation systems. The neural network architecture consists of two hidden layers with 128 neurons each and ReLU activation functions, which offer sufficient representational capacity while mitigating vanishing gradient issues common in deeper networks.

We used a PPO clipping parameter of $\epsilon = 0.2$ to ensure stable policy updates by preventing excessive divergence from the previous policy, thus maintaining training robustness. A discount factor of $\gamma = 0.99$ was employed to balance short-term and long-term rewards, which is particularly important in urban freight logistics where repositioning and pricing decisions may influence outcomes over extended horizons.

The learning rates were tuned separately for the actor and critic networks: 5×10^{-5} for the actor to reduce the risk of destabilizing the policy updates, and 1×10^{-4} for the critic to allow faster convergence of the value function. These values were selected after evaluating several configurations and identifying those that yielded the best convergence behavior and reward performance across multiple training runs. Policy updates were performed every 64 timesteps, a setting that provides a good trade-off between responsiveness to environmental dynamics and sample efficiency.

In the output layer, we employ the softmax activation function for the matching and dispatching agents, while the pricing agent uses a sigmoid activation function to constrain outputs within the range of zero to one. The output dimension layers are configured as follows: the pricing agent has a single neuron, the matching agent has neurons equal to the pool size (N = 10), and the dispatching agent has neurons corresponding to the number of involved GHU6s.

Operational Parameters

To enhance simulation realism, we employed the following operational parameters:

- **Vehicle and Cargo Types:** Categorized as *Light*, *Medium*, and *Heavy*.
- **Pricing Parameters:** Basic price per kilometer P_k was set as 1.5, 2.25, and 3 Canadian dollars for *Light*, *Medium*, and *Heavy* vehicle types, respectively, with $P_k^{\max}$ being 10 times P_k. The cost per time step C_{rv}^{w} was defined as $7.45 \times P_k$, based on our assumption

that each vehicle can traverse one GHU6 per time step. The empty travel cost C_v^e was $0.05 \times P_k$.

- **Penalty Parameters:** $\alpha = 0.05$ and $\beta = 0.05$ penalized mismatches in vehicle quality and capacity.
- **Penalty and Fixed Costs:** A penalty U of -1000 discouraged matches with fictitious vehicles, and fixed costs f_v were 30, 40, 50 Canadian dollars for *Light*, *Medium*, and *Heavy* vehicles, respectively. Additionally, a penalty $\acute{U}$ of -1000 was imposed to discourage infeasible assignments during dispatching.
- **Additional Costs:** The cost per distance unit for dispatching, $C^{\acute{l}}_{\acute{v}\acute{h}}$ was set as $0.05 \times P_k$.

6.7.3 Performance Analysis

Overall Performance Metrics

In this study, we use the "Matching with a Fixed Price" configuration as our baseline for assessing the effectiveness of the freight logistics model. We then evaluate the power and capabilities of two advanced configurations: (1) Joint Pricing and Matching(two-agent model) and (2) Joint Pricing, Matching, and Dispatching (three-agent model). Key performance metrics include sequential average on rewards, match with fictitious vehicle counts, vehicle idle time and shipment waiting time, average distance per match, revenue, and successful matches. The successful matches metric measures the number of successful pairings between requests and suitable real vehicles, considering factors like capacity, quality, and feasibility within operational constraints. These metrics evaluate the efficiency and effectiveness of each configuration.

Montréal Case Study

As illustrated in Figure 6.7, the comparison of metrics convergence across the baseline, two-agent, and three-agent models reveals the effectiveness of each configuration throughout the training. The figure shows how the metrics stabilize over time, highlighting the improvements made by the two-agent and three-agent models compared to the baseline. Table 6.1 summarizes these metrics across all three configurations.

Using the baseline as a reference, the two-agent model demonstrates a substantial leap in both reward and revenue. The reward in the two-agent model rises to 2,017,288, almost ten times higher than the baseline configuration, which is 203,881. Likewise, revenue increases to \$2,163,248 from a baseline revenue of \$308,375, underscoring the value of the joint matching

Table 6.1 Performance metrics for different configurations (Montréal).

Metric	**Fixed Price (1 Agent)**	**Joint Matching & Pricing (2 Agents)**	**Matching, Pricing & Dispatching (3 Agents)**
Reward	203,881	2,017,288	2,104,112
Revenue	308,375	2,163,248	2,227,110
Successful Matches	4,132	4,068	4,278
Match with Fictitious Vehicle	35	131	79
Vehicle Idle Time	0.55	0.17	0.18
Shipment Waiting Time	0.74	0.37	0.38
Mileage to Pickup (km)	7.46	7.46	7.48

and dynamic pricing approach. These improvements highlight the power of a model that not only matches vehicles with shipments but also optimizes pricing to capture greater revenue.

The three-agent model, which adds dispatching to the pricing and matching functionality, surpasses both previous configurations, achieving a reward of 2,104,112 and a revenue of $2,227,110. These figures reflect a 4.3% increase in reward and a 2.9% increase in revenue over the two-agent model. The substantial increase in both reward and revenue—especially in the three-agent configuration—highlights the added value of incorporating dispatching to align vehicles strategically with high-revenue demand hotspots.

The successful match indicator shows a significant improvement with the three-agent model. The two-agent model reaches 4,068 successful matches, only slightly below the baseline at 4,132. However, the three-agent model significantly enhances matching accuracy, achieving 4,278 successful matches—a 3.5% increase over the baseline and a 5.1% increase over the two-agent model. This slight decrease in the number of successful matches in the two-agent model (4,068 compared to 4,132 in the baseline) may initially appear counterintuitive. However, it is compensated by substantial improvements in both platform revenue and reward. This trade-off reflects the model's prioritization of higher-value, more profitable matches over merely maximizing the number of fulfilled requests.

The match with fictitious vehicle count reflects instances where matches are made to fictitious vehicles due to the unavailability of appropriate real vehicles. In the baseline configuration, this rate is relatively low at 35 matches. However, it increases significantly to 131 in the two-agent model, largely due to the complexity introduced by dynamic pricing. The pricing agent in the two-agent model dynamically adjusts vehicle rates based on local supply-demand imbalances, which can create pricing disparities across zones. Simultaneously, the matching agent prioritizes maximizing revenue by pairing cargo with the most profitable vehicles,

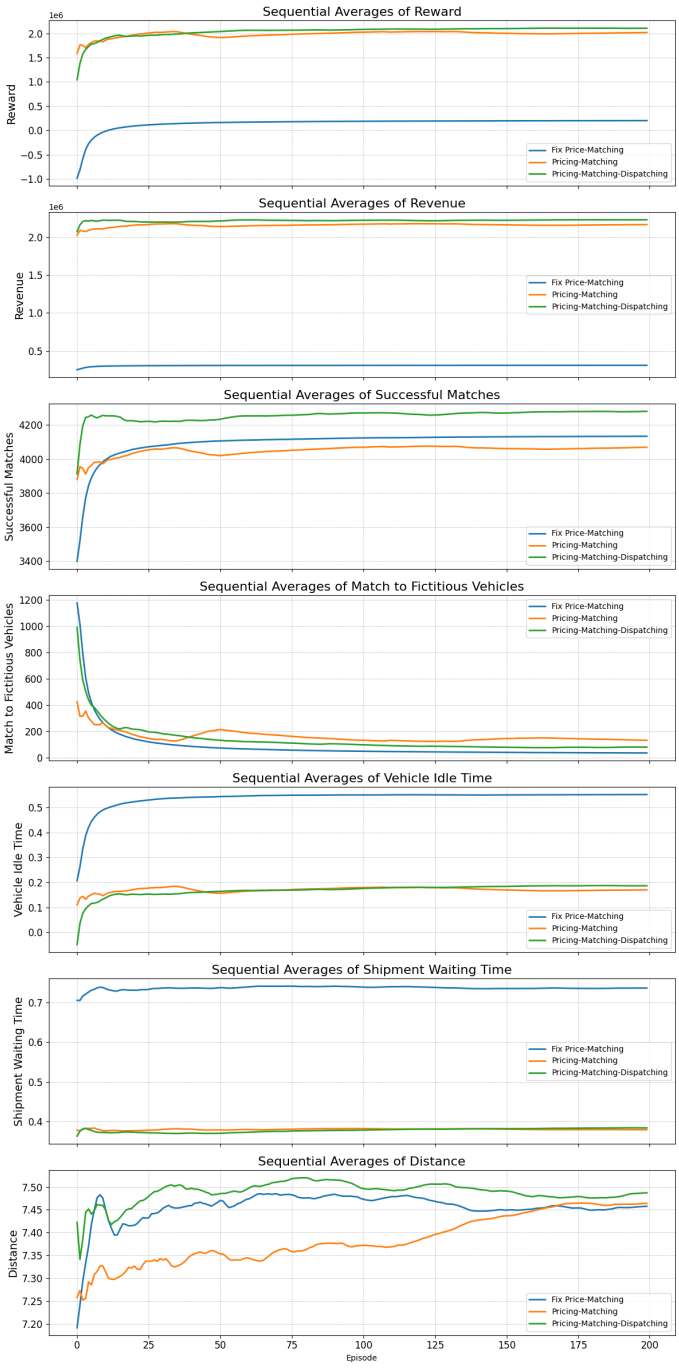

Figure 6.7 Training Convergence of Metrics for Different Agent Configurations in Montréal (Fixed Price, Joint Pricing-Matching, Full NHRL)

without addressing supply-demand imbalances in other regions. This results in localized shortages of real vehicles in high-demand zones, increasing the reliance on fictitious vehicles to maintain the matching pool.

The three-agent model mitigates this issue by incorporating a dispatching agent, which strategically repositions idle vehicles to high-demand areas. This reduces the reliance on fictitious vehicles, as reflected by the decrease in the match with fictitious vehicle rate to 79. The inclusion of the dispatching agent enhances vehicle distribution and operational reliability, effectively addressing imbalances and boosting the overall efficiency of the system.

The addition of agents has a marked impact on both vehicle idle time and shipment waiting time. In the two-agent model, the vehicle idle time per match decreases dramatically from the baseline's 0.55 to 0.17 of the time step —a significant reduction of 69.1%. The three-agent model slightly increases this idle time to 0.18, showing that the introduction of dispatching maintains the low vehicle idle time achieved in the two-agent configuration.

Similarly, the shipment waiting time per match drops by 48.6% in the two-agent model, from a baseline of 0.74 to 0.38 of the time step, and this reduction is maintained in the three-agent model with an average wait time of 0.38 per match. These noticeable decreases in both vehicle and shipment waiting times demonstrate that the two-agent model already achieves a high level of operational efficiency and responsiveness, and the three-agent model sustains these improvements while adding the dispatching benefits. The average distance per

Table 6.2 Performance metrics for different configurations (Toronto).

Metric	**Fixed Price (1 Agent)**	**Joint Matching & Pricing (2 Agents)**	**Matching, Pricing & Dispatching (3 Agents)**
Reward	634,934	4,162,329	5,947,154
Revenue	884,285	4,706,687	6,274,512
Successful Matches	15,453	14,268	15,332
Match to Fictitious Vehicle	20	100	166
Vehicle Idle Time	0.40	0.38	0.07
Shipment Waiting Time	0.63	0.58	0.32
Mileage to Pickup (km)	7.08	6.95	7.15

match remains relatively consistent across all configurations, with the two-agent and baseline models both averaging 7.46 km. The three-agent model shows a slight increase to 7.48 km, potentially due to the dispatching agent's strategy of directing vehicles toward the center of GHU6s, thus slightly extending travel distances. This marginal increase is offset by the more efficient service coverage, ensuring optimal vehicle allocation within the service area.

In conclusion, both the two-agent and three-agent models exhibit significant improvements over the baseline, demonstrating the power of integrating joint matching and pricing capabilities. The two-agent model yields substantial revenue and profitability gains over the baseline, showing the effectiveness of a coordinated real-time pricing strategy. However, the three-agent model stands out as the optimal configuration, delivering the highest reward and revenue, the greatest accuracy in matches, and reduced waiting times. The inclusion of the dispatching agent enhances the model's ability to adapt to dynamic demand patterns, effectively transforming freight logistics into a more responsive, efficient, and profitable system.

These findings from the Montréal case study validate the effectiveness of our NHRL framework in a dense urban freight context. To further evaluate generalizability, we conduct a parallel analysis using data simulated for the city of Toronto.

Toronto Case Study

The Toronto case study provides an additional validation of the proposed NHRL framework in a larger and more spatially dispersed urban context. As shown in Table 6.2 and Figure 6.8, the results reinforce the generalizability and scalability of our model across diverse urban environments.

In the baseline configuration—Fixed Price with Matching—the cumulative reward reaches 634,934, and total revenue is $884,285. This configuration results in 15,453 successful matches, with only 20 fictitious vehicle assignments, reflecting a relatively high match rate but limited profitability due to static pricing and absence of proactive dispatching. Vehicle idle time and shipment waiting time per match are recorded at 0.40 and 0.63 time steps, respectively, while the average pickup distance is 7.08 km.

The introduction of real-time pricing in the two-agent configuration significantly enhances economic performance. The cumulative reward increases to 4,162,329—over 6.5 times the baseline—and revenue surges to $4,706,687. However, the number of successful matches slightly drops to 14,268, and fictitious vehicle matches increase to 100. This trade-off, also observed in the Montréal case, stems from localized vehicle shortages due to dynamic pricing-induced imbalances not yet addressed by dispatching. Nonetheless, vehicle idle time drops to 0.38, and shipment waiting time improves to 0.58, indicating more efficient resource utilization and better responsiveness.

The three-agent configuration, which adds a dispatching component to the system, demonstrates further improvements across all metrics. The cumulative reward peaks at 5,947,154—a 42.9% increase over the two-agent model—while revenue climbs to $6,274,512. Successful

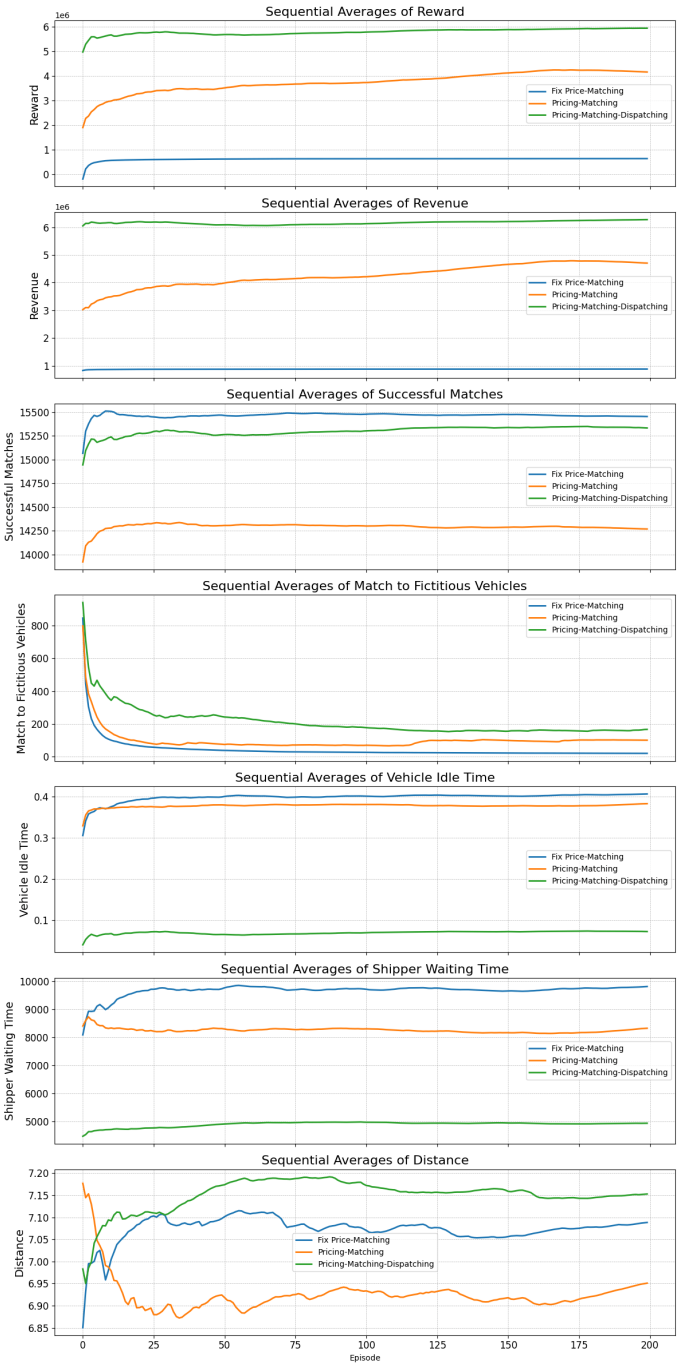

Figure 6.8 Training Convergence of Metrics for Different Agent Configurations in Toronto (Fixed Price, Joint Pricing-Matching, Full NHRL)

matches increase to 15,332, nearly recovering the baseline value, despite the system's more selective and revenue-oriented matching strategy. The number of fictitious matches rises to 166, likely due to wider dynamic price variation and temporal mismatches, but this is counterbalanced by improved system performance in other key dimensions.

Notably, vehicle idle time is reduced drastically to 0.07 time steps—an 82.5% reduction compared to the baseline—demonstrating the impact of proactive vehicle repositioning. Shipment waiting time also sees significant gains, dropping to 0.32, which is nearly half the waiting time observed in the baseline. These results show that the dispatching agent enhances the system's responsiveness by aligning idle vehicle supply with forecasted demand.

The average pickup distance in the three-agent configuration slightly increases to 7.15 km, compared to 6.95 km in the two-agent model. This modest rise is attributable to the dispatching agent's strategy of positioning vehicles in anticipation of demand rather than minimizing immediate travel distance. However, the improved matching rate and reduced idle time suggest that this trade-off leads to overall system efficiency gains.

In summary, the Toronto case study confirms that the proposed NHRL framework consistently outperforms other configurations. The three-agent model delivers the highest cumulative reward and revenue while simultaneously improving match quality, reducing idle resources, and minimizing delays. These results further validate the framework's robustness and its practical applicability to various urban freight contexts.

6.7.4 Comparison with Alternative Algorithms

Table 6.3 presents the cumulative rewards achieved by three reinforcement learning algorithms—DDPG, TRPO, and PPO—in both Toronto and Montréal case studies. The cumulative reward serves as a comprehensive indicator of system performance, encapsulating revenue generation, matching quality, vehicle utilization, and dispatching efficiency throughout the simulation horizon.

In Montréal, PPO achieves the highest cumulative reward of 2,104,112.99, outperforming TRPO (1,961,177.50) and DDPG (1,806,764.40). Similarly, in Toronto, PPO maintains superior performance with a reward of 5,947,154.80, compared to TRPO (5,552,985.92) and DDPG (2,847,380.35). These results consistently demonstrate PPO's effectiveness in learning robust policies across different urban freight environments.

The performance gap between PPO and the other algorithms can be attributed to several key factors. First, PPO's clipped surrogate objective stabilizes policy updates by constraining the deviation between the new and old policies at each iteration, thereby preventing destruc-

Table 6.3 Final Cumulative Reward for Selected Algorithms in Montréal and Toronto

Algorithm	**Montréal Reward**	**Toronto Reward**
DDPG	1,806,764.40	2,847,380.35
TRPO	1,961,177.50	5,552,985.92
PPO	**2,104,112.99**	**5,947,154.80**

tive performance shifts during training [42], which is particularly beneficial in the stochastic and highly dynamic freight logistics environment. This mechanism provides greater training reliability compared to unconstrained updates in DDPG [44] and the complex trust region computations in TRPO [43]. Second, PPO balances exploration and exploitation more effectively than DDPG, which is highly sensitive to hyperparameter tuning—such as critic learning rates, target network update speeds, and noise processes—and can easily converge to suboptimal deterministic policies in high-dimensional, continuous control problems. Third, while TRPO offers theoretical guarantees of monotonic policy improvement, its reliance on second-order conjugate gradient optimization introduces substantial computational overhead and makes it less scalable for practical applications. In contrast, PPO achieves comparable or superior performance with simpler, first-order updates. Empirically, PPO has consistently demonstrated superior robustness across diverse continuous control benchmarks, achieving strong performance without extensive hyperparameter tuning [139]. These properties collectively make PPO a particularly suitable choice for complex, dynamic environments such as real-time urban freight logistics. The consistent advantage of PPO across both cities highlights the generalizability and robustness of the NHRL framework when paired with PPO. These findings reinforce the suitability of PPO as the primary learning algorithm for joint optimization of pricing, matching, and dispatching tasks in real-time freight logistics platforms, delivering superior operational and financial outcomes.

Table 6.4 Two-Way ANOVA Results for Sensitivity Analysis on α and β

City	**Metric**	**p-value(α)**	**p-value(β)**
Montreal	Revenue generation	0.47	0.58
	Vehicle idle time	0.36	0.31
	Shipment waiting time	0.48	0.94
Toronto	Revenue generation	0.20	0.91
	Vehicle idle time	0.40	0.13
	Shipment waiting time	0.15	0.25

6.7.5 Sensitivity Analysis

To evaluate the robustness of our NHRL framework under parameter perturbations, we conducted a comprehensive sensitivity analysis on the penalty parameters α and β, which penalize mismatches in vehicle quality and capacity, respectively. A total of 64 combinations of these parameters were tested, with values ranging from 0.05 to 0.4. We performed this analysis separately for Montréal and Toronto to assess consistency across distinct urban freight environments.

Table 6.4 presents the results of two-way ANOVA tests evaluating the statistical significance of α and β on key performance metrics. For both case study, none of the p-values fall below the 0.05 threshold, indicating that variations in penalty parameters do not significantly affect revenue generation, vehicle idle time, or shipment waiting time.

The model demonstrates strong stability, with a mean revenue of \$2,283,704 and a standard deviation of \$151,557 in Montréal, yielding a coefficient of variation (CV) of only 6.64%. Similarly, in the Toronto case, revenue generation and other operational metrics remain largely unaffected. The Toronto simulation yields a mean revenue of \$6,334,907 with a standard deviation of \$361,037, corresponding to an even lower CV of 5.7%.

As illustrated in Figure 6.9, the cumulative revenue heatmap across α and β values reaffirms the model's resilience in both cities. These findings validate the practical robustness of our framework, reducing the need for extensive hyperparameter tuning during real-world implementation and simplifying deployment in diverse operational scenarios.

6.7.6 Discussion and Practical Implications

The extensive simulations conducted in both Montréal and Toronto consistently validate the proposed NHRL framework as a robust and scalable solution for urban freight coordination. While both cities demonstrate significant performance improvements over baseline and two-agent models, a comparative analysis reveals fresh insights into the framework's adaptability across diverse urban environments.

General Findings Across Both Case Studies

Across both Montréal and Toronto, the NHRL framework consistently achieved superior performance in key operational and economic metrics. The full 3-agent NHRL model delivered the highest cumulative reward and revenue in both cities, significantly outperforming the fixed-price (1-agent) and joint pricing-matching (2-agent) configurations, highlighting the

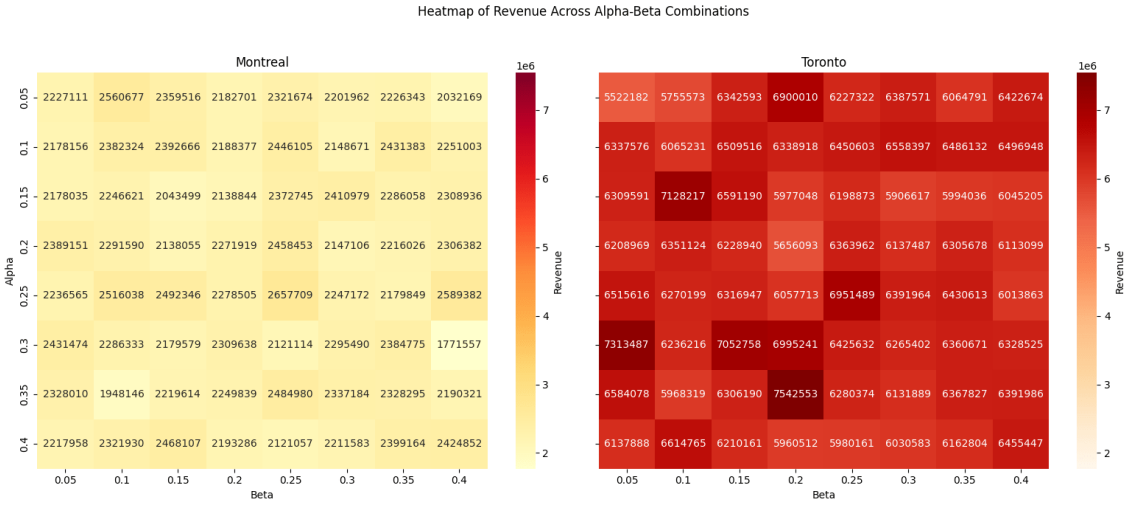

Figure 6.9 Heatmap of Cumulative Average Revenues Across α and β Parameters for Both Cities(Montréal on the left, Toronto on the right)

critical value of jointly optimizing pricing, matching, and dispatching decisions in real-time to capture greater profitability. Both Montréal and Toronto saw substantial reductions in vehicle idle time and shipper waiting time with the NHRL framework, highlighting the framework's effectiveness in enhancing resource utilization and improving service responsiveness by proactively repositioning idle vehicles to anticipated demand hotspots. Despite dynamic pricing complexities, the 3-agent model in both cities demonstrated improved matching accuracy and successful matches, mitigating the initial slight dip observed in the 2-agent model, which confirms the dispatching agent's role in balancing supply and demand, leading to more feasible and successful pairings. The consistent performance of the NHRL framework in two structurally different urban environments, Montréal and Toronto, confirms its generalizability and scalability. The framework's ability to maintain stability and effectiveness despite varying geographic layouts and demand densities is a key finding, affirming its practical applicability to diverse urban freight networks.

Underlying Data Differences and Impact on Results

While both cities serve as valuable testbeds, their inherent structural and operational differences contribute to the observed variations in raw performance metrics and convergence patterns. Toronto (cumulative reward of $5,947,154$) is a significantly larger metropolitan area than Montréal (cumulative reward of $2,104,112$), with a more spatially dispersed demand and supply landscape, as illustrated in Figures 6.5 and 6.6. This larger scale and distributed nature mean that the Toronto simulation involves a greater number of demand

points and a more dynamic fleet distribution over a broader area. Consequently, the absolute values for revenue, reward, and successful matches are notably higher in Toronto, reflecting the larger market size. The more dispersed nature of Toronto's urban freight network means that achieving optimal supply-demand balance is inherently more complex than in Montréal. This can lead to a higher number of "match to fictitious vehicles" instances in Toronto (166 for 3-agent model) compared to Montréal (79 for 3-agent model), as the system works harder to bridge gaps between available vehicles and distributed demand, especially under dynamic pricing. While both cities show robustness to penalty parameters (Table 6.4), the Toronto case exhibits a slightly lower Coefficient of Variation for revenue (5.7% compared to Montréal's 6.64%). This suggests that in a larger, more complex environment, the model's strategies might be even more critical in maintaining stable revenue generation amidst greater variability.

Explaining Differences in Training Convergence

The training convergence plots (Figures 6.7 and 6.8) show variations between Montréal and Toronto, which can be attributed to the inherent differences in their simulated environments. Toronto's plots exhibit more initial volatility and a longer "settling" period for metrics like reward and revenue compared to Montréal. This is likely due to the more dispersed network, the agents must explore in Toronto's more expansive and complex environment. The learning process involves navigating a higher number of possible pricing, matching, and dispatching combinations. For the 3-agent model, Toronto shows a more noticeable increase in reward and revenue metrics over the training episodes, particularly after initial exploration. This indicates that the added complexity and greater potential for inefficiency in a larger city (if not managed optimally) also translate into larger potential gains when an effective hierarchical framework like NHRL is applied. The dispatching agent, in particular, has a greater impact in optimizing resource allocation across a broader and more diverse set of GHU6 zones. Despite these differences, PPO consistently demonstrates smooth convergence with minimal oscillations in both cities, validating its stability and suitability for such large-scale, continuous control tasks. The slight variations in convergence speed and magnitude between the cities further show the framework's adaptive learning capabilities across different data distributions and operational scales.

New Managerial Insights from Cross-Case Analysis

The comparative analysis of Montréal and Toronto yields several enhanced managerial insights. The consistent revenue and reward improvements in both cities, regardless of their

size and demand characteristics, provide strong evidence that the NHRL framework is a viable investment for freight platforms operating in diverse urban settings. The substantial revenue increase is not an anomaly tied to a specific city, but a repeatable outcome of integrated, intelligent decision-making. While dynamic pricing and matching (2-agent model) offer significant benefits, the Toronto case study particularly highlights the disproportionate impact of the dispatching agent in larger, more dispersed networks. The reduction in vehicle idle time (82.5% in Toronto vs. 67.3% in Montréal) and shipper waiting time (nearly split in both) demonstrates that proactive repositioning is even more crucial for maintaining service quality and efficiency as the operational scale increases. This suggests that in emerging markets or smaller cities, a 2-agent system might offer strong initial gains, but for established or large-scale urban operations, the full 3-agent NHRL framework becomes essential for maximizing efficiency and minimizing service gaps. The observed trade-off between maximizing the number of successful matches and maximizing profitability (as seen by the slight initial dip in successful matches in the 2-agent model for both cities) is a key insight. The NHRL framework's ability to recover and even surpass baseline match rates while significantly boosting revenue indicates that platforms can achieve both high profitability and robust service delivery, even in dynamic and geographically varied markets. This is particularly valuable for strategic planning, allowing managers to fine-tune the balance between market share (successful matches) and financial performance (revenue). The sensitivity analysis showing robustness to penalty parameters in both cities implies that real-world deployment of this framework would require less extensive hyperparameter tuning. This significantly reduces the operational overhead and time-to-market for logistics platforms seeking to implement such advanced RL systems across multiple cities or regions, thereby simplifying the rollout path (Phases 1-3). The demonstrated generalizability reinforces the framework's potential for future innovations. As urban freight evolves (e.g., with autonomous vehicles, stricter sustainability targets), multi-resolution, modular architecture can readily integrate new objectives (e.g., carbon-aware reward structures) and technologies, ensuring long-term adaptability and competitive advantage across diverse operational contexts. In conclusion, the NHRL framework not only advances the state of integrated decision-making in urban logistics but also provides a practical, scalable, and adaptable solution. The cross-case analysis of Montréal and Toronto offers insights into its consistent performance benefits and strategic implications for logistics decision-makers in an increasingly complex and dynamic freight environment.

6.7.7 Limitations and Future Work

While the simulation environment was carefully designed to reflect realistic freight operations in Montréal and Toronto, it inherently involves simplifications. One key simplification is the

modeling of vehicle behavior. In our framework, vehicles are treated as passive entities without incorporating driver preferences, or the ability to accept or reject assignments. In real-world platforms, these human-centered factors can significantly impact match success, fulfillment reliability, and dispatch feasibility. For instance, driver preferences for certain zones, or load types could lead to assignment rejections, which the current model does not capture. Future work should consider integrating agent-based driver behavior models or historical driver acceptance data to better reflect platform dynamics and improve policy robustness.

The simulation also does not account for exogenous factors that affect real-world freight operations, such as weather conditions, road closures, or policy/regulatory changes. These disruptions can cause route infeasibility, sudden demand shifts, or temporary bottlenecks—factors that real-time systems must be able to adapt to. Incorporating predictive external data streams (e.g., weather forecasts, traffic reports, or public event schedules) could enhance the situational awareness of agents and lead to more resilient dispatching and pricing decisions.

Addressing these limitations offers promising directions for advancing both the theoretical foundations and practical applications of intelligent freight transportation systems.

6.8 Conclusion

This study introduced a Nested Hierarchical Reinforcement Learning framework for the joint optimization of real-time pricing, shipment matching, and vehicle dispatching in urban freight transportation. By employing three coordinated reinforcement learning agents across multiple H3-based spatial resolutions, the framework captures the complex interdependencies between market-driven pricing, efficient matching, and proactive dispatching.

To enhance scalability and decision-making precision, the framework integrates modular prefiltering algorithms—PAMA for matching and DEZE for dispatching—that effectively reduce action space dimensionality while preserving operational flexibility. Simulation results in two major Canadian cities, Montréal and Toronto, show that our NHRL model significantly outperforms traditional fixed-price matching and two-agent models across critical performance metrics including revenue, match success rate, vehicle idle time, and shipment delay.

In comparative evaluation, Proximal Policy Optimization demonstrated superior reward performance over baseline algorithms such as TRPO and DDPG. The sensitivity analysis further validated the framework's robustness to variations in reward shaping parameters, affirming its adaptability in dynamic freight environments.

The proposed NHRL framework offers a practical and generalizable solution for digital freight

platforms aiming to improve system-wide efficiency, responsiveness, and profitability. Its modular and scalable design makes it suitable for deployment in diverse urban logistics contexts. Future research will explore real-world implementation (like weather conditions or road closures), incorporation of sustainability indicators (e.g., emissions) into the reward functions, and extensions to electric and autonomous vehicle fleets to further advance smart and sustainable freight mobility.

Acknowledgment

The authors gratefully acknowledge Asad Yarahmadi for his contribution in preparing the Toronto dataset utilized in this study.

This work is supported by Shiphual Logistics and Mitacs through the Mitacs Accelerate program IT30680.

CHAPTER 7 ARTICLE 4: DECENTRALIZED VEHICLE-LEVEL AUTONOMY FOR URBAN FREIGHT: A DYNAMIC TASK-SWITCHING MULTI-AGENT REINFORCEMENT LEARNING APPROACH

Ali Shiri, **Samira Keivanpour**.

IEEE Transactions on Intelligent Vehicles
Submitted on August 25, 2025

Abstract

Centralized control systems in urban freight logistics struggle with scalability, adaptability, and responsiveness in dynamic environments. To address these limitations, this paper presents a decentralized, context-aware, task-switching multi-agent reinforcement learning framework that enables freight vehicles to make autonomous operational decisions at the vehicle level. Each vehicle dynamically switches among shipment matching, routing, and dispatching policies based on interpretable context variables, including load ratio, shipment count, and idle duration. These variables are designed to be transparent to support trust and decision traceability in industrial logistics settings. To manage decision complexity at scale, the framework integrates H3 geospatial partitioning with two pre-filtering modules: ShipScan for shipment selection and PADA for vehicle repositioning. This combination ensures computational tractability while maintaining high decision quality. Extensive experiments across two urban regions—Montréal and Toronto—demonstrate that the proposed approach improves successful matches by up to 25.9%, increases fleet utilization by 8.21%, and significantly reduces shipment waiting times, vehicle idle times and fictitious assignments. These results confirm the effectiveness of decentralized, interpretable autonomy in enhancing the scalability and responsiveness of next-generation urban freight systems

Keywords: Multi-Agent Reinforcement Learning, Vehicle Autonomy, Dynamic task switching Systems, Urban Freight, Context-Aware Computing, Operational Intelligence.

7.1 Introduction

Urban freight systems are experiencing a paradigm shift toward decentralized operational autonomy, driven by the limitations of centralized control mechanisms in handling dynamic, high-frequency decision scenarios. Digital freight platforms increasingly delegate decisions

to individual vehicle agents rather than relying solely on global optimization engines. This transformation creates unprecedented opportunities for vehicle-level intelligence that can adapt to local conditions while maintaining system-wide coordination.

The fundamental challenge lies in enabling freight vehicles to autonomously reconfigure their operational behavior based on contextual tasks—switching from shipment matching to route optimization to repositioning—without compromising system efficiency or requiring centralized oversight. This represents a departure from traditional freight management paradigms that assign fixed functional tasks to vehicles or rely on centralized dispatchers for all decisions.

This paper introduces a decentralized vehicle-level operational autonomy architecture that redefines how urban freight vehicles make different decisions in real time. Unlike prior works [107, 118, 140], which rely on fixed-task agents or centralized matching without dynamic adaptation, our approach enables each vehicle to dynamically switch among matching, routing, and dispatching tasks based on local operational context. The architectural novelty lies in the seamless integration of task-switching logic, context-aware prioritization, and spatially constrained pre-filtering mechanisms into a scalable, decentralized reinforcement learning framework.

Key Contributions:

- **Architectural Innovation**: A novel vehicle-level MARL framework that enables dynamic reconfiguration of decision-making structures based on local operational context.
- **Context-Aware Operational Intelligence**: Development of interpretable context-scoring mechanisms that prioritize vehicles based on multi-dimensional operational relevance (size, capacity utilization, time constraints), improving system responsiveness and load balancing.
- **Scalable and Tractable Decision-Making**: The framework integrates H3 geospatial partitioning with local pre-filtering mechanisms (ShipScan, PADA) to constrain the action space of each vehicle, maintaining decision quality while significantly reducing computational load. Additionally, this decentralized architecture mitigates the bottlenecks of centralized control by enabling real-time decisions at vehicle scale, supporting large-scale deployment in dynamic, high-demand environments.
- **Empirical Validation of Necessity**: The proposed approach is evaluated across two large-scale urban freight environments—Montréal and Toronto—comprising over 37,000 vehicle-shipment interactions. Results show that the Full MARL configuration improves successful matches by up to 25.9% and 9.3%, and fleet utilization by 3.92%

and 7.19%, respectively. These improvements are accompanied by reduced shipment waiting times, significant declines from 46–93 to 4–13 in fictitious assignments(instances where the system is forced to assign placeholder due to an exhausted candidate pool) respectively, and meaningful reductions in pickup distances in dense urban settings.

The remainder of this paper is organized as follows: Section II reviews relevant literature; Section III describes the problem context; Section IV details our methodology; Section V outlines the simulation setup; Section VI presents and discusses results; and Section VII concludes with implications and future research directions.

7.2 Literature Review

Recent comprehensive reviews highlight the evolving challenges in urban logistics environments. Nikola et al. [141] systematically analyze stochastic dynamic vehicle routing problems, emphasizing the need for adaptive decision-making frameworks that can handle real-time uncertainties. Hildebrandt et al. [142] identify key opportunities for reinforcement learning in stochastic dynamic vehicle routing, emphasizing the need for adaptive policies that can handle multiple operational objectives simultaneously. Their framework validates the potential of RL approaches but does not address the task-switching capabilities essential for modern urban freight operations.

The predominant approach in existing literature assigns fixed operational tasks to agents, fundamentally limiting their adaptability to changing operational contexts. Haliem et al. [84] introduced RL algorithms for pricing and dispatching in ride-sharing. While their approach uses coordinated agents, the agents are assigned fixed tasks, meaning each vehicle follows a static task regardless of its operational state. This design is inadequate for urban freight, where vehicles must transition between tasks such as shipment acquisition, delivery routing, or vehicle repositioning, depending on their real-time operational context. Without this task flexibility, agents cannot respond to evolving system needs, leading to inefficient resource utilization and degraded service responsiveness.

Similarly, Wang et al. [10] implemented DRL for ride-sharing with passenger transfers, but the architecture lacked mechanisms for dynamically switching agent tasks based on context. Singh et al. [59] allowed for local decision-making in multi-hop ride-sharing, yet agents still adhered to static decision logic. Guo and Xu [66] used DRL to balance cost and service quality in autonomous mobility systems but did not incorporate structural adaptability—vehicles followed fixed behavior patterns regardless of demand context or system state. Li et al. [47] used value-based learning with task specialization that were still preassigned and could not

change dynamically. These structural limitations are not isolated. In fact, many other frameworks continue to generalize agent behavior through centralized or monolithic DRL policies.

Several studies continue to rely on centralized frameworks or monolithic DRL policies that generalize across all vehicles and situations. For instance, Liu et al. [12] proposed a centralized DRL model optimized for income, but used a single policy across all vehicles. Such monolithic structures are inadequate for freight, where task differentiation is essential due to heterogeneous vehicle states and complex service constraints. Xu et al. [11] and Gao et al. [65] attempted hybrid approaches integrating repositioning and matching under unified DRL frameworks but maintained centralized decision-making paradigms that hinder scalability and responsiveness to local conditions.

Li et al. [47] addressed uncertainty in travel times with robust dispatching but relied on fixed-task, centralized logic. More holistic frameworks like Manchella et al.'s [67, 69] and Chen et al. [48] explored passenger-freight integration using DRL but suffered from centralized dispatching architectures less suitable for large-scale, real-time freight scenarios.

The third major limitation involves approaches that optimize single-task workflows or rely on predetermined task sequences, lacking the contextual intelligence necessary for dynamic urban freight environments. Tian et al. [71] proposed a probabilistic freight matching algorithm using Bayesian networks, which modeled uncertainty but lacked any capacity for policy switching or adaptive structure activation. This represents a fundamental misalignment with the dynamic nature of freight operations, where optimal task prioritization should depend on real-time operational context such as vehicle capacity utilization, spatial demand patterns, and service time constraints.

Recent efforts have explored temporal and spatial context to inform dispatch decisions. Liu et al. [12] and Qin et al. [56] integrated time-based information, while Kumar et al. [143] introduced fairness metrics into zone-level matching without efficiency loss. These works validate the benefit of context-aware modeling and represent important progress toward context-aware decision-making.

In terms of spatial partitioning, earlier studies relied on simple grid discretization [137, 144], whereas recent advances in geospatial computing, particularly hierarchical spatial indexing systems like H3, offer opportunities for more sophisticated spatial reasoning that remain underexplored in the freight logistics literature [107]. Our method adopts H3-based hierarchical indexing to align with city topology and enhance decision locality.

While modular MARL architectures have gained theoretical traction in general reinforce-

ment learning literature, their application to urban freight logistics remains critically limited. Haliem et al. [84] attempted joint decision-making using multiple global agents but did not implement dynamic task switching at the vehicle level. Systems like PRide [68] emphasized predictive matching and privacy, yet lacked vehicle-level autonomy and adaptability.

Our analysis reveals three fundamental architectural gaps that limit the effectiveness of existing approaches in urban freight contexts:

1- Task Assignment Rigidity: agents are constrained to fixed operational tasks, preventing optimal adaptation to dynamic operational contexts;

2- Centralized Control Bottlenecks: hampers responsiveness to transient local events such as demand surges or idleness; and

3- Context Insensitivity: existing systems lack mechanisms for context-aware task prioritization and dynamic structural reconfiguration, failing to leverage real-time operational intelligence for optimal decision-making.

These limitations collectively represent a fundamental mismatch between the static, centralized architectures prevalent in current literature and the dynamic, distributed nature of urban freight operations.

To address these issues, we propose a MARL framework with embedded structural flexibility, allowing vehicle agents to switch among matching, routing, and dispatching tasks based on operational context. Unlike prior works that bind decision logic to centralized planners or static tasks, our approach enables each agent to autonomously switch task structures based on interpretable real-time variables, bridging the gap between adaptability and scalability.

Our proposed approach integrates: dynamic task-switching mechanisms enabling context-driven policy selection; decentralized decision-making with shared learning to maintain scalability while preserving coordination benefits; and H3 geospatial indexing to enable locally-optimal yet globally-coherent decision-making. This approach is further supported by prioritization scoring, enhancing real-time urban freight vehicle utilization.

In the following sections, we formalize the proposed architecture and empirically demonstrate its superiority across two large-scale urban freight environments.

7.3 Problem Context

7.3.1 Challenges in Existing Architectures

Modern urban freight platforms encounter three major architectural limitations that highlight the need for vehicle-level operational autonomy. The first challenge is the scalability bottleneck of centralized control. As the number of vehicles (L) and shipment requests (K) increases, centralized coordination becomes computationally infeasible. Specifically, each decision cycle incurs a computational complexity of $\mathcal{O}(L \times K)$ in the worst case where all vehicles evaluate all available shipments, requires $\mathcal{O}(L)$ communication exchanges for assignment, and demands $\mathcal{O}(L \times K)$ memory to maintain the global system state. As fleets grow to thousands of vehicles and daily shipments, these overheads introduce significant decision latency, which undermines the system's ability to operate in real time.

Second, existing architectures suffer from operational task rigidity and context insensitivity, leading to inefficient decision-making. Static task assignments—where agents are dedicated solely to one task—fail to reflect the dynamic operational context of freight vehicles. For instance, an empty vehicle should focus on shipment acquisition (a discrete matching problem), a loaded vehicle must prioritize delivery sequencing (a combinatorial routing problem), and an idle vehicle should reposition strategically (a location optimization problem). Without the ability to switch tasks based on current context, vehicles operate suboptimally during transitional phases. Moreover, fixed tasks inhibit responsiveness to local demand variations, leading to underutilization of capacity in high-demand zones and oversupply in low-demand areas.

Finally, centralized systems exhibit limited responsiveness to local context. By aggregating local data into a global system state, they may obscure vital operational details such as vehicle-specific capacity and service capabilities, and immediate spatial demand patterns. This abstraction results in degraded decision quality and diminished responsiveness to localized events, further weakening system performance in dynamic urban environments. Collectively, these limitations underscore the need for decentralized, context-aware, and adaptable decision-making architectures at the vehicle level.

7.3.2 Proposed Solution: Vehicle-Level Operational Autonomy

Our approach addresses these limitations through a novel architectural paradigm that embeds operational intelligence directly within individual vehicle agents. Each vehicle operates as a general-purpose freight agent equipped with three specialized internal structures. The term "general-purpose" in this context refers to the agent's capability to handle the diverse

operational tasks of urban freight operations without being limited to a single operational task or vehicle type. Specifically, this general-purpose capability encompasses:

- *Matching Structure:* Evaluates and selects shipment offers based on spatial proximity, available vehicle capacity, required service quality, and pickup time windows. This structure is activated when the vehicle is empty or operating under a LTL condition.
- *TSR Structure:* Optimizes the delivery sequence for onboard shipments to minimize total travel distance and improve routing efficiency. It is activated when the vehicle is carrying multiple cargoes.
- *Dispatching Structure:* Repositions idle vehicles toward GHUs with anticipated demand surpluses based on learned predictive signals. This structure is activated when a vehicle remains idle beyond a predefined time threshold.

The platform environment is partitioned using H3 hierarchical geospatial index at resolution level 7, enabling geographically localized and computationally tractable decision-making. Two pre-filtering mechanisms reduce decision space complexity: ShipScan filters top-N shipment candidates based on local context, and PADA evaluates GHU eligibility for dispatch targets.

7.3.3 Constraints and Assumptions

The system operates under a set of operational constraints and practical assumptions grounded in real-world urban freight logistics:

- *Capacity Constraint:* A vehicle may only accept shipments if its remaining capacity is sufficient to accommodate the required load, ensuring feasible and efficient utilization of space.
- *Service Time Constraint:* Each vehicle must operate or reach a dispatch destination within its available service time window, reflecting operational time limits.
- *One-to-One Assignment:* Each shipment request is matched to a single vehicle only, preventing redundant or conflicting assignments.
- *Repositioning Feasibility:* Dispatch actions are constrained by the vehicle's remaining service time and its reachable H3 zones, ensuring that relocations are physically and temporally viable.

7.4 Methodology

To address the aforementioned limitations, we now detail the architecture and training procedures of our proposed MARL framework. This section formalizes the design of our MARL framework, where each vehicle operates as a general-purpose agent equipped with three internal RL structures. These task structures correspond to key urban freight tasks. The vehicle autonomously switches among them based on local operational context.

7.4.1 Architectural Overview: Vehicle-Level Operational Autonomy

Let $\mathcal{V}$ denote the set of vehicles and $\mathcal{R}$ the set of shipment requests. Each vehicle $v \in \mathcal{V}$ is modeled as an independent agent with access to a modular decision policy. The task-specific policy $\pi^{(v)}_{\text{type}}$ is activated according to interpretable context conditions:

$$\pi^{(v)} = \begin{cases} \pi^{(v)}_{\text{match}} & \text{if load}(v) < \theta \\ \pi^{(v)}_{\text{tsr}} & \text{if } |\text{cargoes}(v)| \geq 2 \\ \pi^{(v)}_{\text{dispatch}} & \text{if idle\_time}(v) > \tau \end{cases} \tag{7.1}$$

Here, θ represents a load ratio threshold, below which a vehicle is considered underutilized and eligible for shipment matching. Similarly, τ denotes the idle time threshold, used to identify inactivity and activate dispatching actions when no task engagement occurs within a reasonable time frame. Each policy is trained using PPO with customized state representations, reward functions, and action definitions that align with the specific goals of the respective operational task structure.

7.4.2 Dynamic task switching Logic

The core innovation of our approach lies in its dynamic task switching capability. Each vehicle continuously monitors its operational context and transitions between policies in real time. The switching logic functions as a reactive state machine governed by the vehicle's load level, shipment count, and idle duration:

- *Matching Task Activation:* When a vehicle's current load ratio is below θ, the agent activates the matching structure. This condition typically occurs when the vehicle is either empty or operating under LTL configuration, enabling it to accept new shipments.

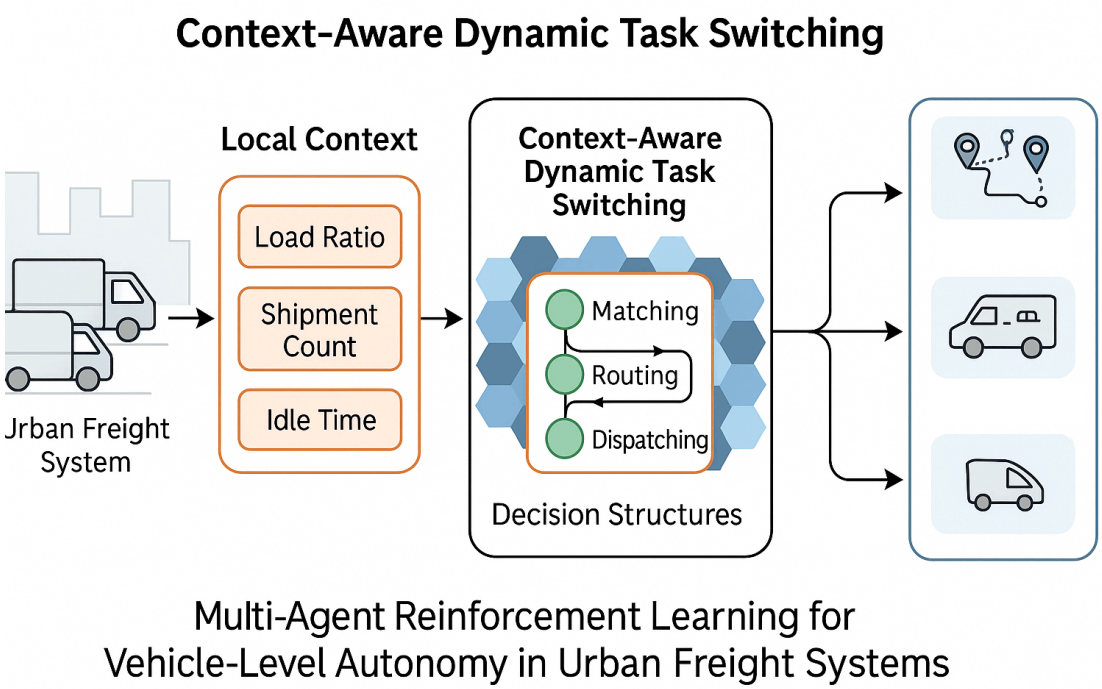

Figure 7.1 Context-Aware Dynamic Task Switching MARL Framework

- *TSR Task Activation:* Once a vehicle is assigned at least two shipments (i.e., $|\text{cargoes}(v)| \geq 2$), it switches to the TSR structure. The agent computes the optimal delivery sequence to minimize total travel distance. The immediate transition to the TSR task ensures that vehicles dynamically optimize their delivery path as soon as a load is onboarded.
- *Dispatching Task Activation:* If a vehicle remains idle for a duration exceeding τ, it activates the dispatching task structure. This task structure repositions the vehicle toward forecasted high-demand GHUs using spatial pre-filtering and learned relocation policies.

The conditions are evaluated in the order presented in Equation 7.1, enforcing a priority hierarchy where accepting new shipments (matching) takes precedence over completing deliveries (TSR), which in turn takes precedence over idle repositioning (dispatching). Thus, if a vehicle is both idle and has capacity below the threshold θ, the matching structure is activated instead of the dispatching one. Figure 7.1 illustrates the context-aware dynamic task switching in the urban freight system. The arrows from decision structures indicate dynamic activation of the corresponding task based on vehicle real-time operational context."

This modular decision architecture offers three core benefits: firstly, vehicles adapt their

task in response to dynamic system needs without requiring central coordination; secondly, decision policies are applied with precision and contextual relevance; and thirdly, fleet-wide coordination emerges through decentralized, dynamic task switching logic that scales with system size.

7.4.3 Prioritization Module

To enhance decision efficiency while maintaining true decentralization, each vehicle autonomously computes its own dynamic prioritization score based on locally observable operational context. This self-assessed score determines the vehicle's aggressiveness for shipments and influences its decision-making behavior without requiring centralized coordination. Vehicles with higher total capacity, lower unused capacity, and limited remaining service time adopt more assertive matching strategies, improving load consolidation and resource utilization through emergent coordination rather than explicit control.

The framework maintains its decentralized architecture by eliminating any centralized prioritization authority. When multiple vehicles operate within the same geographical region, each independently evaluates its contextual relevance using the same scoring function. They engage in lightweight peer-to-peer communication within their local H3 hexagonal zones, exchanging only computed scores and basic operational status. This minimal information sharing enables vehicles to establish temporary local consensus on assignment precedence without centralized oversight.

The communication protocol operates through brief, localized message exchanges. Vehicles broadcast their prioritization scores to immediate neighbors, compare received scores, and autonomously defer to higher-prioritization vehicles within a predefined time window. This bounded and ephemeral communication preserves the scalability advantages of decentralized decision-making while enabling efficient conflict resolution in high-demand areas.

Score Strategy 1 (Capacity-focused):

$$\text{Score} = a\left(\frac{C_T}{C_M}\right) + b\left(1 - \frac{C_A}{C_T}\right) \tag{7.2}$$

Score Strategy 2 (Capacity + Service Time):

$$\text{Score} = a\left(\frac{C_T}{C_M}\right) + b\left(1 - \frac{C_A}{C_T}\right) + c\left(\frac{T_{p^*}}{T_p}\right) \tag{7.3}$$

Here, C_T denotes the vehicle's total capacity, C_M is the maximum possible capacity across

all vehicles in the system, and C_A represents the vehicle's remaining available capacity. T_{p^*} indicates the elapsed service time, while T_P corresponds to the full service time window. The coefficients a, b, and c are weighting factors that control the relative influence of each component in the scoring formula. The coefficients a, b, and c are weighting factors that balance the contribution of capacity normalization, unused capacity ratio, and time urgency in the prioritization score. These coefficients are empirically tuned during the training phase based on operational objectives. They remain fixed during deployment but can be reconfigured to reflect platform-specific priorities or updated business goals.

The term $\left(\frac{C_T}{C_M}\right)$ prioritizes vehicles with higher total size, promoting better utilization of high-volume assets. The term $\left(1 - \frac{C_A}{C_T}\right)$ captures the vehicle's current load ratio—favoring those that are more loaded. The time-based component $\left(\frac{T_{p^*}}{T_p}\right)$ enables urgency-aware scheduling by emphasizing vehicles closer to their operational deadlines.

This prioritization mechanism ensures that vehicles with high operational impact are evaluated first, improving overall system responsiveness and resource allocation efficiency.

To illustrate the operational value of the Prioritization Module, consider a high-demand urban freight environment during peak hours, where shipment requests arrive continuously and multiple heterogeneous vehicles may be simultaneously eligible to fulfill the same request. Without a prioritization mechanism, vehicles would operate in random order, potentially resulting in suboptimal assignments that underutilize fleet capacity and miss opportunities for efficient load consolidation.

The Prioritization Module addresses this challenge through dynamic scoring that reflects real-time operational relevance. For instance, consider two candidate vehicles for a shipment: Vehicle A has a large total size, is 40% empty, and is nearing the end of its service window; Vehicle B has a smaller total size, is empty, but has just started its service shift. Despite both being eligible for the shipment, Vehicle A would receive a higher prioritization score due to its higher capacity utilization and time urgency.

This Prioritization mechanism directs computational attention to operationally valuable vehicles, enabling the system to maximize load utilization, minimize idle time, and improve responsiveness. By ensuring that vehicles best positioned to make immediate and efficient contributions receive priority consideration, the module enhances overall fleet efficiency and service quality in real-time urban freight environments.

7.4.4 Matching Task: Shipment Allocation

The matching task activates when a vehicle is empty or operates under an LTL condition.

Shipment Scanner Algorithm

The ShipScan enables each vehicle to identify and evaluate the most contextually relevant shipment candidates within its surrounding area, based on operational constraints such as proximity, capacity, time, and service requirements. The workflow is as below:

1. *Initial Search within the GHU of Vehicle:* The algorithm starts by evaluating all shipments located in the same GHU region as the vehicle. It filters those shipments based on availability window, quality threshold, and the same vehicle capacity. If more than N suitable shipments are found, a random subset of N is selected. If exactly N, all are retained. If fewer than N, the algorithm stores all and continues searching.

2. *Expansion to Neighboring Rings:* The search expands to the adjacent two GHU rings. The ShipScan assesses all candidate shipments in those regions against capacity compatibility, service time windows, and quality scores. New valid shipments are added to the candidate pool until the total reaches N.

3. *Low Availability — Urgent Vehicle assignment:* If the vehicle is marked as low availability or in urgent need of assignment, ShipScan relaxes the quality score threshold (which operates on discrete levels) and re-scans the initial and neighboring GHU regions to identify shipment requests with more flexible quality constraints. If a sufficient number of candidates is found, the pool is completed; otherwise, the algorithm proceeds to apply further relaxation steps.

4. *Further Relaxation — Capacity Flexibility:* The algorithm includes shipments requiring up to the vehicle's remaining available capacity, assuming the remaining volume can be matched later. This strategy maximizes load consolidation opportunities while expanding the shipment candidate pool.

5. *Time Window Tolerance:* If the vehicle can tolerate a broader idle window, the search includes shipments with later service start times or longer pickup buffers. The algorithm then readjusts selection priorities based on proximity and compatibility.

6. *Fallback and Fictitious Shipments:* If a sufficient number of shipments still cannot be found, ShipScan inserts fictitious (placeholder) shipment entries to preserve consistency in the top-N format. A fully fictitious set is flagged as invalid; otherwise, forwarded to the RL agent for final selection.

The RL policy for matching task then evaluates the pre-filtered top-N shipments, incorporating spatial proximity, waiting and idle times, capacity volumes, quality parity, delivery distance, and reward expectation to select the most suitable shipment.

State Space

The state space for the matching structure captures both vehicle-specific attributes and the features of candidate shipments in the pre-filtered pool. For each vehicle v, the state is defined as follows:

$$s_v^F = \{\underbrace{x_v, y_v, \text{sat}_v, \text{cap}_v, \text{idle}_v}_{\text{Vehicle Features}}; \underbrace{\{x_r^o, y_r^o, x_r^{de}, y_r^{de}, \text{cap}_r^{\text{req}}, \text{sat}_r^{\text{req}}, \text{wait}_r\}_{d=1}^{N}}_{\text{Top-}N\text{ Shipment Features}}\} \tag{7.4}$$

The first part of the state encodes vehicle-level information: (x_v, y_v) are the current spatial coordinates of vehicle v, sat_v denotes the vehicle's satisfaction level, cap_v indicates the remaining capacity units available in the vehicle, and idle_v captures how long the vehicle has remained idle.

The second part of the state comprises attributes for each of the top-N shipment candidates pre-filtered by the ShipScan. For each shipment request r, (x_r^o, y_r^o) and (x_r^{de}, y_r^{de}) represent the coordinates of the origin (pickup) and destination (drop-off) locations, respectively. The term $\text{cap}_r^{\text{req}}$ refers to the shipment's required capacity units, $\text{sat}_r^{\text{req}}$ reflects the shipment's required satisfaction level, and wait_r is the waiting time since the shipment was requested or entered the system.

Action Space

The agent selects one shipment r from the pre-filtered top-N pool $\mathcal{P}_v^{(N)}$. Each action corresponds to committing the vehicle to that shipment, considering factors such as proximity, capacity fit, idle and waiting times, delivery distance, and satisfaction score.

Reward Function

The reward function for the matching task is designed to capture several key operational objectives. First, efficiency is promoted by minimizing pickup and delivery distances, which in turn reduces both operational costs and environmental impact. Second, the reward function encourages responsiveness by penalizing long idle and waiting times, ensuring that shipments are assigned promptly and service delays are minimized. Third, utilization is improved by fa-

voring shipment assignments that closely align with the vehicle's capacity, thereby increasing load factors and reducing underutilized trips. Lastly, service quality is addressed by prioritizing shipments with higher satisfaction scores, enhancing the overall customer experience. Each term in the reward function is associated with one of these goals and is weighted according to its relative importance. The agent is trained to minimize these penalized components, effectively maximizing the overall operational value.

$$F(X_{vr}) = -\sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}}\left(d_{vr}^{pick}W_{vr}^{pick} + d_{vr}^{dist}W_{vr}^{dist} + (T_v + T_r)W_{vr}^{t} + (C_v - C_r)W_{vr}^{c} + W_{vr}^{q}\right)X_{vr} + U(1 - X_{vr}) \tag{7.5}$$

In the function 7.5, the term d_{vr}^{pick} represents the distance between vehicle v and origin of shipment r, weighted by W_{vr}^{pick} to penalize long pickup distances that consume fuel and time. Similarly, d_{vr}^{dist} captures the delivery distance from shipment origin to destination, with W_{vr}^{dist} penalizing long-haul deliveries that may be inefficient. This encourages shorter delivery trips, thereby improving vehicle availability for future shipments and enabling the platform to serve more requests within the same operational window. The time component $(T_v + T_r)W_{vr}^{t}$ addresses system responsiveness by considering both vehicle idle time T_v and shipment waiting time T_r, and weighted by W_{vr}^{t} to prioritize matches that reduce overall waiting.

Capacity utilization is optimized through $(C_v - C_r)W_{vr}^{c}$, which penalizes differences between vehicle available capacity C_v and shipment required capacity C_r, thereby encouraging assignments that maximize load efficiency and minimize unused space. The quality factor W_{vr}^{q} ensures that shipments receive preferential treatment in the matching process. Each potential match is represented by the binary decision variable X_{vr}, while fictitious matches incur an additional penalty U to encourage real, value-generating assignments. The negative sign in the formula ensures the agent works to minimize these weighted costs, effectively maximizing the true reward. Through this reward structure, the system simultaneously pursues sustainability (shorter distances), profitability (better capacity utilization), and customer satisfaction (reduced idle, waiting times, and quality handling), creating a balanced and efficient logistics operation.

7.4.5 TSR Task: Routing Optimization

The TSR task activates when a vehicle carries multiple shipments and must determine the optimal delivery sequence to minimize travel distance. This task addresses the dynamic vehicle routing problem within a localized delivery context.

State Space

The state space for the TSR structure captures the spatial configuration of the current vehicle position and the set of destinations that must be visited to complete all active deliveries:

$$s_v^B = \{(x_v, y_v), (x_1, y_1), \ldots, (x_n, y_n)\} \tag{7.6}$$

Where (x_v, y_v) represents the current vehicle coordinates. $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ is the set of coordinates for all remaining delivery locations.

This representation enables the agent to reason about spatial relationships and distances between the current position and all pending delivery points, allowing for optimized route planning.

Action Space

The action space for the TSR structure consists of selecting the next drop-off location from the set of onboard shipments. At each step, the agent chooses one destination:

$$A_v^B = \{d_1, d_2, \ldots, d_n\} \tag{7.7}$$

where each d_i is a pending delivery point.

The agent employs sequential decision-making, selecting one destination at each time step, moving there, and then selecting the next destination from the remaining delivery points. This incremental approach reduces computational complexity from $O(n!)$ to $O(n^2)$ while enabling dynamic reoptimization as conditions change during delivery execution.

Reward Function

The reward function for the TSR structure directly incentivizes route optimization by minimizing the total travel distance. It is defined as:

$$B_v = -\sum_{\substack{j=1 \\ j \neq i}}^{n} \text{dist}(i, j) \tag{7.8}$$

Here, $\text{dist}(i, j)$ represents the distance between location i and location j in the planned delivery sequence of n total stops for vehicle v. The negative sign converts the distance minimization problem into a reward maximization framework, ensuring the agent learns to select routes that minimize total travel distance. This formulation directly addresses the classic traveling salesman problem in the context of multiple deliveries, encouraging efficient routing patterns that reduce mileage, fuel consumption, and delivery time.

7.4.6 Dispatching Task: Vehicle Repositioning

The dispatching task activates when a vehicle remains idle beyond a predefined threshold. It uses a PADA to identify eligible GHUs for relocation, then applies a learned policy to select the optimal destination.

PADA

The PADA algorithm is designed to identify eligible GHUs for dispatching idle vehicles. It evaluates demand-supply imbalances while accounting for vehicle capacity and quality compatibility with forecasted shipments. The algorithm proceeds as follows:

1. *Initial Evaluation*: Each idle vehicle is assessed based on its current H3 tag, along with its capacity, quality score, and remaining service time.

2. *Spatial Search Expansion*: We assume that vehicles can traverse three GHUs at a resolution 7 during each timestamp. The vehicle's search area expands to adjacent GHUs within three hexagonal rings in resolution 7 per each available service time, reflecting the maximum relocation range per time step.

3. *Demand-Supply Analysis*: For each candidate GHU, the algorithm estimates future shipment demand and anticipated vehicle supply by leveraging predictive signals from the simulation environment. These forecasts are time-aligned with the estimated arrival time of the repositioned vehicle, enabling identification of zones likely to experience demand surpluses or supply shortages at the time of arrival.

4. *GHU Eligibility Assignment*: GHUs with projected surplus capacity demand are marked as eligible for relocation with a value of 1, while those without surplus demand or outside

the selected GHUs are assigned a value of -1.

Figure 7.2 illustrates an example of the PADA algorithm's output, showing the origin GHU (in red), eligible GHUs (in blue), and non-eligible GHUs (in black).

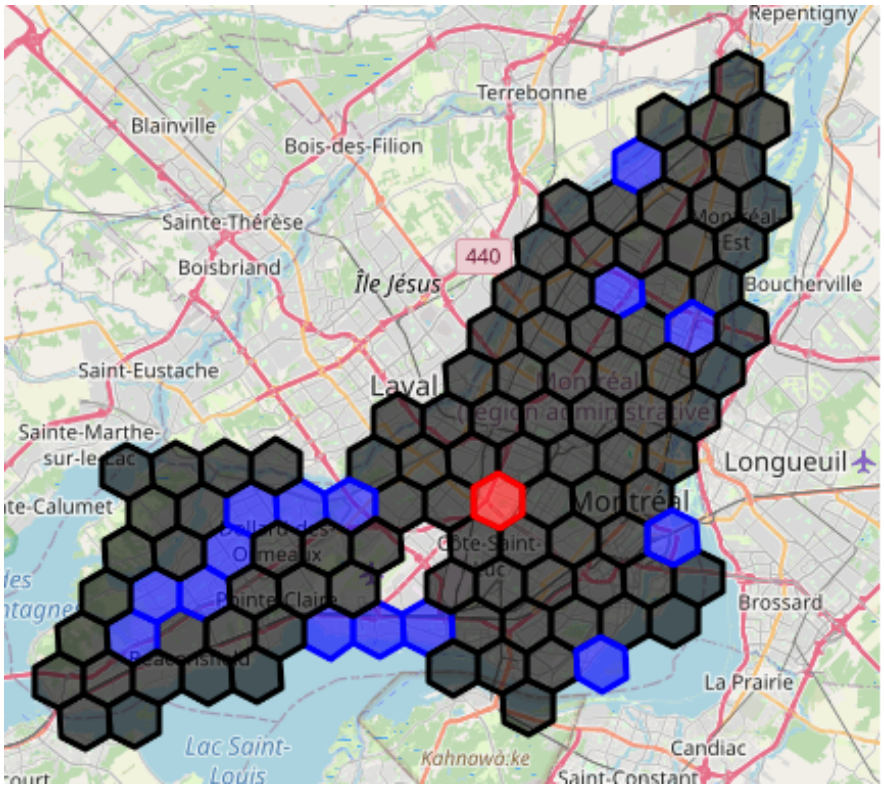

Figure 7.2 Origin GHU (red), Eligible GHUs (blue), and Non-eligible GHUs (black) as determined by the PADA algorithm.

State Space

The state space of the dispatching structure encodes the spatial and contextual information required to make informed relocation decisions. It is defined as:

$$s_{\acute{v}}^{G} = \left(E_{\acute{v}}, \left(latH_{\acute{h}}, lonH_{\acute{h}}\right), \left(latV_{\acute{v}}, lonV_{\acute{v}}\right)\right) \tag{7.9}$$

This vector captures the agent's current knowledge of eligible relocation zones and its own position, enabling optimal decision-making. It includes the following components:

- *Hexagon Eligibility Vector ($E_{\acute{v}}$)*: A binary vector generated, indicating which GHUs are eligible for vehicle dispatch.

- *Hexagon Central Points Vector ($latH_{\acute{h}}, lonH_{\acute{h}}$)*: It stores the latitude and longitude of the central points of all GHUs, providing spatial references for distance calculations.
- *Vehicle Positions Vector ($latV_{\acute{v}}, lonV_{\acute{v}}$)*: Records the latitude and longitude of an idle vehicle, crucial for evaluating remaining service time, identifying reachable GHUs, and computing distances.

Action Space

The action space consists of Q discrete options, each corresponding to a potential GHU relocation target:

$$A_{\acute{v}}^{G} = \{1, 2, ..., Q\} \tag{7.10}$$

Reward Function

The reward function is designed to encourage efficient and valid repositioning while discouraging costly or irrelevant moves:

$$G(Y_{\acute{v}\acute{h}}) = -\sum_{\acute{v}\in\acute{\mathcal{V}}}\sum_{\acute{h}\in\acute{\mathcal{H}}} d_{\acute{v}\acute{h}}^{\acute{l}} Y_{\acute{v}\acute{h}} + \acute{U}\left(1 - Y_{\acute{v}\acute{h}}\right) \tag{7.11}$$

In the formulation (7.11), $Y_{\acute{v}\acute{h}}$ is a binary decision variable equal to 1 if idle vehicle $\acute{v}$ moves to eligible GHU $\acute{h}$, and 0 otherwise. The term $d_{\acute{v}\acute{h}}^{\acute{l}}$ represents the spatial distance between vehicle $\acute{v}$ and the center of GHU $\acute{h}$. The penalty coefficient $\acute{U}$ is introduced to discourage relocation to non-eligible or suboptimal zones.

The negative sign ensures that the agent maximizes its reward by minimizing total relocation distances. When combined with the eligibility constraints from the PADA module, this reward function encourages the agent to select nearby, demand-rich GHUs while avoiding inefficient dispatches.

7.4.7 Training and Algorithm Selection

All three structures are trained using PPO, which provides stable policy updates and sample efficiency. While each structure maintains dedicated actor-critic networks, we adopt a shared policy training paradigm to improve generalization and computational scalability.

Task-specific Actor-Critic Networks

Each decision structure in the proposed method is implemented with its own dedicated neural architecture. The framework consists of three distinct actor networks: π_{match}, π_{tsr}, and π_{dispatch}, each with its own parameters and specialized to handle the unique operational requirements of its respective decision structure. Similarly, the framework maintains task-specific value networks V_{match}, V_{tsr}, and V_{dispatch} that estimate expected returns for their respective operational contexts, enabling precise advantage estimation for stable PPO-based policy updates.

Each actor network processes vehicle-specific contextual inputs and generates actions tailored to the operational logic of its respective structure. The parameter sharing occurs at the fleet level, where vehicles utilize identical task-specific networks rather than having individually parameterized networks per vehicle.

Each vehicle dynamically selects its current policy based on contextual thresholds defined by load level, cargo count, and idle duration, following the task-switching logic previously formalized in Equation (7.1). Notably, the framework employs a shared-policy architecture: all vehicles utilize the same instance of the matching policy, the same instance of the TSR policy, and the same instance of the dispatching policy. These shared policies are independently trained using PPO with task-specific state spaces, reward structures, and action definitions. This design enables decentralized scalability while maintaining structural specialization.

At runtime, each vehicle agent dynamically switches between these policies based on its operational context. This approach ensures that each vehicle utilizes the most appropriate decision-making structure according to its current state while benefiting from fleet-wide learning experiences.

Experience Pooling and Sampling

To align with task-specific training, the framework employs independent replay buffers for each decision structure. Specifically, $\mathcal{I}_{\text{match}}$ stores experiences related to shipment allocation, $\mathcal{I}_{\text{tsr}}$ contains transitions from routing among pickups and drop-offs, and $\mathcal{I}_{\text{dispatch}}$ holds data for idle vehicle repositioning tasks. During training, mini-batches are sampled separately from each buffer to update the corresponding policy and value network, ensuring that each structure learns from task-relevant interactions.

Modified PPO Training Algorithm

The training process begins by collecting task-specific trajectories in the form of (s_t, a_t, r_t, s_{t+1}). Advantage estimates are computed using GAE, and policy parameters are updated by minimizing the clipped surrogate loss:

$$L^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{7.12}$$

Simultaneously, the value network is optimized using the MSE between predicted and target returns:

$$L^{\text{VF}}(\theta) = \hat{\mathbb{E}}_t \left[(V(s_t;\theta) - \hat{V}_t)^2 \right] \tag{7.13}$$

The complete training process, including structure-switching, trajectory sorting, and parameter updates, is outlined in Algorithm 7.

Algorithm 7 Shared Policy Training with Structure-Switching PPO

Require: Initial parameters $\theta_{\text{match}}, \theta_{\text{tsr}}, \theta_{\text{dispatch}}$

for each training iteration **do**
  for each vehicle agent $v \in \mathcal{V}$ **do**
    Determine structure type
    Run corresponding policy π_{type}
    Store transitions in buffer $\mathcal{B}_{\text{type}}$
  end for
  for each structure type $\in \{\text{match}, \text{tsr}, \text{dispatch}\}$ **do**
    Sample mini-batch from $\mathcal{B}_{\text{type}}$
    Compute $\hat{A}_t$ via GAE
    for K epochs **do**
      Update policy using clipped objective L^{CLIP}
      Update critic using value loss L^{VF}
    end for
    Update old policy weights: $\pi_{\text{type}}^{\text{old}} \leftarrow \pi_{\text{type}}$
  end for
end for

7.4.8 Scalability Considerations

The proposed MARL framework is designed to address scalability challenges that hinder centralized approaches in urban freight systems. As the number of vehicles and requests grows, centralized coordination becomes computationally intensive. Our decentralized structure

alleviates this through four key mechanisms:

Decentralized Decision-Making

Each vehicle acts independently, selecting and applying the appropriate decision structure based on local context. This parallelism eliminates the bottleneck of centralized assignment and allows the system to scale naturally with fleet size.

Pre-filtering Mechanisms

The ShipScan and PADA modules play a critical task in reducing the action space before policy evaluation. ShipScan filters the set of candidate shipments by selecting only the top-N options based on local context, such as proximity, available capacity, and service constraints. Similarly, PADA narrows down dispatching choices by identifying eligible GHUs for relocation using demand-supply forecasts and geographic feasibility. This targeted pre-filtering ensures that each vehicle evaluates only the most relevant options, significantly improving computational efficiency.

H3 Spatial Partitioning

To maintain computational tractability in dense urban freight environments, the decision space of each vehicle is geographically bounded using H3 spatial indexing at resolution 7. The H3 framework partitions the operational landscape into uniformly sized GHUs, offering notable advantages over traditional grid systems due to their spatial continuity, minimized edge effects, and scale-invariant properties.

By leveraging this structure, each vehicle's set of admissible actions—whether selecting shipments or determining repositioning targets—is restricted to candidates located within a finite subset of H3-indexed zones. Formally, the dimensionality of the action space for any given vehicle v is upper-bounded by the number of reachable hexagons at the chosen resolution:

$$|\mathcal{A}_v| \leq |\mathcal{H}_7| \ll |\mathcal{R}| \tag{7.14}$$

Here, $\mathcal{A}_v|$ denotes the size of the vehicle's action space, $|\mathcal{H}_7|$ represents the number of H3 cells at resolution level 7 that are reachable given operational constraints, and $\mathcal{R}$ is the total number of shipment requests in the system. This inequality ensures that each vehicle evaluates a geographically localized and significantly reduced subset of the global request pool, even in high-density, large-scale freight environments.

This bounded spatial abstraction yields two core advantages. First, it preserves local decision optimality by aligning vehicle-level reasoning with spatially relevant demand patterns. Second, it mitigates combinatorial complexity by preventing the exponential growth of the decision space with respect to system scale. When integrated with context-driven pre-filtering modules such as ShipScan and PADA, the H3-based partitioning becomes a critical enabler for scalable, real-time vehicle-level autonomy in complex urban logistics networks.

Shared Policy Architecture

Vehicles utilize shared task-specific policy networks, which offers several key advantages. First, this approach ensures memory efficiency, as the number of parameters remains fixed regardless of the fleet size. Second, it improves training efficiency by enabling shared experience buffers, allowing vehicles to learn from a more diverse set of interactions and accelerating policy convergence. Together, these mechanisms enable real-time, scalable decision-making across large metropolitan freight networks.

7.5 Simulation Setup

We describe the experimental setup used to evaluate the effectiveness of the proposed decision architecture in this section.

7.5.1 City Models

To ensure consistency and comparability across our research stream, we adopt the same simulated urban freight environments used in our prior studies [107, 118, 140]. Specifically, we model two major Canadian metropolitan areas—Montréal and Toronto—using synthetic yet realistic shipment and vehicle activity profiles. Each city is spatially discretized into H3 geospatial units at resolution level 7, allowing for localized decision-making and scalable simulation.

Shipment requests are generated dynamically with attributes that reflect real-world conditions based on [108] and [109], including random origin-destination pairs, variable load sizes, service quality levels, and time window constraints. These requests are designed to mimic diverse real-life logistics scenarios, from last-mile deliveries to medium-range intra-urban transport.

Vehicle fleets in both cities are initialized with heterogeneous capacity units and levels (ranging from light to heavy-duty trucks) and assigned randomized initial locations within the

H3 grid. Their operational constraints, including available service time, fuel capacity, and quality compliance, are consistently applied across all experiments. Demand density, vehicle distribution, and request clustering follow empirical patterns based on historical freight movement data from prior urban logistics studies, enabling a controlled yet realistic evaluation of model performance. Figures 7.3 and 7.4 illustrate the spatial distribution of shipment origins, destinations, and vehicle availability in Montréal and Toronto, respectively, based on the H3 resolution 7 grid.

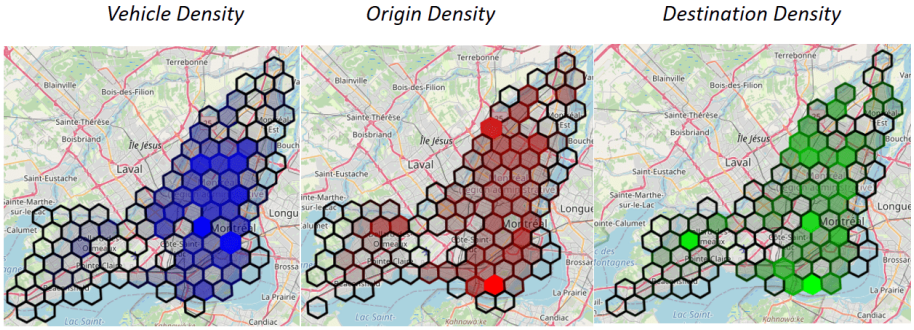

Figure 7.3 Spatial distribution of shipment origins, destinations, and vehicle availability in Montréal at resolution level 7.

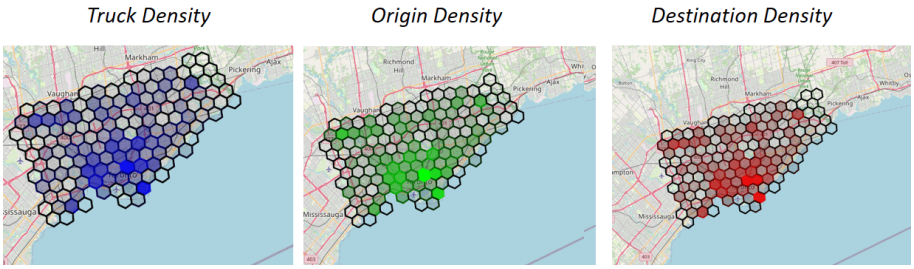

Figure 7.4 Spatial distribution of shipment origins, destinations, and vehicle availability in Toronto at resolution level 7.

7.5.2 Hyperparameter Tuning

We manually selected and adjusted critical PPO hyperparameters for each of the three decision structures based on empirical observations of training stability, policy performance, and computational efficiency. All experiments were conducted on a workstation equipped with an Intel Core i7 processor, 64 GB RAM, and an 8 GB VRAM graphics card. The actor-critic architecture was implemented using PyTorch in Python, and each configuration was evaluated over 200 simulation episodes.

The policy network is updated every 64 time steps, and each PPO update involves 20 training epochs to ensure stable yet frequent refinement of policy parameters. The PPO clipping parameter is set to 0.2 to control the magnitude of policy updates and improve convergence reliability. A discount factor of $\gamma = 0.99$ is used to balance short-term and long-term rewards during learning.

Learning rates are differentiated across network components: the actor network uses a learning rate of 0.00005, while the critic network is trained at 0.0001. For reward shaping, penalty values U and $\acute{U}$ are both set to 1000, providing strong negative incentives against fictitious matches and infeasible dispatch actions.

In the Matching structure, the candidate shipment pool size is set to $N = 10$ in the ShipScan module, striking a balance between computational tractability and decision diversity.

7.6 Results and Discussion

We evaluate our framework on key operational metrics across six configurations to assess the contribution of each architectural component.

7.6.1 baselines

We evaluate six key system variants, each progressively adding components to enhance decision-making capabilities:

1. **Baseline Global:** The centralized baseline corresponds to a global RL agent that manages shipment-vehicle assignments for the entire fleet. It uses a monolithic policy trained for the matching task and lacks structural decomposition for context-aware task differentiation such as routing or dispatching. All decisions are made centrally, without vehicle-level autonomy.

2. **Baseline MARL:** This decentralized baseline represents a general-purpose vehicle agent architecture restricted to the matching task. Each vehicle operates independently using the same trained matching policy and the ShipScan module for local shipment selection. However, it lacks task-switching logic, prioritization scoring, and dispatching capabilities. This setup serves as a transitional benchmark between centralized coordination and the fully modular MARL configuration.

3. **LTL + TSR:** Introduces two-task adaptive between Matching and TSR policies based solely on the vehicle's load state. This configuration does not include any prioritization,

PADA for pre-filtering, or dispatching components, representing a reactive decision-making baseline.

4. **Ctx1:** Extends LTL + TSR by incorporating a capacity-based prioritization scoring mechanism. Vehicles are prioritized for matching based on total shipment size and load ratio, enabling more informed decision-making under partial load conditions.

5. **Ctx2:** Builds upon Ctx1 by integrating time-service constraints into the scoring logic. Vehicles are ranked not only by capacity but also by urgency of service, improving responsiveness to time-critical shipments.

6. **Full MARL:** The most comprehensive configuration. It combines Dynamic task switching, ShipScan, and both capacity- and time-aware scoring modules. Additionally, it includes PADA for pre-filtering and introduces a dedicated Dispatching task structure, allowing idle vehicles to proactively relocate based on predicted spatial demand patterns.

7.6.2 Performance Metrics

To evaluate the effectiveness of our dynamic task switching MARL framework, we employ a comprehensive set of performance metrics that capture operational efficiency, service quality, and system-wide sustainability:

Table 7.1 Performance Comparison of MARL Variants in Montréal and Toronto

Model	Successful	Fictitious	Wait	Idle	Pickup Dist	Trucks	Util. Gain (%)
Montréal Case Study							
Baseline Global	3705	46	0.20	0.54	7.62	3705	0.00
Baseline MARL	4042	7	0.07	0.42	7.69	4042	0.00
LTL + TSR	4284	6	0.10	0.42	7.75	4107	4.31
Ctx1	4209	8	0.10	0.39	7.68	3957	6.37
Ctx2	4343	4	0.10	0.41	7.58	4112	5.62
Full MARL	**4667**	**4**	**0.08**	**0.42**	**7.51**	**4491**	**3.92**
Toronto Case Study							
Baseline Global	12347	93	0.22	0.46	7.21	12347	0.00
Baseline MARL	12863	13	0.06	0.30	7.01	12863	0.00
LTL + TSR	13581	12	0.05	0.32	7.85	12706	6.89
Ctx1	13315	14	0.05	0.32	7.4	12305	8.21
Ctx2	13494	14	0.05	0.34	7.3	12577	7.29
Full MARL	**13491**	**13**	**0.05**	**0.35**	**7.31**	**12586**	**7.19**

- **Successful Matches:** Total number of shipments successfully matched to vehicles, reflecting overall system throughput.

- **Fictitious Matches:** Number of instances where vehicles were assigned fictitious shipments due to insufficient available options. These assignments incur a significant penalty of 1000 units (as detailed in Section IV) to strongly discourage the system from relying on non-existent shipments, making the reduction of fictitious matches a critical optimization objective.
- **Shipment Waiting Time:** Average time shipments wait before matching, a critical customer service metric.
- **Vehicle Idle Time:** The duration vehicles remain without assignments, a key indicator of fleet utilization efficiency.
- **Distance to Pickup:** Average distance traveled by vehicles to reach shipment pickup locations, a critical sustainability and operational cost metric.
- **Used Trucks:** Total number of unique vehicles utilized throughout the operational period, reflecting fleet deployment efficiency.
- **Utilization Gain:** Percentage improvement in fleet utilization, computed as the relative increase in the match-to-truck ratio compared to the baseline.

These metrics collectively provide a holistic view of how dynamic task switching, prioritization scoring, and pre-filtering mechanisms influence the performance of urban freight operations.

7.6.3 Montréal Case Study Quantitative Results

Table 7.1 presents the comparative performance of five MARL configurations in the Montréal case study. The results clearly highlight the advantages of incorporating dynamic task switching, prioritization scoring, and dispatching mechanisms.

The Full MARL configuration achieves the highest number of successful matches (4,667), which corresponds to a 25.9% improvement over the Baseline Global model (3,705) and a 15.4% improvement over the Baseline MARL model (4,042). This demonstrates both the value of decentralized decision-making and the additional benefits of dynamic task switching with prioritization and dispatching.

The progression from Baseline Global through each MARL variant reveals several important insights. The shift from centralized (Baseline Global) to basic decentralized control (Baseline MARL) yields a 9.1% increase in successful matches, demonstrating the fundamental value

of vehicle-level autonomy. Introducing two-task adaptation (LTL+TSR) adds another 5.98% improvement over Baseline MARL, while prioritization scoring (Ctx1, Ctx2) provides more nuanced gains of 4.1% and 7.4% respectively over Baseline MARL. Most notably, the Full MARL configuration with integrated dispatching delivers the largest overall performance boost, achieving a 25.9% increase over the traditional centralized approach.

In terms of operational efficiency, the Full MARL model achieves competitive shipment waiting time (0.08) compared to the Baseline Global (0.20) and Baseline MARL (0.07), representing a 60% improvement over centralized control while maintaining near-optimal responsiveness. Regarding idle vehicle times, the results show that Baseline Global maintains the highest idle time at 0.54, while the MARL variants demonstrate more efficient vehicle utilization with lower idle times.

Distance efficiency shows consistent improvements, with Full MARL achieving the shortest average pickup distances (7.51 km), outperforming both Baseline Global (7.62 km) and most other configurations. This reduction in empty travel directly translates to improved sustainability through reduced emissions and operational costs.

Fleet utilization metrics further underscore these advantages, with the Full MARL configuration achieving a 3.92% improvement in the ratio of matches to vehicles used compared to Baseline MARL, and an even more substantial improvement over the centralized approach, indicating significantly more efficient resource utilization.

7.6.4 Toronto Case Study Quantitative Results

As shown in Table 7.1, the performance trends in Toronto are largely consistent with those observed in Montréal, with the MARL variants demonstrating robust improvements over both centralized and basic decentralized approaches across this distinct urban environment.

The Full MARL configuration achieves 13,491 successful matches, representing a 9.3% improvement over the Baseline Global model (12,347) and a 4.9% improvement over the Baseline MARL model (12,863). These results demonstrate consistent value across diverse urban environments, with the decentralization benefit (Baseline Global to Baseline MARL) showing a 4.2% improvement in the Toronto context.

The progression through model variants in Toronto highlights distinct contributions of each component. The fundamental shift from centralized to decentralized control (Baseline Global to Baseline MARL) provides a 4.2% increase in successful matches. The LTL+TSR configuration yields an additional 5.6% increase over Baseline MARL, while prioritization variants (Ctx1, Ctx2) achieve moderate improvements of 3.5% and 4.9% respectively over Baseline

MARL. The Full MARL configuration demonstrates the cumulative benefit of all components, achieving a 9.3% improvement over the centralized approach while maintaining excellent operational efficiency metrics.

Fictitious matches show a dramatic improvement trajectory: from 93 in Baseline Global to 13 in both Baseline MARL and Full MARL, indicating that the transition to decentralized decision-making fundamentally improves matching precision in Toronto's larger, more spatially distributed freight system. The consistency between Baseline MARL and Full MARL in fictitious matches (both at 13) suggests that the throughput gains in Full MARL are achieved through genuine efficiency improvements rather than relaxed matching criteria.

Operational efficiency metrics reveal substantial improvements over centralized control. Full MARL achieves excellent shipment waiting time (0.05) compared to Baseline Global (0.22), representing a 77% improvement in responsiveness. The idle time patterns show that Baseline Global maintains higher idle time (0.46), while MARL variants achieve lower idle times ranging from 0.30-0.35. The lower idle times in MARL configurations demonstrate improved vehicle utilization and more efficient fleet deployment compared to the centralized approach.

Pickup distance metrics show some variation across configurations, with Baseline Global achieving the shortest average pickup distance (7.21 km) and Baseline MARL maintaining similar efficiency (7.01 km). The advanced MARL variants show slightly increased pickup distances: LTL + TSR (7.85 km), Ctx1 (7.4 km), Ctx2 (7.3 km), and Full MARL (7.31 km). These marginal increases in pickup distances reflect the strategic trade-off where vehicles are willing to travel slightly further to secure higher-value matches and access high-demand areas, ultimately contributing to the substantial improvements in overall system throughput.

Fleet utilization gains are particularly notable, with Ctx1 achieving the highest utilization improvement (8.21%) and Full MARL maintaining strong performance (7.19%), both substantially outperforming the centralized baseline and demonstrating more efficient vehicle deployment across Toronto's extensive urban geography.

7.6.5 Managerial Insights

The proposed MARL framework delivers practical and measurable benefits across the urban freight logistics ecosystem. It supports operational optimization for platform, vehicles, and shippers.

For platforms, the decentralized MARL framework offers substantial operational and strategic advantages that directly translate to improved business performance. The transition from centralized to decentralized architecture provides immediate operational benefits. The

comparison between Baseline Global and Baseline MARL demonstrates 9.1% and 4.2% improvements in successful matches for Montréal and Toronto respectively. This shift significantly reduces dependence on centralized control infrastructure and associated computational overhead. The Full MARL configuration further enhances these benefits through dynamic task switching, achieving up to 25.9% improvement over traditional centralized systems. Critically, the framework's modular design enables risk-managed deployment strategies—platforms can implement basic decentralization first to capture immediate gains, then progressively integrate advanced components such as prioritization scoring and dispatching modules. This staged approach minimizes implementation risks while maximizing return on technology investment. The shared policy architecture ensures consistent decision-making across heterogeneous vehicle fleets without requiring individual vehicle training or customization, significantly reducing deployment complexity and maintenance costs. Furthermore, the inherent scalability of the decentralized approach positions platforms to handle fleet expansion and geographic growth without proportional increases in computational infrastructure or coordination overhead.

From the vehicle-level perspective, the framework delivers measurable efficiency gains and operational flexibility that directly benefit fleet operators and individual vehicle performance. The dynamic task switching architecture enables vehicles to autonomously adapt their operational focus based on real-time context, transitioning seamlessly between shipment acquisition, route optimization, and strategic repositioning without external coordination. This adaptability results in fleet utilization improvements of up to 8.21% compared to centralized approaches, with vehicles spending less time idle. While pickup distances may increase slightly due to broader geographic coverage and strategic positioning, the overall system throughput improvements of 15-25.9% substantially outweigh these marginal distance costs through increased earning opportunities. The context-aware prioritization system ensures that vehicles with higher operational relevance—those with greater capacity utilization or time urgency—receive priority consideration for valuable shipments, leading to more efficient resource allocation across the fleet. Additionally, the shared policy architecture eliminates the need for individual vehicle training or customization, allowing fleet operators to deploy consistent decision-making capabilities across diverse vehicle types and operational contexts while maintaining performance standards.

For shippers, the framework offers substantially improved service reliability. The dramatic reduction in fictitious matches (from 46-93 in centralized systems to 4-13 in Full MARL) indicates more dependable shipment acceptance and reduced service disruptions. Waiting time improvements of 60-77% over centralized systems demonstrate enhanced responsiveness, enabling shippers to achieve more predictable delivery schedules and improved customer

service. Additionally, the decentralized architecture reduces single points of failure inherent in centralized systems, providing shippers with more resilient logistics services that maintain performance even during peak demand periods or system disturbances.

Strategic implications suggest that logistics platforms should prioritize decentralization as a foundational capability, with the basic transition providing immediate operational benefits. The progressive enhancement through task-switching and prioritization mechanisms offers additional competitive advantages, particularly in dense urban environments where operational flexibility becomes critical. The framework's scalability characteristics make it particularly suitable for growing platforms that need to maintain service quality while expanding geographically or increasing fleet sizes. Its decentralized architecture inherently reduces computational complexity by parallelizing decision-making across vehicles, thereby avoiding bottlenecks associated with centralized coordination.

7.7 Conclusion

This paper proposed a context-aware, dynamic task-switching MARL framework for enhancing vehicle-level operational autonomy in urban freight systems. The core innovation lies in empowering each vehicle to adaptively switch among three dedicated RL structures—Matching, TSR, and Dispatching—based on interpretable local context variables such as load ratio, shipment count, and idle time. This design departs from fixed-task or centralized coordination models and enables real-time, decentralized decision-making at fleet scale.

Extensive simulations across Montréal and Toronto demonstrate that the proposed framework consistently improves system performance: up to 25.9% increase in successful matches, 8.21% gain in fleet utilization, and substantial reductions in shipment waiting time and fictitious assignments. The architecture also scales efficiently by combining H3 geospatial partitioning with modular pre-filtering (ShipScan, PADA), which constrain the vehicle's candidate action space based on local context, ensuring real-time decision-making computationally feasible at scale without loss of performance.

From an industrial perspective, the proposed approach provides concrete implementation pathways for logistics platforms seeking to transition from centralized to decentralized operational control. The modular structure enables incremental deployment, platforms can begin with basic vehicle autonomy (Baseline MARL) and progressively integrate task-switching and prioritization components, capturing performance gains at each stage. By distributing decision-making intelligence to individual vehicles rather than concentrating it in central control systems, the framework reduces operational latency, improves resource utilization, and

enhances service reliability.

Future work will explore three key directions. First, we aim to extend the framework to support heterogeneous reward objectives across fleets, allowing different vehicle operators to optimize for sustainability metrics, cost minimization, or service quality under varying operational policies. Second, we plan to incorporate domain adaptation mechanisms to enable cross-city policy transfer without retraining, leveraging learned behavioral patterns from one urban environment to accelerate deployment in new cities. Third, we will investigate federated multi-agent reinforcement learning architectures that enable privacy-preserving collaborative learning across competing logistics platforms while maintaining operational autonomy.

Acknowledgment

This work is supported by Shiphual Logistics and Mitacs through the Mitacs Accelerate program IT30680.

CHAPTER 8 GENERAL DISCUSSION

8.1 Integration of Research Contributions

This dissertation presents a comprehensive exploration of RL architectures tailored to the challenges of real-time decision-making in urban freight logistics. Through four articles, the research progressively advances from centralized single-agent models to decentralized multi-agent frameworks, each addressing core logistics functions with increasing complexity and adaptability. The model evolution follows a systematic progression where each subsequent framework directly addresses the limitations identified in the previous approach.

Article 1 [Chapter 4] introduces a Deep Q-Network for real-time shipment-vehicle matching, demonstrating the feasibility of learning-based approaches in discrete decision spaces.This foundational work validates the core concept of using RL for freight optimization but the single-agent approach cannot coordinate multiple operational tasks simultaneously. The centralized DQN architecture, while effective for individual matching decisions, lacks the modularity needed to handle the interdependent nature of pricing, matching, and dispatching decisions that characterize real-world freight platforms.

Article 2 [Chapter 5] directly addresses these limitations by proposing a Hierarchical Reinforcement Learning framework that enables joint optimization of matching and dispatching.The HRL framework introduces modular pre-filtering algorithms (PAMA and PADA) and H3 spatial partitioning to manage computational complexity. By introducing actor–critic agents and a centralized coordination hub, this architecture enables modular task decomposition and improves match success rates, vehicle utilization, and responsiveness. However, this approach still operates under centralized coordination and lacks dynamic pricing capabilities, limiting its ability to respond to market-driven demand fluctuations.

Article 3 [Chapter 6] extends the coordination scope by introducing a Nested Hierarchical Reinforcement Learning model that integrates pricing, matching, and dispatching across multiple spatial resolutions. This advancement was essential because a fixed pricing structure can not adapt to dynamic supply-demand conditions, leading to suboptimal resource allocation and missed revenue opportunities. The NHRL framework operates three specialized agents at different spatial resolutions—pricing and matching at fine-grained H3 level 7, and dispatching at coarser H3 level 6—enabling multi-scale decision coordination. While this approach achieves comprehensive optimization, the centralized coordination hub creates a bottleneck under high-frequency decision-making scenarios.

Article 4 [Chapter 7] addresses the challenge of Article 3 by presenting a decentralized Multi-Agent Reinforcement Learning model with dynamic task-switching capabilities, enabling vehicle-level autonomy and context-aware decision-making. The fully decentralized approach eliminates the coordination bottleneck by distributing decision-making to individual vehicles while maintaining system coherence through lightweight peer-to-peer communication. This model enables vehicles to switch autonomously between matching, routing, and dispatching tasks based on contextual cues, providing the operational flexibility and scalability needed for real-world deployment at urban scale.

Regarding the first research question (RQ1) on overcoming the limitations of fragmented approaches, the results from the NHRL (Article 3) and MARL (Article 4) frameworks confirm that integrated optimization significantly outperforms sequential methods. Specifically, the NHRL framework demonstrated that jointly optimizing pricing and matching increased revenue by 2.9% and 42.9% in Montréal and Toronto respectively, compared to disjointed baselines. Furthermore, the MARL framework proved that decentralized coordination could achieve a 25.9% increase in successful matches compared to centralized baselines, directly addressing the coordination bottlenecks identified in RQ1.

Together, these contributions form a cohesive framework for intelligent freight logistics, addressing scalability, responsiveness, and operational efficiency in dynamic urban environments while validating the dissertation's thesis that RL—when appropriately architected—can provide scalable, adaptive, and real-time control for complex freight logistics systems.

A summary of the architectural evolution across the four articles is illustrated in figure 8.1, highlighting the progression from centralized decision-making to fully decentralized, context-aware systems.

8.2 Comparative Insights Across Architectures

The evolution of reinforcement learning (RL) architectures across the four articles reveals key comparative insights into their performance, scalability, and operational applicability in urban freight logistics.

- **Centralized DQN (Article 1):** demonstrated strong performance in discrete shipment–vehicle matching tasks. It achieved a reward per match of -98.2 and minimized average pickup distance to 6.15 km using ReLU activation with 1024 neurons (Table 4.1). However, its centralized and single-task design limited adaptability and scalability in multi-agent environments.

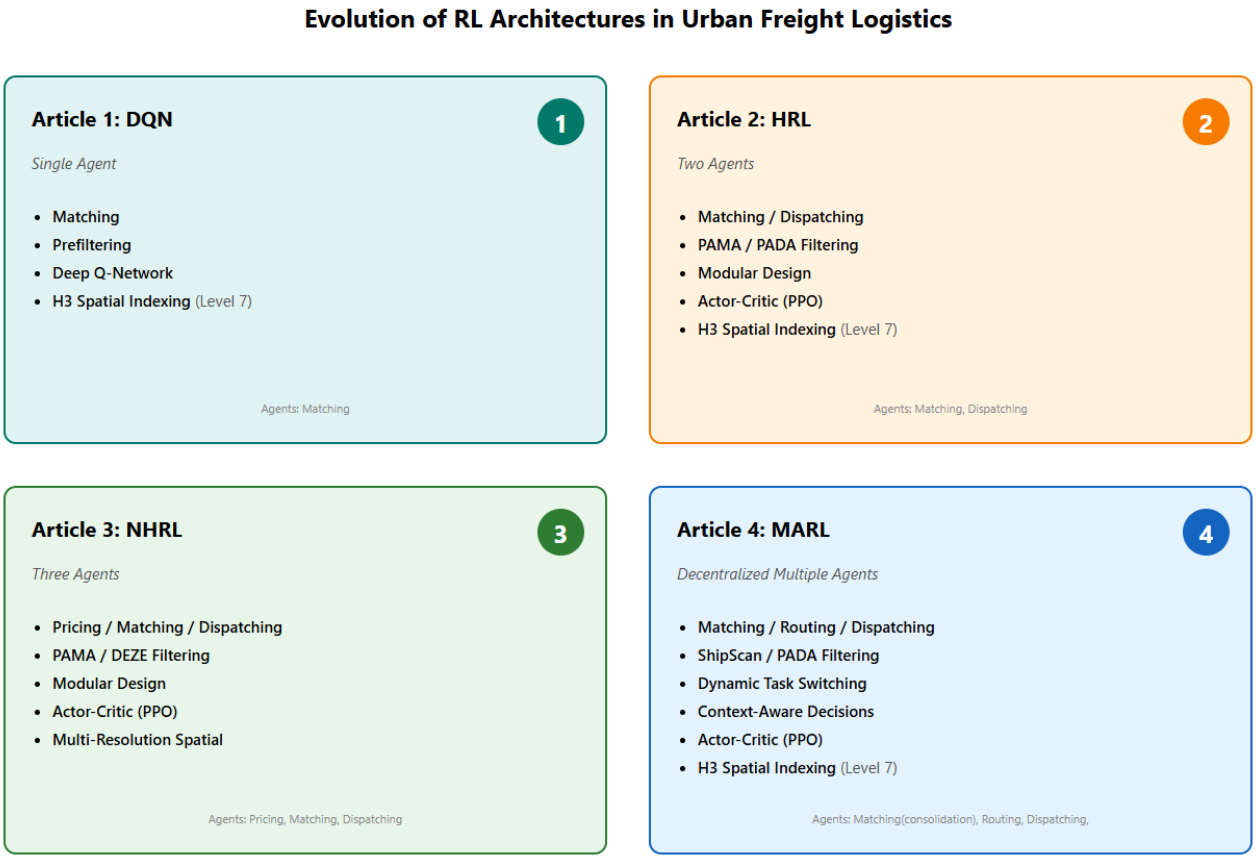

Figure 8.1 A summary of the architectural evolution across the four articles, highlighting the progression from centralized decision-making to fully decentralized, context-aware systems.

- **HRL (Article 2):** introduced modular coordination between matching and dispatching agents using actor–critic models and PPO. With the A512–C128 matching agent and A128–C256 dispatching agent configuration, the system achieved 3,757 successful matches and reduced vehicle idle time by 5.7% (from 0.157 to 0.148 timestamps), while maintaining an average empty mileage of 7.56 km (Tables 5.2-5.6). These results validated the benefits of hierarchical coordination and spatial pre-filtering (PAMA and PADA) for scalable decision-making.

  It is important to note that the DQN model (Article 1) served as a foundational proof-of-concept to validate the integration of H3 spatial indexing with RL. As such, the evaluation focused on internal architectural trade-offs rather than external baselines. Similarly, the HRL framework (Article 2) represents an intermediate architectural step; while it distributes tasks to specific agents, the control logic remains operationally centralized through the Coordination Hub. This centralization was a necessary design choice to validate the joint optimization logic before transitioning to the fully decen-

tralized autonomy presented in Article 4.

- **NHRL (Article 3):** enabled joint optimization of pricing, matching, and dispatching across multi-resolution spatial layers. In the Montréal case study, the three-agent model improved cumulative reward by 4.3% and revenue by 2.9% compared to the two-agent baseline (Table 6.1). It also increased successful matches by 3.5% and reduced shipment waiting time by 48.6% (from 0.74 to 0.38 timestamps). In Toronto, the three-agent model achieved a cumulative reward of 5,947,154 and revenue of $6,274,512, outperforming both fixed-price and two-agent configurations (Table 6.2).
- **Decentralized MARL (Article 4):** The MARL framework introduced vehicle-level autonomy with dynamic task switching among matching, routing, and dispatching. It achieved up to 25.9% improvement in successful matches and 8.21% increase in fleet utilization across Montréal and Toronto (Table 7.1). Shipment waiting times and pickup distances were reduced. These results highlight the superior scalability, responsiveness, and robustness of decentralized architectures in large-scale freight systems.

Across all models, the use of H3 hierarchical spatial indexing consistently outperformed clustering-based methods by providing uniform spatial granularity and improved neighbor relations, contributing to better match accuracy and reduced computational overhead.

8.3 Methodological Innovations and Design Principles

This dissertation introduces several methodological innovations that advance the state of the art in freight logistics optimization:

8.3.1 Technical Innovations

To support the successful implementation of the architectures described, this dissertation introduces a set of technical innovations that enhance performance, scalability, and operational relevance. These innovations include both foundational methods and advanced agent behaviors that collectively form the backbone of the proposed frameworks.

- **H3 Spatial Partitioning** provides uniform, multi-resolution geographic coverage that enables scalable and consistent spatial reasoning across agents. Unlike traditional clustering methods that produce irregular boundaries and inconsistent coverage, H3's hexagonal structure ensures uniform spatial representation and maintains consistent neighbor relationships across different resolution levels. This innovation allows agents

to reason about geographic proximity and spatial relationships in a standardized manner, facilitating coordination between agents operating at different spatial scales while maintaining computational efficiency.

- **Modular Pre-Filtering Algorithms** including PAMA, PADA, DEZE, and ShipScan significantly reduce action space complexity, enhance real-time feasibility, and improve learning efficiency. These algorithms intelligently filter potential actions before the RL decision process, eliminating infeasible or suboptimal options based on domain-specific constraints such as vehicle capacity, geographic proximity, and temporal windows. By reducing the action space from potentially thousands of options to manageable subsets, these pre-filtering mechanisms enable real-time decision-making while improving learning convergence and reducing computational overhead.

- **Dynamic Task Switching** enables vehicles in the MARL framework to autonomously switch between operational tasks based on interpretable context variables such as current load status, geographic location, and market conditions. This innovation moves beyond static role assignments by allowing agents to adapt their behavioral strategies dynamically, switching between matching-focused, routing-focused, or dispatching-focused modes based on current operational context. The interpretable nature of the switching criteria ensures transparency and enables operational managers to understand and influence agent behavior.

- **Multi-Resolution Coordination** allows NHRL agents to operate at different spatial granularities, enabling fine-grained pricing and matching decisions at high resolution while maintaining coarse-grained dispatching strategies for broader geographic coordination. This innovation addresses the inherent multi-scale nature of urban logistics, where local decisions must be coordinated with regional strategies. The hierarchical coordination mechanism ensures that local optimizations remain aligned with system-wide objectives while maintaining computational tractability.

- **Context-Aware Prioritization** enables vehicles to compute prioritization scores based on capacity utilization, load ratio, and service time requirements, facilitating emergent coordination without centralized control. This mechanism allows agents to self-organize and coordinate their actions through distributed decision-making, reducing communication overhead and improving system resilience. The prioritization scores provide a common coordination language that enables agents to make locally optimal decisions that contribute to global system performance.

8.3.2 Architectural Design Principles

A key methodological takeaway is the modular and layered design philosophy adopted throughout the dissertation. Several architectural design principles emerged:

- **Hierarchical Decomposition** proves essential for scalability and interpretability in large urban freight systems by breaking complex decision-making processes into manageable layers with clear responsibilities. This principle enables system designers to address different aspects of the logistics problem at appropriate levels of abstraction, from strategic decisions to matching and operational dispatching, while maintaining clear interfaces between decision layers.
- **Agent Modularity** through separate agents for pricing, matching, and dispatching allows for targeted improvements and easier integration into legacy systems. This design principle enables incremental deployment and system evolution, allowing operators to upgrade specific functionality without disrupting entire systems. The modular approach also facilitates debugging, performance optimization, and system maintenance by isolating different functional components.
- **Dynamic Task Switching** enhances agent autonomy and aligns well with real-world operational fluidity where vehicles and drivers must adapt to changing conditions throughout their operational shifts. This principle recognizes that static role assignments are inefficient in dynamic environments and enables agents to maximize their contribution to system performance by adapting their behavior based on current operational needs and opportunities.

Pre-filtering algorithms, separate training pipelines, and consistent simulation environments were essential to maintaining scalability while improving model interpretability. These insights contribute to methodological best practices for future RL applications in logistics and smart cities.

8.4 Theoretical and Practical Implications

8.4.1 Theoretical Contributions

This dissertation contributes several novel theoretical frameworks that extend the boundaries of reinforcement learning (RL) applications in urban freight logistics.

The concept of dynamic task switching introduced in Article 4 represents a significant advancement in MARL. Unlike traditional MARL models with fixed agent roles, this framework

enables vehicle-level agents to autonomously reconfigure their operational behavior—switching between matching, routing, and dispatching—based on interpretable context variables such as load ratio, shipment count, and idle time. This dynamic structure-switching mechanism contributes a new layer of adaptability and responsiveness to MARL theory, supporting emergent coordination without centralized control.

The multi-resolution coordination mechanism developed in Article 3 offers a theoretical innovation in spatial reasoning within RL. By integrating Uber's H3 hierarchical indexing system, the NHRL framework enables agents to operate across nested spatial layers—fine-grained zones for pricing and matching, and coarser zones for dispatching. This approach formalizes a scalable method for spatial abstraction in RL, allowing for simultaneous optimization across tactical and strategic decision layers.

The use of modular pre-filtering algorithms (PAMA, PADA, DEZE, ShipScan) across all architectures introduces a generalizable technique for action space reduction. These algorithms decouple spatial preprocessing from policy learning, improving sample efficiency and enabling real-time feasibility in high-dimensional logistics environments. Their modularity supports integration with various RL paradigms, including value-based, actor–critic, and multi-agent systems.

Collectively, these contributions advance RL theory by introducing scalable, interpretable, and context-aware mechanisms for decision-making in complex, multi-agent, and spatially distributed environments. They lay the groundwork for future research in adaptive logistics systems, decentralized coordination, and hierarchical learning architectures.

8.4.2 Practical Implications

Beyond its theoretical contributions, this dissertation offers a wide range of practical benefits for various stakeholders within the smart urban freight ecosystem. The proposed models can inform platform operators, drivers, and shippers, while contributing to broader goals of sustainability and urban efficiency.

For freight platforms, the frameworks offer enhanced revenue generation capabilities through dynamic pricing mechanisms that intelligently align demand fluctuations. The improved match success rates and reduced idle time directly translate to operational cost reductions and improved service quality metrics that are critical for platform competitiveness. The modular architectures facilitate phased deployment strategies, allowing platforms to integrate RL components incrementally with existing systems, reducing implementation risks and enabling gradual adaptation to new decision-making paradigms.

For drivers, the frameworks provide substantial benefits through increased vehicle utilization and fairer workload distribution mechanisms enabled by context-aware dispatching algorithms. The proactive dispatching capabilities that anticipate future demand patterns help drivers optimize their positioning and reduce idle periods, directly improving earnings potential and operational efficiency. The dynamic task-switching capabilities in the MARL framework allow drivers to adapt their operational strategies based on real-time conditions, enhancing their autonomy while maintaining system coordination. These improvements address critical driver satisfaction factors that influence platform participation and service quality.

For shippers, the frameworks deliver improved service quality through faster matching times, reduced waiting periods, and enhanced delivery reliability. The dynamic pricing mechanisms ensure fair and responsive cost structures that reflect actual service conditions rather than static pricing models, providing better value alignment between service quality and cost. The multi-objective optimization inherent in the frameworks balances service speed with cost efficiency, addressing the diverse preferences of different shipper segments while maintaining overall system performance.

The environmental implications of the proposed frameworks are particularly significant for sustainable urban logistics development. The reduction in empty mileage through improved matching algorithms and decreased idle time through proactive dispatching directly contributes to lower fuel consumption and reduced emissions per delivery. The optimization of vehicle utilization and route efficiency supports broader urban sustainability goals by environmental impact from freight operations. These environmental benefits position the frameworks as comprehensive solutions that address both economic and sustainability objectives in urban freight systems.

The Montréal and Toronto case studies validate the models' potential for urban-scale implementation, while the agent-based approach offers flexibility for varying market conditions and geographic scales.

8.4.3 Organizational and Technical Requirements

Implementing these RL frameworks requires specific enterprise infrastructure. Technically, companies need real-time data pipelines (e.g., Apache Kafka) to stream GPS vehicle tracking, dynamic order entry, and traffic data. High-throughput API gateways are essential to facilitate the micro-transactions and communication between decentralized vehicle agents. From a data perspective, the system requires clean, historical datasets of demand patterns for pre-training (offline learning) and real-time state observability (location, capacity, status)

for online execution. Organizationally, the shift to the MARL framework (Article 4) requires a transition from command-and-control dispatching to exception-management, where human dispatchers monitor system health and intervene only during critical failures rather than assigning individual trips.

8.5 Limitations and Assumptions

Despite its contributions and promising results, the research faces several limitations.

8.5.1 Behavioral Realism and Rationality Assumptions

A primary limitation of this research is the assumption of rational agent behavior. In the simulated environment, agents—both vehicles and platform operators—strictly maximize the defined reward functions. However, in real-world industrial contexts, stakeholders often exhibit bounded rationality or heterogeneous preferences. For instance, drivers may reject profitable routes due to personal safety concerns, fatigue, or preference for specific geographic zones, regardless of the algorithmic recommendation. While the current frameworks do not explicitly model this irrationality, the modular design allows for future integration of "human-in-the-loop" parameters, such as acceptance probability models or preference-based weighing factors, to better reflect stochastic human behaviors and improve the system's robustness against unpredictable stakeholder actions.

8.5.2 Data and Environment Limitations

All evaluations in this dissertation were conducted in simulated environments, which, while realistic and domain-informed, cannot fully replicate the variability, noise, and behavioral nuances of real-world freight operations. The simulation environments, though carefully designed to mirror urban logistics dynamics, inevitably simplify complex interactions between drivers, shippers, traffic conditions, and regulatory frameworks that characterize live freight platforms. This limitation is particularly significant given that real-world deployment success often depends on handling edge cases and unexpected scenarios that are difficult to anticipate in simulation. Furthermore, the feedback loops between agent decisions and environment responses in simulation may not accurately reflect the dynamic adaptations that occur in real operational settings.

The reliance on synthetic datasets, while necessary for controlled experimentation, introduces additional constraints on the research findings. These datasets, despite being domain-informed and statistically representative, lack the intricate patterns of real-world demand

fluctuations, seasonal variations, and operational complexities that emerge in live logistics platforms. Real freight operations involve complex customer behaviors, varying service requirements, and dynamic pricing sensitivities that synthetic data cannot fully capture. The absence of actual driver decision-making patterns, route preferences, and acceptance behaviors limits the models' ability to accurately predict performance in operational environments where human factors play crucial roles.

8.5.3 Model Assumptions

The proposed models incorporate several simplifying assumptions regarding vehicle and agent behavior that may not hold consistently in real-world applications. Driver behavior is modeled in a simplified manner that does not account for individual preferences, acceptance decision patterns, or behavioral heterogeneity that significantly influence operational outcomes. Real drivers exhibit complex decision-making processes influenced by personal circumstances, earnings targets, geographic preferences, and risk tolerance that are not captured in the current frameworks. These behavioral nuances can substantially impact system performance, particularly in scenarios requiring high driver acceptance rates or specific geographic coverage.

The exclusion of external factors represents a significant limitation in the current research. Real-world freight operations are constantly affected by weather conditions, traffic incidents, construction activities, regulatory changes, and special events that can dramatically alter demand patterns and operational constraints. The models do not account for these disruptions, which can cause substantial deviations from expected performance. Seasonal variations, economic fluctuations, and evolving urban policies also influence logistics operations in ways that are not captured in the current frameworks, potentially limiting their robustness and adaptability in dynamic urban environments.

Regarding the operational assumptions, the framework relies on specific abstractions to maintain computational tractability. First, regarding the sequential decision flow (Chapter 6), the model assumes near-instantaneous information synchronization between the Coordination Hub and agents (e.g., pricing updates are immediately visible to the matching agent). While this simplifies real-world communication latency, it is consistent with modern high-frequency trading and logistics architectures. Second, regarding traffic dynamics, the simulation assumes constant vehicle speeds and standardized loading times. Validating these physical assumptions would require coupling the RL framework with a microscopic traffic simulator (e.g., SUMO), which—while valuable for future deployment testing—would introduce excessive computational noise during the policy learning phase. Therefore, these assumptions

serve to isolate and validate the decision-making logic of the agents independent of IT latency or traffic variance.

8.5.4 Technical Limitations

The computational requirements of the proposed RL frameworks, particularly in multi-agent configurations, present significant practical challenges for real-world deployment. Training time and memory usage can be substantial, especially for the NHRL and MARL architectures that involve multiple agents operating across different spatial and temporal resolutions. These computational demands may pose barriers to real-time retraining, model adaptation, and scaling to larger fleet sizes or geographic areas. The resource-intensive nature of these approaches could limit their accessibility to smaller freight platforms or require significant infrastructure investments that may not be feasible for all operators.

Hyperparameter sensitivity represents another critical technical limitation, particularly for actor-critic variants used in the hierarchical and multi-agent frameworks. The performance of RL models can be sensitive to learning rates, network architectures, and exploration parameters, requiring extensive tuning for optimal performance. Slight deviations in these parameters can lead to training instability, convergence failures, or substantial performance degradation. This sensitivity creates challenges for deployment in diverse operational contexts where optimal hyperparameters may vary based on local conditions, fleet characteristics, and demand patterns.

Addressing these limitations is essential for transitioning from simulation to real-world deployment and represents a key focus for future research and deployment planning.

8.6 Future Research Directions

Building on the current work and addressing the limitations identified in Section 8.5, future research can explore several promising directions. Table 8.1 provides a systematic overview of how the proposed research directions address the specific limitations discussed in the previous section.

8.6.1 Data and Environment Enhancement

Building directly on the simulation and synthetic data limitations identified in section 8.5, incorporating operational data from existing freight platforms represents a critical step toward practical deployment. Current simulation environments, while comprehensive, lack the

Table 8.1 Mapping of Current Limitations to Future Research Directions

Limitation Category	Specific Limitation	Addressing Research Direction	Section
Data & Environment	Simulation vs. reality gap	Real-world data integration	8.6.1
	Synthetic dataset constraints	Pilot real-world implementations	8.6.1
	Missing behavioral nuances	Advanced driver behavior modeling	8.6.2
Model Assumptions	Simplified driver behavior	Incorporating uncertainty & probabilistic models	8.6.2
	Exclusion of external factors	Uncertainty-aware agents	8.6.2
	Missing disruption modeling	Robust optimization techniques	8.6.2
Technical Limitations	Computational costs	Federated learning approaches	8.6.3
	Training scalability	Transfer learning & model adaptation	8.6.3
	Hyperparameter sensitivity	Meta-learning & automated tuning	8.6.3

nuanced behavioral patterns. This integration should include driver acceptance rates, route preferences, and decision-making patterns that significantly influence system performance but are currently simplified in the proposed models.

Addressing the simulation-reality gap requires transitioning from controlled environments to live implementation through pilot programs. Future research should focus on creating hybrid systems that combine RL-based decision-making with rule-based fallbacks, ensuring system reliability during the transition period. Pilot implementations in controlled urban environments could provide valuable insights into practical deployment challenges and help refine the models based on real operational feedback.

8.6.2 Generalizability and Transferability

To further address the generalizability of the proposed frameworks, future work should validate these models on urban topologies significantly different from the North American grid-like structures of Montréal and Toronto. For instance, applying the H3-based indexing to European cities with irregular road networks and strict zoning regulations would test the spa-

tial adaptability of the pre-filtering modules. Additionally, the frameworks could be adapted for different logistics sectors, such as long-haul trucking or hyper-local food delivery. This would require recalibrating the reward functions (e.g., weighting speed over cost for food delivery) and adjusting the time-step granularity, but the core architectural principles of hierarchical decomposition and dynamic task switching remain transferable.

8.7 Model Robustness and Behavioral Realism

Directly addressing the exclusion of external factors, future research should focus on developing uncertainty-aware agents that can explicitly model and adapt to weather disruptions, traffic incidents, and regulatory changes. Real-world freight operations face constant volatility that current models do not capture. This could involve integrating probabilistic models or risk-sensitive RL methods that maintain performance under various uncertainty scenarios while ensuring reliable service delivery.

The simplified vehicle behavior assumptions motivate the development of more sophisticated driver models that account for individual preferences, acceptance decision patterns, and behavioral heterogeneity. This research direction should incorporate behavioral economics principles, preference learning, and personalized decision models that capture the complex factors influencing driver choices in real operational environments.

Combining reinforcement learning with graph neural networks could leverage the inherent network structure of urban freight systems, enabling agents to better understand spatial relationships and traffic patterns beyond current approaches. Similarly, integrating other machine learning approaches could address the behavioral simplifications while maintaining computational tractability.

8.7.1 Technical Scalability and Deployment

Directly addressing computational cost limitations, federated learning could enable multiple freight platforms to collaboratively improve their models while maintaining data privacy and competitive advantages. These approaches would distribute computational loads across platforms while preserving operational autonomy, making the frameworks more accessible to smaller operators and reducing infrastructure requirements.

The hyperparameter sensitivity challenges highlight the importance of transfer learning research. Investigating how pretrained models can be efficiently adapted to new cities would reduce training time, computational costs, and data requirements for deployment. This should explore domain adaptation techniques, meta-learning approaches, and modular trans-

fer strategies that preserve learned patterns while adapting to local conditions.

Addressing the substantial training and memory requirements, future research should focus on model compression, distributed training architectures, and efficient online learning algorithms that enable real-time adaptation without prohibitive computational costs.

8.7.2 System Extensions (Beyond Current Limitations)

Extending frameworks to include drones, autonomous vehicles, and hybrid fleets would require developing new coordination mechanisms while maintaining demonstrated scalability and adaptability.

Embedding carbon costs, energy use, and environmental impact into reward structures would align frameworks with green logistics goals, positioning them as comprehensive solutions for responsible urban freight logistics

These directions aim to bridge the gap between academic models and practical, responsible deployment in live logistics platforms while furthering the applicability and impact of RL in urban freight logistics.

CHAPTER 9 CONCLUSION

This dissertation presented a comprehensive investigation into the design and application of reinforcement learning architectures for addressing real-time pricing, matching, and dispatching challenges in smart urban freight logistics. Through a structured sequence of four articles, each with a distinct methodological focus and level of system complexity, this research has advanced the understanding of how adaptive and decentralized decision-making models can improve the efficiency and responsiveness of urban freight operations.

The study began by introducing a fundamental RL-based framework for real-time dynamic pricing, demonstrating how competitive agent-based mechanisms can promote fair and efficient freight assignments (Article 1). Building upon this foundation, the second article, Hierarchical Reinforcement Learning, extended the scope to platform-centric matching and dispatching using centralized multi-agent RL strategies. These models successfully addressed system-level objectives such as reduced waiting time, optimized vehicle utilization, and cost minimization, reinforcing the potential of learning-based orchestration.

Articles 3 and 4 further deepened the investigation by addressing the complexities of joint decision-making and decentralized operational autonomy in urban freight systems. Article 3 introduced a Nested Hierarchical Reinforcement Learning framework that optimized pricing, matching, and dispatching within a unified, multi-level architecture. This framework captured the interdependencies between strategic and operational decisions while leveraging H3-based spatial partitioning to handle geographic variability. In contrast, Article 4 proposed a decentralized multi-agent reinforcement learning framework that granted vehicle-level autonomy with dynamic task-switching capabilities. This approach allowed individual agents to adaptively alternate between matching, routing, and dispatching tasks based on local context and system priorities, thereby enhancing flexibility and scalability. Collectively, these two articles advanced both the architectural and algorithmic sophistication of RL frameworks and demonstrated how they can enable more robust and responsive urban freight systems.

This dissertation contributes both theoretically and practically to the fields of intelligent transportation systems and urban logistics. Theoretically, it provides novel formulations and algorithmic insights into reinforcement learning under multi-agent, dynamic, and decentralized conditions. Practically, it proposes operational strategies that are adaptable to real-world logistics platforms, especially those emerging in the context of smart cities and on-demand freight services. The progression of this research—from foundational methodologies to system-wide impacts—is summarized in Figure 9.1. This contribution hierarchy illustrates

how core technical innovations such as H3 spatial indexing and modular pre-filtering algorithms support increasingly complex architectures, culminating in real-world improvements in operational efficiency and scalability."

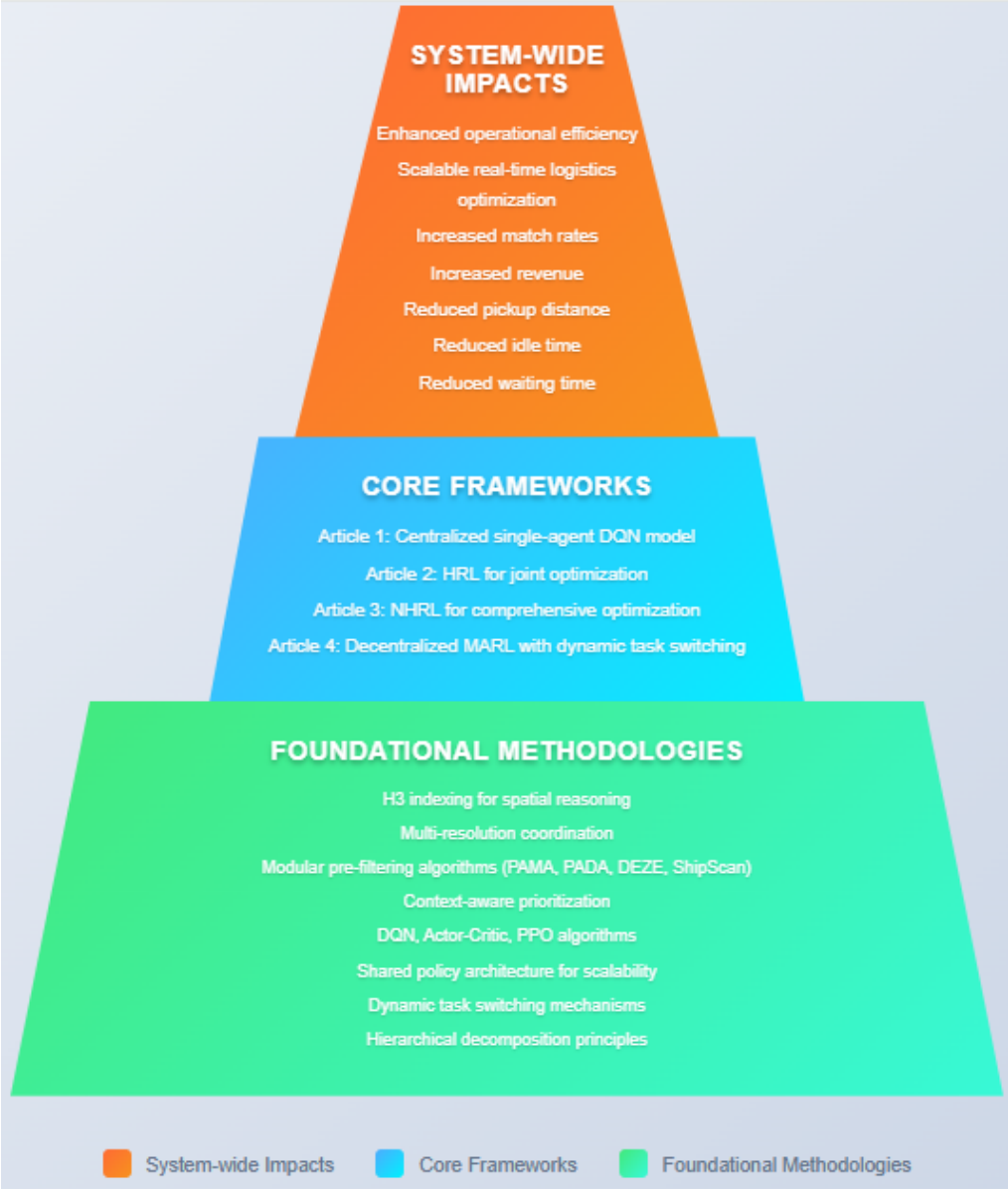

Figure 9.1 Impact Pyramid summarizing the thesis contributions from foundational methodologies to system-wide impacts.

9.1 Academic Implications

This dissertation contributes to the understanding of reinforcement learning (RL) in complex, multi-agent, and spatially distributed systems. It presents a structured evolution from

centralized DQN to decentralized multi-agent architectures, showing how modularity and task decomposition enhance scalability and learning efficiency in logistics. These hierarchical decomposition principles offer a blueprint for managing complexity in large-scale decision-making, where monolithic approaches fall short.

The research advances multi-level decision-making under uncertainty through hierarchical and nested RL architectures, enabling coordinated agent behavior across spatial and temporal scales. A key innovation is dynamic task-switching in MARL, allowing agents to adapt roles based on context variables, fostering decentralized coordination. This has broader relevance in robotics, autonomous systems, and distributed computing.

The multi-resolution coordination framework using H3 spatial indexing enables agents to reason across nested spatial layers, offering scalable spatial abstraction that balances computational efficiency with decision quality. This addresses limitations in traditional grid-based methods and informs GIS, urban computing, and spatial machine learning.

Modular pre-filtering algorithms (PAMA, PADA, DEZE, ShipScan) reduce action space complexity by decoupling spatial preprocessing from policy learning, improving sample efficiency and enabling real-time decision-making. These methods are applicable to resource allocation, scheduling, and optimization in high-dimensional RL environments.

Finally, comparative evaluations of PPO, TRPO, and DDPG across freight scenarios provide empirical guidance for algorithm selection in dynamic, multi-agent logistics systems. These findings help bridge theory and practice, offering insights into trade-offs between learning stability, efficiency, and performance.

9.2 Industrial Implications

The proposed frameworks offer practical value for logistics platforms, urban planners, and freight operators:

The architectures improve match rates, reduce vehicle idle time, and minimize shipment delays, directly supporting key performance indicators in urban freight operations. These improvements translate into cost savings, better resource utilization, and enhanced customer satisfaction.By reducing empty mileage and enabling dynamic resource allocation, the frameworks contribute to environmental sustainability goals. Future extensions could incorporate carbon-aware reward functions and policy constraints, aligning logistics optimization with regulatory and societal objectives.

The modular design supports phased integration into existing logistics platforms. Pricing, matching, and dispatching modules can be deployed independently or jointly, allowing for

flexible adoption based on organizational readiness and infrastructure. The use of interpretable context-aware task switching and pre-filtering algorithms enables real-time decision-making at vehicle level, reducing reliance on centralized control and improving system scalability in high-demand environments.

9.3 Limitations and Future Work

While this research makes significant contributions to reinforcement learning applications in urban freight logistics, several limitations must be acknowledged, offering promising directions for future work.

9.3.1 Limitations

Data and Environment Limitations: The simulation environment, while structurally realistic, remains a controlled approximation of live urban freight operations. The reliance on synthetic datasets—though statistically representative and domain-informed—limits the ability to capture nuanced patterns such as seasonal demand fluctuations, behavioral variability, and emergent operational disruptions. This constraint may reduce the ecological validity of the findings and affect the transferability of learned policies to real-world logistics platforms.

Model Assumptions: The behavioral models used for drivers and shippers are simplified and do not account for individual preferences, acceptance decisions, or behavioral heterogeneity. This abstraction may impact decision efficacy, particularly in scenarios requiring high driver acceptance rates or geographic coverage. It also limits the generalizability of results to live deployment scenarios where human factors—such as earnings targets, risk tolerance, and geographic preferences—play a critical role in system performance.

Technical Limitations: Although scalability was addressed through hierarchical and decentralized designs, computational efficiency remains a challenge for real-time deployment, especially under high-volume shipment scenarios where decision latency directly affects service quality. Further, some models—particularly those in nested or multi-agent settings—require extensive hyperparameter tuning and longer convergence times. The interpretability of agent behavior and the robustness of learned policies under perturbation also remain open challenges.

9.3.2 Future Work

Building upon these insights, several directions can enhance the practical utility and robustness of the proposed RL frameworks:

- **Enhanced Data and Environment Modeling:** Future work should integrate real-time freight transaction data, traffic conditions, and demand forecasts to enrich the state space and enable online learning. This would improve the realism and adaptability of the models, allowing them to respond dynamically to evolving urban conditions. Incorporating simulation-to-reality transfer techniques could further bridge the gap between simulated performance and real-world applicability, accelerating deployment in operational platforms.
- **Adaptive and Robust Learning Models:** Exploring RL under partial observability, robust policy learning, and meta-learning could significantly improve agent performance in environments characterized by uncertainty, sparse feedback, and non-stationarity. These approaches would enhance the resilience of decision-making systems in the face of disruptions, data noise, or unexpected behavioral shifts.
- **Human Behavior Modeling:** Incorporating explicit models of carrier or driver behavior—including preferences, rejection probabilities, and compliance variability—would enhance behavioral realism, particularly in decentralized settings. This would support the development of human-in-the-loop architectures and improve the alignment between algorithmic decisions and real-world operational constraints.
- **Scalable, Lightweight Architectures:** Investigating federated learning and edge-compatible RL models could enable scalable deployment across thousands of vehicles with limited computational resources. This would facilitate privacy-preserving, distributed learning in commercial logistics fleets, supporting real-time decision-making without centralized bottlenecks.
- **Multi-Stakeholder Interaction:** Future frameworks should incorporate the perspectives of shippers, infrastructure planners, and regulators into the decision-making process. This would lead to policy-aware logistics optimization with system-wide coordination, enabling trade-offs between efficiency, equity, and sustainability. Such integration opens the door to new research on incentive design, regulatory compliance, and collaborative governance in smart freight systems.
- **System Extensions:** Potential extensions include multimodal logistics integration (e.g., combining trucks with rail or drones), carbon-aware optimization to support

sustainability goals, and interaction with smart infrastructure elements such as adaptive traffic signals or dynamic tolling systems. These enhancements would expand the scope and societal impact of RL-based logistics platforms.

In conclusion, while this dissertation addresses core challenges in urban freight logistics through advanced RL architectures, extending these models to address behavioral realism, computational tractability, and richer environmental dynamics will be crucial to their successful adoption in real-world smart city contexts. The systematic progression from theoretical foundations to practical applications demonstrated in this work provides a roadmap for continued advancement in intelligent logistics systems and their integration into the broader ecosystem of smart urban infrastructure.

REFERENCES

[1] World Economic Forum. (2020) The future of the last mile ecosystem: Transition roadmaps for public- and private-sector players. Geneva, Switzerland. White Paper. [Online]. Available: https://www.weforum.org/reports/the-future-of-the-last-mile-ecosystem/

[2] McKinsey & Company. (2021) The endgame for postal networks: How to win in the age of e-commerce. Industry Report. [Online]. Available: https://www.mckinsey.com/industries/travel-logistics-and-infrastructure/our-insights/the-endgame-for-postal-networks-how-to-win-in-the-age-of-e-commerce

[3] Y. Liu, Q. Luo, R. Gopalakrishnan, and S. Samaranayake, "A framework for the joint optimization of assignment and pricing in mobility-on-demand systems with shared rides," *arXiv preprint arXiv:2112.14297*, 2021.

[4] B. Turan, R. Pedarsani, and M. Alizadeh, "Dynamic pricing and fleet management for electric autonomous mobility on demand systems," *Transportation Research Part C: Emerging Technologies*, vol. 121, p. 102829, 2020.

[5] Z. He, L. Chen, and B. Liu, "Application of integrating reinforcement learning and intelligent scheduling in logistics distribution," *Intelligent Decision Technologies*, vol. 18, no. 1, pp. 57–74, 2024.

[6] T. G. Crainic, N. Ricciardi, and G. Storchi, "Models for evaluating and planning city logistics systems," *Transportation science*, vol. 43, no. 4, pp. 432–454, 2009.

[7] E. Taniguchi, R. G. Thompson, and A. G. Qureshi, "Modelling city logistics using recent innovative technologies," *Transportation Research Procedia*, vol. 46, pp. 3–12, 2020, the 11th International Conference on City Logistics, Dubrovnik, Croatia, 12th - 14th June 2019.

[8] J. Allen, M. Browne, A. Woodburn, and J. Leonardi, "The role of urban consolidation centres in sustainable freight transport," *Transport reviews*, vol. 32, no. 4, pp. 473–490, 2012.

[9] M. K. Chen and M. Sheldon, "Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform." *Ec*, vol. 16, p. 455, 2016.

[10] D. Wang, Q. Wang, Y. Yin, and T. C. E. Cheng, "Optimization of ride-sharing with passenger transfer via deep reinforcement learning," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 172, 2023.

[11] M. Xu *et al.*, "Multi-agent reinforcement learning to unify order-matching and vehicle-repositioning in ride-hailing services," *International Journal of Geographical Information Science*, vol. 37, no. 2, pp. 380–402, 2023.

[12] Y. Liu *et al.*, "Deep dispatching: A deep reinforcement learning approach for vehicle dispatching on online ride-hailing platform," *Transportation Research Part E: Logistics and Transportation Review*, vol. 161, May 2022.

[13] M. Taiebat, E. Amini, and M. Xu, "Sharing behavior in ride-hailing trips: A machine learning inference approach," *Transportation Research Part D: Transport and Environment*, vol. 103, p. 103166, 2022.

[14] Y.-C. B. Zeinab Shahbazi, "Blockchain and machine learning for intelligent multiple factor-based ride-hailing services," *Computers, Materials & Continua*, vol. 70, no. 3, pp. 4429–4446, 2022. [Online]. Available: http://www.techscience.com/cmc/v70n3/44946

[15] C. Li, D. Parker, and Q. Hao, "A value-based dynamic learning approach for vehicle dispatch in ride-sharing," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, vol. 2022-Octob, 2022, pp. 11 388–11 395.

[16] X. Li, J. Gao, C. Wang, X. Huang, and Y. Nie, "Ride-sharing matching under travel time uncertainty through a data-driven robust optimization approach," in *IEEE Conference on Intelligent Transportation Systems*, 2021, pp. 3420–3425.

[17] D. Kalyanmoy, "Multi-objective optimization using evolutionary algorithms wiley," *J. & Sons 497p*, 2001.

[18] S. Shu and et al., "Modeling freight-sharing platform operations for optimal compensation strategy using markov decision processes," in *IEEE Conference on Intelligent Transportation Systems*, 2022, pp. 1006–1011.

[19] J. Zhang, "Pickup and delivery planning for the crowdsourced freight delivery routing problem," *PloS one*, vol. 20, no. 2, p. e0318432, 2025.

[20] A. H. Sadeghi, Z. Sun, A. Sahebi-Fakhrabad, H. Arzani, and R. Handfield, "A mixed-integer linear formulation for a dynamic modified stochastic p-median problem in a competitive supply chain network design," *Logistics*, vol. 7, no. 1, p. 14, 2023.

[21] D. F. Demirel, A. Alev, B. B. Erturan, E. Bağrıyanık, E. Akkaya, and ğ. Z. Gŭndoğdu, "A mixed-integer programming model for optimizing the distribution network of a packaging company," *Journal of Transportation and Logistics*, vol. 10, no. 1, pp. 18–33, 2025.

[22] Anonymous, "Multi-objective mixed-integer linear programming for dynamic fleet scheduling and route optimization," *Sustainability*, vol. 17, no. 10, p. 4707, 2023.

[23] R. A. Russell, "Hybrid heuristics for the vehicle routing problem with time windows," *Transportation science*, vol. 29, no. 2, pp. 156–166, 1995.

[24] L. Li, T. Pantelidis, J. Y. Chow, and S. E. Jabari, "A real-time dispatching strategy for shared automated electric vehicles with performance guarantees," *Transportation Research Part E: Logistics and Transportation Review*, vol. 152, p. 102392, 2021.

[25] A. H. Sadeghi, Z. Sun, A. Sahebi-Fakhrabad, H. Arzani, and R. Handfield, "A mixed-integer linear formulation for a dynamic modified stochastic p-median problem in a competitive supply chain network design," *Logistics*, vol. 7, no. 1, p. 14, 2023.

[26] D. F. Demirel, A. Alev, B. B. Erturan, E. Bağrıyanık, E. Akkaya, and Ş. Z. Gündoğdu, "A mixed-integer programming model for optimizing the distribution network of a packaging company," *Journal of Transportation and Logistics*, vol. 10, no. 1, pp. 18–33, 2025.

[27] A. Polimeni, A. Donato, and O. M. Belcore, "Urban freight distribution with electric vehicles: comparing some solution procedures," *Frontiers in Future Transportation*, vol. 5, p. 1491799, 2024.

[28] K.-W. Jie, S.-Y. Liu, and X.-J. Sun, "A hybrid algorithm for time-dependent vehicle routing problem with soft time windows and stochastic factors," *Engineering Applications of Artificial Intelligence*, vol. 109, p. 104606, 2022.

[29] G. Li and J. Li, "An improved tabu search algorithm for the stochastic vehicle routing problem with soft time windows," *IEEE Access*, vol. 8, pp. 158 115–158 124, 2020.

[30] S. Ghaemifard and A. Ghannadiasl, "A comparison of metaheuristic algorithms for structural optimization: Performance and efficiency analysis," *Advances in Civil Engineering*, vol. 2024, no. 1, p. 2054173, 2024.

[31] A. Walter, K. Ahsan, and S. Rahman, "Application of artificial intelligence in demand planning for supply chains: a systematic literature review," *The International Journal of Logistics Management*, 2025.

[32] M. Babai, M. Arampatzis, M. Hasni, F. Lolli, and A. Tsadiras, "On the use of machine learning in supply chain management: a systematic review," *IMA Journal of Management Mathematics*, vol. 36, no. 1, pp. 21–49, 2025.

[33] M. Su, S.-H. Bae, and K.-s. Park, "Port congestion and container freight rate dynamics: forecasting with an rbf neural network," *Frontiers in Marine Science*, vol. 12, p. 1545471, 2025.

[34] N. Servos, X. Liu, M. Teucke, and M. Freitag, "Travel time prediction in a multimodal freight transport relation using machine learning algorithms," *Logistics*, vol. 4, no. 1, p. 1, 2019.

[35] A. S. Paramita and T. Hariguna, "Comparison of k-means and dbscan algorithms for customer segmentation in e-commerce," *Journal of Digital Market and Digital Currency*, vol. 1, no. 1, pp. 43–62, 2024.

[36] D. Madhusoodanan, R. Vismaya, H. Santhosh, and S. Jayan, "Comparative study of clustering algorithms for customer segmentation," in *International Conference on Data Science and Applications*. Springer, 2025, pp. 13–24.

[37] P. Eichenseer, L. Hans, and H. Winkler, "A data-driven machine learning model for forecasting delivery positions in logistics for workforce planning," *Supply Chain Analytics*, vol. 9, p. 100099, 2025.

[38] H. Tran-Dang, J.-W. Kim, J.-M. Lee, and D.-S. Kim, "Shaping the future of logistics: Data-driven technology approaches and strategic management," *IETE Technical Review*, vol. 42, no. 1, pp. 44–79, 2025.

[39] Y. Yan, A. H. Chow, C. P. Ho, Y.-H. Kuo, Q. Wu, and C. Ying, "Reinforcement learning for logistics and supply chain management: Methodologies, state of the art, and future opportunities," *Transportation Research Part E: Logistics and Transportation Review*, vol. 162, p. 102712, 2022.

[40] M. Haliem, G. Mani, V. Aggarwal, and B. Bhargava, "A distributed model-free ride-sharing algorithm with pricing using deep reinforcement learning," 2020.

[41] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[42] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," https://arxiv.org/abs/1707.06347, 2017, arXiv preprint arXiv:1707.06347.

[43] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015, pp. 1889–1897.

[44] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," https://arxiv.org/abs/1509.02971, 2015, arXiv preprint arXiv:1509.02971.

[45] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," *Advances in neural information processing systems*, vol. 29, 2016.

[46] W. Guo, B. Atasoy, W. B. van Blokland, and R. R. Negenborn, "A dynamic shipment matching problem in hinterland synchromodal transportation," *Decis. Support Syst.*, vol. 134, 2020.

[47] X. Li, J. Gao, C. Wang, X. Huang, and Y. Nie, "Order dispatching in ride-sharing platform under travel time uncertainty: A data-driven robust optimization approach," in *Proceedings of ICAS 2021 - IEEE International Conference on Autonomous Systems*, Aug. 2021.

[48] J. Chen, A. K. Umrawal, T. Lan, and V. Aggarwal, "Deepfreight: A model-free deep-reinforcement-learning-based algorithm for multi-transfer freight delivery," in *Proceedings International Conference on Automated Planning and Scheduling, ICAPS*, 2021, pp. 510–518.

[49] C. Chen, F. Yao, D. Mo, J. Zhu, and X. M. Chen, "Spatial-temporal pricing for ride-sourcing platform with reinforcement learning," *Transportation Research Part C: Emerging Technologies*, vol. 130, p. 103272, 2021.

[50] T. Wu, "Automated pricing agents in the on-demand economy," Master's thesis, EECS Department, University of California, Berkeley, May 2016. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-57.html

[51] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[52] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.

[53] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[54] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[55] Uber Engineering, "H3: Uber's Hexagonal Hierarchical Spatial Index," https://eng.uber.com/h3, 2018, accessed: 2025-07-05.

[56] G. Qin, Q. Luo, Y. Yin, J. Sun, and J. Ye, "Optimizing matching time intervals for ride-hailing services using reinforcement learning," *Transp. Res. Part C Emerg. Technol.*, vol. 129, 2021.

[57] J. H. Hong and X. Liu, "The optimal pricing for green ride services in the ride-sharing economy," *Transportation Research Part D: Transport and Environment*, vol. 104, p. 103205, 2022.

[58] A. Y. Ng, S. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, no. 2, 2000, p. 2.

[59] A. Singh, A. O. Al-Abbasi, and V. Aggarwal, "A distributed model-free algorithm for multi-hop ride-sharing using deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8595–8605, 2022.

[60] C. Ma, A. Li, Y. Du, H. Dong, and Y. Yang, "Efficient and scalable reinforcement learning for large-scale network control," *Nature Machine Intelligence*, vol. 6, no. 9, pp. 1006–1020, 2024.

[61] M. Abbasi, A. Consilvio, and D. Giglio, "A multi-objective optimization approach for disruption management in an intermodal freight transport network," *Transportation Research Procedia*, vol. 78, pp. 410–417, 2024.

[62] T. Cui, Y. Shi, J. Wang, R. Ding, J. Li, and K. Li, "Practice of an improved many-objective route optimization algorithm in a multimodal transportation case under uncertain demand," *Complex & Intelligent Systems*, vol. 11, no. 2, pp. 1–22, 2025.

[63] Z. He, M. Zhang, Q. Chen, S. Chen, and N. Pan, "Optimization of heterogeneous vehicle logistics scheduling with multi-objectives and multi-centers," *Scientific Reports*, vol. 13, no. 1, p. 14169, 2023.

[64] G. Zhou, W. Tian, R. Buyya, R. Xue, and L. Song, "Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions," *Artificial Intelligence Review*, vol. 57, no. 5, p. 124, 2024.

[65] J. Gao, X. Li, C. Wang, and X. Huang, "Bm-ddpg: An integrated dispatching framework for ride-hailing systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11 666–11 676, Aug. 2022.

[66] G. Guo and Y. Xu, "A deep reinforcement learning approach to ride-sharing vehicle dispatching in autonomous mobility-on-demand systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 1, pp. 128–140, 2020.

[67] K. Manchella, M. Haliem, V. Aggarwal, and B. Bhargava, "Passgoodpool: Joint passengers and goods fleet management with reinforcement learning aided pricing, matching, and route planning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3866–3877, 2022.

[68] J. Huang, Y. Luo, S. Fu, M. Xu, and B. Hu, "Pride: Privacy-preserving online ride hailing matching system with prediction," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 7413–7425, Aug. 2021.

[69] K. Manchella, A. K. Umrawal, and V. Aggarwal, "Flexpool: A distributed model-free deep reinforcement learning algorithm for joint passengers and goods transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2035–2047, 2021.

[70] H. S. Narman, H. Malik, and G. Yatnalkar, "An enhanced ride sharing model based on human characteristics, machine learning recommender system, and user threshold time," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 13–26, 2021.

[71] R. Tian, C. Wang, Z. Ma, Y. Liu, and S. Gao, "Research on vehicle-cargo matching algorithm based on improved dynamic bayesian network," *Comput. Ind. Eng.*, vol. 168, 2022.

[72] J. Song, Y. J. Cho, M. H. Kang, and K. Y. Hwang, "An application of reinforced learning-based dynamic pricing for improvement of ridesharing platform

service in seoul," *Electronics*, vol. 9, no. 11, p. 1818, 2020. [Online]. Available: https://doi.org/10.3390/electronics9111818

[73] E. Mazumdar, L. J. Ratliff, T. Fiez, and S. S. Sastry, "Gradient-based inverse risk-sensitive reinforcement learning," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 5796–5801.

[74] H. Chen, Y. Jiao, Z. Qin, X. Tang, H. Li, B. An, H. Zhu, and J. Ye, "Inbede: Integrating contextual bandit with td learning for joint pricing and dispatch of ride-hailing platforms," in *2019 IEEE International Conference on Data Mining (ICDM)*, 2019, pp. 61–70.

[75] C. Yan, H. Zhu, N. Korolko, and D. Woodard, "Dynamic pricing and matching in ride-hailing platforms," *Naval Research Logistics (NRL)*, vol. 67, no. 8, pp. 705–724, 2020.

[76] J. Li, Y. Zheng, B. Dai, and J. Yu, "Implications of matching and pricing strategies for multiple-delivery-points service in a freight o2o platform," *Transportation Research Part E: Logistics and Transportation Review*, vol. 136, p. 101871, 2020.

[77] G. Guo and M. Kang, "Rebalancing and charging scheduling with price incentives for car sharing systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18 592–18 602, 2022.

[78] B. Turan and M. Alizadeh, "Competition in electric autonomous mobility-on-demand systems," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 1, pp. 295–307, 2021.

[79] D. Fan and K. Heminthavonf, "Heavyweight of the canadian economy, road transportation," Research Publications, Library of Parliament, Canada, 2022.

[80] G. Homsi, B. Gendron, and S. D. Jena, "Two-stage stochastic one-to-many driver matching for ridesharing," *Networks*, vol. 82, no. 4, pp. 414–436, 2023.

[81] Y. L. Hanifa, Y. Satria, and H. Burhan, "Application of parallel tabu search in solving ride-sharing problem with hov lanes," in *Journal of Physics: Conference Series*. Institute of Physics Publishing, 2019.

[82] G. P. Yatnalkar, "A machine learning recommender model for ride sharing based on rider characteristics and user threshold time," 2019. [Online]. Available: https://mds.marshall.edu/etdCapstones.1259

[83] X. Azagirre and et al., "A better match for drivers and riders: Reinforcement learning at lyft," *INFORMS Journal on Applied Analytics*, vol. 54, no. 1, pp. 71–83, 2024.

[84] M. Haliem, G. Mani, V. Aggarwal, and B. Bhargava, "A distributed model-free ride-sharing approach for joint matching, pricing, and dispatching using deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7931–7942, 2021.

[85] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[86] Zion Market Research, "Freight forwarding market size, share report, analysis, trends, growth 2032," https://www.zionmarketresearch.com/report/freight-forwarding-market, 2024.

[87] M. Furuhata, M. Dessouky, F. Ordóñez, M.-J. Brunet, X. Wang, and S. Koenig, "Ridesharing: The state-of-the-art and future directions," *Transportation Research Part B: Methodological*, vol. 57, pp. 28–46, 2013.

[88] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *European Journal of Operational Research*, vol. 223, no. 2, pp. 295–303, 2012.

[89] A. Shiri, A. Yarahmadi, S. Keivanpour, and A. Lamghari, "Learning-based matching algorithm for smart freight platform and sustainability assessment in montreal," in *Proceedings of the 1st International Conference on Smart Mobility and Vehicle Electrification*, 2023.

[90] P. Santi and C. Ratti, "Care to share? using gps fleet data to assess taxi sharing," 2015, geo-Intelligence and Visualization through Big Data Trends.

[91] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement learning and dynamic programming using function approximators*. CRC press, 2017.

[92] X. Tang *et al.*, "A deep value-network based approach for multi-driver order dispatching," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 1780–1790, http://arxiv.org/abs/2106.04493.

[93] J. Ke *et al.*, "Pricing and equilibrium in on-demand ride-pooling markets," *Transportation Research Part B: Methodological*, vol. 139, pp. 411–431, 2020.

[94] Z. Xu *et al.*, "Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2018, pp. 905–913.

[95] Y. Wang *et al.*, "Adaptive dynamic bipartite graph matching: A reinforcement learning approach," in *Proceedings - International Conference on Data Engineering*, 2019, pp. 1478–1489.

[96] J. Wen, J. Zhao, and P. Jaillet, "Rebalancing shared mobility-on-demand systems: A reinforcement learning approach," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 220–225.

[97] N. Garg and S. Ranu, "Route recommendations for idle taxi drivers: Find me the shortest route to a customer!" in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2018, pp. 1425–1434.

[98] H. Rong *et al.*, "The rich and the poor: A markov decision process approach to optimizing taxi driver revenue efficiency," in *International Conference on Information and Knowledge Management, Proceedings*. Association for Computing Machinery, 2016, pp. 2329–2334.

[99] T. Verma, P. Varakantham, S. Kraus, and H. C. Lau, "Augmenting decisions of taxi drivers through reinforcement learning for improving revenues," in *Proceedings International Conference on Automated Planning and Scheduling, ICAPS*, 2017, pp. 409–417, www.aaai.org.

[100] Y. Jiao *et al.*, "A deep value-based policy search approach for real-world vehicle repositioning on mobility-on-demand platforms," Deep RL Workshop, NeurIPS, 2020, https://www.researchgate.net/publication/346032249.

[101] M. Han, P. Senellart, S. Bressan, and H. Wu, "Routing an autonomous taxi with reinforcement learning," in *International Conference on Information and Knowledge Management, Proceedings*. Association for Computing Machinery, 2016, pp. 2421–2424.

[102] Z. Shou and X. Di, "Reward design for driver repositioning using multi-agent reinforcement learning," *Transportation Research Part C: Emerging Technologies*, vol. 119, 2020.

[103] J. Fan, Z. Wang, Y. Xie, and Z. Yang, "A theoretical analysis of deep q-learning," 2019, http://arxiv.org/abs/1901.00137.

[104] K. Schröder, A. Kastius, and R. Schlosser, "Welcome to the jungle: A conceptual comparison of reinforcement learning algorithms," in *International Conference on Operations Research and Enterprise Systems*. Science and Technology Publications, Lda, 2023, pp. 143–150.

[105] C. Mao, Y. Liu, and Z. J. M. Shen, "Dispatch of autonomous vehicles for taxi services: A deep reinforcement learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 115, 2020.

[106] J. Huang *et al.*, "Deep reinforcement learning-based trajectory pricing on ride-hailing platforms," *ACM Transactions on Intelligent Systems and Technology*, vol. 13, no. 3, 2022.

[107] A. Shiri, A. YarAhmadi, S. Keivanpour, and A. Lamghari, "Real-time rl-based matching with h3 geohash partitioning in smart freight platform," in *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*, Washington, DC, USA, 2024, pp. 1–7.

[108] E. Caspersen, "Freight trip generation and consumer preferences for reducing externalities from last mile deliveries," 2021, nMBU Open Research Archive. [Online]. Available: https://hdl.handle.net/11250/2737293

[109] D. Hardy *et al.*, "Freight and land use travel demand evaluation," United States. Department of Transportation. Federal Highway Administration . . . , Tech. Rep., 2018.

[110] A. Lightstone, T. Belony, and J.-F. Cappuccilli, "Understanding goods movement in canada: Trends and best practices," www.tac-atc.ca, 2021.

[111] Grand View Research, "Digital freight brokerage market size, share & trends analysis report by mode of transport (roadways, seaways, airways, railways), by end-use (manufacturing, retail & e-commerce, healthcare), by region, and segment forecasts, 2024 - 2032," 2023.

[112] MarketsandMarkets, "Digital freight forwarding market by mode of transport, end-use industry, and region - global forecast to 2029," 2024.

[113] J. Allen, M. Browne, A. Woodburn, and J. Leonardi, "The role of urban consolidation centres in sustainable freight transport," *Transport Reviews*, 2012.

[114] OECD, "The economic impact of transport delay and congestion in urban freight systems," International Transport Forum, Tech. Rep., 2019.

[115] L. Schultz and V. Sokolov, "Deep reinforcement learning for dynamic urban transportation problems," https://arxiv.org/abs/1806.05310, 2018.

[116] K. Bimpikis, O. Candogan, and D. Saban, "Spatial pricing in ride-sharing networks," *Operations Research*, vol. 67, no. 3, pp. 744–769, 2019.

[117] M. Meskar, S. Aslani, and M. Modarres, "Spatio-temporal pricing algorithm for ride-hailing platforms where drivers can decline ride requests," *Transportation Research Part C: Emerging Technologies*, vol. 153, p. 104200, 2023.

[118] A. Shiri, A. Yarahmadi, and S. Keivanpour, "Real-time matching and dispatching for urban freight transportation: A hierarchical reinforcement learning through actor-critic and h3 spatial partitioning," *IEEE Transactions on Intelligent Transportation Systems*, 2025, submitted for publication.

[119] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[120] J. Sun, H. Jin, Z. Yang, and L. Su, "Optimizing long-term efficiency and fairness in ride-hailing under budget constraint via joint order dispatching and driver repositioning," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[121] Z. Yang, H. Jin, G. Fan, M. Lu, Y. Liu, X. Yue, H. Pan *et al.*, "Rethinking order dispatching in online ride-hailing platforms," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 3863–3873.

[122] J. Li, Y. Zheng, B. Dai, and J. Yu, "Implications of matching and pricing strategies for multiple-delivery-points service in a freight o2o platform," *Transportation Research Part E: Logistics and Transportation Review*, vol. 136, p. 101871, 2020.

[123] Y. Feng, R. Niazadeh, and A. Saberi, "Two-stage stochastic matching and pricing with applications to ride hailing," *Operations Research*, vol. 72, no. 4, pp. 1574–1594, 2024.

[124] K. VinayKumar and N. C. Santosh, "Data analysis and fair price prediction using machine learning algorithms," *Journal of Computer Allied Intelligence*, vol. 2, no. 1, pp. 26–45, 2024.

[125] B. Berger, H. Ma, D. Parkes, and S. Sekar, "Optimal subscription pricing design for ridesharing platforms," 2023, working paper.

[126] Y. Wang, H. Sun, Y. Lv, X. Chang, and J. Wu, "Reinforcement learning-based order-dispatching optimization in the ride-sourcing service," *Computers & Industrial Engineering*, vol. 192, p. 110221, 2024.

[127] B. Shi, Y. Lu, and Z. Cao, "A dynamic region-division based pricing strategy in ride-hailing," *Applied Intelligence*, vol. 54, no. 22, pp. 11 267–11 280, 2024.

[128] S. Heydari and E. Akhondzadeh Noughabi, "Dynamic pricing in ride-hailing intelligent transportation systems by using deep reinforcement learning," https://ssrn.com/abstract=xxxxxx, 2024, sSRN preprint.

[129] Z. Zhou, C. Roncoli, and C. Sipetas, "Optimal matching for coexisting ride-hailing and ridesharing services considering pricing fairness and user choices," *Transportation Research Part C: Emerging Technologies*, vol. 156, p. 104326, 2023.

[130] B. Lartey, W. Bedada, X. Yan, A. Homaifar, A. Karimoddini, and E. Tunstel, "An efficient profit-aware scalable vehicle dispatch framework for on-demand ridesharing," *IEEE Transactions on Industrial Cyber-Physical Systems*, 2024.

[131] S. Ge, X. Zhou, T. Qiu, G. Wu, and X. Wang, "Elasticshare: Ridesharing order dispatching with dynamic supply-demand distribution," in *2023 IEEE/ACM International Symposium on Quality of Service (IWQoS)*, 2023, pp. 1–10.

[132] Y. Guo, W. Li, L. Xiao, A. Choudhary, and H. Allaoui, "Enhancing efficiency and interpretability: A multi-objective dispatching strategy for autonomous service vehicles in ride-hailing," *Computers & Industrial Engineering*, vol. 194, p. 110385, 2024.

[133] D. Peng, G. Wu, and K. Boriboonsomsin, "Bi-objective battery electric truck dispatching problem with backhauls and time windows," *Transportation Research Record*, 2024.

[134] Y. Wang, J. Wu, H. Sun, Y. Lv, and J. Zhang, "Promoting collaborative dispatching in the ride-sourcing market with a third-party integrator," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[135] Z. T. Qin, H. Zhu, and J. Ye, "Reinforcement learning for ridesharing: An extended survey," *Transportation Research Part C: Emerging Technologies*, vol. 144, p. 103852, 2022.

[136] K. Manchella, M. Haliem, V. Aggarwal, and B. Bhargava, "Passgoodpool: Joint passengers and goods fleet management with reinforcement learning aided pricing, matching,

and route planning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3866–3877, 2021.

[137] J. Ke, X. Qin, H. Yang, Z. Zheng, Z. Zhu, and J. Ye, "Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network," *Transportation Research Part C: Emerging Technologies*, vol. 122, p. 102858, Jan. 2021.

[138] G. A. Rummery and M. Niranjan, "On-line q-learning using connectionist systems," University of Cambridge, Department of Engineering, Technical Report CUED/F-INFENG/TR 166, 1994.

[139] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[140] A. Shiri *et al.*, "Nested hierarchical reinforcement learning for real-time joint pricing, matching, and dispatching in urban freight transportation," *International Journal of Production Economics*, 2025, to appear.

[141] N. Mardešić, T. Erdelić, T. Carić, and M. Đurasević, "Review of stochastic dynamic vehicle routing in the evolving urban logistics environment," *Mathematics*, vol. 12, no. 1, p. 28, 2023.

[142] F. D. Hildebrandt, B. W. Thomas, and M. W. Ulmer, "Opportunities for reinforcement learning in stochastic dynamic vehicle routing," *Computers & operations research*, vol. 150, p. 106071, 2023.

[143] A. Kumar, Y. Vorobeychik, and W. Yeoh, "Improving zonal fairness while maintaining efficiency in rideshare matching," in *Proceedings of the CEUR Workshop*, 2021.

[144] S. Guo, H. Li, and Y. Zhang, "Dynamic dispatching and repositioning in ride-hailing platforms: A deep reinforcement learning approach," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[145] M. Padidar, *A Hybrid Modeling Approach to Joint Matching and Pricing in an Intelligent Freight Transportation Platform.* Ecole Polytechnique, Montreal (Canada), 2022.

APPENDIX A SUMMARY OF REVIEWED WORKS

Table A.1 Summary of Reviewed Works on Reinforcement Learning in Ride-Hailing and Freight Contexts

Ref.	**Methodology**	**Objectives**	**Data/Scale**	**Performance Metrics**	**Limitations**
[40]	Deep RL (DQN)	Pricing, Dispatching	NYC taxi data	Profit, wait time, acceptance rate	Limited to ride-sharing, no freight focus
[10]	Multi-agent DRL	Matching, Pricing	Real taxi data	Revenue, wait time, success rate	Scalability not demonstrated
[11]	Context-aware MARL	Demand-aware dispatching	NYC taxi data	Revenue, fulfillment rate	Limited geographical scope
[65]	Multi-agent RL	Revenue optimization	Real-world datasets	Platform revenue, utilization	OD-pair decomposition may miss global effects
[59]	Centralized RL	Dynamic pricing	Real platform data	Revenue, efficiency	Centralized approach limits scalability
[66]	Deep Q-learning	Pricing optimization	Simulated environment	Revenue, demand alignment	Simulation-based validation only
[67]	RL with MDP	Adaptive pricing	Urban mobility data	Revenue, fulfillment rates	Limited to pricing decisions
[12]	Actor-critic RL	Joint pricing/matching	Real ride-sourcing data	Revenue, efficiency, satisfaction	No freight-specific considerations
[13]	MARL with GNN	Spatiotemporal dispatching	NYC data	Matching rates, idle time	Limited to dispatching only
[15]	Cooperative MARL	Large-scale dispatching	Real-world datasets	Fulfillment rate, wait time	No pricing integration
[16]	MARL with CTDE	Real-time ridesharing	Large-scale datasets	System efficiency, utilization	Passenger focus, not freight
[14]	Spatiotemporal RL	Joint matching/pricing	Real ride-hailing data	Profit, matching rate	Grid-based spatial representation
[47]	MADRL	Joint pricing/dispatching	Real ride-hailing data	Revenue, fulfillment, utilization	Limited hierarchical structure
[68]	RL with Dec-POMDP	Dispatching/ repositioning	Real ride-hailing data	Order fulfillment, wait time	Two-stage architecture may be suboptimal
[56]	Spatio-temporal DRL	Joint pricing/dispatching	Real ride-sourcing data	Profit, fulfillment rate	Grid-based spatial model
[69]	Model-free RL (DQN)	Vehicle repositioning	Berlin car-sharing data	Fleet utilization, shortages	Single objective focus
[70]	RL with GNN	Vehicle repositioning	Real mobility data	Service fulfillment, utilization	Repositioning only
[137]	Multi-agent RL	Joint repositioning/pricing	Real ride-hailing data	Fulfillment rates, revenue	Regional decomposition approach
[48]	Two-level RL	Joint pricing/fleet management	Real-world data	Revenue, service rates	Hierarchical coordination challenges
[46]	Decentralized MARL	Joint pricing/matching	Real-world data	Fulfillment rates, revenue	Limited spatial coordination

Continued on next page

Table A.1 Summary of Reviewed Works on Reinforcement Learning in Ride-Hailing and Freight Contexts

Ref.	Methodology	Objectives	Data/Scale	Performance Metrics	Limitations
[18]	MARL with spatiotemporal	Joint pricing/matching	Real ride-sourcing data	Revenue, wait time, fulfillment	Partial observability challenges
[71]	MARL with graph attention	Joint repositioning/pricing	Real ride-hailing data	Service rate, profitability	Complexity of graph attention scaling
[50]	MARL with CTDE	Joint pricing/dispatching	Real ride-hailing data	Profit, service rate, efficiency	Dual-policy complexity
[49]	Hierarchical MARL	Fleet management/pricing	Real ride-hailing data	Revenue, satisfaction, utilization	Two-stage hierarchy limitations
[72]	Graph-based MARL	Joint pricing/matching	Real ride-hailing data	Revenue, service rate, balance	Graph construction complexity
[73]	Graph-structured MARL	Joint pricing/dispatching	Urban ride-hailing data	Revenue, fulfillment, efficiency	Regional agent limitations
[74]	Graph-based MADRL	Joint matching/pricing	Synthetic and real data	Profit, match rate, fairness	Graph representation complexity
[4]	Hierarchical MARL	Dynamic matching/pricing	Real-world data	Profit, match success, stability	Two-layer hierarchy constraints
[75]	Hierarchical RL	Joint pricing/dispatching	Real ride-hailing data	Revenue, fulfillment, balance	Macro-micro coordination challenges
[145]	Spatio-temporal HRL	Joint pricing/matching	Real ride-sourcing data	Service rates, profit, satisfaction	Spatial-temporal hierarchy complexity
[76]	Hierarchical MARL	Joint matching/pricing	Real-world data	Revenue, matching success, balance	Multi-layered coordination overhead
[57]	Spatio-temporal graph MARL	Joint pricing/dispatching	Urban ride-hailing data	Fulfillment rates, revenue	Dual-agent coordination complexity
[77]	MARL with attention	Joint pricing/dispatching	Real ride-hailing data	Revenue, acceptance, efficiency	Attention mechanism scalability
[58]	Hierarchical graph RL	Joint pricing/dispatching	Real-world datasets	Revenue, service rate, utilization	Two-level hierarchy limitations
[78]	Cooperative hierarchical MARL	Joint pricing/dispatching	Real-world datasets	Profit, fulfillment, efficiency	Hierarchical cooperation complexity