Intelligent Multi-Agent Reinforcement Learning Architectures for Coordinated Autonomous Logistics and Real-Time Network Optimization

Kannan Avalurpet Loganathan[1],Arunraju Chinnaraju[2]

[1]Independent Researcher, California, USA.

[2]Doctorate in Business Administration, Westcliff University, USA.

DOI: https://doi.org/10.47772/IJRISS.2026.10100211

Received: 15 January 2026; Accepted: 21 January 2026; Published: 31 January 2026

ABSTRACT

The complexity and variability of large-scale global logistics networks demonstrate the inherent limits to the potential of both optimized centralization and automated rules based on the present state of knowledge. Logistics systems today function in decentralized, stochastic and partially observable environments, comprising autonomous, however, dependent upon each other, entities such as trucks, warehouses and transportation hubs. This paper provides an overall theoretical and architectural base for the application of Intelligent Multi-Agent Reinforcement Learning (MARL) as a platform for the development of autonomous logistics and the dynamic optimization of logistics networks in real time. Logistics operations are defined as decentralized decision-making processes and stochastic games, which allow agents to develop adaptive coordination policies, through decentralized execution of policies developed during centralized training. An additional layered MARL structure is described to separate perception, coordination, decision-making and optimization, to ensure the ability to scale, modularize and optimize logistics networks in a stable manner. Graph-based communication, message-passing mechanisms and bandwidth-efficient policy-sharing are used to coordinate the actions among agents; whereas, the stability of learning is addressed using value decomposition, structured credit assignment and reward shaping. Advanced learning strategies including actor-critic methods, proximal policy optimization, meta-learning and continual learning are analyzed for multi-objective optimization of logistics networks over time, cost, energy and carbon footprint constraints. In addition, this paper demonstrates how the proposed framework can be integrated with high-fidelity simulation and multiagent digital twins to safely train and validate policies under realistic disruptions, along with cloud-edge infrastructure and distributed data pipelines to deploy these policies in real time. Additionally, the paper addresses the issues of interoperability between the proposed MARL framework and enterprise supply chain systems, as well as the governance issues related to transparency, accountability and regulatory compliance. Finally, the paper outlines future research directions, combining MARL with graph neural networks, generative models and predictive digital twins to enable scalable, resilient and self-optimizing logistics ecosystems.

Keywords: Multi-Agent Reinforcement Learning, Autonomous Logistics Systems, Real-Time Network Optimization, Decentralized Control, Centralized Training Decentralized Execution, Graph-Based Coordination, Digital Twin Logistics, Federated Learning, Cloud-Edge AI Infrastructure

Introduction to Intelligent Logistics Systems

From Deterministic Logistics to Autonomous, Learning Driven Systems Traditional logistics has evolved and developed as deterministic or weakly stochastic pipelines, which are governed centrally through a planning process and controlled hierarchically. Static representations of logistics networks were used in classical logistics optimization, where nodes represent warehouses or hubs, edges represent the transportation route, and optimization problems were solved off-line using Operations Research techniques (e.g., linear programming, mixed integer programming, network flow optimization, and vehicle routing heuristics) (Dantzig and Ramser, 1959; Clarke and Wright, 1964; Laporte, 1992). However, these models have several assumptions: (i) the demand is stationary; (ii) the transit times are predictable; and (iii) the states of the system are fully observable. Therefore, tractability was ensured but the response to real world variability was severely limited (Dror and Trudeau, 1989). As logistics networks have become larger and have gone global, the traditional centralized model has become less reliable. The dimensionality of the decision space increases exponentially with the number of vehicles, routes, warehouse locations and time slots (Laporte, 2007). For computational reasons, simplification was necessary to decouple optimization results from the operational realities. Furthermore, the time lag between the planning process and the actual execution resulted in outdated solutions when facing real-time disruptions like traffic jams, labor shortages, port delays, weather

anomalies or sudden spikes in demand (Winkelhaus and Grosse, 2020; El Hamdi et al., 2022). The development towards intelligent logistics systems presents a radical new way of thinking about how optimization and control are integrated into logistics operations (Winkelhaus and Grosse, 2020). Instead of treating intelligence as an additional planning layer, intelligence is embedded into the operational units themselves. Vehicles, robots, warehouses and hubs are seen as autonomous decision making entities that interact with their environment in real time, perceive local states and adjust their actions through learning (Corvello et al., 2025). This development is part of the broader trend towards cyber-physical systems and distributed control, in which centralized command is substituted by decentralized autonomy, achieved through feedback and communication (Bernstein et al., 2002). Modern logistics systems are therefore not static networks that are optimized periodically but dynamic systems that evolve over time. Optimization is transformed from the search for a globally optimal solution to the learning of policies, and resilience is generated not by redundancy alone but by the ability of agents to dynamically reorganize their behavior under uncertain conditions (Ning et al., 2024).

Logistics as Distributed Multi-Agent Decision System

At its most basic level, modern logistics networks have all the characteristic properties of distributed multiagent systems. Each of the many and diverse entities (vehicles, warehouses, hubs) that make up the logistics network operate simultaneously with the goal of achieving their own local goals, they have only partial visibility of the system, and they have limited capacity to communicate with other entities (Beynier, 2013; Bernstein et al., 2002). Vehicles optimize routes and travel speed subject to uncertainty in the traffic. Warehouses allocate labor, storage and picking resources subject to fluctuations in the volume of orders received. Hubs manage the synchronization of the incoming and outgoing material flows while minimizing congestion and meeting service-level agreements. Entities are interconnected and mutually dependent because they share common resources and/or are temporally linked. A decision regarding the routing of a vehicle affects the pattern of congestion experienced by all the other vehicles. The replenishment of stock at a node will affect the flow of goods through the network. Thus, decisions taken locally have a nonlocal influence, and create complex feedback loops that cannot be efficiently managed by separate optimization (Corvello et al., 2025). Multi-Agent Reinforcement Learning is a systematic approach to modeling such systems (Ning et al., 2024; Zhu et al., 2024). In MARL, each entity is defined by a policy that takes an observation and returns an action, and learning occurs through interaction with the same environment. The environment describes both the physical laws governing the system and the cumulative effect of the actions of all the other entities. More formally, distributed logistics systems can be viewed as decentralized partially observable Markov decision processes or stochastic games in which the global state is not directly visible to any single entity and the rewards reflect both local and systemic performance (Beynier, 2013; Bernstein et al., 2002). This formulation allows logistics to be formulated as a learning problem rather than an optimization problem. Entities continually update their policies through experience, and thus, the coordination strategies are discovered internally rather than being determined prior to the start of operation (Ning et al., 2024). Additionally, MARL permits the treatment of different learning dynamics for the various types of entities present in a logistics network (e.g., vehicles, warehouses, and hubs), and yet, achieves collectively optimal outcomes (Zhu et al., 2024).

Real-Time Network Optimization as Continuous Control

Real-Time Network Optimization is a departure from traditional planning-based optimization towards continuous, feedback driven control (Corvello et al., 2025). Decisions are made at each decision point, based upon the current perception and the learned expectations of the future dynamics of the system. The goal of RTNO is not to determine a globally optimal plan, but to maintain near-optimal performance under nonstationary conditions. Viewed from a control theory perspective, logistics networks behave as high dimensional, nonlinear stochastic systems with delayed feedback and coupled state transitions (Bernstein et al., 2002). Centralized controllers suffer from severe observability and latency constraints, while decentralized controllers may experience instability resulting from uncoordinated actions. To achieve effective real-time optimization, a balance between autonomy and coordination is required. MARL meets this requirement by incorporating coordination into the learning objective itself (Zhu et al., 2024). Agents learn policies that take into account the implications of their actions on the future state of the network and the behaviors of other agents. The overall objectives of the system (throughput, service reliability, etc.) are encoded in the reward structure of the system, and credit assignment mechanisms ensure that individual agents receive meaningful learning signals (Foerster et al., 2018). Coordination develops in multiple ways. Direct communication allows agents to exchange summaries of their local state or intent signals (Zhu et al., 2024). Indirect coordination develops through shared environmental feedback, where agents develop the ability to predict the impact of their actions on congestion, resource contention and cascading effects in the system. Ultimately, over time, the policies of the agents converge to stable equilibria that strike a balance between local optimization and global performance, and enable the network to regulate itself in real time (Ning et al., 2024).

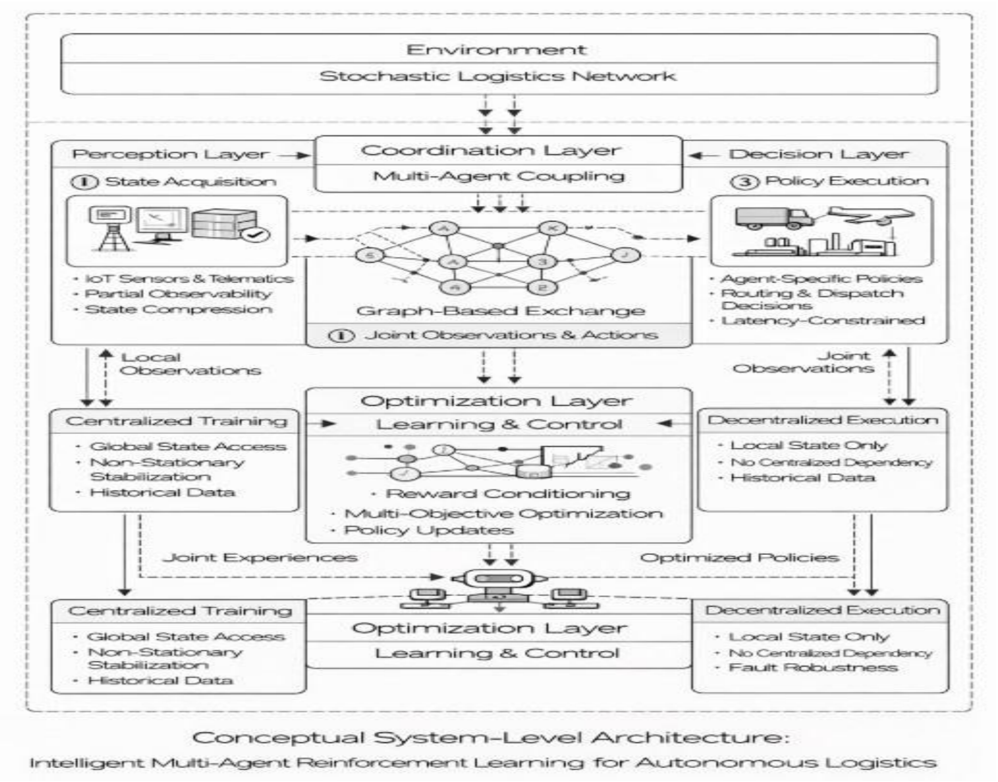
Centralized Training and Decentralized Execution in Logistics Environments

A key architectural feature of MARL in logistics is centralized training with decentralized execution (Amato, 2024). During training, agents have access to more information (including global representations of the system state and joint rewards), which facilitates learning algorithms to identify relationships and achieve stable convergence (Wang et al., 2022). During execution, each agent uses only local information and limited communication, which ensures scalability and robustness (Amato, 2024). Separating training from execution is especially important in logistics environments because it is impossible to execute in a centralized manner due to the latency and bandwidth limitations and due to organizational boundaries (Winkelhaus and Grosse, 2020). Vehicles cannot rely on continuous global coordination, and warehouses must function independently regardless of whether there is communication (Winkelhaus and Grosse, 2020). The CTDE architecture allows for the learning of shared value functions, factorized policies, or coordination priors that direct agent behavior during execution without needing real-time centralized control (Sunehag et al., 2017; Rashid et al., 2018). CTDE also matches well with the constraints of logistics enterprises, which can use historical data and digital twins for training while executing at the operational edge (Abideen et al., 2021).

Research Objective and Foundational Contributions

This research aims to provide an adaptable, scalable and resilient architecturally solid base for intelligent logistics systems utilizing Multi-Agent Reinforcement Learning (Ning et al., 2024). Unlike previous works, which proposed new algorithms, this work treats MARL as a holistic system architecture that encompasses theoretical formulations, architectural designs, communication protocols, learning dynamics, infrastructure needs, and enterprise-wide integration (Zhu et al., 2024). This research targets some of the main challenges currently preventing the widespread adoption of MARL in logistics, including scaling to thousands of agents, maintaining stability under changing conditions, coordinating under limited communication, and matching with the existing systems and governance frameworks of logistics enterprises (Winkelhaus and Grosse, 2020; Corvello et al., 2025). By representing logistics as a living multi-agent ecosystem, as opposed to a static optimization problem, this work supports a conceptual shift to self-optimizing and continuously-learning supply networks (El Hamdi et al., 2022). This foundational contribution serves as a basis for the following sections that successively formulate the learning dynamics, architectural elements, algorithmic components, simulation environments, and real-time implementation aspects (Abideen et al., 2021).

Figure 1: Conceptual system level architecture for Intelligent Multi-Agent Reinforcement Learning.



The layer-by-layer presentation of the system shown in figure 1 illustrates a layered control and learning architecture that supports multi agent reinforcement learning applied to autonomous logistics. Each layer is assigned a different role in the system and together they provide a way in which decentralized agents can operate autonomously yet be governed, coordinated, and learnable within the bounds of real-world constraints (Zhu et al., 2024). The architecture is represented in terms of formal blocks and flows rather than representational diagrams so that formal system behavior is emphasized over metaphorical representation. At the very top of the architecture is the environment layer, representing the global logistics network as a stochastic and partially observable system (Beynier, 2013). As a layer, the environment includes physical infrastructure (transportation links, warehouses and hubs), as well as external uncertainties due to factors such as weather and demand variability.

Additionally, the layer represents internal coupling effects, where the action of one agent causes changes in the condition experienced by other agents via congestion propagation and/or resource contention (Corvello et al., 2025). From a technical standpoint, the environment layer acts as the state transition function of a multi agent decision process, where joint actions cause a change in the global state of the network over time (Bernstein et al., 2002).

Below the environment layer is the perception layer, responsible for acquiring states given information limitations. Agents receive local, noisy and incomplete observations from sensor, telematic and operational systems. The perception layer performs observation mapping, feature extraction and state compression to create agent-specific representations of state that can be used as inputs to policy. Importantly, the perception layer imposes partial observability on the agents, preventing them from accessing global state information in real-time (Beynier, 2013). This limits the learning problem to the actual information available to the agents in deployment scenarios and eliminates unrealistic coordination that would not be possible in the operational context. The coordination layer provides the architectural core of the system and facilitates interaction among agents without relying on centralized control (Zhu et al., 2024). The coordination layer describes multi agent coupling via graph-based communication and message passing structures, where agents exchange limited amounts of information with their immediate neighbors or within constrained communication graphs.

Collective signals (e.g. congestion, demand imbalance, etc.) are encoded in shared latent representations. Coordination can also occur implicitly through learned value factorization or joint embedding (Sunehag et al., 2017; Rashid et al., 2018). Feedback from the environment to this layer indicates the continuous feedback loop between the evolving state of the system and agent coordination.

To the right of the architecture, the decision layer is responsible for executing decentralized policies. Each agent possesses a policy that maps local observations and coordination signals to actions. Actions consist of routing decisions, scheduling, dispatching and resource allocations and are taken in real-time while adhering to latency, bandwidth and reliability constraints. This layer embodies decentralized execution, implying that agents take independent actions in real-time and do not rely on centralized decision-making, but their actions are indirectly coordinated through the learned structures (Wang et al., 2022). At the bottom center of the architecture resides the optimization layer, which is responsible for performing learning and control. This layer collects feedback from the perception, coordination and decision layers, including rewards, performance metrics and constraint violations. The layer supports reward shaping, multi-objective optimization, and credit assignment across agents, allowing for trade-offs between objectives such as cost-efficiency, delivery times, emission reductions and system resilience (Foerster et al., 2018). Through reinforcement learning updates, the layer generates improved policies that are provided back to the decision layer and close the learning loop.

The optimization layer is supported by the centralized training component, which operates in either simulation or offline modes (Abideen et al., 2021). The centralized training component has access to the global state, joint rewards and historical data that are inaccessible during execution. Centralized training stabilizes learning in the non-stationary multi-agent environments by providing synchronized policy updates and explorations in digital twins or controlled simulations (Abideen et al., 2021). The centralized training component allows the system to learn about global structure and cooperative strategies that would be unachievable or unstable using solely decentralized learning (Ning et al., 2024). Conversely, the decentralized execution component enforces the deployment constraints in the production environments. While operating in execution mode, agents will utilize only their local observations and learned coordination mechanisms to make decisions, and will not have access to centralized control or global state (Amato, 2024). The decoupling of these two components ensures scalability, fault-tolerance to communication failures, and operational feasibility in large-scale logistics networks. Decentralized execution ensures that policies developed using centralized training can be reliably executed under real-world conditions (Wang et al., 2022).

Collectively, the architecture establishes a closed-loop system where autonomous agents operate in accordance with a structured learning and control framework. The environment generates state transitions, perception constrains information, coordination enables collective behavior, decision executes autonomy, optimization improves performance, centralized training stabilizes learning, and decentralized execution ensures scalability. Therefore, the architecture formally defines autonomous logistics as an engineered multi-agent control system, as opposed to an ad-hoc combination of intelligent components, thereby enabling rigorous theoretical study and practical deployment (Zhu et al., 2024).

Theoretical Foundations of Multi Agent Reinforcement Learning

Key Principles of Reinforcement Learning

Environment Policy Reward and Exploration Exploitation Trade-off Reinforcement learning theory starts with defining the decision-making process that produces experience. An RL environment is not only a simulation environment, it is also a stochastic dynamic system that describes how actions affect the environment (Sutton, 1988; Watkins & Dayan, 1992). For single-agent environments, an MDP model is typically utilized. The MDP model is described by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathrm{r}, \gamma \rangle$, which represents the state space $\mathcal{S}$ containing the hidden variables describing the state of the system at a particular decision epoch; the action space $\mathcal{A}$, containing all possible actions that can be taken; the transition probabilities

$\mathcal{P}(s'|s,a)$, indicating how the environment transitions between states following the execution of action a in state s; the reward function $r(s,a,s')$ representing instantaneous utility and providing a mapping between the operational objective of the agent and the learnable signal; and the discount factor γ, which is used to provide a weight to future outcomes relative to present outcomes, with a higher discount factor indicating longer-term optimization (Sutton et al., 2009).

Each of these components will have significant meanings in logistics. The state must be capable of representing the complex network conditions, including vehicle locations, queue sizes, inventory levels, backorders, link travel times, and service level agreements. The transition kernel includes the physical aspects of the system, including the build-up and dissipation of traffic congestion, the loading and unloading rates, and stochastic arrival processes of orders. The reward function captures the objectives of the system, including on time delivery, cost, energy usage, CO2 emissions, and resilience to disruptions. If the Markovian assumption is violated because of unobservable latent factors or delayed effects, the environment should be viewed as partially observable, and the agent must use its history of observations or belief states to estimate the Markovian structure of the environment (Bernstein et al., 2002). Policy learning aims to find the policy that maximizes the expected discounted return:

$$J(\pi) = \mathbb{E}\left[\sum^{\infty} \gamma^{t} r_{t}\right]$$

t=0

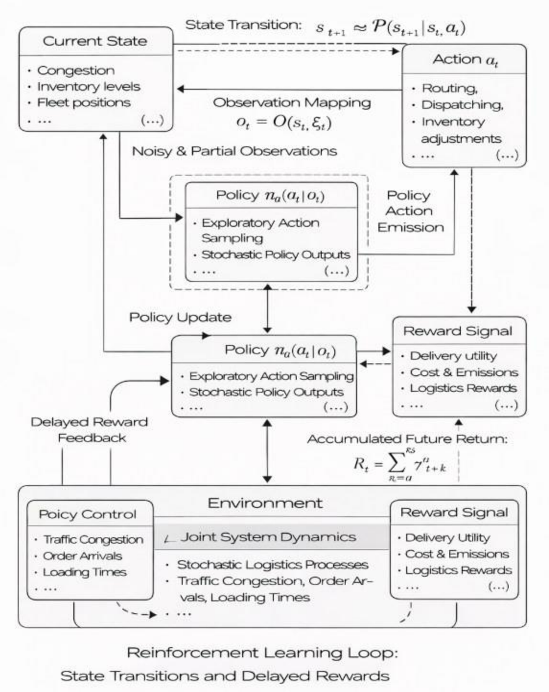
Policy is the primary mathematical construct that governs control and learning. The policy π(a|s) is a probability distribution over actions given the state. Stochastic policies are generally needed in logistics problems due to the presence of stochasticity and the need for exploration. Therefore, stochastic policies are used both for learning stability and for learning robustness (Williams, 1992). Tabular policy parameterizations are typically sufficient for small systems; however, in most logistics problems, function approximations are necessary for the policy πθ, and neural networks are used for this purpose, as they operate on structured inputs, including graphs, time-series data, and multimodal sensor data (Mnih et al., 2015). The goal of policy learning is to find the policy that maximizes the expected return J(π), where the expected return is the expectation of the discounted cumulative reward. The value function Vπ(s) is defined as the expected return starting from state s and following policy π, and the action-value function Qπ(s,a) is defined as the expected return starting from state s, taking action a, and then following policy π (Watkins & Dayan, 1992). These two functions allow learning algorithms to map long-term utility to short-term actions and are essential for both value-based and policy-gradient learning methods.

An exploration-exploitation trade-off is not simply a generic heuristic; it is a technical restriction that controls the statistical quality of the learning process. Exploitation selects actions that are believed to produce the highest estimated value, while exploration selects actions that decrease the uncertainty associated with the value landscape (Auer et al., 2002). In stochastic control terms, exploration is an information-gathering control policy that provides immediate performance to obtain reduced posterior uncertainty. In logistics, there is a structural constraint on this trade-off because exploratory actions can generate real operational costs, including missing delivery windows, increased fuel consumption, or customer dissatisfaction. Thus, exploration must frequently be risk-aware and constrained. Mechanisms for implementing these include using entropy regularization in policy gradients, adding parameter noise to value methods, optimistic initialization for value methods, posterior sampling, and safe exploration constraints that limit actions to the feasibility envelopes (Schulman et al., 2017; Achiam et al., 2017). In multi-agent environments, exploration is coupled because one agent's exploratory actions can change the data distribution experienced by other agents. Consequently, coordination among the agents for exploration or structured exploration policies, where the exploration is conducted in a way that maintains compatibility with system-level constraints, is required (Busoniu et al., 2008).

Another problem in logistics is that the reward is often sparse, delayed, and noisy. For example, the consequences of a routing decision may be the increase in congestion and subsequent costs to other parts of the supply chain hours later. Similarly, repositioning inventory may positively impact service levels days later.

These temporal delays make credit assignment difficult and amplify the variance in estimating returns. Discounting, selection, reward shaping, and auxiliary prediction tasks become critical theoretical tools for transforming delayed operational objectives into signals that can be learned efficiently without distorting the optimal policy (Sutton et al., 2009; Bellemare et al., 2017).

Figure 2: Reinforcement Learning Loop.



This is depicted graphically and demonstrates the stochastic decision-making process of autonomous logistics in a form of a stochastic decision process with both delayed and incomplete feedback (i.e., with partial observability). In the autonomous logistics environment at each decision point, there exists a current latent state describing the current logistics environment state (e.g., network conditions (i.e., congestions), inventory levels, positions of the fleets) that follows a stochastic transition probability based on the agents' actions, but that agents do not have direct access to. Instead of observing this latent state directly, the agents obtain a noisy, partial observation of the latent state through an observation function and pass it to a parameterized stochastic policy to generate an action (e.g., route, dispatch, adjust inventory). This action is then executed within the logistics environment to induce a new state and create a reward signal that reflects multiple objectives of logistics performance metrics (e.g., delivery utility, costs, emissions). Many logistics phenomena are delayed and therefore rewards will be accumulated and propagated back to the learner through discounted returns. The optimization part uses the delayed rewards to modify the policy parameters to improve the decisions on future actions. From a technical standpoint, the diagram illustrates all the major components of a partially observable MDP with function approximation and highlights the interaction between state transitions, observation noise, policy execution, and delayed reward feedback to allow for learning in complex, stochastic logistics systems.

Types of Multi Agent Reinforcement Learning: Cooperative Competitive and Mixed Mode Learning

Multi-agent reinforcement learning extends single-agent control through the introduction of multiple decisionmakers interacting simultaneously in a shared environment (Busoniu et al., 2008; Shoham et al., 2007). A basic theoretical differentiation among MARL problem types exists regarding how agent utilities coincide. Cooperative Multi-Agent Reinforcement Learning (MARL) assumes all agents have the same objective and seek to maximize a single overall reward. This is commonly expressed as a Decentralized Partially Observable Markov Decision Process (DEC-POMDP), wherein the environment has a global state, but each individual agent receives a local observation. Each agent makes decisions solely based upon their own local knowledge of the environment and the global state is updated according to the collective actions taken by all agents. Rewards are provided to agents to represent system performance and the reward is shared by all agents. This is an example of a single enterprise attempting to optimize total end-to-end service level and cost with subsystems (vehicles, warehouses, hubs, etc.) which must work together versus competitively.

A major difficulty in cooperative MARL is the growth rate of the joint action space as the number of agents increases resulting in exponential complexity in the computation required for solving the problem (Bernstein et al., 2002). Consequently, learning the value function as a function of joint actions Q(s,a1…aN) is computationally intractable for large-scale networks and thus a motivation for employing factorization-based assumptions as well as coordination mechanisms such as Centralized Training Decentralized Execution (CTDE) (Sunehag et al., 2017; Rashid et al., 2018). CTDE employs global knowledge during training to enable structured critics or decomposed value functions to be learned while allowing for independent action selection by agents at run-time. Furthermore, cooperative learning must address coordination failures which occur when agents converge to locally optimal yet globally suboptimal behaviors (i.e. conventions). Examples of this include vehicles selecting route alternatives to avoid local traffic congestion while creating congestion at other locations throughout the network; and/or warehouses individually optimizing their local throughput while starving downstream nodes.

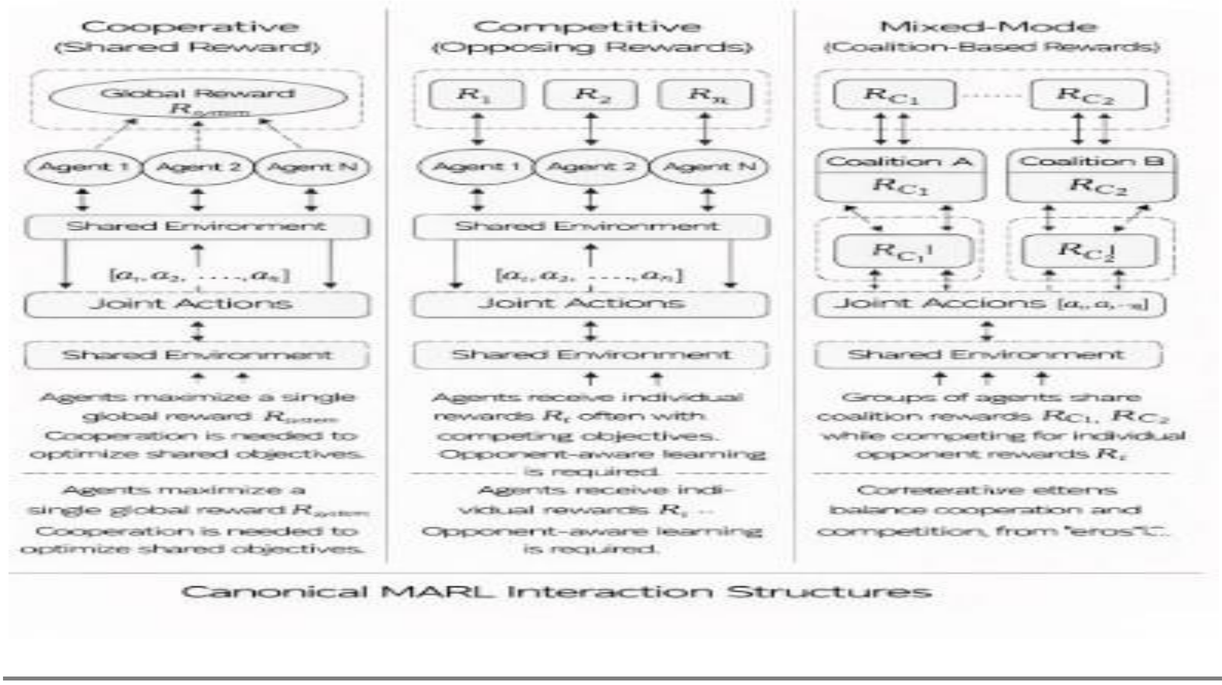
In contrast, competitive MARL models agents with competing goals, which are generally represented using either zero-sum or generalized sum stochastic games. Zero-sum games are those in which an agent's gain is equal to the loss experienced by another agent and the solution concept is minimax optimality. Competitive formulations arise in logistics when multiple carriers are competing for limited hub capacity or when autonomous vehicle fleets, which may be owned by different firms, use the same road infrastructure and strategically react to each other (Roughgarden & Tardos, 2002). For competitive MARL, it is necessary to employ learning mechanisms that account for the behavior of opponents, and equilibrium-seeking algorithms are typically employed. Theoretically, convergence to Nash Equilibrium may be established under specific restrictive conditions, e.g. two-player zero-sum games (Lowe et al., 2017).

Finally, mixed-mode MARL represents hybrid configurations where some subset of agents cooperate with each other while others compete with additional groups of agents. This is likely the most accurate representation of logistics multi-stakeholder ecosystems (Tuyls & Weiss, 2012). Mixed-mode learning introduces mechanism design concerns because the structure of rewards and incentives will ultimately determine if cooperative behavior will emerge or collapse into self-interested strategies. More formally, mixed mode systems may be modeled as generalized sum stochastic games with coalition structures or hierarchical reward compositions. To balance the local and global objectives of each agent, the reward received by each agent i is represented by:

$$r_i = \alpha r_{\text{local}} + (1 - \alpha)_{\text{system}}$$

Where α determines the weights assigned to the local and system-wide objectives, respectively, and influences the equilibrium properties, fairness, and stability (Shoham et al., 2007). From a learning perspective, the differing requirements for information-sharing between cooperative, competitive, and mixed-mode environments necessitates the development of federated and privacy-preserving learning architectures (Ning et al., 2024).

Figure 3: Canonical MARL interaction structures diagram



Agents maximize a single global reward R_{system}. Cooperation is needed to optimize shared objectives.

Agents maximize a single global reward R_{system}. Cooperation is needed to optimize shared objectives.

Agents receive individual rewards R_i often with competing objectives. Opponent-aware learning is required.

Agents receive individual rewards R_i -- Opponent-aware learning is required.

Groups of agents share coalition rewards R_{C1}, R_{C_2} while competing for individual opponent rewards R_i

Cortetetative ettens balance cooperation and competition, from "eros"[...].

In Figure 3, we see three types of standard forms in multi-agent reinforcement learning which have been identified based on the way that agents' incentives are connected with their rewards (incentives). The first form, the cooperative type, has all agents acting in the same environment and all agents get the same, global, reward. Therefore, the incentive for individual learning is based on overall system performance and not on individual success. The focus is on coordination between agents as the individual actions of agents contribute to a common outcome and less-than-optimal local decision making could negatively affect the overall performance of the system. The second form, the competitive type, also has agents acting in the same environment and producing joint actions, but each agent gets its own reward that may be in opposition to the other agents' rewards. Agents therefore need to learn about opponents and achieve an equilibrium. An example of this would be where an increase in performance for one agent occurs at the cost of another; in this case the performance of the two agents is a zero sum or general sum relationship. The third form, the mixed mode type, has agents grouped into coalitions that produce group rewards while they are also competing with other coalitions or agents using individual reward components. The mixed mode structure provides a good representation of many real world multi-stakeholder environments where there is partial cooperation among parties operating under some contractural or organizational agreement but there is competition among the different groups. Overall, the three panels provide examples of how differences in the coupling of the incentives of agents directly impacts the dynamics of learning, the amount of information exchanged, and the stability of multi-agent reinforcement learning systems.

Game Theory and Nash Equilibrium Applications in Agent Coordination

Game theory offers a formal framework for examining multi-agent strategic behavior (Shoham et al., 2007). In MARL the game is defined by the environment and reward functions used; each agent chooses a policy; the resultant joint policy determines the distribution of trajectories' possible outcomes. The Nash Equilibrium (NE) is a fundamental solution concept in game theory; it is a joint policy profile where no agent can improve its expected returns through unilateral action if all other agents maintain their policies unchanged (Zhang et al., 2021). Because MARL algorithms are likely to learn towards fixed points that are often interpretable as equilibria (even though they may be solving an objective function that is not NE), understanding NE is important for understanding how MARL systems will behave.

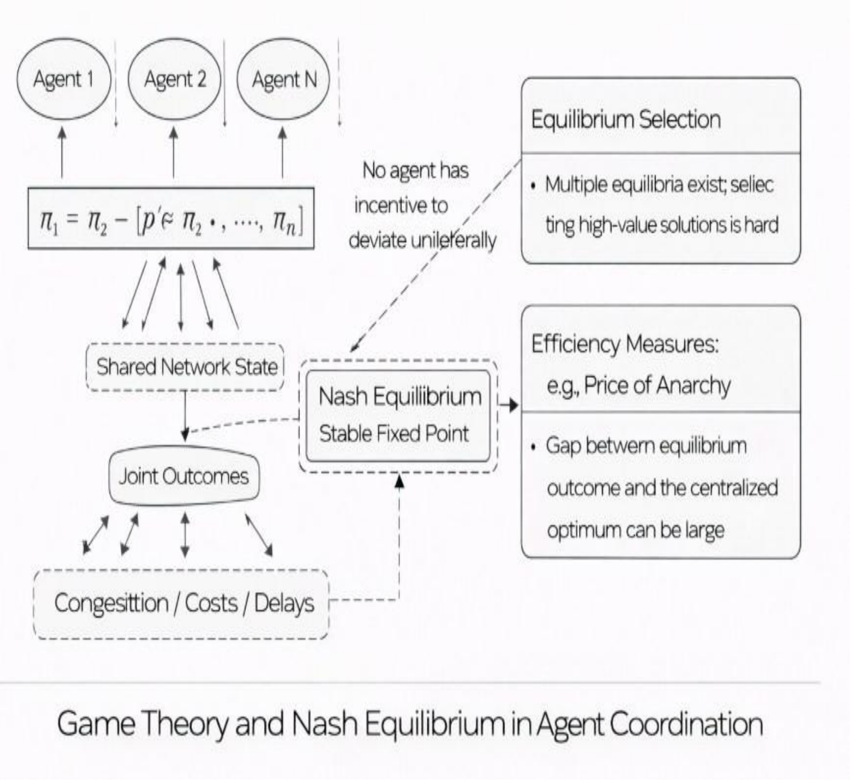
The reason why Nash Equilibrium is relevant to cooperative logistics is due to the fact that when there is decentralized implementation of logistics decisions, agents make decisions based upon local objectives or estimates of local values. Therefore, poor design of the learning system may cause agents to reach stable conventions (equilibria) that optimize unilateral objectives with respect to local knowledge while failing to achieve optimal global objectives. This is an example of a coordination equilibrium problem where there are multiple equilibria and the learning system needs to choose from them those that maximize social welfare.

Furthermore, in competitive or mixed mode logistics, Nash Equilibrium is an even more explicit part of the analysis. For instance, competing carriers allocating trucks to shared delivery zones can be viewed as a stochastic game where the equilibrium represents the stable allocation patterns of vehicles in the delivery zone. Additionally, efficiency analysis relies on concepts such as Price of Anarchy which measures the difference between equilibrium outcomes and the optimal outcomes that would have been achieved had a single entity planned the entire logistics system (Roughgarden & Tardos, 2002).

An additional aspect of MARL systems is that many MARL techniques can be thought of as being approximate equilibrium solvers. Policy Gradient Ascent can be seen as an approximation of gradient play; and in certain environments this method may converge; however, in general sum games, this method may cycle (Bowling & Veloso, 2002). Actor Critic methods with a centralized actor critic will stabilize the gradients; however, these methods do not provide any guarantees about achieving equilibrium (Foerster et al.,

2018). Furthermore, Nash Equilibrium analysis in logistics coordination problems intersect with Network Flow Theory through the concept of Wardrop Equilibrium in traffic networks; and MARL generalizes Wardrop Equilibrium under uncertainty and learning (Yau et al., 2017).

Figure 4: Game Theory and Nash Equilibrium in Agent Coordination


Agent 1
Agent 2
Agent N
$\pi_1 = \pi_2 - [p' \approx \pi_2 \bullet, \dots, \pi_n]$
No agent has incentive to deviate unileferally
Equilibrium Selection
• Multiple equilibria exist; seliec ting high-value solutions is hard
Shared Network State
Nash Equilibrium
Stable Fixed Point
Efficiency Measures:
e.g., Price of Anarchy
• Gap between equilibrium outcome and the centralized optimum can be large
Joint Outcomes
Congesittion / Costs / Delays
Game Theory and Nash Equilibrium in Agent Coordination

In figure 4 we show the development of a game theoretic structure, through the interaction of the decentralized nature of policies, shared system state and equilibrium conditions, from Multi Agent Reinforcement Learning (MARL). Multiple independent agents select policies; each agent has a strategic decision-making process regarding how to operate within the same shared environment. These individual agent's policy decisions result in a common network state (i.e., delay, congestion, resource utilization, etc.) and due to the fact that each agent's return is influenced by both their own policy and the policies of all other agents; the learning problem becomes a stochastic game. The Nash Equilibrium is located at the center of the diagram and represents the stable fixed point where no single agent can increase their expected return by making a unilateral deviation in their current policy as long as all other agents maintain their current policies. The Nash Equilibrium develops as a result of the feedback loop between joint outcome, system wide cost/delay and agent policy updates. In addition to the Nash Equilibrium the diagram highlights two significant theoretical issues; first, equilibrium selection, with respect to multiple equilibria existing and potential for the convergence of learning dynamics to efficient but unstable solutions. Secondly, efficiency analysis through measures such as the "price of anarchy," which measures the difference between equilibrium performance and the globally optimal performance of a centralized solution. Together the diagram illustrates how the learning dynamics of MARL, strategic interaction and network effects are combined to create stable yet potentially suboptimal coordination in large-scale logistics systems.

Credit Assignment Reward Sharing and Stability Challenges

Learning Credit Assignment in Multi-Agent Systems

Credit assignment is the problem of identifying how an agent's actions affect overall outcomes (Busoniu et al., 2008). Reward sharing among cooperative multi-agent reinforcement learning (MARL) logistics systems is typical since the primary goal of these systems is to achieve system-level performance. A shared global reward can create a very large variance learning signal because an agent experiences fluctuations in observed rewards due to both the actions of other agents and exogenous noise. The variance of this learning signal can increase with the number of agents and make learning unstable (Henderson et al., 2018).

Reward Shaping: Intermediate rewards are introduced through reward shaping to generate denser learning signals that may support the same optimal policy as unshaped rewards (potential-based shaping conditions).

However, shaping can modify the topology of the equilibrium landscape, and thus careful design is necessary (Foerster et al., 2018).

Difference Rewards: A difference reward provides a principled method of determining the marginal contribution of each agent's actions toward achieving a particular global state, but it requires estimating counterfactuals, which is computationally expensive (Foerster et al., 2018).

Value Decomposition Methods: Value decomposition methods decompose the joint value function into individual value functions corresponding to each agent's perspective, and do so subject to constraints that ensure that the resulting policies are decentralized and greedy (Sunehag et al., 2017; Rashid et al., 2018). The stability challenge associated with value decomposition arises due to the fact that each agent's learning affects the learning environment of every other agent. In logistics applications, this manifests as oscillatory routing or rescheduling patterns. To stabilize learning in such systems, one must use centralized critics, slow target updates, entropy regularization, and constrain policy updates, as well as structured communication protocols (Schulman et al., 2017; Achiam et al., 2017).

Another stability challenge is the existence of multiple equilibria, and convergence to suboptimal conventions. Curriculum learning, staged training, and explicit coordination mechanisms have been proposed to mitigate these challenges (Gronauer & Diepold, 2022). Finally, stability must be considered over both fast operational and slow learning time scales. Therefore production-grade multi-agent reinforcement learning systems typically decouple learning from execution through the use of shadow evaluation, canary deployment, and safety-constrained updates, to maintain operational stability (Henderson et al., 2018).

Architecture of Intelligent Multi-Agent Reinforcement Learning Systems

Architectural Foundations of Intelligent MARL Systems in Logistics

An architecture of an intelligent multi-agent reinforcement learning system represents the structural implementation of theoretical principles of autonomous decision-making, strategic interactions, and adaptive controls in decentralized systems. Architecture in logistics environments is not simply an engineering consideration, it is one of the primary determinants of the feasibility, scalability, and learning stability of a system. Unlike a central pipeline for optimization which assumes complete system observability and fixed models, MARL architectures must directly support partial observability, delayed feedback, non-stationary environments, and heterogeneous capabilities of individual agents (Busoniu et al., 2008; Hernández Leal et al., 2019; Gronauer & Diepold, 2022; Zhang et al., 2021). Therefore, the architecture establishes how abstract concepts such as policies, value functions, coordination equilibria, and learning processes are implemented in realistic environments, particularly those that are cyber-physical where computational decisions are executed using physical resources with dynamic characteristics that cannot be abstracted (Lee, 2008; Rajkumar et al., 2010). Logistics systems are paradigmatic examples of cyber-physical systems due to tight coupling of sensing, computation, communication, and actuation. Failures in timing, accuracy, and coordination cause cascading failures in services, safety risks, and economic losses (Lee, 2008; Rajkumar et al., 2010).

From the perspective of control theory, a MARL architecture describes how authority, information, and adaptation are distributed throughout a system. Assets in logistics networks operate under rigid latency and reliability constraints. Local decisions have global implications through shared infrastructure, demand coupling, and congestion effects. The architecture therefore reconciles decentralization with coordination by establishing explicit interfaces between perception, communication, action execution, and learning while satisfying the hard real-time distributed computing constraints that are inherent in cyber-physical systems (Lee, 2008; Rajkumar et al., 2010). The separation of concerns within the layered architecture allows for modular analysis of stability, convergence, and scalability while maintaining end-to-end consistency. Theoretically, MARL architectures can be viewed as structured approximations of decentralized stochastic games where each layer of the architecture introduces inductive biases that limit the number of possible policies and coordination schemes (Shoham et al., 2007; Tuyls & Weiss, 2012; Zhang et al., 2021). These biases are not arbitrary. They capture structural aspects of logistics systems including locality of interaction, hierarchical organization, and temporal separation between control and learning. An effective architecture matches the inductive biases introduced by the architecture with the structural properties of the problem, resulting in improved sample efficiency, convergence properties, and robustness (Busoniu et al., 2008; Hernández Leal et al., 2019; Gronauer & Diepold, 2022).

Organizational and enterprise constraints in logistics also require consideration of architectural design. Integration with existing operational systems, compliance with regulations, and incremental deployments require architectures that support hybrid operation between learning-enabled agents and rule-based systems as well as humans. Explainability and auditable decision traces should be treated as architectural requirements rather than afterthoughts (Doshi Velez & Kim, 2017). Furthermore, layered/hierarchical architectures provide a means of enforcing governance and operational stability when sensing, coordination, decision-making, and optimization responsibilities are clearly delineated and validated at their interfaces.

Theoretical, the architecture maps the abstract MARL objective to its operational realization. The global learning objective can be represented as the maximization of an expected long-horizon return over joint policies, recognizing that the objective is realized via approximate function learning and constrained updates rather than exact dynamic programming (Sutton & Barto, 1998; Mnih et al., 2015; Henderson et al., 2018):

$$\max_{\{\pi_i\}} \ \mathbb{E}\left[\sum^{\infty} \gamma^{t}\,(s_t, a_1{}^{t}, \dots, a_N{}^{t})\right]_{t=0}$$

The architecture determines how this joint objective is decomposed, approximated, and optimized through decentralized components, and how learning stability is preserved under non stationarity induced by interacting learners (Busoniu et al., 2008; Hernández Leal et al., 2019; Zhang et al., 2021). The remainder of this chapter elaborates how this decomposition is achieved through a layered architectural framework.

The architecture defines how this joint objective is decomposed, approximated, and optimized using decentralized components, and how learning stability is maintained under non-stationarity caused by interacting learners (Busoniu et al., 2008; Hernández Leal et al., 2019; Zhang et al., 2021). The remainder of this chapter will describe how this decomposition occurs through a layered architectural framework.

Layered Architecture: Sensing, Coordination, Decision-Making, and Optimization

Layered architectural paradigms allow for a systematic method to decompose the complexity of intelligent logistics systems while ensuring coordination. The proposed architecture is composed of four interconnected layers: sensing, coordination, decision-making, and optimization. Each layer has addressed a unique set of technical challenges and corresponds to a specific level of abstraction in the MARL formulation. The sensing layer serves as the perceptual interface between the physical logistics environment and the learning system. The primary responsibility of the sensing layer is to gather raw data from diverse sources, such as Internet-of things devices, telematics streams, inventory databases, and enterprise platforms (Atzori et al., 2010; Gubbi et al., 2013). Theoretically, the sensing layer represents the observation function of a decentralized decision process. It limits the amount of information available to each agent and thus influences the policy space (Busoniu et al., 2008; Zhang et al., 2021). In logistics systems, sensing is partially observable, noisy, and asynchronous and is representative of the cyber-physical nature of the domain where sensors, networks, and devices operate at different frequencies and levels of reliability (Lee, 2008; Rajkumar et al., 2010). Agents observe local traffic conditions but do not know the global congestion state. Warehouses observe the internal queue but not the downstream demand realization. The sensing layer must therefore perform abstraction, aggregation, and filtering to create concise, semantically meaningful representations.

The coordination layer is the central element of the MARL system architecture. The role of the coordination layer is to enable inter-agent dependency mediation without the need for centralized control. Theoretically, this layer approximates the coupling structure of the underlying stochastic game by allowing agents to base their decisions on common latent variables rather than the explicit global state (Zhang et al., 2021; Zhu et al., 2024). The coordination layer may implement graph-based message passing, learned intent embeddings, or implicit coordination through shared value representations, demonstrating the principle that coordination can be achieved through limiting information flow and representation structure, not through transmitting the entire state (Shoham et al., 2007; Tuyls & Weiss, 2012). The coordination layer is particularly relevant to logistics networks where there is significant interdependence among agents due to congestion, shared resources, and shared infrastructure.

The decision layer implements decentralized policy execution. Each agent selects actions based upon local observations and coordination signals, operating independently and subject to strict latency constraints. This layer satisfies the constraint of decentralized execution and reflects logistical realities of deployment in logistics environments that must meet timing and robustness constraints associated with real-time distributed systems (Lee, 2008; Rajkumar et al., 2010). Theoretically, the decision layer corresponds to factorized policy representations that approximate the joint policy using only local information (Busoniu et al., 2008; Zhang et al., 2021). However, the challenge remains to ensure that locally optimal actions maintain global coherence even though they were selected based on incomplete information and asynchronously.

The optimization layer governs learning and adaptation. It operates on a longer timescale than the decision layer and is responsible for updating policies, shaping rewards, and assigning credits. Theoretically, the optimization layer implements the learning dynamics that approximate equilibrium-seeking or welfare maximizing behavior under multi-agent coupling (Shoham et al., 2007; Tuyls & Weiss, 2012). In logistics systems, optimization must address delayed rewards, multiple objectives, and non-stationarity. Architecturally, decoupling the decision layer from the optimization layer is necessary to preserve operational stability and to continue to improve performance, particularly since large updates can disrupt both learning and real-world operations, an issue discussed in the literature related to reproducibility and stability of deep reinforcement learning (Henderson et al., 2018). The interaction between layers can be viewed as a closed-loop control system. Data flows from the sensing layer up to the decision layer, while learning signals flow from the optimization layer down to the policy execution layer. This separation enables independent analysis of perception accuracy, coordination fidelity, decision optimality, and learning stability. A compact representation of the interaction of the layers is:

A compact representation of layered interaction can be expressed as:

$\pi(a_i \mid o_i, c_i), c_i = f_{\text{coord}}(o_1, \dots, o_N)$

where c_i represents coordination signals generated from the shared information (Zhang et al., 2021; Zhu et al., 2024).

Agent Design: Autonomy, Communication, and Shared Policy Learning

Autonomy is an absolute necessity in logistics because logistics happens in highly dynamic, geographically distributed environments. As such, agents must operate autonomously and make real-time decisions when they have little or no knowledge of the status of other agents in the system. While theoretically, autonomy means that each agent has its own policy and local belief of the environment, it does not mean that an agent is independent. Rather, effective agents can take into account the effect of other agents in the system through the mechanisms of coordination incorporated into their decision process. Therefore, the design of an agent's ability to communicate is very important to the overall architecture of the agents in a multi-agent reinforcement learning (MARL) system.

Direct communication enables the agents to explicitly share information. However, direct communication has scalability problems due to the bandwidth and synchronization constraints present in many systems. Thus, there is a strong motivation to implement structured communication and/or selective information propagation as part of the architectural choices of MARL systems with communication (Zhu et al., 2024). An example of indirect communication includes shared latent representations of the environment or coordination through environmental signals. In both cases, the agents do not require explicit messages to understand what the other agents are doing. From a theoretical perspective, the mechanisms of indirect communication enable the agents to coordinate with each other without requiring direct information exchange. In particular, the mechanisms of indirect communication can embed coordination into the representation of the policy and values of the agents, thus reducing their dependency on direct information exchange (Zhu et al., 2024; Zhang et al., 2021).

In addition to the mechanisms of communication and coordination, a significant architectural aspect of MARL is the ability of multiple agents to learn and utilize a common policy. In systems consisting of homogeneous agents, the sharing of parameters can provide a common policy to the agents. Sharing a common policy can provide several benefits including reduced sample complexity for the agents and improved generalization performance. Theoretically, the sharing of a common policy imposes symmetry constraints on the learning problem. The symmetry constraints restrict the policy space to those functions that are invariant to permutations of the agents, thus providing better convergence behavior (Busoniu et al., 2008; Gronauer & Diepold, 2022). For example, in a logistics fleet, the agents can learn a common policy to navigate new routes or new geographic areas without having to be re-trained from scratch.

An additional architectural aspect of MARL systems is temporal abstraction. Many logistics decisions are made based on historical context including changes in traffic congestion, changing demand trends and so on. Agents must be able to abstract over varying lengths of time to consider these types of historical context. The attention-based component of the agent can provide a mechanism to abstract over varying lengths of time and to selectively focus the computation of the agent on salient structures of the environment. Attention-based components are aligned with recent advances in routing-oriented learning methods that use attention to model the combinatorial structure of the routing problem (Kool et al., 2019).

A representative agent-level objective can be expressed as follows:

$$\pi_i{}^* = \arg\ \max\ [\sum \gamma^t r_i(s_t, a_i{}^t, a_-{}^t{}_i)]$$

$\pi_i\, t$

where the dependence on other agents' actions reflects strategic coupling.

Control Hierarchy Between Centralized Training and Decentralized Execution (CTDE)

One of the major architectural challenges in designing intelligent multi-agent reinforcement learning systems is the trade off between learning efficiency and operational feasibility. One way to address this trade-off is through the use of a control hierarchy between centralized training and decentralized execution. From a theoretical standpoint, centralized training can provide a global view of the state of the system, the joint dependencies of the actions of the agents, and the complete reward signal generated by all the agents. However, from a deployment standpoint, decentralized execution is necessary in order to satisfy the operational constraints of latency, communication unreliability, and local autonomy that are typical of cyber physical and real-time distributed systems (Lee, 2008; Rajkumar et al., 2010). The CTDE paradigm addresses this trade-off by decoupling the learning authority from the execution authority while maintaining the coordination structure of the system. CTDE is a scalable and governance-compatible approach to deploying enterprise-wide intelligent multi-agent reinforcement learning systems.

Centralized training takes place in a regime where learning algorithms have access to the full joint system state and the actions of all agents. However, this regime is not intended to reflect operational reality, but rather to stabilize the learning dynamics. From a theoretical perspective, centralized training approximates a more fully observable stochastic game, enabling learning updates to condition on the full interaction structure of the agents. This is important because each agent experiences a non-stationary environment as the other agents in the system update their policies. Centralized training reduces this non-stationarity by conditioning updates on joint information, reducing variance and improving convergence behavior (Busoniu et al., 2008; Gronauer & Diepold, 2022; Zhang et al., 2021).

As a result of CTDE, a key architectural construct that emerges is the centralized critic. The centralized critic evaluates joint actions in the context of the global state, providing learning signals that reflect the inter-agent dependencies of the system. It is important to note that this critic is not executed during deployment. Rather, it serves as a training time mechanism that shapes the decentralized policies. Through the centralized critic, the system learns the coupling structure of the logistics system, and captures how individual actions contribute to congestion, resource contention, and demand fulfillment. From a theoretical perspective, the centralized critic approximates the joint-action value function of the underlying stochastic game, which is consistent with the broader actor-critic perspective on stabilizing policy learning through structured value estimation and variance reduction (Foerster et al., 2018; Lowe et al., 2017).

Decentralized execution enforces the operational constraint that each agent must act based solely on locally available information. Each agent executes a policy that is conditioned on local observations and limited coordination signals, and has no knowledge of the global state or centralized control. This is not simply a practical consideration, but rather a foundational consideration that defines the admissible policy class and ensures that the learned behaviors can be realized in real-world logistics environments. The architecture must therefore ensure that any information utilized during training can be reconstructed by the agents during execution. Consistency between centralized training and decentralized execution is a non-trivial architectural consideration. If training utilizes information that cannot be reconstructed during execution, a training execution mismatch results, which leads to either brittle or unrealizable policies, a failure mode that is exacerbated as deep reinforcement learning systems become increasingly complex and sensitive to experimental conditions (Henderson et al., 2018). From a theoretical perspective, this requires that the centralized critic provide shaping signals while the decentralized actors execute greedily based only on local information. Architectural mechanisms such as value decomposition and monotonic mixing provide one method to ensure this consistency by bounding how global value is represented and optimized, while preserving decentralized greedy execution (Rashid et al., 2018).

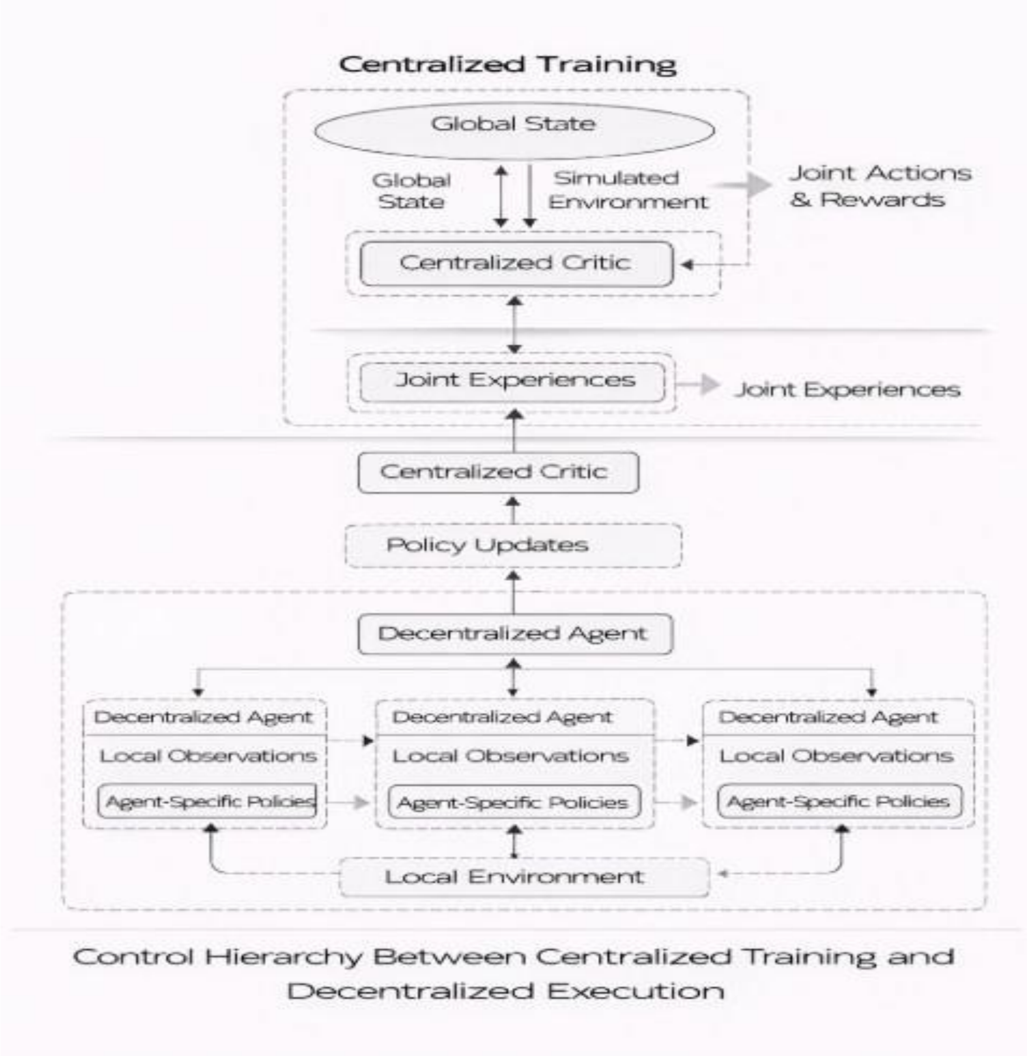
In addition to ensuring consistency between centralized training and decentralized execution, CTDE also introduces a temporal separation between learning and control. Typically, training is performed offline or asynchronously using simulated environments, historical data, or digital twins. Execution is performed in real time under tight latency constraints. This temporal separation is essential for logistics systems since frequent policy updates can destabilize operations. The architecture must therefore support slow, controlled policy updates that do not disrupt ongoing operations, and align the learning dynamics with the organization's risk tolerance. From a control-theoretic perspective, CTDE can be viewed as a two-layer control hierarchy. The higher layer performs strategic optimization and policy synthesis, while the lower layer performs real-time control. This is analogous to classical hierarchical control architectures, but extends them to include learning based systems. The learning layer adapts policies to meet long-horizon objectives, while the execution layer ensures responsiveness and stability at operational timescales (Lee, 2008; Rajkumar et al., 2010). CTDE architectures also support the needs of safety and governance. The decoupling of learning from execution permits the organization to validate and audit policies prior to their deployment. This is particularly important in regulated logistics environments where decisions have consequences related to safety, labor and compliance. The CTDE architecture enables the development of staged rollout strategies such as shadow evaluation and limited deployment, which reduce the risks associated with autonomous decision-making.

A simplified abstraction of the CTDE consistency relationship can be written as follows:

$$\pi(a_i \mid o_i) \approx \arg\max_{a_i} \mathbb{E}_{a-i}[Q_{\text{central}}(s, a_i, a-i)]$$

where the centralized critic shapes decentralized policy behavior without being required at execution time (Lowe et al., 2017; Foerster et al., 2018; Zhang et al., 2021).

Figure 5: Controlled Hierarchy architecture



The illustration demonstrates a two-layered architecture representing a control hierarchy between centralized training and decentralized execution. The control hierarchy has been separated into two distinct areas: a top area (centralized) where learning occurs, and an area below (decentralized) where execution occurs. The top layer is a centralized training regime that includes the learning system having access to global state information, simulated environments and joint actions/rewards generated by all agents. In this regime, joint experiences are combined and evaluated by a centralized critic that identifies how the interactions between individual agents' actions affect one another due to shared system dynamics such as congestion, resource competition and demand dissemination. A centralized critic is used as a training-time construct to stabilize learning in a non-stationary multi-agent environment, by having the critic update policy based on the full joint information, enabling the correct assignment of credits and the identification of the interdependencies among agents that would have otherwise remained hidden to purely local learning. Policy updates are produced during centralized training, and include the global interaction structure encoded in decentralized policies. The bottom layer illustrates decentralized execution, where each agent operates independently in real-time using only their local observations and agent-specific policies. Agents interact with their local environment and with each other only indirectly via environment-mediated effects, without access to global state or centralized control. Separating the layers ensures that operationally feasible constraints related to latency, communication and robustness are respected, while ensuring that learned behaviors can be realized during deployment. The link between the two layers emphasizes that, the information flows down from centralized training to decentralized execution only in terms of valid policy parameters, and not real-time control signals, thus avoiding training-execution mismatches. Overall, the diagram describes CTDE as a hierarchical control system where strategic learning and coordination occur centrally and off-line, while tactical decision-making and control occur locally and on-line, while providing scalability, stability and governance compatibility in large-scale autonomous logistics systems.

Scalability Mechanisms for Large-Scale Logistics Systems

Scalability is the principal architectural constraint that distinguishes theoretically appealing MARL formulations from systems that can operate in real logistics networks. Many modern logistics systems consist of thousands of vehicles, hundred of warehouses and millions of daily decisions. Without explicit architectural structures to address scalability issues, MARL systems will encounter combinatorial explosions in the number of possible joint actions, communications overload and unstable learning dynamics (Busoniu et al., 2008; Hernández Leal et al., 2019; Gronauer & Diepold, 2022). Scalability problems in MARL arise primarily from the exponential growth of the joint action space. As the number of agents increases, the number of possible joint actions increases exponentially. Thus, naive representations of joint policies or value functions become infeasible. Therefore, scalability architectures must be designed to exploit locality, symmetry and hierarchy existing in logistics networks (Tuyls & Weiss, 2012; Zhang et al., 2021).

Sharing parameters is a fundamental scalability mechanism. In logistics systems consisting of homogeneous agents (e.g., delivery trucks or warehouse robots), sharing policy parameters among agents greatly reduces the dimensionality of the learning problem. Sharing parameters introduces permutation invariance to the policy space, which improves generalizability and decreases variance. By sharing parameters, the learning process can use aggregated experiences of multiple agents rather than separate trajectories (Gronauer & Diepold, 2022; Busoniu et al., 2008). Another important scalability mechanism is the hierarchical decomposition of decision making and learning. Logistics systems typically possess natural hierarchical structure at various levels of time and/or space. Examples of hierarchical structure include decision-making at the fleet level (hours/days) versus decision-making at the vehicle level (seconds/minutes). Hierarchical decompositions provide a way to structure the learning and control processes according to the appropriate scales of decision-making.

Hierarchical decompositions allow learning within regions/hubs, but also reduce coupling among higher-level aggregates. Furthermore, hierarchical decompositions naturally correspond to temporal abstraction in RL, where decisions are made at different timescales, and policies can be structured accordingly (Kulkarni et al., 2016).

Architectures that provide sparse communication enhance scalability by limiting coordination among agents to those that are relevant to the task. Instead of communicating the entire state of the world, agents communicate only what is necessary about their local environment. Communication among agents is typically restricted based on the proximity of the agents, the agents' access to common resources, or the agents' history of interaction (Zhu et al., 2024; Zhang et al., 2021). Structured communication is a stability mechanism because it limits the amount of nonstationarity each agent experiences from other learners, yet still allows for coordinated behavior among agents.

Finally, the representation of global utility in value functions contributes to scalability. Representations of global utility that can be factored and decomposed enable scalability. Monotonic mixing-based methods, such as QMIX (Rashid et al., 2018), can represent global utility as a function of decentralized action selection, while approximating the global value function, which is particularly useful at scale where explicit representations of joint values are infeasible.

In addition to the aforementioned architectural mechanisms for managing scale, computational architecture plays an important role in scalability. Cloud-edge hybrid architectures are emerging as an important approach to supporting large-scale logistics networks. In such an architecture, large-scale training and simulation occur on centralized cloud infrastructure, while policy execution occurs on edge devices with low latency. Edge computing is particularly important for logistics decision-making loops, which are often time-sensitive and require rapid feedback to make decisions. Therefore, execution of policies must occur on edge devices, while training, aggregation, and heavy simulation occur on centralized cloud infrastructure (Satyanarayanan, 2017; Shi et al., 2016; Mach & Becvar, 2017; Mao et al., 2017). Research on computation offloading based on reinforcement learning (RL) supports this type of architecture by showing how the decision of where to perform computation can adapt to trade-off latency, energy, and reliability under varying network conditions (Hortelano et al., 2023).

From a learning-theoretic perspective, scalability mechanisms introduce inductive biases that reflect the structure of the environment. These biases limit the hypothesis space of policies and value functions, and improve sample efficiency. In logistics, these biases ensure that agents learn behaviors that respect locality, capacity constraints and flow conservation without requiring explicit encoding of these rules. Scalability must be able to accommodate the continuous growth and changes in organization of logistics networks. Logistics networks continue to evolve over time as new routes, facilities, and partners are added. Architectures that rely on rigid global coordination fail to accommodate such evolution. Scalable MARL architectures support incremental expansion, and allow new agents to join with little retraining, using shared policies and local coordination mechanisms. The computational complexity of scalable MARL architectures can be expressed abstractly as:

$$\text{Complexity} \propto (N \cdot d)$$

where N is the number of agents and d is a bounded local interaction degree, reflecting sparse interaction structure (Zhu et al., 2024; Zhang et al., 2021).

Business Impact and Strategic Value of MARL Architecture in Logistics Systems

Intelligent MARL Systems' Architectural Design has significant and substantial Business Implications, thus constituting a central contribution versus secondary considerations. While Algorithmic Improvements provide small increments under controlled conditions, Architectural Choices will either allow for the deployment of MARL in Enterprise Scale environments and produce long term Value or prevent such deployments. From a Cost Optimization Perspective, MARL Architectures enable continuous, Decentralized Adaptation to Demand Variability, Congestion, and Disruptions, whereas Traditional Logistics Systems are reliant upon Static Optimization or Periodic Replanning, neither of which effectively responds to Real Time Disturbances. The Intelligence embedded within MARL Architectures exists at the Agent Level, thereby enabling continuous adaptation of Decisions, resulting in Reduced Operational Slack, Improved Asset Utilization, Lowered Fuel, Labor, and Inventory Holding Costs. These Benefits are Strengthened when the System is Built as a Cyber Physical Control Loop with Verified Timing and Reliable Execution Pathways, as the Loss of Business Value in Logistics is typically due to Delay, Mismatch, and Poor Coordination, rather than Lack of Optimization Theory (Lee, 2008; Rajkumar et al., 2010).

Another primary Business Impact is Service Level Performance. Through Autonomous Responses to Local Disruptions while Maintaining Global Coordination among Agents, MARL Architectures Improve Delivery Reliability and Reduce Lead Time Variability. This is Particularly Valuable in Time Sensitive Logistics such as Ecommerce Fulfillment and Perishable Goods Distribution, where Service Failures have Proportionately Larger Downstream Consequences. A Strategic Dimension Where MARL Architectures Provide a Significant Advantage is Resilience. As Logistical Networks Experience Increasing Exposure to Disruptions including Traffic Incidents, Supply Shocks, and Extreme Weather Events, MARL Architectures Support Adaptive Reconfiguration of Routes, Schedules, and Resource Allocations in Response to Such Disruptions.

Furthermore, Adaptive Resilience Cannot be Achieved through Static Optimization Alone and Represents a Fundamental Shift in Logistics Capability.

A Strategic Dimension Where MARL Architectures Enable the Transition Toward Autonomous, Self-Regulating Logistics Networks. By Reducing Dependence upon Centralized Planning and Manual Intervention, Organizations Can Scale Operations without Linear Increases in Managerial Overhead. This Creates a Structural Competitive Advantage That Is Difficult to Replicate Using Traditional Systems.

Additionally, MARL Architectures Support Sustainability Objectives by Enabling Multi Objective Optimization that Explicitly Incorporate Emissions and Energy Efficiency Alongside Cost and Service Quality. In contrast to Rule Based Sustainability Initiatives, MARL Systems Continuously Learn Tradeoffs and Adapt Behavior as Operational Conditions Change. Organizationally, Layered MARL Architectures Facilitate Incremental Adoption. Components can be Deployed Gradually, Integrated with Existing Enterprise

Systems, and Evaluated Under Controlled Conditions. This Reduces Adoption Risk and Aligns with Enterprise Governance Requirements. Explainability and Accountability Further Influence Business Adoption Because Stakeholders Must Be Able to Audit and Justify Autonomous Decisions, Which Motivates Embedding Interpretability and Monitoring Interfaces Into the Architecture Itself Rather Than Relying Upon Post Hoc Explanations (Doshi Velez & Kim, 2017). Additionally, Fairness and Subgroup Performance Can Also Become Business Critical When Optimization Affects Service Allocation, Prioritization, or Customer Experience Across Diverse Segments, Thus Making Fairness Aware Evaluation and Governance a Strategic Consideration in System Design (Kearns et al., 2019).

The business objective underlying MARL-enabled logistics can be abstractly expressed as:

$$\max\, [\text{Service Reliability} - \text{Operational Cost} - \text{Environmental Impact}]$$

This objective highlight that MARL architecture is not merely a technical artifact but a strategic enabler.

Agent Communication and Coordination Protocols

Direct vs. Indirect Communication Models in Multi-Agent Logistics Systems

Coordination in Multi-Agent Reinforcement Learning (MARL) depends heavily on Communication. In Logistics Environments, Agents have partial Observations, are Spatially Separated, and have different Capabilities. Without an effective way to communicate, Agents cannot take into account the Externalities caused by their own Actions on Others, resulting in poor Coordination, Congestion Cascades, and Unstable Learning Dynamics (Sayde, 2014; Bucsoniu et al., 2008; Zhang et al., 2021). Although Communication is costly due to Bandwidth, Latency, Reliability, and Organizational Boundaries, there still needs to be an Architectural and Theoretical Design of Communication Mechanisms in MARL for Logistics (Ren & Beard, 2008; Nowzari et al., 2019; Olfati Saber et al., 2007).

Agents that use Direct Communication Models, send Messages to one another that include State Information, Intent Signals, or Coordinated Variables (Zhu et al., 2024). Theoretically, Direct Communication adds to each Agent's Observation Space with the Messages they receive from Other Agents, thereby Increasing the

Informational Richness of the Decision Process of the Agent, Reducing Uncertainty About the Global System State (Sayed, 2014; Cover & Thomas, 2006). In Logistics, Vehicles can share Congestion Alerts, Warehouses can Share Capacity Signals, Hubs can Coordinate Scheduling Decisions Using Direct Communication. With Reliable and Timely Communication, Direct Messaging Can Improve Coordination Efficiency and Convergence Speed Significantly (Ren & Beard, 2008; Olfati Saber et al., 2007).

Although Direct Communication Has Advantages, it Does Not Scale Well in Large Logistics Networks. As the Number of Agents Increases, the Communication Graph Becomes Dense and the Volume of Messages Grows Combinatorially, Causing Bandwidth Saturation, Synchronization Overhead, and Increased Latency. From A Theoretical Standpoint, Dense Communication Introduces Tight Coupling Between Agents, Which Can Destabilize Learning by Amplifying Feedback Loops (Nowzari et al., 2019; Sayed, 2014; Zhang et al., 2021).

Additionally, Operational Failures or Delays in Communication Can Further Reduce Performance, Making Direct Communication Brittle Under Real World Conditions (Nowzari et al., 2019; Ren & Beard, 2008).

Indirect Communication Models Address These Limitations by Embedding Coordination Signals Implicitly Within Shared Representations or Environmental Feedback Rather Than Explicit Messages (Zhu et al., 2024).

Agents Infer Behavior or Intent of Other Agents Through Observed System Dynamics, Shared Latent Variables, or Learned Coordination Embeddings in Indirect Communication. For Example, Congestion Patterns in a Network Can Serve as an Implicit Coordination Signal, Allowing Agents to Adapt Routing Decisions Without Explicit Messaging. Theoretically, Indirect Communication Uses the Environment as a Medium for Communication, Reducing Dependence on Explicit Channels (Sayed, 2014; Olfati Saber et al., 2007).

Indirect Communication is Particularly Attractive in Logistics Systems Because It Is More Representative of Physical Realities. Many Coordination Signals Are Naturally Embedded in the Environment Such as Queue Lengths, Travel Times, Resource Utilization. Agents Can Coordinate Implicitly at Scale While Learning to Interpret These Signals. However, Indirect Communication Adds Complexity to the Learning Process for Agents to Disentangle Causal Relationships Between Observed Dynamics and the Actions of Other Agents. This Can Increase Sample Complexity and Slow Down Convergence If the Architectural Inference Bias Does Not Support the Architecture Appropriately (Zhang et al., 2021; Zhu et al., 2024; Bucsoniu et al., 2008). Hybrid Communication Models Combine Direct and Indirect Mechanisms to Balance Scalability and

Coordination Fidelity (Zhu et al., 2024). In Hybrid Models, Direct Communication Is Reserved for High Impact or Time-Critical Signals While Indirect Communication Handles Routine Coordination. Selective Communication Strategies Reflect a Key Insight: Not All Coordination Requires Explicit Messaging. By Prioritizing Communication Resources, Hybrid Models Achieve Scalability Without Sacrificing Performance (Nowzari et al., 2019; Sayed, 2014). Communication Models Determine the Coupling Structure of the Decentralized Control Problem Theoretically. Direct Communication Increases the Strength of Coupling Among Agents While Indirect Communication Weakens Coupling But Relies on Shared Dynamics (Ren & Beard, 2008; Olfati Saber et al., 2007). Therefore, Effective MARL Architectures Must Be Designed to Stabilize Coordination Under Uncertainty and Optimize the Coupling Among Agents (Nowzari et al., 2019; Sayed, 2014; Zhang et al., 2021).

A conceptual abstraction of communication-augmented policy execution can be written as:

$\pi_i(a_i \mid o_i, m_i)$

where $m_i$represents either explicit messages or implicitly inferred coordination signals.

Graph-Based Communication Networks and Message-Passing Mechanisms

Graph-based communication has provided a formal basis to describe the relationships between the agents within a logistics system (Zhou et al., 2020; Zhang et al., 2021; Wu et al., 2021; Zhu et al., 2024) because all logistics networks have an inherent relational structure based on both physical proximity and operational dependencies. Agents are modeled as nodes in a graph and communication channels as edges, and this model enables both structured and scalable coordination in alignment with the problem domain (Zhou et al., 2020; Wu et al., 2021). Graph-based MARL models represent the interaction topology of the underlying stochastic game (Zhu et al., 2024; Zhang et al., 2021). Rather than assuming full connectivity between agents, the graph models which agents influence each other. Therefore, it limits the number of possible actions in a given state (Zhu et al., 2024; Wu et al., 2021), and limits the amount of information available to each agent to the agents they directly communicate with (Wu et al., 2021; Zhou et al., 2020).

The Message-Passing mechanism used in these models operate on this graph structure by enabling agents to pass messages to and receive messages from neighboring agents. An agent uses the messages received from its adjacent nodes to create a coordination context which will inform its decisions. The process of aggregating messages from different agents is analogous to a distributed consensus or fusion of information across multiple agents (Olfati-Saber et al., 2007; Boyd et al., 2006; Dimakis et al., 2010). Theoretically, message passing enables agents to reach an agreement on a global solution by exchanging messages repeatedly in a localized manner as information does in physical networks (Ren & Beard, 2008; Boyd et al., 2006; Dimakis et al., 2010). Furthermore, graph-based communication allows for heterogeneity in the agents themselves. Different agents (e.g., vehicles, warehouse, hubs) can be represented as different types of nodes in a graph with different message passing functions (Zhu et al., 2024; Wu et al., 2021; Zhou et al., 2020). This allows for different coordination protocols to adhere to the specific limitations and capabilities of each type of agent (Zhu et al., 2024; Wu et al., 2021; Zhou et al., 2020). Heterogeneous agents are common in logistics, so this capability makes graph-based communication a natural fit for logistics.

Another theoretical challenge in graph-based communication is achieving a balance between expressiveness and stability. Excessive use of deep or unstructured message passing can cause over-smoothing, where the representation of individual agents becomes indistinct and thus decision quality decreases. Shallow message passing may fail to capture long range dependency in logistics interactions. Therefore, careful consideration of the graph's depth, aggregation function, and update frequency is required to match the spatial and temporal characteristics of logistics interactions (Wu et al., 2021; Zhou et al., 2020). Additionally, graph-based communication allows for dynamic updates to the topology of the graph itself. If logistics networks change in response to changing traffic conditions, shifting demands, or failures, the graph can be dynamically updated to reflect changes to the interaction patterns between agents (Nowzari et al., 2019; Wu et al., 2021; Zhu et al., 2024). These dynamic changes provide additional resiliency to the coordination mechanisms being employed.

A generic message-passing update can be abstractly represented as:

$$h_i^{(k+1)} = f\left(h_i^{(k)},\ \sum\ g\,(h_j^{(k)})\right)$$

$j \in \mathcal{N}(i)$

where $h_i$denotes agent representations and $\mathcal{N}(i)$the neighborhood of agent i.

Bandwidth-Efficient Policy Sharing and Compressed State Exchange

The bandwidth limitations of MARL are considered one of the major limitations of MARL applications in the real world. Many logistics agents have to work through un-reliable wireless networks with limited bandwidth and varying latency. Therefore, it will be impossible to continuously send/receive large amounts of high dimension data regarding the states or policies of other agents in order to coordinate their actions. As a result, communications between the agents need to be optimized for bandwidth (Nowzari et al., 2019; Sayed, 2014).

From a theoretical perspective, bandwidth optimization is trying to minimize the amount of information that is exchanged about the agents, while still optimizing the performance of coordination (Cover & Thomas, 2006). The first step to do this is to identify what information needs to be communicated in order to facilitate coordination. Some of the information that the agents need to share can be abstracted away or replaced with some form of approximation (Cover & Thomas, 2006). For example, in many logistics systems, not all of the state variables that are available to each agent are relevant to the coordination of the agents' activities.

One way to reduce the communication costs associated with coordinating multiple agents is to use policy sharing mechanisms. With these mechanisms, instead of communicating the raw state information between the agents, they can communicate compact representations of the policies or the intentions of the other agents. This would enable the agents to communicate at a lower frequency (i.e., less frequently) than if they were to continuously exchange state information. According to the theory, this should allow the agents to anticipate how the other agents will behave (Sayed, 2014; Zhang et al., 2021), but it can also introduce new problems (such as synchronizing versions, establishing trust, etc.) especially in cases where there are multiple competing agents (Zhang et al., 2021; Zhu et al., 2024).

Compressing state information can further reduce the communication costs, by encoding the observed costs, by encoding the observed information into a set of low dimensional latent features. These latent features

can be learned simultaneously with the policies of the agents, so that the agents only communicate the most important information that is necessary for them to coordinate their actions. From a learning-theory viewpoint, compressing the state information introduces an information bottleneck that can help to regularize the coordination between the agents, and improve their ability to generalize to different scenarios (Cover & Thomas, 2006; Sayed, 2014). However, if the compression is too strong, it can lead to the loss of important information that is required to perform well, and thus can degrade the performance of the system (Cover & Thomas, 2006).

Finally, another mechanism to optimize bandwidth usage in multi-agent systems is based on event triggered communication. In this type of communication mechanism, the agents only communicate when a certain threshold has been reached, i.e., a significant change in the level of congestion or capacity. Thus, the communication is tied to the actual operational requirements of the system, rather than to the fixed time intervals used in periodic communication schemes (Nowzari et al., 2019; Li et al., 2024). From a theoretical point of view, event triggered communication leads to the reduction of the transmission of irrelevant information, and to the stabilization of the coordination between the agents under changing conditions (Nowzari et al., 2019; Wang et al., 2023).

A high-level abstraction of compressed communication can be written as:

$$m_i = \phi(o_i),\ \dim(m_i) \ll \dim(o_i)$$

where ϕdenotes a learned compression mapping.

Federated Learning for Distributed Coordination Across Logistics Nodes

Distributed coordination based on Federated Learning allows for the preservation of data locality, privacy and organizational boundaries (Yang et al., 2019; Xia et al., 2021; Li, T., et al., 2020; Aledhari et al., 2020; Huang et al., 2022; Yang et al., 2019). Due to regulatory constraints, competitive concerns, and data volume, centralizing data aggregation in logistics systems covering various regional areas, partners, or companies is generally unfeasible. Therefore, Federated Learning provides an environment for agents or nodes to collaboratively enhance shared models without having to share their original data (Aledhari et al., 2020; Yang et al., 2019). On a theoretical basis, federated Multi-Agent Reinforcement Learning (MARL) expands upon decentralized learning with the introduction of periodic synchronizations of model parameters instead of state or trajectory data (Sayed, 2014; Li, T., et al., 2020). Agents perform local learning using their individual experiences and the collected updates are combined into a global model. The combination of the updates can be viewed as a type of distributed stochastic optimization (Li, T., et al., 2020; Sayed, 2014). Federated Learning facilitates coordination among geographically dispersed hubs or fleets in logistics while preserving the autonomy and privacy of each hub or fleet (Xia et al., 2021; Aledhari et al., 2020).

In the context of heterogeneous logistics environments including various stakeholders, federated coordination is especially beneficial because each agent operates under different constraints, data distributions, and objectives. With federated learning, each agent can provide input to a shared coordination model while maintaining their local specialization. Theoretically, this promotes transfer learning and improves the ability to generalize over a variety of operational settings (Huang et al., 2022; Yang et al., 2019). One of the largest challenges in federated MARL is dealing with non-identical data distributions. For example, logistics nodes typically have different demand levels, traffic conditions, and operational constraints. When combining updates from different distributions, the rate of convergence can slow down and/or cause bias in the global model (Zhao et al., 2018; Li, T., et al., 2020). There are several architectural solutions available such as weighted aggregation, adaptive learning rates, and clustering-based federation (Li, T., et al., 2020; Zhao et al., 2018).

Additionally, federated learning has implications regarding communication limitations. While federated learning reduces the need to send raw data, sending model updates can be large and thus compression and sparseness of updates become essential parts of federated coordination architectures (Xia et al., 2021; Aledhari et al., 2020). Moreover, asynchronous aggregation can help to reduce latency and synchronize overheads (He et al., 2020; Li, T., et al., 2020). From a governance perspective, federated MARL architectures align well with regulatory and organizational expectations such as auditability, data sovereignty and controlled sharing of information (Yang et al., 2019; Aledhari et al., 2020). This makes federated coordination especially appealing for large-scale logistics systems which span across many jurisdictions.

A simplified federated update can be expressed as:

$$\theta(t+1) = \sum_{i=1}^{N} w_i\, \theta_{i(t)}$$

where $\theta_i$denotes locally learned parameters and $w_i$aggregation weights.

Learning Strategies and Algorithmic Mechanisms

Deep Q-Networks, Actor–Critic, and Proximal Policy Optimization in Multi-Agent Logistics

Learning Strategies in Multi-Agent Reinforcement Learning (MARL) will define how Agents adjust their policies over time in reaction to Rewards that are Delayed, Stochastic and Coupled Strategically (Busoniu et al., 2008; Hernández Leal et al., 2019; Zhang et al., 2021). In Logistics Environments, Learning Algorithms have to work under High-Dimensional State Spaces, Multi-Objective Reward Structures and Non-Stationarity Induced by Other Learning Agents (Henderson et al., 2018; Hernández Leal et al., 2019). Therefore, Classical Reinforcement Learning Algorithms have to be Extended and Reinterpreted so that they Function as

Coordination Mechanisms instead of Isolated Optimizers (Shoham et al., 2007; Zhang et al., 2021). Deep QNetworks, Actor-Critic Methods and Proximal Policy Optimization Represent Three Foundational Algorithmic Paradigms That Support this Transition (Mnih et al., 2015; Konda & Tsitsiklis, 2003; Schulman et al., 2017).

Deep Q-Networks Approximate Action-Value Functions Using Deep Neural Networks, Allowing Agents to Learn Long-Horizon Utility Estimates from Raw Observations (Mnih et al., 2015). For Logistics Tasks Such as Routing and Dispatching, DQN Allows Agents to Associate Immediate Actions with Delayed Outcomes such as Congestion Buildup or Delivery Delays (Mnih et al., 2015; Henderson et al., 2018). Theoretically, DQNs Approximate Fixed Points of the Bellman Optimality Operator (Mnih et al., 2015). However, in Multi-Agent Settings, the Bellman Operator Becomes Non-Stationary Because Transition Dynamics Depend on Other Agents' Evolving Policies (Busoniu et al., 2008; Hernández Leal et al., 2019; Zhang et al., 2021). Naïve Application of DQNs in MARL Often Leads to Instability Unless Architectural Constraints or Training Constraints Are Applied (Henderson et al., 2018; Hernández Leal et al., 2019). Stabilization Mechanisms Developed in Deep RL, Including Double Q-Learning, Dueling Architectures, Prioritized Experience Replay, Distributional RL, and Integrated Approaches Such as Rainbow Are Directly Relevant in this Context Because They Reduce Overestimation Bias, Improve Representation, and Enhance Sample Efficiency Under Noisy, High-Dimensional Dynamics (van Hasselt et al., 2016; Wang et al., 2016; Schaul et al., 2016; Bellemare et al., 2017; Dabney et al., 2018; Hessel et al., 2018).

Actor-Critic Methods Address Some of These Limitations by Separating Policy Representation (Actor) from Value Estimation (Critic) (Konda & Tsitsiklis, 2003). This Separation is Theoretically Significant Because It Decouples Policy Improvement from Value Approximation Error (Konda & Tsitsiklis, 2003). In Logistics Environments, Actor-Critic Methods Enable Continuous Action Spaces and Stochastic Policies, Which Are Essential for Modeling Routing Probabilities, Inventory Adjustments, and Scheduling Priorities (Lillicrap et al., 2016; Haarnoja et al., 2018). The Critic Provides a Low-Variance Learning Signal That Stabilizes Policy Updates Even Under Partial Observability and Delayed Rewards (Konda & Tsitsiklis, 2003; Schulman et al., 2015). This Family Also Scales Naturally to Distributed Execution and Asynchronous Learning Regimes That Are Operationally Aligned with Logistics Systems Where Data Arrives from Many Concurrent Actors, by Using Asynchronous Updates and Decoupled Actor-Learner Pipelines (Mnih et al., 2016; Espeholt et al., 2018).

Proximal Policy Optimization Introduces an Additional Theoretical Safeguard by Constraining the Magnitude of Policy Updates (Schulman et al., 2017). In Logistics Systems, Abrupt Policy Changes Can Destabilize Operations, Leading to Oscillatory Routing or Rescheduling (Henderson et al., 2018). PPO Enforces Trust Region – Like Constraints That Limit How Far a Policy Can Deviate from Its Previous Iteration (Schulman et al., 2017). This Aligns Naturally with Operational Risk Constraints in Logistics, Where Gradual Adaptation is Preferred Over Aggressive Optimization (Achiam et al., 2017). From a Theoretical Perspective, PPO Improves Stability by Approximating Monotonic Policy Improvement Under Bounded Updates (Schulman et al., 2017), Closely Related in Intent to Trust Region Policy Optimization (Schulman et al., 2015).

In MARL Contexts, these Algorithms Must Be Interpreted as Components of a Coupled Learning System Rather Than Standalone Optimizers (Busoniu et al., 2008; Shoham et al., 2007). Each Agent's Learning Update Alters the Effective Environment Experienced by Others, Creating Feedback Loops (Hernández Leal et al., 2019; Zhang et al., 2021). Algorithms That Are Stable in Single-Agent Settings May Fail When These

Couplings Are Ignored (Henderson et al., 2018; Hernández Leal et al., 2019). Thus, Architectural Constructs Such as Centralized Critics, Shared Replay Buffers or Coordination-Aware Value Estimation are Often Required to Restore Theoretical Stability (Lowe et al., 2017; Foerster et al., 2018; Hernández Leal et al., 2019). The Multi-Objective Nature of Logistics Further Complicates Learning. Routing Decisions Affect Time, Cost, Emissions and Reliability Simultaneously. Actor–Critic and PPO Frameworks Naturally Support Multi-Objective Learning Through Weighted Reward Aggregation or Vector-Valued Critics (Roijers et al., 2014; Vamplew et al., 2011). This Flexibility Allows Learning Strategies to Encode Organizational Priorities Explicitly Instead of Relying on Post Hoc Tradeoff Tuning (Vamplew et al., 2011; Roijers et al., 2014). A Representative Learning Objective Underlying these Algorithms can be Expressed as:

J(π) = ∑_{t=0}^{∞}γ^t(r_t + \phi(t))

A representative learning objective underlying these algorithms can be expressed as:

$\gamma_t r_i(s_t, a_{it}, a_{-t\,i})]$

$_{i\,t=0}$

$$\max_{\pi} \ \mathbb{E}\Bigg[\sum^{\infty}$$

where coupling to other agents' actions reflects coordination dynamics.

Value Decomposition Networks for Cooperative Multi-Agent Learning

A primary challenge in Cooperative Multi-Agent Reinforcement Learning (MARL) is enabling agents to learn how to work together and act cooperatively, but execute their actions independently (Busoniu et al., 2008; Zhang et al., 2021). Agents within Logistics Systems have a common goal and thus want to minimize the overall delivery time and/or costs across all agents, however each agent will only act based upon information that it has available locally (Bernstein et al., 2002; Hernández Leal et al., 2019). Value Decomposition offers a theoretical framework that allows for the reconciliation of these two opposing objectives by providing a method for decomposing the global value function into multiple functions associated with each agent (Rashid et al., 2018). The core theoretical contribution of Value Decomposition is that if the joint action-value function can be decomposed into a structure composed of individual value functions, then agents acting independently based upon their own values will lead to a solution that is globally optimal (Rashid et al., 2018). Thus, in Logistics, it suggests that Vehicles or Warehouses can make Local Decisions that result in Optimal Performance at the Network Level. It is essential for both Scalability and Feasibility (Bernstein et al., 2002; Busoniu et al., 2008).

VDN uses an Additive Structure to Decompose the Joint Action-Value Function (Busoniu et al., 2008). While Additive Structure simplifies the Problem-Solving Process, it places constraints on how Agents Interact with Each Other (Hernández Leal et al., 2019). QMIX relaxes the assumption of Additive Structure by using a Monotonic Mixing Function to Mix Individual Values into a Single Global Estimate of Value (Rashid et al., 2018). The Monotonicity Constraint Ensures that Increasing an Agent's Local Value Cannot Decrease the Overall System Value, which Preserves the Ability of Agents to Act Independently (Rashid et al., 2018).

Theoretical Considerations for Value Decomposition Place Inductive Biases on the Space of Represented Coordination Strategies (Hernández Leal et al., 2019; Zhang et al., 2021). These Biases Are Beneficial in Logistics Systems Where Agent Interactions Are Structured and Local Rather Than Arbitrary (Zhang et al., 2021). Encoding the Assumptions Architecturally Improves Sample Efficiency and Learning Stability (Rashid et al., 2018; Henderson et al., 2018).

Value Decomposition Also Addresses Credit Assignment. By Assigning Portions of the Global Value to Individual Agents, It Provides More Informative Learning Signals Than Shared Global Rewards. This Reduces Variance and Accelerates Convergence in Large-Scale Logistics Networks (Rashid et al., 2018; Foerster et al., 2018). Nonetheless, Value Decomposition Is Not Without Limitations. The Monotonicity Constraint May Restrict the Capacity of Representation and Prevent the Development of Coordination Strategies That Require Non-Monotonic Interactions (Son et al., 2019; Hernández Leal et al., 2019). Therefore, Logistics Contexts with Complex Interdependencies Such as Tightly Coupled Hub Scheduling Must Be Carefully Evaluated (Hernández Leal et al., 2019; Zhang et al., 2021). Although There Are Limitations to Value Decomposition, It

Represents One of the Most Practically Successful Paradigms for Cooperative MARL in Logistics Due to Its Balance Between Theoretical Guarantees and Operational Feasibility (Rashid et al., 2018; Hernández Leal et al., 2019).

A canonical abstraction of value decomposition is:

$$Q_{\text{total}}(s, \mathbf{a}) = f(Q_1(s, a_1), \ldots, Q_N(s, a_N))$$

where f is constrained to preserve monotonicity.

Reward Shaping and Credit Assignment for Joint Optimization

Designing Rewards is arguably the most important theoretical element of Multi-Agent Reinforcement Learning (MARL) (Busoniu et al., 2008; Zhang et al., 2021), since in supply chain logistics, Rewards express organizational goals for which poorly designed Rewards can undermine learning and induce self-serving behavior (Henderson et al., 2018; Shoham et al., 2007) and/or sub-optimal equilibria (Shoham et al., 2007; Busoniu et al., 2008). Thus, designing Rewards is a fundamental aspect of designing Learning Strategies

(Foerster et al., 2018; Hernandez-Leal et al., 2019) including both Reward Shaping and Credit Assignment. In Cooperative MARL, typically Agents receive a single Global Reward. Although Global Rewards are conceptually simple, they create serious problems of Credit Assignment. Specifically, because every Agent's learning signal is influenced by all other Agents' Actions, the variance of every Agent's learning signal grows as the number of Agents increases, resulting in slow learning and unstable coordination among Agents (Hernandez-Leal et al., 2019; Henderson et al., 2018; Busoniu et al., 2008). Reward Shaping introduces Intermediate Rewards that guide learning toward a better outcome without changing the optimal policy under appropriate conditions (Henderson et al., 2018). Examples of Reward Shaping in Supply Chain Logistics include penalizing Congestion Generation, Encouraging Early Delivery, and Maintaining Inventory Balance.

Theoretical analysis views Reward Shaping as a form of Potential-Based Transformation that preserves Policy Optimality while Improving Learning Efficiency (Zhang et al., 2021; Busoniu et al., 2008). Difference Rewards represent a Principled Solution to the Problem of Credit Assignment by Measuring the Marginal Contribution of each Agent to the Global Outcome (Foerster et al., 2018). In a Supply Chain Context, this represents an estimation of how a specific Vehicle's Routing Decision Influenced Overall Network Delay. Difference Rewards reduce Variance but are Computationally Expensive because they Require Counterfactual Reasoning (Foerster et al., 2018). Value-Based Credit Assignment Mechanisms, such as Centralized Critics and Decomposed Value Functions, Offer Scalable Approximations to Difference Rewards (Rashid et al., 2018; Lowe et al., 2017). These Value-Based Credit Assignment Mechanisms Infer Contribution Implicitly Through Learned Representations Rather Than Explicit Counterfactual Simulation (Hernandez-Leal et al., 2019; Rashid et al., 2018). Finally, Credit Assignment is Tightly Coupled with Stability. Poorly Attributed Rewards Can Cause Oscillatory Behavior, Where Agents Overreact to Noisy Signals (Henderson et al., 2018). Successful Credit Assignment Dampens These Oscillations By Providing Consistent Gradients Aligned With Global Objectives (Foerster et al., 2018; Henderson et al., 2018).

A conceptual representation of difference rewards is:

$$D_i = R(\mathbf{a}) - R(\mathbf{a}_{-i})$$

which isolates marginal contribution.

Meta-Learning and Continual Learning for Adaptive Policy Refinement

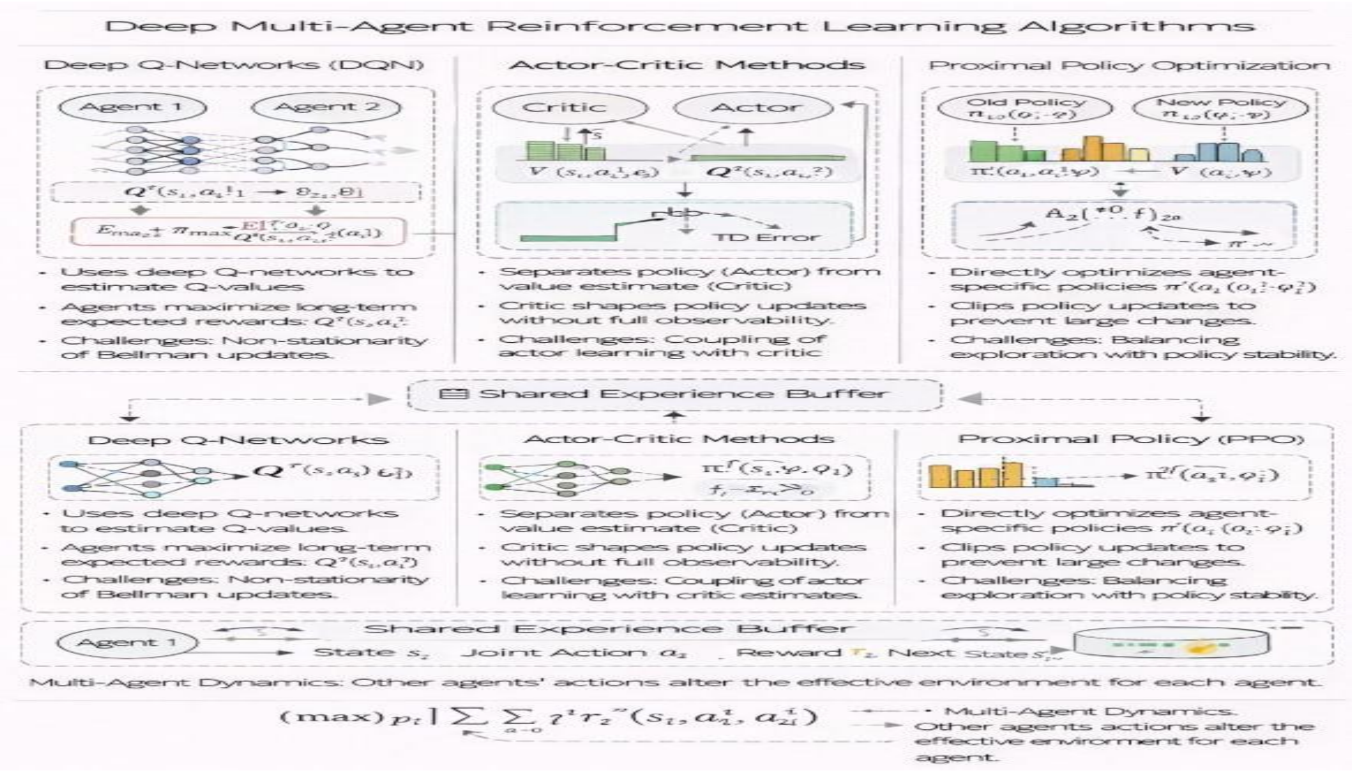
The environments in which logistics take place are always changing — the patterns of demand fluctuate, the infrastructure is continuously developing, and unforeseen disturbances will occur (Henderson et al., 2018; Parisi et al., 2019). Policies created to be used statically (or to remain unchanged) as they were originally designed (with certain assumptions that do not change), quickly deteriorate when working in non-static environments (Henderson et al., 2018; Parisi et al., 2019). There are both theoretical and practical methodologies associated with meta-learning and continual learning that address the need for continued adaptation within MARL systems (Finn et al., 2017; Parisi et al., 2019; Lesort et al., 2020). Meta-learning is concerned with learning "how to learn." Rather than designing an optimal static policy for an environment, meta-learning creates policies that are able to quickly adapt to new environments (Finn et al., 2017; Nichol et al., 2018). For example, in logistics, it would allow the ability to transfer knowledge between different locations, seasons or types of demand (Pan & Yang, 2010; Finn et al., 2017). Meta-learning theoretically finds the best way to initially set up or update a policy so as to reduce the amount of time required to adapt to the new conditions (Finn et al., 2017; Nichol et al., 2018). Continual learning provides the means for learning over an extended operational horizon, providing solutions to the problem of catastrophic forgetting (Kirkpatrick et al., 2017; Parisi et al., 2019). For example, in logistics systems, agents may need to learn new things while maintaining their existing knowledge of the system's dynamics that have not changed. Continual learning methods add constraints or create memories that prevent an agent from losing the representation of useful aspects of its knowledge base while adapting to new conditions (Kirkpatrick et al., 2017; Parisi et al., 2019; Lesort et al., 2020). Continual learning is especially difficult in MARL environments because each agent's adaptation changes the environment for all other agents (Hernández Leal et al., 2019; Zhang et al., 2021). Therefore, meta-learning frameworks must consider strategic non-stationarity, so that agents can adapt not just to the changes in the environment, but also to the changes in the coordination strategies employed by the other agents (Wang et al., 2016; Hernández Leal et al., 2019). Meta-learning also enhances the robustness of MARL systems. Exposing agents to a variety of training scenarios, including those that represent rare disruptions, helps improve the resilience and generalization of agents (Finn et al., 2017; Nichol et al., 2018). Rare disruptions can have significant impacts on logistics systems, so the improvement in generalization and resilience provided by meta-learning is important. Meta-learning and continual learning provide a means to reduce the cost of retraining MARL systems and to deploy them for longer periods of time (Parisi et al., 2019; Lesort et al., 2020). These two methodologies transform the learning process from periodic retraining to continuous refinement of the MARL systems, which matches the operational cycle of enterprises (Parisi et al., 2019; Lesort et al., 2020).

A simplified meta-learning objective can be expressed as:

$$\min \, _{\mathcal{T}}[\mathscr{L}_{\mathcal{T}}(\theta - \alpha\nabla_{\theta}\mathscr{L}_{\mathcal{T}}(\theta))] \ _{\theta}$$

capturing rapid adaptation across tasks.

Figure 6: Multi-Agent Reinforcement Learning Algorithms


Deep Multi-Agent Reinforcement Learning Algorithms
Deep Q-Networks (DQN)
Actor-Critic Methods
Proximal Policy Optimization
Agent 1
Agent 2
Critic
Actor
Old Policy
New Policy
TD Error
Uses deep Q-networks to estimate Q-values
Agents maximize long-term expected rewards
Challenges: Non-stationarity of Bellman updates.
Separates policy (Actor) from value estimate (Critic)
Critic shapes policy updates without full observability.
Challenges: Coupling of actor learning with critic
Directly optimizes agent-specific policies
Clips policy updates to prevent large changes.
Challenges: Balancing exploration with policy stability.
Shared Experience Buffer
Deep Q-Networks
Actor-Critic Methods
Proximal Policy (PPO)
Uses deep Q-networks to estimate Q-values.
Agents maximize long-term expected rewards
Challenges: Non-stationarity of Bellman updates.
Separates policy (Actor) from value estimate (Critic)
Critic shapes policy updates without full observability.
Challenges: Coupling of actor learning with critic estimates.
Directly optimizes agent-specific policies
Clips policy updates to prevent large changes.
Challenges: Balancing exploration with policy stability.
Shared Experience Buffer
Agent 1
State
Joint Action
Reward
Next State
Multi-Agent Dynamics: Other agents' actions alter the effective environment for each agent.
Multi-Agent Dynamics.
Other agents actions alter the effective environment for each agent.

The Diagram 6 represents a single technical approach for understanding how three basic paradigms of

Deep Reinforcement Learning (DRL) - Deep Q Networks (DQNs), Actor-Critic Methods and Proximal Policy Optimization (PPO) - can be understood as interconnected learning processes in multi-agent logistics environments rather than isolated algorithms. On the left is a representation of the DQN block, illustrating agents that learn action-value functions mapping high-dimensional logistics states (e.g., inventory position; fleet location; level of congestion; etc.) to low-horizon utility estimates. The block emphasizes that updates to Q-values are determined by delayed rewards and by the shared use of experience replay, a mechanism that can become unstable when multiple agents interact because of policy-induced non-stationarity. In the center is a representation of the Actor-Critic block, explicitly separating the actor from the critic. The actor generates stochastic or continuous logistics actions (e.g., routing probabilities; dispatch priority; etc.). The critic assesses the quality of joint state-action pairs and provides low-variance learning signals to the actor, thus enabling coordination among agents even under partial observability and delayed system-level feedback. On the right is a representation of the PPO block, highlighting the constrained nature of policy updates via trust-region-style clipping. This visually reinforces the idea that logistics systems require slow policy adaptation to avoid oscillatory or unsafe operational behavior. Across all three paradigms, the shared experience buffer and joint reward paths depicted at the bottom illustrate that learning in a coupled environment occurs where each agent's update changes the effective dynamics experienced by other agents, making coordination an emergent property of the learning architecture, not an explicit control signal. Diagram 6 also captures the theoretical insight that the primary differences among these algorithms arise from their differing approaches to managing the bias-variance trade-off and stability under coupling. DQN relies on approximating value-iteration-based learning; Actor-Critic relies on structured gradient information to improve performance; and PPO constrains policy updates based on the agent's risk tolerance for operational failures. Overall, Diagram 6 conveys that in multi-agent logistics, learning algorithms function as system-wide coordination mechanisms that translate delayed, multi-objective rewards into stable, decentralized policies within the constraints imposed by non-stationarity, scalability, and feasibility of deployment in the real-world.

Simulation and Environment Modeling

Synthetic Simulation of Logistics Networks

Simulation of synthetic systems is the primary methodological approach for assessing intelligent multi-agent reinforcement learning systems in logistics (Shapiro et al., 2014; Powell, 2019). In contrast to traditional optimization approaches which assume static parameters and deterministic dynamics, MARL systems are inherently dynamic, nonlinear, and sensitive to interaction effects between agents (Dulac-Arnold et al., 2021; Henderson et al., 2018). Therefore, analytical solutions cannot be found in a realistic setting and simulation is the only viable option for controlled experimentation (Tako & Robinson, 2012). From a theoretical point of view, simulation represents an approximation of the underlying stochastic game, enabling the examination of policy behavior, coordination dynamics and scalability properties in repeated conditions (Bonabeau, 2002; Macal & North, 2010). Synthetic simulation environments in logistics research are created to simulate the basic structure of a network of real-world logistics systems (e.g., warehouses, vehicle fleets, transportation routes, etc.) (Toth & Vigo, 2014). Each element is abstracted into a representation that retains the causal relationships inherent in each element while providing for controlled variation in complexity (Davidsson et al.,

2005). Warehouses are represented as service nodes with limitations in capacity, queueing delay and processing time (Tako & Robinson, 2012). Vehicle fleets are represented as moving agents with routing, scheduling and energy limitations (Toth & Vigo, 2014). Travel routes represent the physical connections between locations and include both spatial distances and travel time distributions along with the possibility of congestion (Vlahogianni et al., 2014; Clark & Watling, 2005). The ultimate goal of such abstraction is not to model reality, but rather to accurately capture the structure of interactions so that the behaviors learned through simulation will generalize to other environments (Henderson et al., 2018).

Synthetic simulations provide the ability to systematically examine and compare coordination phenomena that cannot be isolated or compared in real world systems (Clark & Watling, 2005). For example, congestion cascades, oscillatory routing behavior and coordination failures can be caused intentionally through changes in network topology or demand levels (Clark & Watling, 2005). Researchers can then examine how various

MARL architectures respond to such "stress" conditions and determine if the learned policies demonstrate stability, robustness or unstable/patological behavior (Dulac-Arnold et al., 2021; Henderson et al., 2018). From a control theoretic perspective, simulation provides the ability to analyze the closed-loop system behavior under different control policies to reveal emergent behaviors that are not evident from the local decision making rules (Powell, 2019).

Another key function of synthetic simulation is the evaluation of scalability. While real logistics networks can contain thousands of agents, it is practically difficult to deploy experimental MARL systems of this size during the development stage. Simulation allows for gradual scaling of the number of agents, the size of the network and the level of interaction among agents to allow researchers to study the evolution of learning dynamics as system complexity grows (Henderson et al., 2018). This is critical in validating design decisions regarding the architecture of MARL systems, such as value decomposition, sparse communication and hierarchical coordination (Dulac-Arnold et al., 2021).

Synthetic simulation environments provide researchers with the ability to conduct controlled ablation experiments. Components of a MARL system, such as coordination mechanisms and/or reward structures, can be selectively enabled or disabled to evaluate their effect on the performance and stability of the overall system (Henderson et al., 2018). Such controlled experimentation is in line with scientific principles of establishing cause-and-effect and supports the validity of experimental conclusions. Controlled experimentation in operational logistics settings would not be possible without simulation (Tako & Robinson, 2012). From a learning perspective, simulation functions as a data generator producing diverse trajectories across a wide range of operating conditions (Henderson et al., 2018; Dulac-Arnold et al., 2021). Diverse trajectories are necessary for developing robust policies that generalize beyond narrow scenarios. Through manipulating network topologies, demand patterns and the frequency of disruptions, simulation provides agents with experience with rare but potentially significant events that are under-represented in historical data (Henderson et al., 2018; Fu et al., 2020). Theoretical rigor in designing synthetic simulation environments necessitates careful consideration of time discretization, event scheduling and synchronization among agents (Tako & Robinson, 2012). Logistics systems typically consist of combinations of continuous-time processes (e.g., movement of vehicles) and discrete events (e.g., arrival of orders). Simulation frameworks must reconcile these differing time structures without introducing artifacts that affect the learning dynamics (Tako & Robinson, 2012). Lastly, synthetic simulation provides a risk management capability for containing the risks associated with failure during development, identifying failure modes that would be catastrophic in production (Dulac-Arnold et al., 2021). As logistics systems errors result in financial, safety and reputational implications, the ability to fail safely during the development phase is essential.

A high-level representation of the simulated environment can be represented as a stochastic transition process:

$$s_{t+1} \sim (s_t, \mathbf{a}_t, \xi_t)$$

where $\xi_t$represents exogenous stochastic influences.

Environment Design Using OpenAI Gym or Unity ML-Agents

A key function of Simulation environments for Multi-Agent Reinforcement Learning (MARL) is to provide an abstraction layer that separates the learning algorithm and the System Dynamics (Brockman et al., 2016; Juliani et al., 2018). For example, both OpenAI Gym and Unity ML-Agents are best viewed as abstraction layers that formally define the interface between the system dynamics and the agent logic (Brockman et al., 2016; Juliani et al., 2018). As abstraction layers, they provide a significant theoretical contribution by providing a strict separation of concerns, between the Agent Logic and the Environment Dynamics that provides the basis for reproducible and modular MARL Research (Henderson et al., 2018). When designing environments for Logistics, this includes defining Observation Spaces, Action Spaces, Reward Signals, and Transition Dynamics in a way that accurately captures operational constraints (Clark & Watling, 2005; Toth & Vigo, 2014). Furthermore, Agents must receive observations based on local and partial information available at each agent location, for example, local Inventory Levels or Congestion at neighboring intersections (Clark & Watling, 2005). The Action Space defines the possible actions an Agent can take, including Routing Choices, Dispatch Timing, and Inventory Repositioning (Toth & Vigo, 2014). Designing the Interface poorly can result in unintended leakage of Global Information or Unintended Control Authority to the Agent, resulting in Misleading Results (Henderson et al., 2018). OpenAI Gym Environments emphasize Discrete-Time Decision Making and Episodic Interaction, which matches the nature of many Logistics Problems, including Routing and Scheduling (Brockman et al., 2016; Toth & Vigo, 2014). By comparison, Unity ML-Agents provides support for Continuous-Time Simulation and Physics-Based Environments, allowing for the Modeling of Vehicle Motion, Robotic Coordination, and Spatial Effects due to Congestion (Tobin et al., 2017; Peng et al., 2018; Juliani et al., 2018). Therefore, the selection of a specific Framework is a reflection of the Theoretical Assumptions made regarding Time, Continuity, and Observability (Brockman et al., 2016; Juliani et al., 2018).

One of the most important Design Considerations when Developing Environments is how to Define Episode Boundaries. In Logistics Systems, Operations are Continuous, and Artificial Termination of Episodes can Introduce Learning Artifacts (Henderson et al., 2018; Pardo et al., 2018). Therefore, Environment Designers Must Choose Between Defining Episodic Formulations for Tractability and Continuing Tasks that Better Represent Reality. This Choice has Theoretical Implications for Convergence Guarantees and Policy Evaluation (Schulman et al., 2015; Henderson et al., 2018). Another Important Aspect of Environment Design is Handling Multi-Agent Synchronization. Environments Must Specify Whether Agents Act Simultaneously or Asynchronously and How Conflicts Are Resolved (Macal & North, 2010; Davidsson et al., 2005). These Choices Can Influence the Effective Game Structure and Significantly Alter Learning Dynamics (Henderson et al., 2018). Reset Conditions and Initial State Distributions are Other Important Environmental Design Considerations. These Choices Shape the Training Distribution and Influence Generalization. Narrow Initial Distributions May Lead to Over-Fitting, While Random Initialization May Slow Learning (Henderson et al., 2018). Finally, from a Software Architecture Perspective, Environment Frameworks Provide Modular Experimentation by Decoupling the Simulation Logic from the Learning Algorithm (Brockman et al., 2016; Juliani et al., 2018). This Modularity Supports Comparative Evaluation of Different MARL Strategies Under Identical Conditions and Increases Empirical Validity (Henderson et al., 2018).

The environment–agent interaction can be abstracted as:

$(o_{it+1}, r_{it}) = \mathscr{E}(s_t, a_{it})$

where $\mathscr{E}$denotes the environment interface.

Incorporation of Stochastic Elements in Logistics Simulation

To test how well MARL systems adapt to unpredictable real-world logistics conditions, it is crucial to include stochastic elements into your simulation environment to analyze their ability to operate robustly (Powell, 2019; Clark & Watling, 2005; Shapiro et al., 2014). Theoretically speaking, when you include stochastic variables in your model, they create uncertainty for both the agent's transition from one state to another, and the reward function itself, thereby converting what was once a deterministic problem into a stochastic decision process (Shapiro et al., 2014; Powell, 2019). The incorporation of stochastic variables creates the need for the agent to reason probabilistically and develop policies which will perform well "on average" versus optimal (Henderson et al., 2018; Dulac-Arnold et al., 2021). Traffic variability is commonly simulated by applying stochastic travel-time distributions to traffic conditions based upon time-of-day, level of congestion, and random events (Vlahogianni et al., 2014; Clark & Watling, 2005). Demand spikes can be represented by using non-stationary arrival processes, including time-varying Poisson or bursty processes (Powell, 2019). Weather-induced disruptions have a synergistic effect on all agents within an area experiencing a disruption, requiring coordination among agents operating in areas affected by the same event (Dulac-Arnold et al., 2021).

The inclusion of stochastic variables into a simulation environment also allows you to analyze the susceptibility of various learning algorithms to stochastic variability (Henderson et al., 2018). A high-variance reward structure can cause instability during training and/or learning, whereas an excessively smoothed stochastic representation can result in understatement of risks. Therefore, the designer of a simulation environment must carefully calibrate the stochastic processes to achieve a reasonable trade-off between realism and learnability (Henderson et al., 2018; Dulac-Arnold et al., 2021). The use of stochastic variables in a simulation environment allows for the stress-testing of MARL architectures. By either increasing the amount of stochastic variability introduced into the simulation or by simulating rare but catastrophic events, you can evaluate if the policies developed will degrade gracefully or catastrophically fail (Dulac-Arnold et al., 2021). These are both critical aspects to evaluate for assessing the resilience and safety of MARL architectures.

Theoretical studies of stochastic environments focus on developing and analyzing expectations-based performance measures instead of providing deterministic guarantees (Shapiro et al., 2014; Powell, 2019). Consequently, learning objectives must take into consideration the likelihood of probabilistic outcomes, and evaluation procedures should also account for the variance of outcomes across simulation runs (Henderson et al., 2018). Finally, the inclusion of stochastic variables into a simulation environment has implications for exploration strategies used by agents. For example, in very stochastic environments, it can become increasingly difficult to distinguish between structural changes in the environment versus the inherent randomness of the stochastic environment, which can complicate learning (Henderson et al., 2018). Using a simulation environment provides a mechanism to experimentally investigate these interactions (Henderson et al., 2018).

A stochastic demand model can be abstractly represented as:

$D_t \sim \mathcal{D}(\lambda_t)$

where $\lambda_t$varies over time.

Multi-Agent Digital Twin Environments for Real-World Testing

A Multi-Agent Digital Twin Environment is the most advanced and theoretically supported method to bridge the gap between simulated learning and real-world implementation of Intelligent MARL Systems in Logistics (Grieves & Vickers, 2017; Minerva et al., 2020). A Digital Twin is not simply a High-Fidelity Simulator but rather a Continuously Synchronized Virtual Representation of a Physical Logistics System that Mirrors

Operational States, Constraints and Dynamics in Near Real Time (Grieves & Vickers, 2017; Tao et al., 2019; Fuller et al., 2020). In MARL systems, the Digital Twin serves as a Controlled Experimental Substrate in which Decentralized Learning Agents Can Be Evaluated, Stress Tested and Refined Without Risking Operations (Wang et al., 2022; Le & Fan, 2024). This Changes Simulation from Being an Offline Research Tool to a Live Component of the Learning/Governance Architecture (Minerva et al., 2020; Lu et al., 2020).

Theoretically speaking, a Multi-Agent Digital Twin can be viewed as a Dynamic Approximation of the Underlying Stochastic Game Controlling the Logistics Network (Grieves & Vickers, 2017; Minerva et al., 2020). Static Simulations Assume Fixed Transition Kernels Whereas the Digital Twin Updates Its State Using Real-Time Data Which Allows the Transition Dynamics to Change Over Time (Minerva et al., 2020; Rasheed et al., 2020). This is Important Because Logistics Environments are Non-Stationary: Demand Patterns Shift, Infrastructure Changes, Human Interventions Alter the Behavior of the System. The Digital Twin Thereby Approximates a Time-Varying Stochastic Game Allowing MARL Agents to Learn Policies That Are Valid Under Changing Conditions Rather Than Converging to Fragile Solutions Optimized for Assumptions Based on Stationarity (Le & Fan, 2024; van der Valk et al., 2022).

An important aspect of Multi-Agent Digital Twins is Bidirectional Coupling Between the Physical System and the Virtual Environment (Fuller et al., 2020; Minerva et al., 2020). The Digital Twin's State Representation Is Continuously Updated Using Sensor Data, Telematics Streams, Inventory Updates and Enterprise Transaction Logs (Minerva et al., 2020). At the Same Time, the Digital Twin Can Be Used to Evaluate Candidate Policies, Coordination Strategies, Architectural Modifications Etc. Prior to Their Deployment (Grieves & Vickers, 2017; Wang et al., 2022). This Bidirectional Coupling Enables Counterfactual Reasoning: the System Can Ask How Alternative Joint Policies Would Have Performed Under the Same Observed Conditions (Rasheed et al., 2020; Le & Fan, 2024). From a Learning-Theoretic Perspective, This Supports Off-Policy Evaluation and Policy Improvement Without Violating Safety Constraints (Thomas & Brunskill, 2016; Levine et al., 2020).

In MARL Systems, Coordination Failures Often Only Emerge Under Specific Combinations of Agent Density, Demand Surges or Correlated Disruptions. These Rare But Highly Impactful Regimes are Difficult to Observe During Limited Offline Simulation or Historical Replay. Digital Twins Allow for Targeted Exploration of Such Regimes by Injecting Hypothetical Disruptions/Stress Scenarios While Maintaining Realistic Interaction Structure (Wang et al., 2022; Le & Fan, 2024). For Example, Correlated Warehouse Outages or Cascading

Traffic Incidents Can Be Simulated to Determine Whether Coordination Mechanisms Remain Stable. This Capability is Necessary for Assessing Resilience Which Cannot be Inferred from Average-Case Performance Alone (Ivanov & Dolgui, 2020; Hosseini et al., 2019).

Another Key Theoretical Role of Digital Twins is to Validate Policies Under Decentralized Execution Constraints. Policies Trained Using Centralized Training Mechanisms Must Ultimately be Executed by Agents with Partial Observability and Limited Communication. The Digital Twin Imposes these Execution Constraints While Still Permitting Full Observability for Evaluation (Minerva et al., 2020; Le & Fan, 2024).

This Separation Enables Researchers to Detect Training – Execution Mismatches Where Policies Perform Well Under Training Assumptions but Fail Under Deployment Constraints (Henderson et al., 2018; Dulac-Arnold et al., 2021). Detection and Correction of Such Mismatches Before Actual Rollout is One of the Strongest Arguments in Favor of Integrating Digital Twins into MARL Architectures (Le & Fan, 2024; Fuller et al., 2020).

Digital Twins Also Enable Closed-Loop Learning Architectures, Where Policy Updates Are Driven by Both Simulated Results and Real-Time Feedback from Operations (Grieves & Vickers, 2017; Minerva et al., 2020). Rather Than Retraining Policies Episodically from Scratch, MARL Systems Can Be Incrementally Improved by Testing Candidate Updates in the Twin, Measuring Their Impact and Selectively Deploying Improvements (Le & Fan, 2024; Wang et al., 2022). This Enables Ongoing Learning to Occur Without Jeopardizing Operational Stability (Levine et al., 2020; Dulac-Arnold et al., 2021). Theoretically, This Represents a Two-Timescale Learning Process: Rapid Policy Evaluation in the Twin and Gradual Policy Adaptation in the Physical System, Reducing the Likelihood of Unstable Feedback Loops (Grieves & Vickers, 2017; Rasheed et al., 2020).

From an Architectural Point of View, Multi-Agent Digital Twins Must Preserve Agent Heterogeneity and Topology of Interactions (Wang et al., 2022; Lu et al., 2020). Logistics Networks Include Different Types of Agents with Diverse Action Spaces, Constraints and Objectives. Vehicles, Warehouses, Hubs and Planners Interact Through Shared Resources but Operate Under Different Decision Horizons. A Digital Twin That Collapses Agent Heterogeneity Into Homogeneous Agents May Misrepresent Coordination Dynamics.

Therefore, High-Quality Digital Twins Maintain Explicit Agent Roles, Interaction Graphs and Constraint Sets Ensuring That Learned Coordination Strategies Remain Structurally Valid (Minerva et al., 2020; Le & Fan, 2024).

One of the Most Significant Theoretical Challenges in Developing Digital Twins is Finding the Right Balance Between Model Fidelity and Computational Feasibility (Barricelli et al., 2019; Tao et al., 2019). Digital Twins That Are Too Detailed May Capture the Physical Dynamics Accurately But Become Infeasible to Run Large-Scale MARL Experiments Due to Computational Requirements. On the Other Hand, Digital Twins That are Too Abstract May Miss Critical Effects of Interaction Leading to False Confidence in the Policies That Were Learned. Theoretical Guidance Suggests That Fidelity Should Be Allocated Preferentially to Components of Interaction Which Will Influence Outcome (Rasheed et al., 2020; Fuller et al., 2020). Examples of These Types of Components Include Congestion Propagation, Capacity Constraints and Demand Coupling Rather Than Low-Impact Details (Le & Fan, 2024; van der Valk et al., 2022). This Selective Allocation of Fidelity Preserves Learning Relevance While Maintaining Scalability.

Digital Twins Also Play a Key Role in Governance, Explainability and Trust (Grieves & Vickers, 2017; Fuller et al., 2020). Autonomous Logistics Decisions Affect Cost, Safety, Labor and Regulatory Compliance. Digital Twins Provide a Transparent Environment Where Decision Logic Can be Inspected, Stress-Tested and Audited

(Minerva et al., 2020; Wang et al., 2022). For MARL Systems, Where Emergent Behavior Can be Difficult to Interpret, Ability to Replay Scenarios and Trace Coordination Outcomes is Essential for Acceptance Within Organizations (Le & Fan, 2024; Rasheed et al., 2020). From a Theoretical Perspective, This Provides Post-Hoc Analysis of Equilibrium Selection, Stability Properties and Failure Modes (Henderson et al., 2018).

Finally, Multi-Agent Digital Twins Extend the Role of MARL Beyond Operational Optimization into Strategic Planning Support (van der Valk et al., 2022; Wang et al., 2022). By Allowing for Scenario Analysis and Long-Horizon Experimentation, Digital Twins Allow Organizations to Assess Infrastructure Investments, Policy Changes or Coordination Strategies Prior to Committing Resources (Grieves & Vickers, 2017; Le & Fan, 2024). This Transforms MARL Systems from Reactive Controllers into Proactive Planning Tools.

Theoretically, This Represents a Shift from Episodic Control Optimization to Adaptive System Design, Where Learning Agents Inform Strategic Decisions Regarding the Structure of the Logistics Network Itself (Tao et al., 2019; Lu et al., 2020).

A simplified digital twin update can be expressed as:

$$s_t\text{twin} \leftarrow s_t\text{real} + \epsilon_t$$

where $\epsilon_t$captures modeling error.

Figure 7: Multi-Agent Twin environment for real-world testing

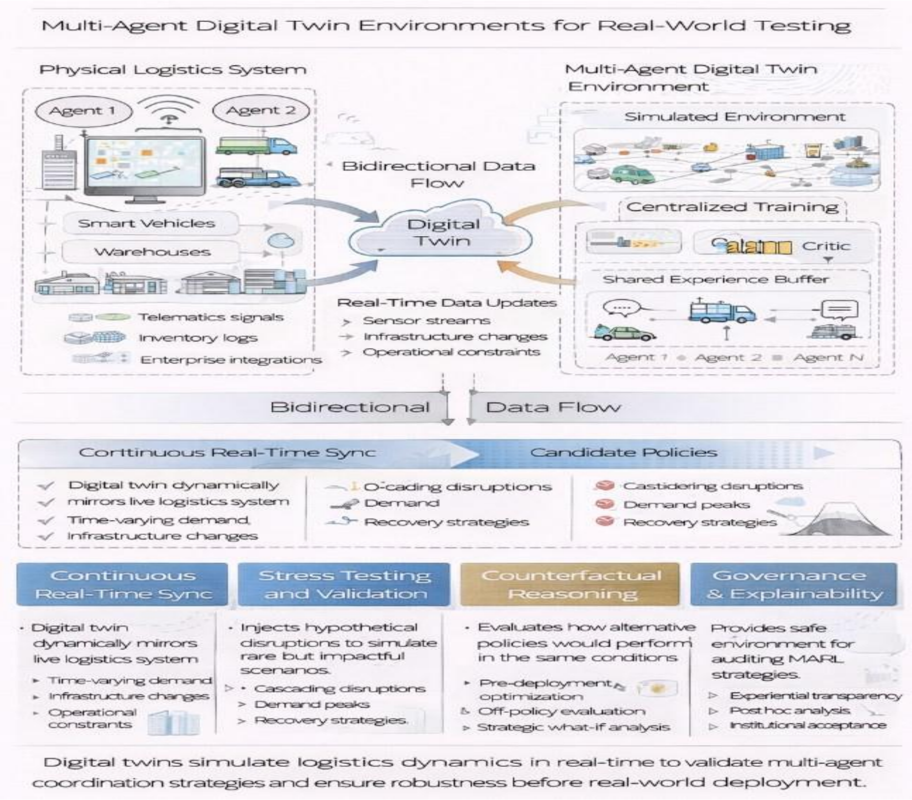

Multi-Agent Digital Twin Environments for Real-World Testing
Physical Logistics System
Agent 1
Agent 2
Smart Vehicles
Warehouses
Telematics signals
Inventory logs
Enterprise integrations
Bidirectional Data Flow
Digital Twin
Real-Time Data Updates
Sensor streams
Infrastructure changes
Operational constraints
Multi-Agent Digital Twin Environment
Simulated Environment
Centralized Training
Critic
Shared Experience Buffer
Agent 1
Agent 2
Agent N
Bidirectional
Data Flow
Corttinuous Real-Time Sync
Candidate Policies
Digital twin dynamically mirrors live logistics system
Time-varying demand,
Infrastructure changes
O-cading disruptions
Demand
Recovery strategies
Castidering disruptions
Demand peaks
Recovery strategies
Continuous Real-Time Sync
Stress Testing and Validation
Counterfactual Reasoning
Governance & Explainability
Digital twin dynamically mirrors live logistics system
Time-varying demand
Infrastructure changes
Operational constraints
Injects hypothetical disruptions to simulate rare but impactful scenarios.
Cascading disruptions
Demand peaks
Recovery strategies.
Evaluates how alternative policies would perform in the same conditions
Pre-deployment optimization
Off-policy evaluation
Strategic what-if analysis
Provides safe environment for auditing MARL strategies.
Experiential transparency
Post hoc analysis
Institutional acceptance
Digital twins simulate logistics dynamics in real-time to validate multi-agent coordination strategies and ensure robustness before real-world deployment.

Diagram 7 illustrates the system-level structure of an integrated multi-agent digital twin and physical logistics system, to support safe, adaptive and theoretically-grounded multi-agent reinforcement learning (MARL) applications. The physical logistics system provides an ongoing flow of operational information about its operation via a variety of sources including sensors, telematics, inventory systems and enterprise transaction records. Each of these types of information provide the digital twin with a dynamic picture of the current status of all vehicles, warehouses, hubs, demand, and constraints imposed by the infrastructure. As a result, this type of operational information is fed bidirectionally into the digital twin which exists as a synchronous simulation model of the real logistics system. The environmental dynamics within the digital twin are continuously updated to represent the

nonstationarity found in the real world; agents can therefore interact in a simulated environment that models the true nature of congestion propagation, capacity coupling and disruption effects. In addition, the centralized training elements exist within the digital twin, providing global access to the state of the entire logistics system, joint rewards, and centralized critics to allow for the development of stable learning and coordination analysis to occur within a reasonable timeframe, and without making unrealistic assumptions about the way in which the agents will be deployed. Candidate policies and coordination strategies are tested in a controlled yet representative manner, which includes a wide range of stressful scenarios such as correlated disruptions, demand surges, or failure of the infrastructure. Such testing enables counterfactual reasoning and off-policy evaluation, and provides the ability to develop and implement decision-making processes that have been validated against a large number of potential scenarios. Concurrently, the digital twin enforces the constraints of decentralized execution so that agents are able to only act upon local observations and limited coordination signals, and exposes training-execution mismatches prior to the agent being deployed in a real-world scenario. Finally, feedback from policy evaluation is returned to the learning loop, to facilitate incremental policy refinement rather than requiring the implementation of disruptive retraining, and thus creates two timescales of learning process. Rapid experimentation occurs in a virtual environment, while slow, risk aware updates occur in the real-world environment. The diagram also highlights the governance role of the digital twin, and shows how audit trails, explanation artifacts, and performance diagnostic tools arise naturally from the same architecture that is used for learning/testing. Overall, the figure illustrates that the digital twin is not simply a supportive tool, but rather an integral component of the MARL control and governance stack, and supports continuous alignment between learned coordination strategies, real-world operational constraints, and organizational risk tolerance; and transforms autonomous logistics control into a measurable, testable, and auditable system.

Real-Time Network Optimization

Real-time network optimization serves as the operational core of intelligent logistics systems; multi-agent reinforcement learning enables continuous, adaptive control over routing, scheduling and resource allocation under uncertainty (Lin et al., 2018). Most traditional methods of logistics optimization create static, or periodically revised plans, based upon either deterministic or stochastic programming models that assume relatively stationary system behavior. However, real-time logistics are defined by non-stationarity of demand, stochastic travel times, infrastructure failures, and tightly coupled agent interactions that make stationary optimization fragile (Clark & Watling, 2005). Within this framework, MARL reformulates routing and load balancing as sequential decision-making processes in which agents continually sense changing network states, and choose actions that affect both current and future system behaviors (Lowe et al., 2017). Additionally, dynamic routing driven by reinforcement learning, permits agents to account for congestion externalities by learning how their individual route choice affects overall traffic flow patterns, queue formation and service reliability (Roughgarden & Tardos, 2002). Similarly, load balancing is a distributed coordination problem among agents, who implicitly negotiate the sharing of available capacity among hubs, vehicles, and time slots, using learned policies instead of explicit centralized control. Theoretical implications stem from the transition from computing equilibria to tracking equilibria, where policies continually adjust to moving operating points, as opposed to converging to a single static point (Hall, 1978). This ability to adapt enables logistics networks to react dynamically to unanticipated events such as vehicle breakdowns, sudden demand spikes, or weather induced reductions in capacity, while maintaining their level of performance as system conditions continue to evolve unpredictably.

Adaptive scheduling and fleet management represent additional extensions of real-time optimization beyond just routing decisions into the temporal and organizational dimensions of logistics control (Wang et al., 2020). Scheduling in MARL frameworks treats delivery scheduling as a rolling horizon decision problem in which agents learn to rank tasks, assign vehicles, and schedule deliveries based on the current state of the system, rather than a predetermined plan (Joe & Lau, 2020). Coordination of fleets is a multi-agent control problem in which vehicles, depots, and hubs operate as independent decision-makers, whose policies must be coordinated in order to avoid inefficiencies such as idle time, missed time windows, or cascading delays. From a control theory perspective, effective scheduling policies reduce variability and limit the propagation of disturbances throughout the network and serve as stabilizing feedback mechanisms (Konda & Tsitsiklis, 2003).

Optimization with multiple objectives is inherent within this process as real-time decisions need to address multiple conflicting objectives, including delivery time, operational costs, energy consumption and environmental emissions (Roijers et al., 2014). MARL accommodates these trade-offs by learning policies that incorporate weighted or vector valued reward structures, permitting agents to dynamically adjust their priorities for objectives as the operating conditions change. For example, during periods of congestion, or high demand, a policy may prioritize service reliability over minimizing costs; whereas, when there is no congestion or other issues, a policy may focus on reducing energy consumption and environmental emissions. The adaptive weighting of objectives in MARL, thus differentiates it from fixed objective optimization, and links operational control with broader goals related to sustainability and resilience.

Feedback loops provide the mechanism that enable real-time optimization to remain viable over long-term operation (Schulman et al., 2017). Feedback loops exist at multiple time scales in MARL-driven logistics systems: short time scale loops govern immediate control actions, such as routing and dispatching, while longer time-scale loops update policies based on accumulated experience and observed performance. These feedback loops permit autonomous improvement of policies by providing agents with the opportunity to learn from discrepancies between expected and actual outcomes, such as unexpected congestion, or delayed deliveries (Mnih et al., 2015). Theoretically, this results in a coupled learning-control system in which policy updates affect the underlying environment, which in turn affects future learning signals. Maintaining this coupling is important for stability as overly aggressive updates may lead to oscillating behavior, while overly conservative updates will reduce the rate of adaptation. From the perspective of the organization, continuous feedback enables systems to evolve with the organization, incorporating new constraints, service objectives, or regulatory requirements without requiring complete redesign. Therefore, real-time network optimization transforms logistics operations from static execution engines to adaptive cyber physical systems that can learn, self-correct and improve over time. The strategic implications include a shift towards logistics networks that are not merely optimized but also self-optimizing, continually adjusting their operational decisions relative to the organization’s objectives, under uncertain conditions.

Integration with Supply Chain Management Systems

Interfacing MARL Models with ERP, SCM, and WMS Platforms

The convergence of multi-agent reinforcement learning (MARL) systems and existing Enterprise Resource Planning (ERP), Supply Chain Management (SCM), and Warehouse Management Systems (WMS) represents a paradigmatic shift in how supply chain intelligence is operationalized (Jacobs & Weston, 2007). ERP, SCM, and WMS are not simply repositories of information, they are encoded embodiments of organizational policy, contractually enforceable logic, compliance requirements, and financial obligations (Lambert & Cooper, 2000). From a theoretical point-of-view, the introduction of MARL systems into the environment of traditional deterministic workflows and pre-determined rules creates an opportunity for adaptive, probabilistic decision making processes to occur (Vernadat, 2007). Therefore, the primary challenge is not establishing technical connectivity between MARL and Enterprise Systems, but reconciling two fundamentally different control paradigms: learning-based adaptive control and rule-based enterprise governance (Vernadat, 2007).

In order for successful integration to occur, an architectural mediation layer will need to be established so that MARL agents can make decisions that are bounded by the feasibility constraints defined by the enterprise, while still being able to learn and adapt under conditions of uncertainty (Chen et al., 2008).

Enterprise Systems operate over a variety of planning time horizons, including long-term financial planning and procurement, as well as short-term operational execution. By contrast, MARL systems operate over very fine time scales, where decisions are made continuously based on a stream of observation. Therefore, theoretically, a hierarchical control architecture needs to be established whereby ERP and SCM Systems provide high level objective definitions, constraint definitions, and priority definitions for the MARL Agents to execute against at the operational execution layer in real-time (MacCarthy et al., 2016). Establishing this hierarchy ensures that the learned policies developed by MARL Agents aligns with the overall strategic objectives of the organization, thereby limiting the potential for local optimum behaviors that conflict with longer-term commitments such as Service Level Agreements (SLAs), contractual delivery windows, and Inventory Valuation Rules (Gunasekaran et al., 2001).

An additional theoretical requirement for successful integration is the establishment of semantic alignment between the state representation used by MARL Agents and the domain semantics embodied in the data models of ERP and WMS Platforms. In particular, ERP and WMS Platforms embody domain semantics such as Order Lifecycle States, Ownership Boundaries, Inventory Classifications, and Fulfillment Priorities. Therefore, MARL Agents must accurately internalize these semantics so that they do not develop policies that optimize surrogates that do not represent true business performance. To achieve this, explicit Schema Mapping, Ontology Alignment, and Semantic Validation Layers must be developed to map enterprise concepts into agent-interpretable State Variables (Chen et al., 2008). If semantic alignment does not exist, Reinforcement Learning may converge to policies that look like they are numerically efficient but produce operational outcomes that are misaligned with organizational intent.

Integrating WMS Platforms introduces additional layers of complexity since Warehouse Operations are governed by Physical Layout Constraints, Labor Rules, Safety Protocols, and Equipment Availability (Alyahya et al., 2016). Therefore, MARL Agents operating with WMS Systems must be able to operate under dynamic feasibility constraints. From a theoretical viewpoint, this means that Constraint-Aware Learning Mechanisms such as Action Masking or Feasibility-Condiioned Policies will need to be implemented to prevent MARL Agents from proposing actions that would violate physical or regulatory limits. Establishing constraint awareness is critical both from a safety standpoint, as well as from establishing trust in Autonomous Decision-Making Systems within Operational Environments (Lim et al., 2013).

SCM Platforms typically include Deterministic Planning Modules for Forecasting, Network Design, and Capacity Planning. MARL Agents do not replace these modules, rather they augment them through Adaptive Execution Intelligence that Compensate for Errors in Forecasts and Real-Time Volatility (Wang et al., 2016).

Therefore, this Layered Intelligence Architecture preserves the Strengths of Classical Optimization while Addressing its Limitations Under Conditions of Uncertainty. From a Systems Theory Perspective, this represents a Hybrid Control Architecture where Planning Determines Feasible Regions and Learning Optimizes Trajectories Within Those Regions (He et al., 2020).

From a Business Impact Standpoint, Effective Integration Enables Organizations to Move from Reactive Exception Handling to Proactive Self-Optimizing Operations. Decision Latency is Reduced, Manual Intervention Decreases, and Operational Resilience Improves (Wamba et al., 2017). However, these benefits can only be achieved if MARL Decisions are Transparent, Auditable, and Reversible Within Enterprise Systems. Therefore, Integration Architectures Must Support Detailed Logging, Explainability Hooks, and Governance Workflows That Allow Human Oversight While Preserving Automation Benefits (Rahimi et al., 2016).

Integration Also Redefines Organizational Decision Authority. Integrating MARL into ERP-Driven Workflows Transfers Control from Static Rules to Adaptive Policies. Therefore, The Socio-Technical Shift Requires Phased Deployment, Human-In-The-Loop Validation, and Gradual Trust Building. Theoretical Frameworks from Organizational Systems Emphasize that Autonomy Must be Introduced Incrementally to Avoid Resistance and Unintended Consequences (MacCarthy et al., 2016). Ultimately, Interfacing MARL with ERP, SCM, and WMS Platforms Determines Whether Reinforcement Learning Remains an Experimental Optimization Tool or Becomes a Core Operational Capability. Therefore, Integration is Not an Implementation Detail, Rather It is a Central Theoretical and Architectural Concern That Defines the Viability of Intelligent Logistics Systems.

Data Ingestion Pipelines from IoT Sensors and Telematics

The perceptual base of MARL-based Supply Chain Systems are Data Ingestion Pipelines (Ben-Daya et al., 2019), which are different from traditional analytical pipelines, used for historical reporting, that have to send continuous low-latency high fidelity data-streams to enable decision making in real time. From a theoretical point of view, they represent the Observability Structure of the Learning Environment (Atzori et al., 2010) defining how parts of the system-state are observable by agents and with what time precision. These structures influence learning-stability, learning-speed and policy robustness in partially observable environments, like logistics, because the ingestion design determines how agents receive information about their environment. The IoT sensors (Gubbi et al., 2013) offer detailed insight into inventory level, equipment condition, environmental factors and process-states in warehouses and transport assets. Additionally, telematics (Gubbi et al., 2013) systems monitor the position, speed, fuel consumption, driving behavior and route-compliance of mobile agents. The integration of the different heterogeneous data-sources requires careful normalization, synchronization and noise-filtering. Theoretically, this process can be seen as State Estimation under

Uncertainty (He et al., 2020), transforming unstructured raw signals to structured observations, which can be used for Reinforcement Learning under Stochastic Dynamics (He et al., 2020).

Latency is the dominant bottleneck in the ingestion pipeline-design. Late observations will lead to a phase-lag in the control-loop, which will reduce the quality of the decisions made and can even cause instability in the learning-dynamics. As a result, ingestion architectures have to prioritize real-time ingestion, edge-processing and priority-routing of critical-signals to make sure that agents take decisions based on current instead of outdated information (Shi et al., 2016). Additionally, ingestion-pipelines need to be reliable and fault-tolerant. Failures of sensors, communication-breakdowns and irregularities in reporting are inherent in logistics environments. For this reason, MARL-systems have to be able to function properly in spite of missing or impaired data. Therefore, ingestion-pipelines have to include redundancy-mechanism, probabilistic imputation and confidence-aware observation-models (Brous et al., 2020), allowing agents to reason about the uncertainty they perceive instead of relying on perfect knowledge (Brous et al., 2020).

For businesses, well-functioning ingestion-pipelines provide real-time insights into all areas of the supply chain, enabling faster reactions to disturbances and better monitoring of performances (Wang et al., 2016). Poor quality of ingestion-data results in distrust towards autonomous systems, causing manual interventions and undermining the benefits of automation (Wang et al., 2016). Therefore, the quality of ingestion-data has a direct effect on the operational-performance as well as on the organizational-adoption of MARL-based systems. Another major architectural-challenge related to scalability. Large logistics-networks generate massive amounts of data from thousands of assets. To manage these large amounts of data ingestion-pipelines need to scale-outwards, while still guaranteeing ordering of the events, needed for learning. To achieve this scaling, distributed streaming architectures and event-driven processing are necessary to prevent bottlenecks (Shi et al., 2016).

In addition to the challenges mentioned above, ingestion-pipeline design needs to address security and privacy concerns. Telematics- and IoT-data contains operational and personal information, which are subject to various regulations. Therefore, ingestion-pipelines need to implement encryption, access-control and anonymization, resulting in trade-offs between data density and compliance (Brous et al., 2020). Finally, ingestion-pipelines need to address data-drift (Wang et al., 2016). Due to changes in operational-behavior, the statistical properties of the new data differ from the statistical properties of the data used during training. Therefore, ingestion pipelines need continuous monitoring and adaptation-mechanisms to maintain the distribution of the training- and deployment-environment. If no such mechanisms exist, MARL-policies will deteriorate silently over time.

API-Driven Real-Time Coordination Between Warehouse and Transport Nodes

API-driven coordination forms the operational control-surface through which MARL-based systems exert influence over distributed logistics-subsystems (Papazoglou & van den Heuvel, 2007). In an enterprise context, APIs are not only integration-tools, but formally defined control-surfaces translating learned policies into executable-actions within warehouse-management-systems, transportation-management-systems and fleet control-platforms. Theoretically, APIs form the interface between adaptive-learning-processes and deterministic-operational-executions. Therefore, this interface needs to be carefully designed to preserve the original intention of the MARL-policies, while enforcing feasibility, safety and compliance-constraints specific to physical-logistics-operation. A poor design of the APIs introduces latencies, ambiguities or distortions that can undermine both learning-stability and operational-reliability. Historically, warehouse and transport subsystems were decoupled and had to coordinate manually. Routing-decisions depend on docking-plans, allocation-plans and picking-orders, whereas transport-delays in the warehouse-propagate downstream into fleet-planning and delivery-commitments. With API-driven-coordination, these subsystems can now exchange state-information and commitment-to-action in real-time, establishing a distributed control-loop that facilitates synchronized-decision-making (Eugster et al., 2003). From a control-theory-perspective, the relationship between the subsystems can be considered as a coupled MIMO-system (Multi-Input Multi-Output System), in which APIs are the signal-channels transmitting state-feedback and control-actions across the subsystems.

A significant theoretical problem occurs due to the mismatch between MARL-policy-outputs and the operational-expectations of the enterprise-execution. MARL-agents typically produce probabilistic, high-level action-recommendations that are optimized for maximizing the long-term rewards. In contrast, operational systems require deterministic, accountable and executable commands that can be performed reliably. Therefore, the mediation-layer between the MARL-policy-outputs and the operational-execution has to perform action-concretization, constraint-enforcement and conflict-resolution. This mediation does not occur neutrally; it defines the actual action-space available to the MARL-agents and has to be explicitly modelled to prevent unintended feedback-loops that may deteriorate the learning-performance. A second key problem is the temporal-coordination. Since APIs operate under hard-latency constraints, delayed execution can invalidate or worsen decisions. Even minor delays can cause cascading effects in real-time logistics, such as missing time-windows or congestive-amplifications. Theoretical models of real-time-control highlight the necessity of bounded-delay and deterministically-predictable execution-time. Therefore, API-architectures need to focus on minimizing-latency communication, using asynchronous processing when possible, and providing deterministic-response-guarantees to maintain closed-loop stability (Vogels, 2009).

In addition to the problems mentioned so far, temporal-coordination is crucial when coordinating activities between multiple agents in logistics environments. Transactional-integrity is essential when executing coordinated activities across multiple agents and systems. Activities in logistics often consist of sequential steps, such as dispatching a vehicle and at the same time preparing the inventory. Partial execution of such an activity can lead to inconsistencies that are difficult and expensive to correct. Therefore, APIs need to provide atomicity, idempotence and rollback-mechanisms to ensure that coordinated activities are either completely executed or fail safely. Theoretically, this means that APIs have to provide additional distributed-transaction semantics in the learning-execution-cycle, increasing the complexity of designing these systems. The scalability of API-driven coordination adds to the complexity of designing these systems. As the number of agents in a MARL-based system grows, the amount of coordination traffic increases significantly. Therefore, the API-infrastructure needs to be able to handle high concurrency, avoiding to become a bottleneck. Principles from queueing-theory and distributed-systems can help here, as improperly scaled APIs can cause congestion that negatively impacts the learning- and execution-performance of the system. Therefore, scalable designs of API-driven coordination should use load-balancing, back-pressure-mechanisms and prioritization of critical control-messages.

Security considerations are an integral part of the design of APIs. APIs expose control-surfaces that, if compromised, can adversely affect physical operation, financial flows or safety-criteria. Therefore, APIs need to include authentication, authorization and ongoing-monitoring, to protect against unauthorized usage of APIs, and to account for accountability and traceability. From a business impact viewpoint, API-driven coordination decreases manual interventions, speeds up responses to disruptions and synchronizes warehouse and transport operations. This leads to increased throughput, lower dwell times and improved reliability of services (Wamba et al., 2017). However, these advantages can only be achieved, if APIs are designed as first class control-interfaces and not as ad-hoc integration points. Ultimately, APIs realize intelligence. They determine whether MARL-systems can work together seamlessly across distributed logistics-nodes or remain limited to independent decision-support.

System Interoperability with Digital Enterprise Ecosystems

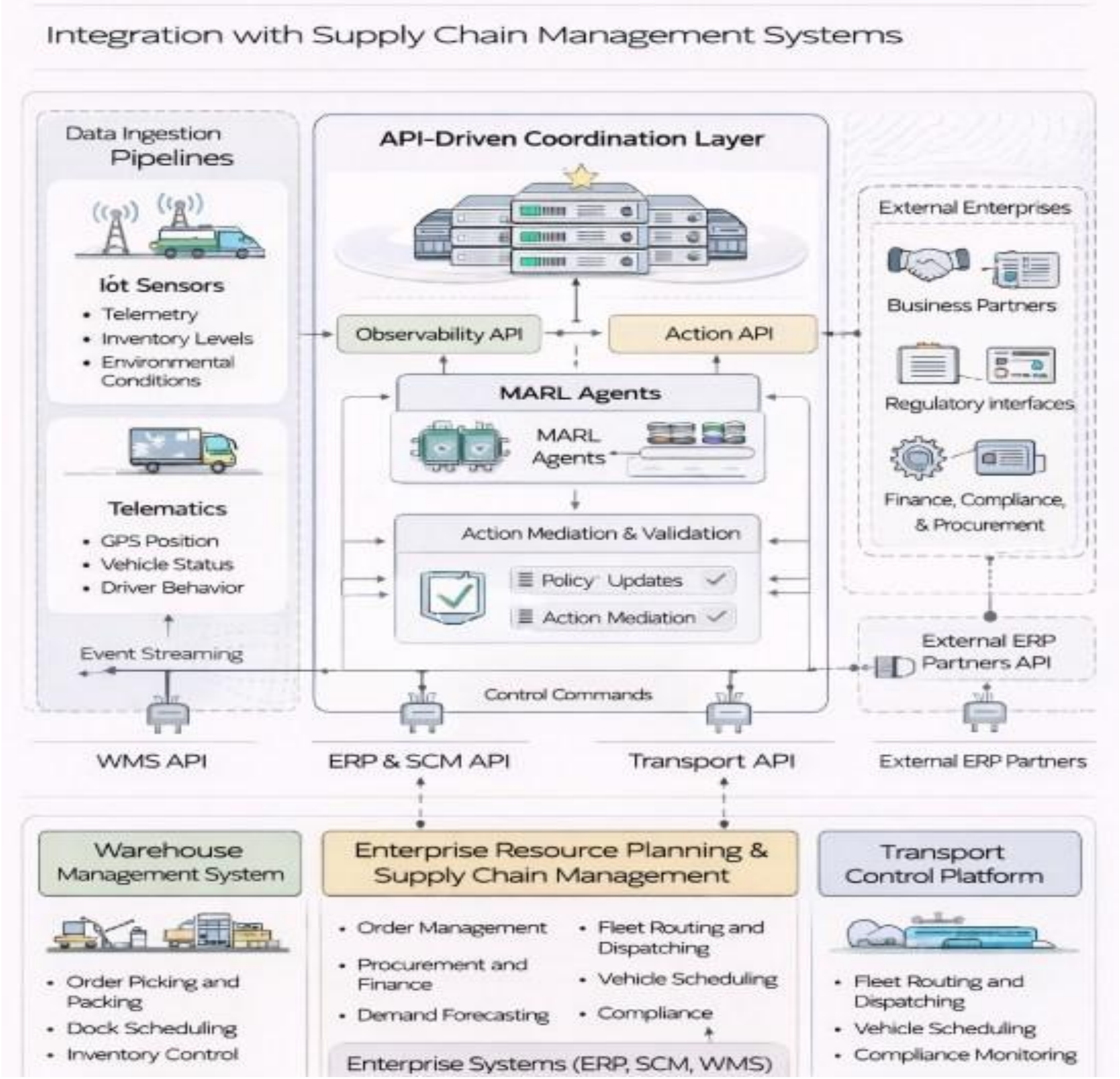
MARL will not only add to the internal logistical functions but will also provide integration into the wider digital enterprise environment; including, purchasing, accounting, compliance, partner networks, and regulatory interfaces (Panetto & Molina, 2008). Today's supply chains exist as large-scale socio-technical systems. No one platform has complete control and/or visibility. From an abstract view point, the issue of interoperability is associated with the matching of the logical control processes, semantic data representations and governance frameworks of disparate systems, that were not developed to support adaptive control based on learning (Chen et al., 2008). Therefore, MARL systems must be integrated in a way that respects existing institutional boundaries, yet allows for collaborative optimization. The enterprise platforms such as SAP,

Oracle and IBM contain strong assumptions about sequence of process steps, data ownership and compliance. These platforms have been optimized for consistency, audibility and financial accuracy, as opposed to adaptability (Jacobs & Weston, 2007). In contrast, MARL systems will optimize expected long term performance given uncertainty. Therefore, interoperability will require architectural patterns allowing MARL to effect the execution of workflow without altering the invariant constraints imposed by enterprise platforms (Vernadat, 2007).

The standards-based approach to interoperability is critical in addressing the interplay between these two competing views. The use of message-oriented middleware, canonical data models and event driven architectures help decouple tightly coupled systems and allow for incremental integration of MARL systems. From a systems theory perspective, modular interoperability increases the resilience of systems by confining failures locally and reducing the risk of cascading disruption (Hohenstein et al., 2015) and is especially important in cases where the interactions with MARL systems occur within financial and compliance sensitive work flows. Additionally, modular interoperability facilitates cross-enterprise coordination in multi-enterprise logistics networks. Many supply chain environments include multiple entities, each operating independently with their own systems and incentive structures. MARL systems can facilitate cross-entity coordination through the exchange of high level signals (e.g. alert vs. raw data) thereby providing confidentiality to the participating entities while facilitating collective optimization. This aligns with game theoretic considerations for multi-agent systems, in which partial information sharing can lead to better equilibrium outcomes then full transparency would.

From a learning perspective, modular interoperability provides additional contextual information to agents. Information from upstream suppliers or downstream customers enhances the situational awareness of agents and enables them to make more anticipatory decisions (He et al., 2020). However, the greater interconnectedness of systems via modular interoperability introduces fragility. The propagation of errors/delays quickly throughout systems is possible due to tight coupling. Architectures for modular interoperability must therefore include fault isolation, graceful degradation and fallback plans to preserve stability under failure (Brewer, 2012). Governance and compliance are essential components of modular interoperability. Autonomous decisions made by MARL systems that could potentially affect the value of inventory, the amount billed to a customer or the submission of regulatory reports must be traceable and explainable across all systems. Modular interoperability layers must therefore support detailed logging, versioning and auditing trails to demonstrate accountability. Without these mechanisms, organizations are unlikely to accept the responsibility for granting operational authority to MARL systems (Rahimi et al., 2016). Business impact will arise from the operation of the supply chain ecosystem. An effective modular interoperability enables end-to-end visibility, decreases the coordination friction among participants in the ecosystem and enables the strategic alignment of participants in the supply chain. Participants in the supply chain can therefore act collectively in response to disruptions, optimize shared resources and negotiate dynamic trade-offs (Lambert & Cooper, 2000). MARL systems therefore become strategic coordinators instead of just optimizing localized operations. Additionally, modular interoperability influences the long-term evolution of the system architecture. As MARL systems influence decisions across multiple platforms, they can also contribute to the redesign of business processes, contractual relationships and infrastructure. The closed-loop interaction of learning systems and organizational evolution transform MARL systems into drivers of organizational adaptation and not merely as passive tools. Ultimately, the ability of MARL systems to achieve strategic relevance in supply chain management is determined by the degree of modular interoperability achieved. Without modular interoperability, MARL systems remain limited to isolated operational optimizations. With modular interoperability, MARL systems become a fundamental capability that changes how businesses cooperate, compete and collaborate in digital ecosystems.

Figure 8: API Integration with Enterprise supply chain system



The illustration shows a complete end to end, API centric, architecture for using multi agent reinforcement learning in conjunction with an enterprise supply chain management system. The architecture is designed to clearly separate the learning, coordination, and execution responsibilities. On the left side of the diagram there are the data ingestion pipeline(s) collecting and processing real time data feeds from IOT sensor systems and telematics systems. In addition to collecting and processing data from IOT sensor systems and telematics systems, the pipeline(s) capture telemetry data related to the status of inventory, environmental condition, vehicle location, and driver behavior. All of these collected streams define the "observable" state of the logistics environment and feed upward into the system through event driven interfaces and not through a direct interface to the logistics systems. The API Driven Coordination Layer is located at the middle of the diagram and represents the institutional control surface between the learning and the enterprise level execution. The API Driven Coordination Layer provides an Observability API (API which provides access to observables) for MARL agents to receive observation(s), an Action API (API providing access to actions) for MARL agents to provide policy based decisions, and an Action Mediation & Validation Component. The action mediation and validation component validates all generated actions against a set of predefined policy constraint(s), feasibility check(s), governance rule(s), and auditability criteria prior to issuing any control commands to the enterprise systems. Thus ensuring that all learning behavior produced by the MARL agents will be constrained by the enterprise level logic and compliance requirements. Enterprise systems appear only once on the lower portion of the diagram and represent the authoritative execution platforms. The enterprise systems include the Warehouse Management System, ERP and Supply Chain Management

Platform, and the Transport Control Platform. Each of the enterprise systems exposes standardized API's and maintain full control over the deterministic execution, financial accountability, and regulatory compliance.

External enterprises and partners can coordinate with the enterprise systems through controlled external ERP APIs, thus allowing for coordination with regulators, business partners, and financial systems without exposing the internal learning logic. Overall, the diagram illustrates that the MARL intelligence is operating above the enterprise systems as a decision optimization layer and that the APIs act as formal governance and control boundaries that translate the probabilistic policies into safe, auditable and executable supply chain actions.

Data Engineering and Infrastructure Requirements

Distributed Pipelines of Data for Multi-Agents Learning

Intelligent logistics systems using Multiple Agents Reinforcement Learning (MARL) rely on the distributed data pipeline for their computational core for continuous learning, coordination, and adaptation in complex large-scale networks (Dean & Ghemawat, 2008). Compared to classical Machine Learning (ML) pipelines for static datasets, MARL pipelines have to deal with concurrent data generation from thousands of agents in interaction with each other and producing temporally correlated experience streams (Chen et al., 2014). A theoretical point of view, these pipelines give a formal representation of the Stochastic Game formulation of the problem of MARL by converting the raw interaction trajectories into learning signals that respect the temporal order, the causal structure, and the identity of each agent (Lamport, 1978). If poorly implemented, the temporal order, the causal structure, and the identity of the agents can be distorted, causing unstable and inefficient policy performances in logistics environments. Agents produce heterogeneous data types (e.g., states, actions, rewards, messages exchanged, and transitions of the environment), and the goal is to ingest, index, and partition these data streams so that they can be used both for the learning of each individual agent and for the coordination of the whole system. It has been theoretically demonstrated that MARL learning depends not only on the amount of data available (volume), but also on how well the data is aligned (alignment). Experiences should be synchronized among agents to represent joint actions and common results. Therefore, the distributed pipelines should guarantee the temporal consistency and the joint context reconstruction, which is much more difficult to achieve compared to the independent data ingestion of single agents.

Scalability is probably the major difficulty for the distributed pipelines of MARL. When the number of agents grows, the volume of data generated by the interactions between them grows even faster because of the coupling effects between agents. Pipelines of data must be able to scale horizontally and not introduce bottleneck that slow down the learning process or increase the latency (Isard et al., 2007). From a systems theory point of view, it is necessary to carefully define the partitioning strategy in order to distribute the load and keep the proximity of the interactions. Simply partitioning the data by agent identity is not enough; the pipelines must also consider the spatial and temporal coupling of agents working in the same area of the network. Reliability and fault-tolerance are important theoretical and practical challenges. In fact, data loss or corruption can silently deteriorate the quality of the learning process of the logistics system. Therefore, pipelines of data of MARL must include redundancy, mechanism of checking points, and replay mechanisms to allow the learning to proceed continuously even when there are failures of the infrastructure (Akidau et al., 2013). Convergence guarantees are undermined if the failure of the infrastructure introduces biases in the sampling of the experience distribution. From a learning dynamic point of view, the design of the infrastructure of the pipelines also influences the exploratory behavior of the agents. Experiences delayed or lost can cause the agents to adopt an outdated policy, thereby reducing the effectiveness of the exploratory behavior. This shows that the design of the infrastructure of the pipelines is tightly coupled to the algorithmic properties of the MARL systems, which means that the data engineering cannot be separated from the learning theory. There is a significant business impact. In fact, the use of robust distributed pipelines allows the learning and adaptation to take place almost in real-time, which decreases the reaction time to disruptions and increases the operational resilience. On the contrary, weak pipelines increase the operational risk and reduce the confidence in the autonomous systems. Therefore, companies deploying MARL must treat the pipelines of data as critical infrastructures of the mission and not as secondary components of analytics.

Distributed pipelines can also facilitate the experimentation and the governance. In fact, by recording and replaying the interaction data, the companies can audit the decisions made, reproduce the results obtained, and validate the new policies before implementing them. This capability is essential to meet the regulatory compliance and the organizational accountability. Finally, distributed pipelines are the substrate through which the intelligence flows. In fact, the design of the distributed pipelines will determine whether the MARL systems scale up properly or fail under the complexity of the operation.

Hybrid Deployment on Cloud-Edge for Low-Latency Logistics

Hybrid cloud-edge architectures are solutions to one of the most basic conflicts in intelligent logistics systems: the relationship between the computing power required by the MARL algorithms and the latency required for the decision making (Shi et al., 2016). In fact, MARL algorithms often require high computing power for the training and coordination of the agents, and they prefer to be deployed in centralized cloud infrastructure (Varghese & Buyya, 2018). However, logistics operations require rapid responses to events and decisions need to be made quickly and close to the physical system. From a theoretical point of view, hybrid architecture solves this conflict by dividing the learning and the execution in the spatial and temporal scales. In fact, edge nodes (such as onboard vehicle controllers or gateway of warehouse) provide fast access to the local state information. These nodes enable the decentralized execution of learned policies under partial observability. On the contrary, cloud infrastructure provides the centralized training of the agents, global coordination, and long horizon optimization. MARL frameworks, such as centralized training with decentralized execution, assume this division of responsibilities, but hybrid architecture makes this assumption concrete and operational.

An important challenge is to synchronize the cloud and edge components. In fact, policies learned centrally must be disseminated to the edge nodes quickly and safely. Similarly, experience data generated at the edge must be aggregated and sent to the cloud quickly and efficiently without overloading the network capacity (Akidau et al., 2015). Control theory emphasizes the importance of bounded delay and consistency in the feedback loops to maintain the stability (Lamport, 1978). Hybrid architectures also support hierarchical learning. Fast-reacting edge policies solve immediately control problems, while slower cloud-based learning processes solve strategically behaviors on longer timescales. This multi-temporal organization corresponds to the natural temporal hierarchy of logistics operations, in which routing decisions are taken faster than the configuration of the network and/or the planning of capacities.

From a business perspective, cloud-edge hybrid architecture reduces the operational latency, improves the service reliability and enables scalability without excessive costs of the infrastructure (Abbas et al., 2018). Companies can deploy intelligence incrementally, starting with critical nodes and expanding the coverage progressively. Security considerations are increased in hybrid architectures. Edge nodes are usually physically exposed and vulnerable to attacks. Therefore, secure communication, authentication, and isolation between cloud and edge nodes are required to protect the learning integrity and the operational safety (Roman et al., 2013). Hybrid architecture also increases the resilience. Edge nodes can continue to work independently during the outage of the cloud, while the cloud can recover and refine the policies offline. This redundancy improves the resilience under adverse conditions. In summary, cloud-edge hybrid architecture is not simply a choice of deployment, but a theoretical facilitator of scalable and low-latency MARL systems in logistics.

Data Lakes and Stream Processing for Real-Time Learning

Data lakes and stream processing frameworks provide the memory and the nervous system of intelligent logistics platforms, allowing both real-time learning and long-term optimization (Harby & Zulkernine, 2025). Data lakes allow storing in a centralized way all the different types of data, such as raw streams of sensors, data of the company organized in tables, and derived features. Stream processing frameworks allow performing in real-time the transformations and aggregations of the continuous data streams into formats ready to be used for learning (Akidau et al., 2013). Together, they enable the analysis of the data offline and the learning online. From a theoretical point of view, the data lakes preserve the complete interaction history of the MARL system, and allow analyzing retrospectively the system, evaluating counterfactually the decisions, and auditing the policies adopted. The preservation of the interaction history is fundamental to understand the dynamics of the learning, diagnose failures, and validate improvements. Unlike the traditional datasets, the MARL data are non-iid and temporally correlated, and the storage systems used must preserve the ordering and the context of the data (Lamport, 1978). Stream processing frameworks allow the computation in real-time of the continuous data streams. Technologies like Apache Kafka and Apache Spark enable scalable ingestion, transformation, and routing of the data in real time. These systems allow the MARL agents to receive the timely updates, and also to perform the batch analytics for the training and the evaluation (Akidau et al., 2015).

Theoretically speaking, stream processing must preserve the causality. In fact, aggregations or windowing operations that obscure the temporal relationships can distort the learning signals. Therefore, the pipeline must be designed in accordance with the temporal assumptions of the algorithms of the reinforcement learning. Business wise, data lakes and streaming enable the unified view of the entire supply chain. In fact, companies can analyze the trends of the performance, identify the anomalies, and simulate alternative scenarios using the historical data. This allows to improve both the operational aspects and the strategic ones. Scalability is a fundamental issue. In fact, the volume of data grows very quickly as the number of agents grows. Architectures of data lakes must support the elastic scalability while keeping the performance and the efficiency of the costs (Chen et al., 2014). Governance and data quality management are also very important issues. In fact, without a strong management of the schema and the validation of the data, the data lakes can degenerate into unmanageable repositories. Good governance will allow the learning systems to operate on reliable and understandable data.

Security and Privacy Protocols in Multi-Agent Collaboration

Enterprise logistics is an ideal area for MARL systems, however, security and privacy are fundamental needs before MARL systems can be implemented in this field (Sicari et al., 2015). The continuous flow of information between agents creates a large amount of potential vulnerabilities and complex data flow paths. From a theoretical viewpoint, the limitations created by security constraints provide the boundaries for the possible information exchange structure of the MARL environment that determine how agents will coordinate and learn from one another. MARL systems need to protect against unauthorized access to data and models, and the intentional manipulation of data and models to create incorrect results, which could potentially cause unsafe behaviors and reveal confidential information (Kouicem et al., 2018). Therefore, security measures need to be included in the design of the learning process, as opposed to being used as an afterthought to protect the learning process. Privacy concerns exist in multi-enterprise logistics networks, due to the fact that each agent represents an organization that has their own set of interests. As a result, there must be careful consideration given to the way information is shared among agents to ensure that the positive effects of collaboration do not outweigh the negative effects of sharing too much information. This adds a new layer of complexity to the design of security, and introduces a new class of problems related to game theory; as the incentives provided to the agents play a critical role in determining what information they will share. In addition to using techniques like encryption, access control, and secure communication channels, MARL systems also need to have privacy preserving learning methods that allow for agents to collaborate without exposing private information. Examples of these types of methods include federated learning and secure aggregation methods (Bonawitz et al., 2017). Enterprise-wide systems are unlikely to adopt MARL-based solutions unless there are adequate levels of security and privacy measures in place, and regulatory compliance requirements, including data protection regulations, place additional constraints on the design of MARL systems. Data protection regulations govern the use of personal data and other sensitive data. For example, if an organization is required to store data for a certain number of years, then the MARL system must be designed to meet those requirements. Compliance with data protection regulations requires that organizations design their systems to comply with regulations at all times. Additionally, security measures provide organizations with the ability to establish governance and accountability for the actions of autonomous systems, through audit trails, anomaly detection, and incident response procedures. To summarize, security and privacy are not secondary concerns in the implementation of MARL systems, they are primary considerations that directly affect the way that agents will interact with one another and the level of trust that organizations will have in the autonomous systems.

Evaluation Metrics and Performance Assessment

Learning Metrics: Convergence Rate, Stability, and Reward Efficiency

Learning-centric metrics evaluate the learning aspects of Multi-Agent Reinforcement Learning (MARL) because MARL represents a coupled adaptive system rather than a single optimizer solving a specific problem (Ning & Xie, 2024). The environment in logistics is uncertain, partially observable, and subject to ongoing change; however, the additional layer of nonstationary created by MARL arises from the fact that while the environment itself may change, each agent adjusts its policy in response to interactions with other learning agents. As such, performance should not be viewed solely as evidence that learning is taking place appropriately because high short-term rewards can be generated through fragile conventions, over-fitting to a specific subset of the scenario distribution, or exploiting characteristics of the simulator used for training (Henderson et al., 2018). Learning metrics thus operationalize the broader question of whether policies are being learned in ways that are consistent across interactions, stable across uncertainty, and consistent with the organizational objectives embodied in rewards. Often, convergence rate is defined as the rate at which the system converges toward a stable state; however, in MARL, convergence rate is best understood as the rate at which the entire system converges toward a state in which all agents behave consistently (Watkins & Dayan, 1992). Consequently, convergence metrics in MARL are generally concerned with bounding policy updates, reducing oscillation in action distributions, reducing variance in returns, and demonstrating consistent performance across time periods rather than strictly achieving convergence toward an optimal solution (in the traditional sense). In logistics terms, the learning question is whether the system will consistently generate a repeatable operational regime that continues to perform effectively when there are changes in demand, congestion patterns, and the behaviors of coordination partners.

Stability is essential in MARL because the learning environment produced by MARL is endogenously determined by the agents themselves. When one agent updates its routing, dispatch, or scheduling behavior, the state transition dynamics experienced by other agents change instantaneously, as congestion, capacity contention, and queue dynamics depend upon the behavior of all agents (Williams, 1992). Consequently, the learning process in MARL is not simply one of adapting to a stochastic external process, but is instead one of reshaping the very process it is attempting to model, making instability a primary theoretical failure mode (Williams, 1992). Thus, stability metrics are designed to identify whether training trajectories exhibit oscillatory behavior, divergent behavior, regime switching, or extreme sensitivity to minor perturbations (e.g., slight variations in demand, noisy observations, etc.) caused by factors such as minor changes in communication bandwidth. Instability manifests in logistics as operational thrashing; for example, when trucks repeatedly route around predicted areas of congestion, when warehouses repeatedly reschedule pick/dock events, and when hubs oscillate between periods of overloading and underutilization. Such behaviors can paradoxically occur even when the total reward appears high, because rewards shared among agents can obscure localized volatility, and because policies can exploit short term gains in order to cause long term disruption. Therefore, the application of stability metrics provides a direct link between learning theory and operational reliability, and requires that high reward levels be associated with smooth control, predictable coordination, and limited reactions to disturbances, which are all necessary for adoption in safety and service level constrained environments.

Reward efficiency refers to the degree to which experience contributes to lasting improvements, and it is relevant in large-scale logistics MARL systems because data collection is costly in multiple senses: it is expensive in terms of simulation computing resources, expensive in terms of the time required by engineers to design simulations, and potentially expensive in terms of actual service degradation in the field (Agarwal et al., 2021). While sample efficiency refers to the number of experiences collected relative to the time required to collect them, reward efficiency captures the quality of those experiences as well as their quantity. Agents collect data under a policy distribution of behavior that changes due to exploration by other agents collecting data simultaneously, resulting in noisier and less representative learning signals. Reward efficiency is therefore concerned with the variance of learning updates, the stability of advantage estimates, and the degree to which reward shaping accurately communicates the true objective rather than a distorted representation of that objective. Credit assignment theory provides the underlying justification for evaluating reward efficiency in MARL systems, because agents cannot learn efficiently unless rewards are able to isolate marginal contributions of individual agents and are not delayed, sparse, and confounded by exogenous factors (such as traffic and weather). Rewards that do not provide efficient credit assignment result in high variance gradients, policies that pursue noise, and slower or brittle convergence, which can ultimately produce policies that appear to work well in training but fail when subject to the slightest shift in the distribution of the operating environment. From an enterprise perspective, low reward efficiency increases the cost and risk associated with experimentation, because more training cycles, more simulation runs, and more tuning of parameters are required prior to achieving confidence in performance, thereby delaying deployment and increasing the likelihood of costly failures during rollout.

Robustness across random seedings and initial conditions is not simply a minor detail in the evaluation of MARL methods, but rather is a fundamental theoretical requirement for asserting that a MARL method has learned a stable coordination mechanism rather than simply memorizing a particular training sequence (Engstrom et al., 2020). Equilibrium selection in MARL is typically path-dependent, meaning that two separate training runs using the same hyperparameters but initialized differently can converge to different conventions, some of which are efficient and others of which are wasteful. Therefore, evaluation protocols must view randomness as part of the scientific object being studied, and not as an error source that must be suppressed. Evaluation of robustness examines distributional properties such as the mean performance, variance, worst-case quantiles, and probability of catastrophic failure across runs, because enterprises cannot permit systems that function only under favorable initial conditions. Robust evaluation of MARL methods therefore aligns with stochastic control theory, where policies are evaluated under uncertainty and disturbance, and with safe evaluation of MARL methods, where the risk of failure in the tails of the distribution is as important as the expected mean. In logistics systems, the need for robustness is magnified further by the fact that operational conditions change every day, and therefore a policy that is stable only in a small portion of the state space is operationally dangerous. Consequently, metrics must explicitly measure the consistency of convergence times, the stability of the final action distributions, and the replicability of the coordination patterns across training runs, in order to provide evidence that the learning process has identified structural relationships rather than noise.

Temporal consistency extends robustness from the training horizon to the deployment horizon, and recognizes that MARL systems can appear to be converged during training but then degrade when exposed to longer operational sequences (Pardo et al., 2018). Reasons for degradation include compounding approximation errors in value functions, un-modeled slow dynamics such as gradual changes in congestion levels, feedback loops between policy actions and demand responses, and drift caused by non-stationary exogenous processes. Metrics for temporal consistency therefore examine whether the returns, constraint violations, and coordination quality of a policy remain stable across extended rollouts, whether performance degrades over time, and whether policies become brittle when the environment moves into rarely visited portions of the state space. Temporal consistency is particularly important in logistics because operations are continuous, and therefore performance degradation over several days can be significantly more impactful than modest under-performance that persists indefinitely. Metrics for temporal consistency often include measures of the stability of returns, constraint violation rates, and measures of the entropy of policies and the drift of action distributions over time. Ultimately, temporal consistency directly affects the reliability of MARL systems over long time horizons, and therefore affects the level of trust placed in these systems by organizations.

The learning metrics outlined above represent the scientific foundation for the evaluation of MARL methods in autonomous logistics. They determine whether the learning process is reliable, stable, efficient, and reproducible under the interaction dynamics that characterize autonomous logistics.

Operational Metrics: Delivery Accuracy, Cost Reduction, and Lead Time Variability

Operational metrics, like delivery accuracy and cost savings, are quantitative measurements of how well an algorithm behaves. They provide a theoretical link between the goals of a Reinforcement Learning algorithm and the constraints of Service Systems (Beamon, 1999). For example, delivery accuracy measures how well the algorithm satisfies the time windows and service levels required by the customer. These requirements are viewed as "hard" constraints from the customer's viewpoint regardless of their encoding as "soft" penalties in the RL training framework. The theoretical significance of measuring delivery accuracy is that it measures how well the algorithm has satisfied the deadlines of the service system, given stochastic travel times, queue dynamics, and coordination dependencies among agents. Moreover, delivery accuracy is rare when there are many agents contributing to the delivery of a shipment; as many agents will have impacted the trajectory of a shipment through various means such as hubs, docks, and routes. Therefore, the delivery accuracy of a learned coordination strategy signifies that the local decisions made by each agent have been aligned with the satisfaction of the overall deadline for the shipment, which is a greater achievement than maximizing the average reward. Furthermore, delivery accuracy impacts the brand trust, contract penalties, and customer retention of the enterprise, making delivery accuracy a non-negotiable criterion for deploying autonomous solutions.

In contrast to delivery accuracy, cost savings represents the economic efficiency of the solution. Cost savings in a Multi Agent Reinforcement Learning setting is primarily related to network-level coordination rather than local optimization (Gunasekaran et al., 2001). Logistics-related costs arise from decisions that are interdependent. For example, a routing decision may decrease transportation cost and labor cost, but also cause traffic congestion and subsequent increases in service time for deliveries downstream, which may result in additional costs associated with overtime, missed delivery windows, and inventory holding costs. Theoretical analysis views cost savings as an emergent property of coordinated policies that address the congestion externality, capacity contention, and task assignment across agents. As such, cost savings must be broken down into measurable components, such as transport cost per mile, asset utilization rates, warehouse labor efficiency, dwell time, and penalty avoidance, because a singular cost metric may obscure trade-offs that negatively affect service quality. In addition, enterprises expect a sustained reduction in cost and therefore cost metrics must be developed to include total cost of ownership and total cost to serve in order to be viewed as credible.

Lead Time Variability is theoretically significant because it provides insight into the degree of sensitivity of the system to uncertainties and the degree to which the system can mitigate disturbances. Lead time variability is a key objective in control theory and can be measured as the standard deviation of the distribution of lead times. While average lead time may decrease and the variance may remain constant, a reduction in the variance of lead time indicates that the system has learned to anticipate disturbances and mitigate them through anticipatory behaviors that avoid congestion formation, balance load, and coordinate arrival at constrained resources such as docks and hubs. In control theory terms, a reduction in the variability of lead time indicates that the system has learned to stabilize itself under stochastic disturbances and thus reduces the amplification of random fluctuations through feedback loops. In an enterprise context, the reduction of variability can be viewed as more beneficial than a reduction of the mean because it allows for leaner safety stock inventories, more reliable appointment schedules, and improved capacity planning leading to compounding financial benefits.

Operational metrics should be tested under a variety of realistic demand patterns and disruption scenarios because nominal performance does not necessarily translate into real world deployment value in logistics networks that frequently undergo shock events. Metrics focused on resilience such as the degree of service degradation under peak demand conditions, recovery time following a disruption, and performance under correlated failures are essential because they indicate whether the system can maintain adequate service under stress. Theoretically, robust policies should exhibit performance generalization across a variety of state distributions, not simply optimal performance in the most typical regime. If the training data for the MARL method contains inadequate diversity of scenarios, then the method may become overly specialized to the specific distribution of the training data, resulting in poor performance under stress testing. As such, stress testing becomes a part of the operational evaluation methodology, rather than a post-hoc check. In addition, these metrics directly relate to the risk management of an enterprise, as large financial and reputational losses can occur due to disruptions and autonomy will only be valued if it results in improved performance in high impact scenarios.

Metrics related to fairness and equity are necessary in MARL to prevent asymmetric burdens being placed upon regions, customer segments, and agent groups as a result of multi-objective optimization. Policies that minimize cost may place a disproportionate number of delays on lower-priority regions, shift congestion to specific corridors, or assign undesirable routes to specific types of vehicles. Such policies may be operationally unacceptable and raise ethical concerns. Theoretically, fairness adds distributional constraints to the evaluation of performance, requiring measurement of performance beyond averages, such as quantiles, group-wise service rates, and disparity indices across geographic areas and/or customer tiers. Fairness also relates to the risk of violating regulatory and reputational standards in logistics networks serving diverse communities. Finally, fairness metrics are used to ensure that autonomy is not viewed as arbitrary or biased, and support internal alignment and external trust.

From the business perspective, operational metrics are the determining factors of adoption because they represent outcomes that directly matter to executives, operations managers, and customers. Delivery accuracy supports the establishment of trust with customers and adherence to contractual obligations, cost savings support margins and lead time stability supports the efficiency of planning and inventory optimization.

However, businesses require predictability as much as improvement, therefore operational metrics must be evaluated across seasons, demand regimes, and disruption patterns to demonstrate consistent value.

Accordingly, operational metrics must include variance and tail risk reporting in addition to point estimates, as a small probability of a severe failure can outweigh average gains. Additionally, operational metrics determine organizational readiness to adopt automation: if frequent human intervention is needed to realize performance gains, then the value proposition of autonomy is diminished independent of the sophistication of the algorithms used.

Finally, operational metrics serve as feedback mechanisms for continuous improvement and lifecycle management of MARL systems. Through performance monitoring, the detection of drift, degradation or emerging failure modes can trigger the need for retraining, updating constraints, or rolling back policies. Theoretically, the use of operational metrics in deployment shifts evaluation from a one-time experiment to an ongoing measurement process, where the learning and the operations are co-evolving. This parallels the concept of continued evaluation, where metrics are calculated online and compared to guard rails to ensure that autonomy remains within acceptable risk thresholds. In enterprise environments, this supports governance models where autonomy is increased or decreased based on measurable performance, thereby enabling metrics to be utilized as operational control instruments. Ultimately, the use of operational metrics ensures that the advancement of MARL translates into tangible, reliable and scalable improvements in real logistics outcomes.

System Metrics: Scalability, Energy Efficiency, and Communication Overhead

The primary reason to report on system-level metrics is that they define whether a MARL solution that has been demonstrated to work in a research setting can be scaled up to be run in a real enterprise setting with thousands of agents, strict latency limits, and limited computing resources (Ning & Xie, 2024). Simply stated, scalability is not about whether your solution still performs well as you add more agents, it's about whether the algorithmic and architectural complexity of your solution grows in a reasonable manner. Theoretically, there is an explosive combinatorial problem here; joint action spaces and interaction graphs grow exponentially with the number of agents and the size of the network, leading to both learning instabilities and infrastructure bottlenecks. To create a scalable MARL solution, therefore, the coordination mechanism, critic, and communication protocol must all be developed to avoid global coupling where possible. In addition, scalability metrics usually determine whether an organization can limit its autonomy to a pilot area or extend it across national operations, and this can have a direct strategic effect.

Similarly, energy efficiency metrics have become important because both training and running MARL systems can consume a lot of computational and environmental resources, and because many organizations now seek to optimize their logistics supply chains using sustainability goals (Strubell et al., 2019). Thus, we need to evaluate energy efficiency over two different dimensions: compute energy consumption (by training and inference), and operational energy (due to decisions regarding routing and scheduling). Theoretically, we want to avoid "cost-shifting," i.e., reduce emissions in the physical network, but substantially increase them through compute, or reduce compute, but then make decisions that result in increased fuel usage. Thus, we need to carefully account for our entire system, including measuring energy per training episode, energy per decision, and the marginal energy cost of communication and coordination layers. For enterprises, energy efficiency means controlling costs and complying with regulations, because both the costs of compute and emissions reporting are becoming increasingly material (Schwartz et al., 2020).

Communication overhead is perhaps the most important metric in MARL, because coordinating agents can quickly become communication-bound. Communication overhead can include message frequency, message size, bandwidth utilization, and the latency sensitivity of coordination messages. Information theoretically speaking, the main question is: how much information must be communicated to get a certain level of coordination, which is essentially an efficiency frontier. Communicating too much can undermine scalability, increase failure susceptibility, and create privacy problems, while communicating too little can lead to coordination failures and inefficiency. In logistics networks, communication constraints are very real due to poor connectivity in the field, varying devices, and the cost of transmitting data; therefore, our overhead metrics should reflect real-world deployment constraints rather than idealized networks.

Although communication overhead is closely related to latency, we believe it necessary to evaluate latency explicitly as an end-to-end metric because timely decisions are crucial to real-time logistics control. Latency can include sensing delay, processing delay, communication delay, inference delay, and actuation delay, and these delays may interact non-linearly with each other as the workload increases. Theoretically, the relationship between latency and closed-loop stability is that delayed feedback can cause oscillations and instability, especially if multiple agents respond to stale congestion signals. Therefore, evaluating latency must include both average latency and tail latency, because infrequent spikes in latency can trigger cascading failures in tightly-coupled systems. In enterprise deployments, latency metrics determine whether autonomous decision-making can be used for time-critical routing versus only for slower planning tasks.

Fault tolerance and availability metrics assess how MARL solutions perform during infrastructure stress, including partial outages, degraded sensors, and communication failures. These are not optional, because logistics networks are always operating and downtime results in both financial and service penalties. Theoretically, the focus should be on graceful degradation: the extent to which the system can revert to safe local heuristics, continue to satisfy constraints, and avoid catastrophic coordination collapse when part of the system fails. Some relevant metrics for assessing fault tolerance and availability include mean time to recovery, performance under node loss, and resilience under partitioned communication. For enterprises, these metrics determine what degree of risk is acceptable because a highly optimal system that fails unpredictably will likely be rejected in favor of one that is less optimal, but reliable.

Ultimately, from a business perspective, system metrics determine the total cost of ownership and operational feasibility. An organization must justify compute costs, network costs, monitoring costs, and engineering overhead against operational gain. System metrics therefore help an organization make design trade-offs, such as whether to use heavy centralized critics, how much communication to permit, and where to deploy edge inference. Practically, organizations will frequently accept slightly less optimal performance in order to obtain significantly better scalability and reliability, because once an organization deploys a stable solution, it generates compounding benefit. System metrics also aid in capacity planning by enabling organizations to determine the amount of infrastructure needed to provision and where to place it. In conclusion, system metrics are required to ensure that MARL solutions are not only intelligent in theory, but deployable, sustainable, and economically rational in the real world of logistics.

Finally, system metrics allow for scientifically comparable architectures at the system level, not just the algorithmic level. Two MARL solutions that produce similar reward outcomes can have greatly differing amounts of compute demanded, communication overhead, and fault tolerance. Reporting system metrics therefore precludes misleading claims that do not take into consideration the actual costs associated with deploying an MARL solution in a production environment. In logistics environments, where the scale of operation is large, these hidden costs can exceed the cost savings from optimization; hence, system metrics are essential for truthfully evaluating MARL solutions. Therefore, system metrics complete the evaluation framework by establishing a link between the success of learning and operational improvements, and the realities of distributed infrastructure.

Benchmark Datasets and Evaluation Environments

Benchmark datasets and standardization of the testing environments are required for MARL due to the nature of the target evaluation: the object of evaluation is not a mapping of static input-output, but rather a policy that interacts with a dynamic environment, generating data in a manner that is endogenously determined

(Bellemare et al., 2013). While supervised learning datasets may be fixed and shared, the evaluation of MARL relies heavily upon the characteristics of the dynamic environment (i.e., environment dynamics, observation models, stochasticity, and reward definitions), which makes benchmarks used in MARL evaluation essentially epistemological instruments; they provide the conditions under which claims of coordination, robustness and scalability are evaluated. Due to this dependency, a comparison between the findings of two studies claiming similar improvements would be rendered invalid, even if the studies were conducted in environments of radically differing difficulties. As a result, a rigorous MARL science relies on the availability of benchmark environments that clearly define the dynamics, constraints, stochastic processes and the evaluation protocol.

Large-scale logistics simulation systems that model enterprise fulfillment networks provide large-dimensional test beds that challenge coordination, capacity constraints, and temporal coupling among other factors. These testbeds include warehouse capacity limitations, dock scheduling constraints, route networks, stochastic demands and congestion propagation, and as such, represent interaction complexities that smaller "toy" problems cannot capture. The theoretical value of these types of benchmarks lies in their ability to subject algorithms to realistic externalities where an agent's decision creates costs or benefits to other agents, a factor central to the study of MARL. They also naturally induce delayed rewards structures where decisions made today affect outcomes tomorrow, representing realistic logistical practices. Enterprise-wide applicability of these types of benchmarks is critical since they assess not only if an algorithm can learn, but if it can learn under realistic operational constraints and noise.

The OR Library provides a wide array of deterministic benchmark problems for vehicle routing, scheduling, and network design, which can serve as a solid baseline for evaluating how competitive MARL is compared to established deterministic methods under static conditions. Although these types of problems do not have the same type of endogenous learning as MARL does, they can provide the ground truth or nearly-optimal solution to validate if the learned MARL policy is significantly inefficient under stable conditions. The theoretical point here is not to suggest that MARL replaces deterministic solvers in static settings, but rather to ensure that

MARL claims are grounded to some degree against known baselines to prevent over-estimation of innovation. When MARL achieves the same level of performance as classical solutions in deterministic regimes, and surpasses classical solutions in uncertain and non-stationary regimes, then the comparative advantages become scientifically credible. From a business standpoint, these baselines also provide assurance that MARL will not perform worse than traditional planning methods under relatively stable conditions.

Urban logistics is characterized by dense interaction, congestion, time-dependent travel times, and multi-modal routing, making simulated city networks important benchmarks. City networks allow the evaluation of coordination under high coupling, where marginal actions can generate system-level congestion waves. The theoretical value of city networks as benchmarks lies in the fact that they naturally encode equilibrium concepts such as Wardrop-style route choices, making them suitable for the study of equilibrium selection, inefficiency, and coordination mechanisms. They also allow for the evaluation of last-mile delivery dynamics, where time windows, parking constraints, and stochastic delays are common. From a business standpoint, city benchmarks are directly relevant to customer-facing operations where service reliability is most visible, and where optimization gains translate directly to improvements in customer experience.

Scenario diversity is required to evaluate benchmarks properly, as MARL systems can over-fit to specific topologies, demand regimes, or disruption patterns. A single environment benchmark could lead to incorrect conclusions if algorithms learn environment-specific tricks or take advantage of reward artifacts. Evaluation protocols that use sets of scenarios that vary in terms of topology, demand intensity, disruption frequency, and observation noise are therefore required (Cobbe et al., 2020). The theoretical connection to generalization under distribution shift is fundamental in control systems deployed in changing real-world environments. Benchmark sets should therefore include stress tests and out-of-distribution scenarios, and not as optional features but as core evaluation components. Scenario diversity is directly related to risk management from an enterprise viewpoint, as operations must function effectively across various regimes throughout the year, city, and business cycle.

From a business perspective, the primary mechanism for reducing the risk associated with adoption before deployment is the provision of benchmark-based evidence. Leaders must be confident that the system will scale, remain stable, and maintain service levels during periods of stress, and benchmarks provide structured evidence for these claims. Additionally, benchmarks help to reduce vendor and research risk by providing a means for independent validation. From a research community viewpoint, standardized benchmarks facilitate rapid progress by allowing for comparisons based on "apples-to-apples," and by determining which innovations improve performance broadly and not just in a single environment. Ultimately, the provision of benchmark datasets and evaluation environments serves as the empirical foundation of MARL science, by connecting theoretical claims to replicable evidence generated under well-specified conditions.

The standardization of benchmarks also enhances scientific integrity by requiring clarity regarding what is being optimized and under what constraints. Results reported in MARL papers can be intentionally or unintentionally modified through small variations in reward scaling, termination conditions, or observation noise. The use of standardized benchmarks helps to eliminate this variability by specifying the environment and the reporting protocols. Standardized benchmarks promote cumulative science, and allow for reliable measurement of progress over time. In logistics, where the stakes of deployment are high, disciplined use of benchmarks also enhances safety and trust by discouraging premature claims of readiness. In conclusion, benchmarks are not optional tools, but are instead a structural component of evaluation that facilitates both scholarly credibility and enterprise deployment confidence.

Applications in Autonomous Logistics Networks

Fleet Coordination and Last Mile Delivery Optimization

At a theoretical level, last-mile logistics can be considered a decentralized partially-observable stochastic control problem where each vehicle agent only sees a small piece of the larger transportation network, yet the actual state of the system is composed of many hidden factors such as congestion and queue dynamics downstream of delivery points, and variations in demand over time (Powell, 2019). As a result of the interconnectedness of vehicles, route choice, dispatch time, and delivery sequence all affect the propagation of congestion, cluster formations due to stops, and ultimately the overall service level achieved by the fleet. As a result, the optimization of last-mile logistics represents a coordination challenge rather than simply a collection of individual shortest path problems, which necessitates the development of policies that account for the interdependencies between the fleet members.

One of the primary theoretical contributions of MARL to the field of fleet coordination is its ability to develop adaptive policies that adapt to non-stationary demand and travel time distributions without having to model those conditions explicitly through parametric forecast models. The traditional methods used to solve routing problems are based on deterministic optimization with stochastic adjustments. However, these traditional methods are frequently unable to effectively manage situations where the environment is changing more quickly than the planning horizon of the traditional routing system (Pillac et al., 2013). By contrast, MARL agents are able to develop policies that continuously re-plan in response to environmental changes, while maintaining global coordination. This continuous adaptation capability is particularly important in situations where demand surges, road incidents occur, or weather events force changes in the available routes and/or time window risks. Additionally, this continuous adaptation can be viewed as developing a control law that maps observed system characteristics to action updates that allow the fleet to follow a dynamic optimum rather than repeatedly solving for a static solution.

In last-mile MARL, the action space is inherently multi-dimensional. For example, a fleet agent may select a routing edge, insert a new order into a route sequence, select a dispatch time, modify the rate of service, or negotiate the transfer of a portion of a load with another vehicle. The richness of the action space introduces several theoretical difficulties related to the abstraction of the actions taken by the agent and hierarchical control. Typically, a doctoral-level architecture decomposes actions into two levels, i.e., a high-level policy determines the region assignments, the boundaries for delivery zones, or the shift schedules, whereas a lower level policy determines the fine-grain route decisions and stop ordering. Hierarchical decomposition has several advantages for learning in MARL including reducing the learning variance and increasing the scalability of the system by allowing different timescales of control to be managed by different portions of the system.

Communication limitations play a major role in last-mile environments. In particular, real-world fleets are subject to bandwidth restrictions, intermittent connectivity, and confidentiality requirements across different subcontracted carriers. Consequently, MARL systems must learn coordination techniques that do not depend on continuous global communication. Examples of such techniques include learning mechanisms for message passing, compressed sharing of intentions, and graph-based coordination among neighbors, which allow vehicles to communicate only the necessary information to avoid duplication of effort and congestion. From a theoretical perspective, this can be viewed as learning sufficient statistics for coordination under limited information exchange where the goal is to approximate global coordination using only local signals.

The reward structure for last-mile optimization is a multi-objective function that must include measures of service quality, costs, and sustainability. While on-time delivery and time window compliance are commonly represented as constraints, minimizing fuel/energy consumption and distance traveled are common objectives. Nonlinear penalties for customer dissatisfaction resulting from missed time windows can reflect the operational realities where late deliveries result in greater negative impacts on business. Theoretical design of the reward structure is therefore closely tied to constrained reinforcement learning, where feasibility and safety must be maintained while optimizing performance. One robust method combines penalty functions for constraint violations with explicit feasibility checks and safe-action filtering to prevent exploration of unsafe delivery sequences during learning.

Evaluation in last-mile MARL must move beyond average performance to focus on tail risk. A policy that consistently provides good average performance but infrequently results in large cascades of delayed deliveries will not be acceptable. Therefore, risk-sensitive evaluation metrics such as maximum possible lateness, highquantile delivery delay, or probability of violating a constraint become operationally relevant. The connection to the theoretical concepts of distributional reinforcement learning (DRL) or risk-aware RL lies in the fact that the policy is optimized to achieve not only an optimal expected value of the reward but also controlled variability and bounded downside risk. Risk-shaping becomes critical in logistics that involve consumers.

In addition to reducing miles driven, fuel cost, and labor overtime while improving service consistency, the strategic impact of improved coordination involves increased operational resilience by providing the means to rapidly respond to disruptions without human dispatch interventions. The combination of lower operating cost and improved customer experience creates significant competitive advantage in markets where delivery speed and reliability are the drivers of customer loyalty. Furthermore, the use of more efficient routing and fewer failed delivery attempts enables organizations to reduce their carbon footprint and achieve other sustainability related goals.

Additionally, the strategic impact of MARL coordination extends to capacity utilization and asset productivity. MARL coordination can improve the effective throughput of an existing fleet by reducing idle time, increasing the density of routes, and smoothing demand over time windows. Reducing the amount of idle time and improving the efficiency of the existing fleet allows organizations to meet growing demands for service without necessarily adding additional assets to their fleets. This reduces the need for capital investment in new assets and helps organizations scale their services without proportionate increases in assets. Ultimately, this translates into improved return-on-investment (ROI) for logistics assets and greater margin stability, both of which are critical for organizational-wide acceptance of MARL technology. Finally, deploying MARL technology in last-mile applications will require strong governance and accountability frameworks, including explanations for the reasons behind the routing decisions made by the autonomous system, auditable records of the causes of late deliveries, and compliance with operational constraints such as federal regulations regarding hours-of-service and labor practices. Autonomous systems must produce transparent output that includes intent signals, summaries of the reasoning behind routing decisions, and confidence values to ensure that humans can evaluate and accept the decisions made by the autonomous system. Ensuring that autonomous systems remain trustworthy and aligned with enterprise-wide principles of accountable decision making are critical.

Warehouse Robot Collaboration and Task Assignment

Robot collaboration in a warehouse is a canonical multi-agent setting due to its unique combination of physical coordination, spatial resource contention and stringent safety constraints in a structured environment (Wurman et al., 2008). In principle, the theoretically optimal method for modeling multi-robot warehouse operation would be as a cooperative decentralized partially observable decision-making process, where agents have a common global objective (such as maximizing warehouse throughput or minimizing makespan) but perceive the world only locally (due to occlusions, sensor noise and lack of communication). However, the key defining characteristic of warehouse robots is the tight coupling of their actions through common pathways, intersections, charging stations, and queues at pick stations. As a result, in practice, learning effective policies will require agents to understand how their individual actions affect those of other agents in the warehouse and develop stable coordination conventions that prevent collisions and deadlocks while achieving high throughput.

Assigning tasks to robots in a warehouse is not merely a scheduling problem; it is a dynamic allocation problem under uncertainty. New incoming orders have varying priorities, items move in response to replenishments, and interactions between humans add stochastic delays to the system. MARL provides the ability to develop policies that adaptively assign tasks to robots based upon both the current state of the system and anticipated congestion (Chen et al., 2022). For instance, deciding which tasks should be assigned to a robot must take into account factors like travel distance, queue length at pick stations, and expected contention on common aisles. From a theoretical perspective, this has similarities to a distributed resource allocation problem in which the system learns to allocate limited time and mobility among competing tasks in such a manner that the global reward is maximized.

In many warehouse settings, there is a need for structured communication among robots. Direct communication allows robots to negotiate who goes first, share intentions, or coordinate hand-offs between tasks. Indirect communication occurs through environmental mediated signals such as occupancy grids, virtual pheromones, or shared blackboards. Graph-based communication networks allow robots to communicate with each other locally, and are scalable since they do not require all robots to communicate directly with each other. From a theoretical perspective, the goal is to find communication policies that are bandwidth efficient, but also enable nearly optimal coordination, particularly in densely populated warehouses where communication overhead could otherwise grow extremely rapidly with the number of robots.

Safety and constraint enforcement are fundamental to autonomous warehouse systems. Robots must avoid colliding with other objects, follow speed limits, keep a safe distance from people, and adhere to operational policies. Therefore, reinforcement learning techniques must be used in a constrained manner, allowing robots to learn from experience while respecting valid feasible action sets that are dynamically restricted. There exist various theoretical frameworks for performing such constrained policy optimization and for developing safe exploration methods that prevent learning unsafe trial actions. In warehouse settings, safe learning is not just an optional feature of warehouse autonomy but a necessary condition for deployment; thus, evaluation of autonomous warehouse systems must assess constraint violations, near misses with people or other objects, and recovery capabilities after encountering unexpected obstacles.

A warehouse collaboration reward function must balance throughput, utilization, and delay costs while maintaining safety and fairness across tasks. Reward functions that only optimize for throughput can cause robots to behave aggressively, i.e., to select paths that increase the likelihood of collision with other robots or objects. Therefore, reward shaping typically includes penalties for creating congestion, blocking, or waiting excessively at intersections. From a theoretical viewpoint, penalty terms must be designed so that they do not create perverse local optima in the robots' behavior, e.g., if robots learn to always avoid areas with high congestion levels, then high-priority pick tasks will starve.

Scalability is a primary issue. Warehouse robotics can potentially involve hundreds or thousands of agents, rendering centralized control infeasible. Decentralized Training Decentralized Execution (CTDE) architectures are commonly employed, where training relies on a global state and centralized critics, but execution remains decentralized and is based on local observations and minimal communication. Theoretical advantages of CTDE architectures include reduced non-stationarities during training and more accurate credit assignment due to global advantage estimation. During execution, robots run lightweight policies that are dependent only on local sensing and minimal communication, ensuring operational viability under latency constraints.

MARL-based task assignment will improve the operational efficiency of a warehouse, thereby improving the throughput of orders, the amount of labor required per unit of throughput, and the utilization of the warehouse space (Boysen et al., 2019). Most importantly, it will improve the robustness of a warehouse to disturbances such as downtime of stations, temporary blockage of aisles, etc. Robustness translates into less operational downtime and increased service reliability for downstream delivery operations. In a competitive fulfillment environment, this will translate into faster shipping promises and higher customer retention.

An additional strategic business benefit of MARL is the flexibility to reconfigure warehouse operations. Traditional warehouse automation technologies typically require significant design effort when warehouse layouts change or when new products with different pick patterns are introduced. On the other hand, MARL systems can quickly adapt policies to new warehouse layouts or changes in demand patterns, thereby reducing reconfiguration costs and enabling rapid scaling. The ability to reconfigure quickly is particularly valuable in high-growth e-commerce environments where fulfillment networks are rapidly evolving. Finally, successful MARL warehouse deployments require close integration with warehouse management systems (WMS) and governance workflows. The policies developed using MARL must be understandable by warehouse operations personnel in order for them to be trusted, and decision log data must be available for auditing and continuous improvement. It is the combination of safety assurances, performance enhancements, and operational transparencies that make MARL-based warehouse collaboration credible as a production technology rather than a research prototype.

Inter-Hub Scheduling and Global Route Synchronization

Inter-hub scheduling and global route synchronization extend MARL from localized operational control to network-wide coordination of multiple distribution centers, cross-docks, and transportation corridors. Theoretically, this domain is characterized by long-horizon dependencies, delayed rewards, and multi-scale coordination, where decisions made at one hub affect congestion, capacity, and service outcomes across the network hours or days later (Powell, 2019). A multi-agent framework is applicable here, as each hub and transportation corridor can be viewed as an agent with local goals and constraints; however, the overall performance of the system depends on the coordinated behavior across the network. Thus, inter-hub operations are transformed into a coupled stochastic control system in which coordination must balance flow conservation and dynamic capacity constraints.

At the hub level, scheduling involves decisions such as when to dispatch trailers, which dock doors to assign, how to allocate labor, and how to sequence cross-docking operations. The decisions made at one hub affect the arrival processes, service times, and capacity constraints at upstream and downstream nodes. Classically, scheduling models assume deterministic arrival processes; however, real-world networks exhibit stochastic variability resulting from traffic, weather, and demand volatility. MARL can learn policies that anticipate these uncertainties and adjust schedules to avoid downstream bottlenecks. Theoretically, the learning of these adaptive policies occur in a partially observable environment, where each hub estimates the state of the entire system through local indicators of global conditions such as delay patterns at incoming docks and buildups at queue locations.

Global route synchronization is a coordination problem because the routes taken by shipments between hubs must be aligned to meet service commitments and minimize dwell time. If a hub dispatches too early, the downstream hubs may not be ready to receive shipments, causing dwell and congestion. Conversely, if a hub dispatches too late, service quality suffers. MARL can learn policies that synchronize dispatch and processing rates across hubs by developing implicit coordination strategies, which are often mediated by shared representations or limited communication. Theoretically, this has similarities to distributed control with coupled constraints where the stability and throughput of the system depend on the coordinated flow control of the system rather than the local optimization of individual components. Information sharing between hubs is a primary constraint at the network level, as hubs may belong to separate business units or partner organizations. Due to restrictions related to privacy, contracts, and technical interoperability, information sharing may be partial. Therefore, MARL architectures must be able to perform coordination with incomplete information sharing. Architectures employing federated or privacy-preserving learning can facilitate the sharing of policy improvements without revealing sensitive operational data. This introduces theoretical challenges related to learning from decentralized data and diverse objectives.

Reward structures for inter-hub scheduling must reflect end-to-end service reliability, total network cost, and congestion avoidance. Rewards are delayed because the impacts of scheduling decisions may not be evident until shipments reach downstream nodes. Therefore, credit assignment is a significant theoretical challenge, as mechanisms such as difference rewards or value decomposition must be employed to attribute the outcome of decisions to specific hub decisions. Without effective credit assignment, learning variance increases and convergence to stable policies slows. Evaluating MARL for inter-hub scheduling must include metrics such as throughput, dwell time, missed connections, and service reliability across lanes. Critically, evaluation must emphasize the system's resilience; specifically, how quickly a network recovers from hub failures, lane closures, and/or sudden demand spikes. MARL is most beneficial in situations where disruptions propagate nonlinearly through the network, and adaptive coordination can mitigate shock propagation by routing flows and reallocating capacity. These ideas are consistent with theories of resilience and network control.

Emergency Logistics and Humanitarian Supply Distribution

The application of multi-agent reinforcement learning (MARL) to emergency logistics and humanitarian distribution represents a very high stakes application area whose objectives are not merely economic, but also relate to human welfare, timely response to crisis, and the availability of resources. From a theoretical perspective, emergency logistics and humanitarian distribution share many characteristics with other uncertain domains, including changing and unknown constraints and lack of complete information (Holguin-Veras et al.,

2012). Emergency logistics and humanitarian distributions are typically conducted in the presence of infrastructure damage, volatile demand, and costs of failure measured in lost human life rather than monetary penalties for missed delivery times. Because there are multiple decision makers involved in emergency logistics and humanitarian distribution (e.g. vehicles, warehouses, medical supply nodes, and coordinating entities), and because of the need for those decision makers to act based upon partial observability, multi-agent reinforcement learning is applicable to this domain.

Humanitarian logistics can be modeled as a mixed-mode multi-agent problem, as agents acting within the same organization can work cooperatively together, yet compete for the same resources as agents acting in different organizations or jurisdictions. This creates a game theoretic component to the problem, since even though all agents seek to assist others, their individual interests will likely not always be aligned (Kovacs & Spens, 2007). As such, MARL architectures must be able to support both cooperative and strategic interaction between agents, and possibly incorporate some mechanism for incentivizing cooperation among agents, or provide some level of coordination protocol to allow agents to cooperate partially without having to fully trust one another. The theoretical contribution in this paper relates to developing policies that continue to function effectively despite poor coordination among agents and changes in the structure of the coalitions of cooperating agents (Balcik et al., 2010).

Decisions regarding routing and allocation during emergencies must be capable of adapting to changing conditions (e.g. road closures, evolving hazards, etc.) and changing demand hotspots. Optimization techniques used in traditional methods require knowledge of fixed maps and known constraints, however, during emergency situations, it is rare that either of these two pieces of information are known (Özdamar & Ertem, 2015). MARL provides a means for developing policies that can adapt to unknown or changing conditions through use of stochastic elements in the simulation environment, allowing agents to respond quickly to new information as it becomes available. In terms of theoretical basis, the development of MARL policies is analogous to robust stochastic control, in that the policy must perform well over a wide variety of possible environmental states, rather than being optimized for a single predicted state (Özdamar & Ertem, 2015).

Resource allocation is a key aspect of emergency logistics and humanitarian distribution. Given the limitations of available resources (e.g. limited supplies, limited number of vehicles and personnel), and the need to allocate these resources across competing demands, a major theoretical challenge is developing a framework for making multi-objective priority decisions (e.g. deciding how to distribute limited resources fairly, efficiently, and with urgency). Developing rewards that capture the relative importance of ethical criteria (e.g. minimizing mortality risk, maximizing the amount of critical supplies covered) is important, as is ensuring that the reward functions do not create bias or inequities in the way that resources are allocated.

In addition to the challenges posed by resource allocation, communication during emergency logistics and humanitarian distribution is severely limited by damaged infrastructure, thus limiting the ability of agents to communicate with one another and/or with a centralized entity. As such, MARL-based systems must be designed to operate in a completely decentralized manner with limited reliance on centralized communication. Edge computing and local coordination are necessary to achieve this goal, as agents may have to operate independently for extended periods of time, communicating only limited status updates when communication with a centralized entity is possible. Theoretically, this necessitates the development of robust, decentralized policies and effective coordination mechanisms that minimize communication overhead. The metrics used to evaluate the performance of MARL-based systems in humanitarian logistics applications are significantly different from those used in commercial logistics. Critical metrics include response time to critical areas of population, percentage of population covered, fairness/equity of allocation, and robustness to infrastructure degradation. Additionally, given the tail-risk nature of humanitarian logistics applications (where a small probability of severe failure is unacceptable), the evaluation of MARL-based systems must take place within a risk-sensitive framework and include safety constraints.

Finally, the business and social implications of deploying MARL-based systems in humanitarian logistics applications are significant, as they enable humanitarian organizations to increase efficiency, reduce waste, and extend coverage with the same resource base, governments to enhance disaster preparedness through the use of MARL-based digital twins to model response strategies and identify potential bottlenecks prior to a crisis, and ultimately lead to faster responses, fairer allocations, and increased public trust (Sheu, 2007). However, deploying MARL-based systems in humanitarian logistics applications is subject to a greater degree of scrutiny in terms of transparency and accountability than deploying them in commercial settings. Thus, stakeholders must be provided with sufficient information to understand how decisions were made, and systems must be capable of providing audit trails. To address this requirement, MARL-based systems must be integrated with explainable mechanisms, decision logs, and governance frameworks that ensure the ethical deployment of these systems (Yi & Ozdamar, 2007). Ultimately, the technical robustness of MARL-based systems, their alignment with ethics and norms of behavior, and their feasibility of operation determine if MARL can make responsible contributions to humanitarian logistics (Mili, 2025).

Business Value and Strategic Implications

Cost Optimization Through Adaptive Automation

Adaptive automation through cost optimization represents one of the most direct and easily defendable areas of business value for multi-agent reinforcement learning in logistics. However, its strategic implications extend far beyond simply saving money. Cost in logistics is an emergent property of a coupled dynamic system where individual agent decisions produce externalities that propagate through congestion, capacity contention and service failures (Tang, 2006). Thus, adaptive automation is important since it allows the system to learn control policies that take these externalities into account instead of optimizing the isolated local objectives. In practical terms, MARL will therefore reduce costs not only by choosing shorter routes and faster schedules, but also by influencing the network level behavior of things like smoothing flow, reducing peak congestion and preventing cascading delays that produce overtime and penalty costs (Waller & Fawcett, 2013).

One of the main mechanisms for cost optimization is the transition from reactive dispatching to anticipatory control. Rule-based systems traditionally react to congestion and delay once they become apparent; however, MARL policies can learn anticipatory patterns of action from past experiences and simulations (Dubey et al., 2019) allowing them to make decisions proactively and earlier in the process, such as earlier route selection, early inventory pre-staging, and earlier fleet rebalancing before bottlenecks develop. From a theoretical basis, policies learn mappings from high-dimensional observation spaces to control actions that maximize expected returns over long horizons, thus embedding an approximate representation of the system's future dynamics in present-day decision-making. From a cost perspective, this results in the elimination of the expensive nonlinear penalties that arise when service levels fail due to operating at or near capacity limits (Ivanov, 2020).

Operational volatility has a disproportionate effect on labor costs. Warehouses and hubs incur overtime when incoming shipments arrive unpredictably in large volumes and drivers incur idle time when their schedules do not coincide with available docking space. By learning synchronized schedules and capacity aware dispatch, MARL can reduce this volatility and act as a controlling element that attenuates disturbances in the system by reducing variance in system output. Less variance in system output results in more predictable staffing needs, better labor utilization and lower overtime premiums, turning cost optimization into a structural rather than episodic activity.

MARL does not limit transportation cost reduction to minimizing distance but also includes fuel efficiency, maximizing asset usage, and eliminating empty miles (Waller & Fawcett, 2013). Empty miles result when decentralized decisions cause unbalanced flows in multi-agent fleets, resulting in vehicles being left without profitable backhaul opportunities or having to operate outside of their preferred pick-up/drop-off zones. MARL policies can learn repositioning strategies that minimize imbalance by treating vehicles as part of a network, rather than independently acting units, thereby reducing deadheading, improving load factors, and increasing revenue per mile, which is typically the greatest contributor to profit in logistics (Wamba et al., 2017).

When MARL is incorporated into supply chain planning systems, it can influence replenishment timing, safety stock placement, and cross-docking routing, leading to inventory related cost optimizations. Not only is inventory holding cost influenced by the average demand, but it is also influenced by the variability in lead time and service risk (Tang, 2006). When MARL reduces variability and improves reliability, companies can safely reduce safety stock without negatively impacting service (Ivanov & Dolgui, 2020), providing another layer of cost reduction through working capital and obsolescence risk savings. Theoretical foundations exist for inventory reductions, which enable the company to implement leaner inventory policies. These benefits are compounded in that inventory reductions free up working capital and reduce obsolescence risk.

There is also a cost aspect of avoiding penalties and complying with contracts. Logistics operations frequently incur nonlinear penalties for missing delivery windows, failing to deliver on time, and not meeting service level agreements (Tang, 2006). MARL policies can learn to assign priority to high-penalty events and allocate resources accordingly, essentially incorporating contractual risk into the reward function. There is great importance in designing objectives correctly to ensure that MARL learns to trade-off cost and service in a systematic manner, rather than relying on inflexible rules. The cost of infrastructure to run MARL itself must also be factored into the cost optimization analysis. Both training and inference require compute resources, data pipelines, and monitoring systems. The value of strategic application is realized when the total operational savings outweigh the total cost of ownership including compute, integration, and governance overhead (Brynjolfsson et al., 2011). Therefore, a formalized evaluation method is necessary in order to measure cost metrics that include both operational savings and the expenditure for digital infrastructure. Companies that view MARL as a product capability versus an experimental technology can achieve the optimal balance between the two by utilizing efficient cloud-edge deployment and implementing policy update governance that minimizes the necessity of redundant retraining (Wamba et al., 2017).

From a strategic management perspective, adaptive automation shifts the cost profile of logistics from variable and reactive costs to predictable and structured investments in infrastructure (Brynjolfsson et al., 2011). As a result, this shift creates greater margin stability and reduces a company's exposure to cost spikes due to logistical disruptions, ultimately enabling scalability without commensurate increases in personnel or managerial overhead. Additionally, the strategic implications of MARL is that it can transform logistics from a cost center into a controllable performance engine that utilizes continuous learning to maintain efficiency regardless of increased demand and/or network complexity (Dubey et al., 2019). Furthermore, cost optimization also provides a strategic advantage in competitive markets by affecting pricing flexibility and service differentiation. Lower unit costs provide either pricing competitiveness or the opportunity to reinvest into faster delivery promises and better customer experience. Since logistics performance is increasingly becoming a defining factor of brand perception in e-commerce and retail, cost optimization through adaptive automation can serve as a strategic lever rather than a back-office efficiency initiative.

Competitive Advantage Through Data Driven Decision Making

Competitive advantage in today's logistics industry is primarily derived from decision intelligence rather than physical assets (Brynjolfsson et al., 2011). By converting raw operational data into adaptive policies that continually improve execution (Wamba et al., 2017), multi-agent reinforcement learning (MARL) enables decision intelligence. Theoretically, logistics networks are complex adaptive systems whose outcomes are dependent upon the interactions between numerous agents and constraints (Ivanov, 2020). The ability to utilize structure in these interactions to anticipate bottlenecks and coordinate resources more effectively than competitors who rely solely on static optimization and manual dispatching enables data-driven decision intelligence to contribute to competitive advantage. One of the primary strategic advantages that derive from decision intelligence is faster decision cycles. Traditional planning methods rely on periodic batch optimization and replanning, which create lag times between environmental change and operational responses.

MARL enables continual decision-making through learned policies that operate in real-time. Technically speaking, policies can provide nearly instant inference once they have been trained, thereby facilitating high frequency control without requiring computationally intensive optimization at each decision point.

Decision intelligence also enhances resource allocation efficiencies. Fleets, hubs and warehouses are all limited resources that must be allocated under conditions of uncertainty. Competitors that employ static policies commonly rely on conservative buffers, such as excess vehicles or higher safety stocks. MARL systems can eliminate the need for buffers by dynamically allocating resources, thereby enhancing capacity utilization. As such, MARL systems provide a competitive advantage by enabling the rapid expansion of service levels and growth with less capital investment than competitors employing static policies. Data network effects enhance this competitive advantage. Companies with extensive operational footprints gather richer data across various routes, demand regimes and types of disruptions. MARL systems can leverage this data to learn more robust policies, establishing a positive feedback loop whereby scale improves learning and learning improves scale (Brynjolfsson et al., 2011). From a theoretical perspective, this is analogous to statistical learning concepts where varied experience distributions reduce over-fitting and enhance generalization. Therefore, smaller competitors may face barriers to entry as they lack sufficient data diversity to train policies that are equivalent in robustness.

Competitive differentiation also exists through sustainable performance. Customers and regulators increasingly reward companies that have lower emission logistics. MARL can optimize for carbon efficiency along with cost and service, thereby enabling companies to demonstrate tangible proof of sustainability commitments.

This is a strategic differentiator, especially in markets where green logistics impacts procurement decisions and consumer brand preference. Another competitive advantage exists through the improved reliability and predictability of logistics operations. Many logistics customers place a greater emphasis on consistent delivery than occasional speed. MARL can reduce lead time variability by stabilizing flows and coordinating capacity, thereby providing superior reliability (Wieland & Wallenburg, 2013). This reliability enables companies to differentiate their services through offering guarantees of delivery windows, premium delivery tiers, or just-intime replenishment, which competitors may find difficult to match without the presence of advanced coordination intelligence.

Governance-wise, decision intelligence also enables companies to improve compliance and risk management. Autonomous policies can be constructed to respect regulatory guidelines, driver hours, and safety standards, thereby enabling consistent compliance at scale. Manual operations may not provide the same level of compliance and risk exposure as autonomous policies. This risk reduction itself is a strategic advantage, as it protects against potential reputational damage and regulatory penalties. Finally, business impact exists in customer retention and customer lifetime value. Rapid, reliable logistics operation improves customer satisfaction and reduces churn. In e-commerce and FMCG, where switching costs are minimal, reliable logistics operation is a key determinant of customer loyalty. Thus, decision intelligence not only provides cost savings, but also revenue protection and growth. Competitive advantage also exists through organizational learning. MARL systems integrate learning into logistics operations, thereby enabling the company to rapidly adapt to new network configurations, new product mixtures, and new market entries. This adaptability reduces the time-to-market associated with entering new geographic regions or launching new delivery modes, thereby providing strategic agility.

Resilience Against Disruptions in Logistics and Transport Chains

Logistical resilience refers to the ability of a logistics network to withstand disruptions (Ivanov, 2020) and continue to provide an acceptable level of service despite being disrupted (Ivanov, 2020). As MARL provides policies that can operate under uncertain circumstances and can react to unforeseen events in real-time, MARL will contribute to a logistics network's resilience. Logistical disruptions are caused by a variety of factors such as traffic accidents, inclement weather, employee shortages, failure of infrastructure, surge in demand, and failure of suppliers; each type of disruption affects the entire logistics network in a cascading manner because logistics networks are tight-coupled systems. For example, when there is a delay at a hub it causes delays in other areas of the logistics network in the form of missed pickups and dropoffs and inventory shortages. MARL can help to minimize the propagation of these cascading effects by providing policies that allow for the coordination of the logistics network to reroute shipments, rebalance capacity, and to prioritize critical deliveries (Ivanov & Dolgui, 2020). Therefore, theoretically speaking, MARL acts like a decentralized adaptive controller that rapidly adjusts system behavior to prevent shock from magnifying in its effect (Ivanov & Dolgui, 2020).

Training of MARL agents in scenario-based simulations is crucial for developing a resilient logistics network. Agents need to be exposed to random elements and disruption models in order for them to develop policies to respond to rare, yet impactful, events (Queiroz et al., 2020). Robust control theory emphasizes that robustness requires training agents on disturbances and their distributions during training (Queiroz et al., 2020). Digital twins of logistics systems enable agents to safely test their resilience strategies prior to deployment. A resilient logistics network should degrade gracefully. When communications fail or data is corrupted, agents need to have a fallback policy that prevents chaos from occurring versus failing to produce useful results.

Architectures of MARL agents that utilize CTDE and local fallback heuristics can ensure that some level of operation continues to exist even in the event of degraded conditions. Evaluating the performance of MARL agents requires evaluating how they perform under failure conditions and how quickly they recover from disruptions. Performance evaluation should assess both the robustness of the algorithms used to support the MARL agent and the resilience of the logistics infrastructure. Coordination among agents under disruptive conditions is especially difficult because agents may be competing for limited resources such as routes, docking space, and personnel. MARL agents must dynamically allocate resources and do so subject to multiobjective constraints. Equilibrium selection under stress is a theoretical consideration since agents must select an equilibrium that does not lead to the convergence of the system to selfish behaviors that ultimately increase the system's performance loss (Wieland & Wallenburg, 2013). The design of rewards and coordination protocols has significant impacts on the resilience of a logistics network supported by MARL agents.

Resilience has a greater business impact than cost savings associated with normal operations because disruptions cause nonlinear loss to a company. Missed deliveries can generate contractual penalties, customer dissatisfaction, and damage to reputation. Inventory shortages can decrease sales and stop production. MARL agents can mitigate the business loss resulting from disruptions thereby protecting a company's revenues and decreasing its risk exposure, making resilience a strategic value driver. Resilience also supports strategic continuity planning. Companies that can deliver services during disruptions gain market share over companies that are unable to continue delivering services during disruptions. This creates competitive advantage during times of crisis. MARL enables companies to rapidly adapt to changing conditions without having to manually coordinate agents that are located throughout the world. Rapid adaptation is critical during disruptions that occur in many different locations at the same time. Another aspect of resilience involves learning from disruptions. MARL agents can use information collected after a disruption occurs to improve future responses to similar disruptions. Therefore, disruptions can be viewed as training signals for continually improving the organization's resilience strategies. This creates an organizational learning loop wherein disruptions are converted into positive events instead of purely negative events. Ultimately, this will increase the organization's robustness and reduce its reliance on static contingency plans.

Transition to Autonomous, Self-Regulating Supply Networks

The transition to autonomous, self-regulating supply networks represents the ultimate strategic implications of adopting MARL. Instead of optimizing individual logistical activities in isolation, MARL enables the creation of distributed intelligence that coordinates across fleets, warehouses, hubs, and enterprise systems.

Theoretically, supply chains become complex adaptive systems with embedded decision-making agents that make adjustments to their behavior in response to state changes in the system (Ivanov, 2020). This represents a shift from centralized planning to distributed control where coordination emerges from the interaction of policy between agents that share common objectives and constraints. Autonomous, self-regulating supply networks require multi-layered autonomy. Agents at the lowest layer make decisions locally, such as determining a route to take or what tasks to assign to vehicles. Meta-controllers at higher layers adjust objectives, constraints, and policy parameters based on strategic goals. This hierarchy of autonomy aligns with multi-scale control theory where fast operational control is nested within slower strategic adaptation.

Ultimately, this creates a supply ecosystem that can reorganize itself dynamically without human intervention. A fundamental characteristic required for self-regulation is stable coordination in the presence of limited communication and partial observability. Because autonomous ecosystems cannot rely on continuous central control due to latency and scale, MARL architectures must support decentralized decision-making with minimal yet informative coordination signals. This creates scalable autonomy while ensuring global alignment, enabling large networks to be created without an exponentially increasing amount of coordination overhead. Governance is a key issue in autonomous ecosystems. Accountability, auditability, and compliance with organizational ethics and regulatory requirements are all critical aspects of autonomy. This means that decision-making agents must be able to demonstrate accountability for their actions, provide explanations for their actions, and provide version control for their policies. Trustworthy autonomy theory emphasizes that self regulation must include oversight mechanisms, i.e., human governance sets constraints and monitors outcomes while agents optimize within those constraints. Another important aspect of autonomous supply ecosystems is interoperability across organizations. Autonomous supply ecosystems typically involve multiple organizations and digital platforms. MARL agents must therefore coordinate across organizational boundaries using standard-based interfaces and privacy-preserving learning techniques. Federated reinforcement learning can facilitate shared learning between agents while preventing the sharing of sensitive data, thus facilitating ecosystem-wide optimization while maintaining boundaries of trust.

Autonomous supply ecosystems can benefit businesses in several ways. They can increase business agility and speed of response to changes in markets, as well as reduce friction in coordinating logistics (Dubey et al., 2019). Autonomous supply ecosystems can add new nodes, delivery modes, or partners much more quickly than traditional supply ecosystems because the intelligence needed to support the ecosystem is distributed and adaptive. This reduces the costs associated with integrating new components of the ecosystem and increases its scalability. Strategically, autonomous supply ecosystems can enable predictive and prescriptive logistics. Instead of responding to disruptions, systems can anticipate disruptions and reconfigure themselves in anticipation of those disruptions (Ivanov & Dolgui, 2020). This enables new business models such as dynamic service pricing, guaranteed delivery windows, and adaptive inventory allocation. Businesses can differentiate themselves from other businesses based on their reliability and responsiveness, not just their size. However, transitioning to autonomous supply ecosystems will change the roles of employees in logistics. Employees will no longer be responsible for manually dispatching vehicles and managing exceptions. Instead, employees will be involved in oversight, exception management, and policy governance. Organizations will need to invest in developing capabilities related to AI operations, monitoring, and compliance, as well as the development of organizational social structures that support the success of autonomous supply ecosystems. Therefore, the strategic success of autonomous supply ecosystems will depend on the successful implementation of both technological innovation and socio-technical transformation.

Ethical, Governance, and Operational Trust Considerations

Decision Transparency in Autonomous Agent Actions

The transparency of the decision process is a basic requirement for the responsible use of multi-agent reinforcement learning systems in logistics due to the fact that the autonomy of the decision-making process changes the way decisions are made, explained, and challenged in organizations. Traditional logistics operations involve decisions that are traceable to human planners, rule-based systems or optimization models with clearly defined objectives and constraints. Multi-Agent Reinforcement Learning (MARL) systems generate policies through learning processes based on data, and the decision logic is encoded implicitly in the parameters of the high dimension. Therefore, theoretically, there exists an epistemological gap between correct behavior and understandable behavior, a problem identified by researchers working on the challenges related to black-box learning systems and explainability (Guidotti et al., 2018; Rudin, 2019). Hence, transparency is not only an issue of usability but also a structural requirement to ensure that autonomous systems comply with the governance of the organization and the social expectations (Floridi et al., 2018; Jobin et al., 2019).

Technically, the transparency of the decision process of the agents implies that the stakeholders of the agents have access to the internal decision process of the agents. This does not mean that the raw weights of the neural networks should be exposed to non-expert users, but rather that structured explanations should be provided of why specific actions were taken in specific conditions, as part of the explainable AI paradigms

(Guidotti et al., 2018). In MARL, this problem is even more challenging since the actions of the agents depend not only on their local observations, but also on the anticipated reactions of the other agents and the coordination behaviors that have been learned. Thus, the transparency mechanisms must be able to operate at two levels simultaneously: the level of each individual agent and the level of coordination of the system, and explain how the local decisions contribute to the global consequences such as the reduction of the congestion or the priority of services.

Theoretically, the approaches to transparency in MARL often refer to explainable reinforcement learning and interpretable control theory. The techniques such as policy distillation into simpler surrogate models, counterfactual reasoning, and the formalized description of the behavior of the model are aligned with the accountability practices proposed for high-risk AI systems (Mitchell et al., 2019; Raji et al., 2020). In logistics applications, the transparency of the decision process may consist of summaries of the decisions that identify the most important factors that determine the routes of the vehicles, the schedules, and the allocations of resources, etc., such as the forecasts of the congestion, the risks of exceeding deadlines, and the constraints of capacities. These summaries will allow the human operators to know if the decisions of the autonomous systems match the intuitions and the policies of the operators.

Moreover, the transparency of the decision process is necessary for the calibration of the trust of the human operators in relation to the autonomous systems. An excessive trust in the autonomous systems can be as problematic as a lack of trust. When the human operators have access to the rationales of the decisions of the autonomous systems, they will be able to decide when to intervene correctly, i.e., when to cancel decisions that are correct and when to pay attention to potential problems of the autonomous systems. From a theoretical point of view, the calibration of the trust requires that the transparency of the decision process is timely, contextual, and proportional to the importance of the decisions, as required by the human-centered frameworks of governance of AI (Floridi et al., 2018; Jobin et al., 2019). In addition, the high-stake decisions, such as the emergency rerouting of vehicles or the priorities of services, require a much more detailed explanation than the routine decisions.

In addition, the transparency of the decision process must consider the emergent behavior of the system of agents. Many of the coordination behaviors of the system are generated by the interactions of the agents, and not by the explicit instructions of the agents. Consequently, it is difficult to assign the consequences of the system to the individual agents. The frameworks of transparency of the decision process must, therefore, provide the explanations of the coordination behaviors of the system, and of the equilibrium selections and of the adaptations of the system, in order to allow the organizations to understand what drives the improvements of performance, and to maintain the confidence in the long-term deployments of the systems of autonomous agents. From the operational point of view, the transparency of the decision process allows the training and the organizational learning of the human planners and the managers, in order to know the new heuristics and the new coordination behaviors that could not be previously considered.

Finally, the transparency of the decision process is also an element that is essential in the resolution of disputes. In case of failures of service, the organizations must justify the decisions taken by the autonomous systems to the clients, the partners, or the regulatory bodies. If the autonomous systems do not have the means to justify the decisions, the organizations will be exposed to the risks of reputation and of legality. Therefore, the mechanisms of transparency of the decision process must be developed so as to support the analyses after the events, and the communications with the stakeholders external to the organization, in addition to the optimization internal to the organization. Finally, the transparency of the decision process is indissociable from the ethics of deployment of the autonomous logistics systems. The ethical principles of fairness, accountability, and proportionality, cannot be implemented, and therefore monitored, without having access to the rationales of the decisions (Dwork et al., 2012; Floridi et al., 2018). Therefore, the transparency of the decision process is the foundation of all the subsequent guarantees of governance and ethics of the systems of autonomous logistics.

Accountability and Traceability in AI-Driven Logistics

Accountability in AI-driven logistics corresponds to the possibility of attributing responsibilities for decisions, results and failures in systems that are increasingly operating autonomously. In the traditional logistics operations, the accountability is organized hierarchically and depends on the roles of humans. The multi-agent reinforcement learning (MARL) systems, however, modify this hierarchy of accountability, and introduce decentralized entities of decision-making that are learned instead of being programmatically determined. From a theoretical point of view, this introduces the question of agency, of responsibility and of control in the systems of autonomous agents that act in a decentralized manner, and which echoes the questions posed by the research on accountability of algorithms (Raji et al., 2020).

The traceability is the technical method that makes possible the accountability. The traceability consists in keeping detailed records of the observations, of the actions, of the communications, and of the versions of the policies that lead to a particular outcome. In MARL, the traceability is complex because the decisions of the agents are interdependent and occur in succession. For example, a delay of delivery can result from a sequence of interactions between several agents who respond to evolving conditions. To be effective, the traceability must reconstruct the causal chains that link the agents in space and in time, as part of the auditing frameworks end-to-end that have been proposed for complex systems of AI (Raji et al., 2020). From a systems point of view, the traceability must be integrated into the architecture of the system, and not be added later.

The logging mechanisms must record sufficient information to allow the analysis of causality without creating an overload of storage, and without compromising the constraints of the privacy. This includes recording the abstractions of the states used by the agents, the key variables of decision, and the messages of coordination.

Mechanisms of preservation of privacy such as the aggregation secure and the differential privacy become increasingly relevant in this context, in order to make possible the traceability, and to protect the data of operation that are sensible (Abadi et al., 2016; Bonawitz et al., 2017). Theoretically, the traceability must retain the sense of the data, and not only the raw data, in order to allow a meaningful analysis.

The frameworks of accountability must also distinguish the different levels of responsibility. Certain consequences of the system may correspond to the individual decisions of the agents, whereas others may correspond to the properties of the system, such as the design of the rewards, the biases of the training data, and the communication protocols. Assigning the accountability solely to the individual agents can hide the design flaws of the system. Therefore, the models of governance must recognize the shared responsibility among the designers of the algorithms, the architects of the systems, and the managers of the operations.

Legally speaking, the mechanisms of accountability support the management of liabilities. As the regulations will evolve to treat the systems autonomous, the organizations will have to show that they have control, supervision, and protection measures reasonable. The traceability will allow to show that the decisions were taken according to the authorized policies and constraints, which is indispensable for the defense against the claims and the audits of compliance. Trust between business partners and clients depends also on the accountability. The partners and clients must be confident that the autonomous logistics systems will not behave in a capricious and unfair manner. The clear accountability structures will reassure the stakeholders that there exist the mechanisms of correction of the errors, of compensation of the damage and of prevention of the recurrences.

Regulatory Compliance in Global Freight and Delivery Data

Compliance with regulatory regimes is one of the most important constraints affecting the use of MARL systems in global logistics networks. In addition to the jurisdictional differences, logistics operations operate under a wide variety of laws and regulations concerning the collection, storage, processing, and sharing of operational data regarding transportation safety, labor standards, trade compliance, and data protection. For autonomous decision-making systems, operationally compliant with all applicable regulatory requirements while continuing to perform well and scale to meet increasing demands of the network will be essential. Practically speaking, regulatory compliance introduces external constraints that limit the possible policy space of MARL agents.

In terms of data protection, the data used in the MARL system must be collected, stored, processed, and shared in a manner that satisfies the requirements of various legal and regulatory requirements. The data required for the MARL system is derived from a wide variety of sources, including sensors, telematics, and enterprise systems. The data must be protected in accordance with the principle of fair information practices, as described by the International Organization for Standardization (ISO) and the European Union's General Data Protection Regulation (GDPR), and using privacy-preserving machine learning techniques (Abadi et al., 2016; Wilkinson et al., 2016). The cross-border movement of data presents significant difficulties. A number of researchers have proposed federated learning paradigms as a method for conducting decentralized training of ML models, which do not require centralized collection of data, and thus, address the difficulties associated with cross border data flow issues (Yang et al., 2019).

In addition to the regulatory requirements for data protection, logistics operations must also comply with transportation-related and labor-related regulations. As a result, the autonomous routing and scheduling decisions made by the MARL system must be made in accordance with various federal, state, local regulations, and labor agreements, and the autonomous routing and scheduling decisions must also be made in accordance with the hours-of-service limits for drivers, vehicle restrictions, and hazardous materials handling requirements. These types of constraints must be incorporated into the learning process of the MARL system through either constrained actions spaces or penalty mechanisms. From a control-theory perspective, this can be considered constrained optimization under uncertainty, where the feasibility of the solution is as important as the optimality of the solution.

In addition to the regulatory compliance requirements, the MARL systems must also satisfy the explainability and auditability requirements imposed by regulatory agencies. There may be a requirement to explain why the MARL system has made certain decisions, especially when those decisions affect safety or the economics of the network. The MARL system must therefore incorporate features that enable the tracking of the decisions made by the MARL system and link those decisions to the relevant regulatory requirements and audit trails. In terms of business considerations, failure to comply with regulatory requirements poses serious risks to the entity operating the MARL system, including financial penalties, loss of operating privileges, and damage to reputation. As a result, compliance with regulatory requirements should be taken into account in the early stages of the design of the MARL system, rather than being considered an afterthought.

Ethical AI Deployment and Fairness in Algorithmic Resource Allocation

The deployment of ethical AI in logistics involves ensuring that the autonomous decisions made by the MARL system align with the values of society, such as fairness, equality, and respect for human dignity. The MARL system allocates scarce resources, prioritizes deliveries, and influences access to services in logistics. Those decisions can have different effects on different regions, customer groups, or communities. From a theoretical viewpoint, there are potential ethical implications of autonomous decision-making in logistics because the optimization objectives of the MARL system may conflict with societal values unless those values are formally included as part of the optimization objective function (Dwork et al., 2012).

Ensuring fairness in algorithmic resource allocation in logistics requires defining what it means to treat customers fairly in logistics contexts. This could include providing similar service levels throughout regions, giving priority to vulnerable populations, or allocating resources proportionately to need. The MARL system must then encode these principles into the rewards or constraints of the MARL system. However, fairness is inherently multi-dimensional and context-dependent, and formalizing it is difficult. From a theoretical viewpoint, researchers have approached fairness in algorithmic resource allocation in logistics using fairness aware learning and multi-objective optimization (Dwork et al., 2012).

Bias can occur in the MARL system through the training data, environment, or rewards. The historical data used in the MARL system may reflect existing inequities, and as a result, the MARL system may learn to develop policies that reinforce those inequities. Ethical deployment of the MARL system, therefore, requires careful consideration of the datasets used to train the MARL system, the design of the scenario(s) used to test the MARL system, and the methods used to evaluate the performance of the MARL system to identify and mitigate bias. Transparency and accountability are required for ethical evaluation of the MARL system. Without knowledge of the decision logic and outcomes of the MARL system, biased behavior may remain undetected (Mitchell et al., 2019; Raji et al., 2020).

From an organizational viewpoint, ethical deployment of the MARL system contributes to long-term trust and legitimacy. Customers, employees, and regulatory agencies are becoming increasingly scrutinized regarding the decisions made by AI systems. Demonstrating alignment with ethics can reduce reputational risk and promote long-term acceptance and utilization of the MARL system. Ultimately, ethical deployment of the MARL system contributes to the positive contribution of the MARL system to society while promoting operational improvements. Fairness, accountability, and transparency are not barriers to achieving optimal solutions but constraints that contribute to developing responsible intelligence.

CONCLUSION AND FUTURE RESEARCH DIRECTIONS

Intelligent multi-agent reinforcement learning (MARL) is demonstrated in this research article as a foundational paradigm for autonomous coordination and real-time network optimization in logistics. It demonstrates the capability of MARL to transform logistics systems from reactive, rule-based systems into adaptable, learning-driven systems. Through modeling logistics networks as distributed, partially observable, and dynamically coupled systems, MARL enables decision-making that considers network externalities, anticipates disruptions, and coordinates resources across spatial and temporal dimensions. The theoretical contributions of this work are in defining logistics as a multi-agent control problem, establishing architectural principles for scalable coordination, and basing evaluations on learning, operational, and system-level metrics to provide scientific validity and relevance to enterprises.

At the architectural level, this research demonstrates that pure MARL is insufficient for implementation in complex, safety-critical logistics environments. Future systems must employ hybrid architectures that combine MARL with classical optimization and graph-based representations. Classical optimization layers provide feasibility guarantees and constraint enforcement, while graph neural networks capture the relational structure of logistics networks, allowing the policy to generalize across topologies and scales. Hybridization is a critical area of future research, as it balances adaptability with stability, learning with control, and flexibility with governance. Theoretical work is required to establish convergence properties, stability guarantees, and error propagation across these layered architectures.

Digital twin technology offers another major research frontier. Digital twins represent the ability to conduct predictive testing, counterfactual analysis, and safe experimentation by maintaining high-fidelity representations of real logistics systems. When coupled with MARL, digital twins enable robust policy development for rare, high-impact disruption scenarios and facilitate ongoing adaptation as system dynamics evolve. However, coupling digital twins with MARL creates a new set of theoretical challenges related to the co-evolution of policies and environment models. Theoretical advances in system identification, robust control, and meta-learning will be required to establish the long-term consistency between simulation and reality.

Federated reinforcement learning is a necessary advancement of MARL deployment as logistics networks increasingly span across organizational and geopolitical borders. Federated approaches to MARL enable learning across distributed nodes without centralized aggregation of data, thereby protecting data sovereignty, privacy, and trust. However, federated MARL introduces deep theoretical challenges, heterogeneous environments, and strategic behavior among participants. Establishing convergence, fairness, incentive compatibility, and robustness against adversarial updates will be essential for enabling ecosystem-wide intelligence across global logistics networks.

Finally, the future of intelligent logistics will be realized through the extension of MARL to include strategic decision-making through integration with multi-agent generative AI. Generative models can generate a wide variety of plausible future scenarios that may involve demand shifts, infrastructure changes, regulatory developments, and climate risk. When paired with MARL, the generation of future scenarios enables the systematic testing of operational policies and facilitates long-term planning under deep uncertainty. The combination of MARL and generative models will shift logistics intelligence from reactive optimization to proactive, scenario-based decision support, creating a vast array of new research areas at the intersection of reinforcement learning, generative modeling, and strategic management.

Together, these research directions suggest the emergence of autonomous, self-regulated supply ecosystems that are comprised of distributed intelligence, hierarchical control, and continuous learning. Such ecosystems require both algorithmic sophistication and strong social and technical governance frameworks that address transparency, accountability, ethics, and regulatory compliance. Therefore, the future of autonomous logistics research must take a socio-technical approach and recognize that successful autonomy requires the alignment of learning systems with organizational structures, human oversight, and societal values.

In summary, this work situates multi-agent reinforcement learning as a central element of a larger intelligent infrastructure for logistics and, as a result, views MARL as a critical element of a broader intelligent infrastructure for logistics. Through specifying architectural principles, evaluation frameworks, and future research areas, this work provides a foundation for developing the science and practice of autonomous logistics. Ultimately, the creation of fully adaptive, explainable, and resilient logistics ecosystems will require sustained interdisciplinary research combining theories of learning, control, optimization, systems engineering, and governance in a single framework.

REFERENCES

1. A multi agent deep reinforcement learning approach for traffic signal control. (2024). IET Intelligent Transport Systems. https://doi.org/10.1049/itr2.12521
2. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 308–318. https://doi.org/10.1145/2976749.2978318
3. Abbas, N., Zhang, Y., Taherkordi, A., & Skeie, T. (2018). Mobile edge computing: A survey. IEEE Internet of Things Journal, 5(1), 450–465. https://doi.org/10.1109/JIOT.2017.2750180
4. Abideen, A. Z., Sundram, V. P. K., Pyeman, J., Othman, A. K., & Sorooshian, S. (2021). Digital twin integrated reinforced learning in supply chain and logistics. Logistics, 5(4), 84. https://doi.org/10.3390/logistics5040084
5. Achiam, J., Held, D., Tamar, A., & Abbeel, P. (2017). Constrained policy optimization. https://doi.org/10.48550/arXiv.1705.10528
6. Adjei, P. K., & others. (2025). A graph attention network-based multi-agent reinforcement learning approach for complex interaction modeling. Scientific Reports, 15, 14032. https://doi.org/10.1038/s41598-025-14032-w
7. Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A., & Bellemare, M. G. (2021). Deep reinforcement learning at the edge of the statistical precipice. arXiv. https://doi.org/10.48550/arXiv.2108.13264
8. Akidau, T., Balikov, A., Bekiroğlu, K., Chernyak, S., Haberman, J., Lax, R., McVeety, S., Mills, D., Nordstrom, P., & Whittle, S. (2013). MillWheel: Fault tolerant stream processing at internet scale. Proceedings of the VLDB Endowment, 6(11), 1033–1044. https://doi.org/10.14778/2536222.2536229
9. Akidau, T., Bradshaw, R., Chambers, C., Chernyak, S., Fernández Moisés, R. J., Lax, R., McVeety, S., Mills, D., Perry, F., Schmidt, E., & Whittle, S. (2015). The Dataflow model: A practical approach to balancing correctness, latency, and cost in massive scale, unbounded, out of order data processing. Proceedings of the VLDB Endowment, 8(12), 1792–1803. https://doi.org/10.14778/2824032.2824076
10. Aledhari, M., Razzak, R., Parizi, R. M., & Saeed, F. (2020). Federated learning: A survey on enabling technologies, protocols, and applications. IEEE Access, 8, 140699–140725. https://doi.org/10.1109/ACCESS.2020.3013541
11. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095
12. Alyahya, S., Qian, W., & Bennett, N. (2016). Application and integration of an RFID-enabled warehousing management system: A feasibility study. Journal of Industrial Information Integration, 4, 15–25. https://doi.org/10.1016/j.jii.2016.08.001
13. Amato, C. (2024). An introduction to centralized training for decentralized execution in cooperative multi agent reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.2409.03052
14. Ammari, K., Bel Mufti, G., & Markou, M. S. (2025). Multi agent reinforcement learning for traffic signal control. IFAC PapersOnLine, 58(10), 65–70. https://doi.org/10.1016/j.ifacol.2025.10.011
15. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. Computer Networks, 54(15), 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010

16. Auer, P., Cesa Bianchi, N., & Fischer, P. (2002). Finite time analysis of the multiarmed bandit problem. Machine Learning, 47(2–3), 235–256. https://doi.org/10.1023/A:1013689704352
17. Balcik, B., Beamon, B. M., Krejci, C. C., Muramatsu, K. M., & Ramirez, M. (2010). Coordination in humanitarian relief chains: Practices, challenges, and opportunities. International Journal of Production Economics, 126(1), 22–34. https://doi.org/10.1016/j.ijpe.2009.09.008
18. Barricelli, B. R., Casiraghi, E., & Fogli, D. (2019). A survey on digital twin: Definitions, characteristics, applications, and design implications. IEEE Access, 7, 167653–167671. https://doi.org/10.1109/ACCESS.2019.2953499
19. Beamon, B. M. (1999). Measuring supply chain performance. International Journal of Operations & Production Management, 19(3), 275–292. https://doi.org/10.1108/01443579910249714
20. Behzadi, G., O'Sullivan, M. J., Olsen, T. L., & Zhang, A. (2018). Agribusiness supply chain risk management: A review of quantitative decision models. Omega, 79, 21–42. https://doi.org/10.1016/j.omega.2017.07.005
21. Bellemare, M. G., Dabney, W., & Munos, R. (2017). A distributional perspective on reinforcement learning. In Proceedings of the 34th International Conference on Machine Learning (pp. 449–458). https://doi.org/10.48550/arXiv.1707.06887
22. Bellemare, M. G., Dabney, W., & Munos, R. (2017). A distributional perspective on reinforcement learning. International Conference on Machine Learning. https://doi.org/10.48550/arXiv.1707.06887
23. Bellemare, M. G., Naddaf, Y., Veness, J., & Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research, 47, 253–279. https://doi.org/10.1613/jair.3912
24. Ben-Daya, M., Hassini, E., & Bahroun, Z. (2019). Internet of things and supply chain management: A literature review. International Journal of Production Research, 57(15–16), 4719–4742. https://doi.org/10.1080/00207543.2017.1402140
25. Bernstein, D. S., Givan, R., Immerman, N., & Zilberstein, S. (2002). The complexity of decentralized control of Markov decision processes. Mathematics of Operations Research, 27(4), 819–840. https://doi.org/10.1287/moor.27.4.819.297
26. Beynier, A. (2013). DEC MDP/POMDP. In Markov decision processes in artificial intelligence (Chapter 9). Wiley. https://doi.org/10.1002/9781118557426.ch9
27. Bonabeau, E. (2002). Agent based modeling: Methods and techniques for simulating human systems. Proceedings of the National Academy of Sciences, 99(Suppl. 3), 7280–7287. https://doi.org/10.1073/pnas.082080899
28. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical secure aggregation for privacy preserving machine learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 1175–1191. https://doi.org/10.1145/3133956.3133982
29. Bonomi, F., Milito, R., Natarajan, P., & Zhu, J. (2012). Fog computing: A platform for Internet of Things and analytics. Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, 13–16. https://doi.org/10.1145/2342509.2342513
30. Bowling, M., & Veloso, M. (2002). Multi agent learning using a variable learning rate. Artificial Intelligence, 136(2), 215–250. https://doi.org/10.1016/S0004-3702(02)00121-2
31. Boyd, S., Ghosh, A., Prabhakar, B., & Shah, D. (2006). Randomized gossip algorithms. IEEE Transactions on Information Theory, 52(6), 2508–2530. https://doi.org/10.1109/TIT.2006.874516
32. Boysen, N., de Koster, R., & Weidinger, F. (2019). Warehousing in the e commerce era: A survey. European Journal of Operational Research, 277(2), 396–411. https://doi.org/10.1016/j.ejor.2018.08.023
33. Brewer, E. A. (2012). CAP twelve years later: How the “rules” have changed. Computer, 45(2), 23–29. https://doi.org/10.1109/MC.2012.37
34. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). OpenAI Gym. arXiv. https://doi.org/10.48550/arXiv.1606.01540
35. Brous, P., Janssen, M., & Herder, P. (2020). The dual effects of the Internet of Things (IoT): A systematic review of the benefits and risks of IoT adoption by organizations. International Journal of Information Management, 51, 101952. https://doi.org/10.1016/j.ijinfomgt.2019.05.008
36. Brynjolfsson, E., Hitt, L. M., & Kim, H. H. (2011). Strength in numbers: How does data-driven decisionmaking affect firm performance? Proceedings of the International Conference on Information Systems (ICIS). https://doi.org/10.2139/ssrn.1819486

37. Bucsoniu, L., Babuška, R., & De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics Part C Applications and Reviews, 38(2), 156– 172. https://doi.org/10.1109/TSMCC.2007.913919
38. Chen, D., Doumeingts, G., & Vernadat, F. (2008). Architectures for enterprise integration and interoperability: Past, present and future. Computers in Industry, 59(7), 647–659. https://doi.org/10.1016/j.compind.2007.12.016
39. Chen, J., Li, Z., & Wang, Y. (2022). Multi robot task allocation in e commerce RMFS based on multi agent deep reinforcement learning. Mathematical Biosciences and Engineering, 20(2), 1–23. https://doi.org/10.3934/mbe.2023087
40. Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. Mobile Networks and Applications, 19(2), 171–209. https://doi.org/10.1007/s11036-013-0489-0
41. Chiang, M., & Zhang, T. (2016). Fog and IoT: An overview of research opportunities. IEEE Internet of Things Journal, 3(6), 854–864. https://doi.org/10.1109/JIOT.2016.2584538
42. Cimino, C., Negri, E., & Fumagalli, L. (2019). Review of digital twin applications in manufacturing. Computers in Industry, 113, 103130. https://doi.org/10.1016/j.compind.2019.103130
43. Clark, S., & Watling, D. (2005). Modelling network travel time reliability under stochastic demand.
44. Transportation Research Part B: Methodological, 39(2), 119–140. https://doi.org/10.1016/j.trb.2003.10.006
45. Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12(4), 568–581. https://doi.org/10.1287/opre.12.4.568
46. Cobbe, K., Klimov, O., Hesse, C., Kim, T., & Schulman, J. (2020). Leveraging procedural generation to benchmark reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.1912.01588
47. Corvello, V., Iazzolino, G., & Verteramo, S. (2025). City logistics 4.0: A reconceptualization of the domain through technology and sustainability perspectives. Annals of Operations Research. https://doi.org/10.1007/s10479-025-06835-x
48. Cover, T. M., & Thomas, J. A. (2006). Elements of information theory (2nd ed.). Wiley. https://doi.org/10.1002/047174882X
49. Dabney, W., Rowland, M., Bellemare, M. G., & Munos, R. (2018). Distributional reinforcement learning with quantile regression. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). https://doi.org/10.1609/aaai.v32i1.11791
50. Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. Management Science, 6(1), 80–91. https://doi.org/10.1287/mnsc.6.1.80
51. Davidsson, P., Henesey, L., Ramstedt, L., Törnquist, J., & Wernstedt, F. (2005). An analysis of agent based approaches to transport logistics. Transportation Research Part C: Emerging Technologies, 13(4), 255–271. https://doi.org/10.1016/j.trc.2005.07.002
52. Davidsson, P., Henesey, L., Ramstedt, L., Törnquist, J., & Wernstedt, F. (2004). Agent based approaches to transport logistics. In Agent and multi agent systems: Technologies and applications (pp. 1–16). Springer. https://doi.org/10.1007/3-7643-7363-6_1
53. Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1), 107–113. https://doi.org/10.1145/1327452.1327492
54. Dimakis, A. G., Kar, S., Moura, J. M. F., Rabbat, M. G., & Scaglione, A. (2010). Gossip algorithms for distributed signal processing. Proceedings of the IEEE, 98(11), 1847–1864. https://doi.org/10.1109/JPROC.2010.2052531
55. Ding, Z., Wang, X., Li, J., & Zhang, Y. (2024). Identifying poisoning attacks in federated learning online. Scientific Reports, 14, 70375. https://doi.org/10.1038/s41598-024-70375-w Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv. https://doi.org/10.48550/arXiv.1702.08608
56. Dragoni, N., Lanese, I., Larsen, S. T., Mazzara, M., Mustafin, R., & Safina, L. (2017). Microservices: Yesterday, today, and tomorrow. Present and Ulterior Software Engineering, 195–216. https://doi.org/10.1007/978-3-319-67425-4_12
57. Dror, M., & Trudeau, P. (1989). Vehicle routing with stochastic demands: Properties and solution frameworks. Transportation Science, 23(3), 166–176. https://doi.org/10.1287/trsc.23.3.166
58. Dubey, R., Gunasekaran, A., Childe, S. J., Wamba, S. F., & Papadopoulos, T. (2019). Big data analytics and artificial intelligence pathway to operational performance under the effects of entrepreneurial

orientation and environmental dynamism: A study of manufacturing organizations. International Journal of Production Economics, 226, 107599. https://doi.org/10.1016/j.ijpe.2019.107599

59. Dulac Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S., & Hester, T. (2021). An empirical investigation of the challenges of real world reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.2003.11881
60. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, 214–226. https://doi.org/10.1145/2090236.2090255
61. El Hamdi, S., Abouabdellah, A., & Bouchentouf, T. (2022). Logistics: Impact of Industry 4.0. Applied Sciences, 12(9), 4209. https://doi.org/10.3390/app12094209
62. Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., & Madry, A. (2020). Implementation matters in deep RL: A case study on PPO and TRPO. arXiv. https://doi.org/10.48550/arXiv.2005.12729
63. Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., Legg, S., & Kavukcuoglu, K. (2018). IMPALA: Scalable distributed deep RL with importance weighted actor learner architectures. International Conference on Machine Learning. https://doi.org/10.48550/arXiv.1802.01561
64. Eugster, P. T., Felber, P. A., Guerraoui, R., & Kermarrec, A.-M. (2003). The many faces of publish/subscribe. ACM Computing Surveys, 35(2), 114–131. https://doi.org/10.1145/857076.857078
65. Finn, C., Abbeel, P., & Levine, S. (2017). Model agnostic meta learning for fast adaptation of deep networks. International Conference on Machine Learning. https://doi.org/10.48550/arXiv.1703.03400
66. Floridi, L., Cowls, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., Luetge, C., Madelin, R., Pagallo, U., Rossi, F., Schafer, B., Valcke, P., & Vayena, E. (2018). AI4People—An ethical framework for a good AI society: Opportunities, risks, principles, and recommendations. Minds and Machines, 28(4), 689–707. https://doi.org/10.1007/s11023-018-9482-5
67. Foerster, J. N., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016). Learning to communicate with deep multi agent reinforcement learning. Advances in Neural Information Processing Systems. https://doi.org/10.48550/arXiv.1605.06676
68. Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., & Whiteson, S. (2018). Counterfactual multi agent policy gradients. In Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). https://doi.org/10.1609/aaai.v32i1.11794
69. Foerster, J., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016). Learning to communicate with deep multi agent reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.1605.06676
70. Fu, J., Kumar, A., Nachum, O., Tucker, G., & Levine, S. (2020). D4RL: Datasets for deep data driven reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.2004.07219
71. Fujimoto, S., Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor critic methods. arXiv. https://doi.org/10.48550/arXiv.1802.09477
72. Fujimoto, S., van Hoof, H., & Meger, D. (2019). Addressing function approximation error in actor critic methods. In Proceedings of the 36th International Conference on Machine Learning (pp. 1587–1596). https://doi.org/10.48550/arXiv.1802.09477
73. Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. IEEE Access, 8, 108952–108971. https://doi.org/10.1109/ACCESS.2020.2998358
74. Gijsbrechts, J., Boute, R., Van Mieghem, J., & Zhang, N. (2022). Can deep reinforcement learning improve inventory management? Performance and caveats. Manufacturing & Service Operations Management, 24(4), 1871–1898. https://doi.org/10.1287/msom.2021.1064
75. Giuseppi, A., & others. (2025). Enhancing federated reinforcement learning: A consensus-based perspective. International Journal of Automation and Computing, 22, 1–22. https://doi.org/10.1007/s11633-025-1550-8
76. Gleave, A., Dennis, M., Wild, C., Kant, N., Levine, S., & Russell, S. (2020). Adversarial policies: Attacking deep reinforcement learning. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1905.10615
77. Grieves, M., & Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In F. J. Kahlen, S. Flumerfelt, & A. Alves (Eds.), Transdisciplinary perspectives on complex systems (pp. 85–113). Springer. https://doi.org/10.1007/978-3-319-38756-7_4

78. Gronauer, S., & Diepold, K. (2022). Multi agent deep reinforcement learning: A survey. Artificial Intelligence Review, 55, 895–943. https://doi.org/10.1007/s10462-021-09996-w
79. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010
80. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. ACM Computing Surveys, 51(5), Article 93. https://doi.org/10.1145/3236009
81. Gunasekaran, A., Patel, C., & Tirtiroglu, E. (2001). Performance measures and metrics in a supply chain environment. International Journal of Operations & Production Management, 21(1/2), 71–87. https://doi.org/10.1108/01443570110358468
82. Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor critic: Off policy maximum entropy deep reinforcement learning with a stochastic actor. https://doi.org/10.48550/arXiv.1801.01290
83. Hall, R. W. (1978). Properties of the equilibrium state in transportation networks. Transportation Science, 12(3), 208–216. https://doi.org/10.1287/trsc.12.3.208
84. Harby, A. A., & Zulkernine, F. (2025). Data lakehouse: A survey and experimental study. Information Systems, 127, 102460. https://doi.org/10.1016/j.is.2024.102460
85. He, C., Annavaram, M., & Avestimehr, S. (2020). Group knowledge transfer: Federated learning of large CNNs at the edge. Advances in Neural Information Processing Systems, 33, 14068–14080. https://doi.org/10.48550/arXiv.2007.14513
86. He, L., Xue, M., & Gu, B. (2020). Internet-of-things enabled supply chain planning and coordination with big data services: Certain theoretic implications. Journal of Management Science and Engineering, 5(1), 1–14. https://doi.org/10.1016/j.jmse.2020.03.002
87. Helo, P., & Hao, Y. (2020). Blockchains in operations and supply chains: A model and reference implementation. Computers & Industrial Engineering, 136, 242–251. https://doi.org/10.1016/j.cie.2019.07.023
88. Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2018). Deep reinforcement learning that matters. In Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). https://doi.org/10.1609/aaai.v32i1.11694
89. Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2018). Deep reinforcement learning that matters. arXiv. https://doi.org/10.48550/arXiv.1709.06560
90. Hernández Leal, P., Kartal, B., & Taylor, M. E. (2019). A survey and critique of multi agent deep reinforcement learning. Autonomous Agents and Multi Agent Systems, 33, 750–797. https://doi.org/10.1007/s10458-019-09421-1
91. Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., & Silver, D. (2018). Rainbow: Combining improvements in deep reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). https://doi.org/10.1609/aaai.v32i1.11796
92. Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., & Silver, D. (2018). Rainbow: Combining improvements in deep reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.1710.02298
93. Hohenstein, N.-O., Feisel, E., Hartmann, E., & Giunipero, L. (2015). Research on the phenomenon of supply chain resilience. International Journal of Physical Distribution & Logistics Management, 45(1/2), 90–117. https://doi.org/10.1108/IJPDLM-05-2013-0128
94. Holguín Veras, J., Jaller, M., Van Wassenhove, L. N., Pérez, N., & Wachtendorf, T. (2012). On the unique features of post disaster humanitarian logistics. Journal of Operations Management, 30(7–8), 494–506. https://doi.org/10.1016/j.jom.2012.08.003
95. Hortelano, D., de Miguel, I., Barroso, R. J., Aguado, J. C., Merayo, N., Ruiz, L., Asensio, A., Masip-Bruin, X., Fernández, P., Lorenzo, R. M., & others. (2023). A comprehensive survey on reinforcement-learning-based
96. computation offloading techniques in edge computing systems. Journal of Network and Computer Applications, 216, 103669. https://doi.org/10.1016/j.jnca.2023.103669
97. Hosseini, S., Ivanov, D., & Dolgui, A. (2019). Review of quantitative methods for supply chain resilience analysis. Transportation Research Part E: Logistics and Transportation Review, 125, 285–307. https://doi.org/10.1016/j.tre.2019.03.001

98. Hsu, B. M., Hsu, L. Y., & Shu, M. H. (2013). Evaluation of supply chain performance using delivery-time performance analysis chart approach. Journal of Statistics and Management Systems, 16(1), 73–87. https://doi.org/10.1080/09720510.2013.777568
99. Huang, J., Liu, J., Zhou, Y., Li, X., Ji, S., Xiong, H., & Dou, D. (2022). From distributed machine learning to federated learning: A survey. Knowledge and Information Systems, 64(4), 885–917. https://doi.org/10.1007/s10115-022-01664-x
100. Improving inventory management quality with reinforcement learning. (2025). Accounting Horizons. https://doi.org/10.2308/HORIZONS-2024-121
101. Isard, M., Budiu, M., Yu, Y., Birrell, A., & Fetterly, D. (2007). Dryad: Distributed data parallel programs from sequential building blocks. Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems, 59–72. https://doi.org/10.1145/1272996.1273005
102. Ivanov, D. (2020). Viability of supply chain networks and supply chain resilience: A systems approach. Annals of Operations Research, 319, 1023–1063. https://doi.org/10.1007/s10479-020-03640-6
103. Ivanov, D., & Dolgui, A. (2020). A digital supply chain twin for managing the disruption risks and resilience in the era of Industry 4.0. Production Planning & Control, 32(9), 775–788. https://doi.org/10.1080/09537287.2020.1768450
104. Ivanov, D., & Dolgui, A. (2020). Viability of intertwined supply networks: Extending the supply chain resilience angles towards survivability. International Journal of Production Research, 58(10), 2904–2915. https://doi.org/10.1080/00207543.2020.1750727
105. Jacobs, F. R., & Weston, F. C., Jr. (2007). Enterprise resource planning (ERP)—A brief history. Journal of Operations Management, 25(2), 357–363. https://doi.org/10.1016/j.jom.2006.11.005
106. Jiang, Q., Shi, S., Zhu, X., & Zhang, X. (2022). Multi agent reinforcement learning for traffic signal control. arXiv. https://doi.org/10.48550/arXiv.2204.12190
107. Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. Nature Machine Intelligence, 1(9), 389–399. https://doi.org/10.1038/s42256-019-0088-2
108. Joe, W., & Lau, H. C. (2020). Deep reinforcement learning approach to solve dynamic vehicle routing problem with stochastic customers. In Proceedings of the Thirtieth International Conference on Automated Planning and Scheduling (ICAPS) (pp. 394–402). AAAI Press. https://doi.org/10.1609/icaps.v30i1.6685
109. Juliani, A., Berges, V. P., Vckay, E., Gao, Y., Henry, H., Mattar, M., & Lange, D. (2018). Unity: A general platform for intelligent agents. arXiv. https://doi.org/10.48550/arXiv.1809.02627
110. Kache, F., & Seuring, S. (2017). Challenges and opportunities of digital information at the intersection of Big Data Analytics and supply chain management. International Journal of Operations & Production Management, 37(1), 10–36. https://doi.org/10.1108/IJOPM-02-2015-0078
111. Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., ... Wright, R. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1–2), 1–210. https://doi.org/10.1561/2200000083
112. Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. Computer, 36(1), 41–50. https://doi.org/10.1109/MC.2003.1160055
113. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska Barwińska, A., Hassabis, D., Clopath, C., Kumaran, D., & Hadsell, R. (2017).
114. Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences, 114(13), 3521–3526. https://doi.org/10.1073/pnas.1611835114
115. Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. The International Journal of Robotics Research, 32(11), 1238–1274. https://doi.org/10.1177/0278364913495721
116. Kolat, M., Maciag, K., & Piotrowski, K. (2023). Multi agent reinforcement learning for traffic signal control. Sustainability, 15(4), 3479. https://doi.org/10.3390/su15043479
117. Konda, V. R., & Tsitsiklis, J. N. (2003). Actor–critic algorithms. SIAM Journal on Control and Optimization, 42(4), 1143–1166. https://doi.org/10.1137/S0363012901385691
118. Kool, W., Van Hoof, H., & Welling, M. (2019). Attention, learn to solve routing problems. https://doi.org/10.48550/arXiv.1803.08475

119. Kouhizadeh, M., Saberi, S., & Sarkis, J. (2021). Blockchain technology and the sustainable supply chain: Theoretically exploring adoption barriers. International Journal of Production Economics, 231, 107831. https://doi.org/10.1016/j.ijpe.2020.107831
120. Kouicem, D. E., Bouabdallah, A., & Lakhlef, H. (2018). Internet of Things security: A top down survey. Computer Networks, 141, 199–221. https://doi.org/10.1016/j.comnet.2018.03.012
121. Kovács, G., & Spens, K. M. (2007). Humanitarian logistics in disaster relief operations. International Journal of Physical Distribution & Logistics Management, 37(2), 99–114. https://doi.org/10.1108/09600030710734820
122. Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review and classification. IFAC PapersOnLine, 51(11), 1016–1022. https://doi.org/10.1016/j.ifacol.2018.08.474
123. Kulkarni, T. D., Narasimhan, K., Saeedi, A., & Tenenbaum, J. B. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. Advances in Neural Information Processing Systems. https://doi.org/10.48550/arXiv.1604.06057
124. Kumar, A., Fu, J., Soh, M., Tucker, G., & Levine, S. (2020). Conservative Q learning for offline reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.2006.04779
125. Kwon, O., Lee, N., & Shin, B. (2014). Data quality management, data usage experience and acquisition intention of big data analytics. International Journal of Information Management, 34(3), 387–394. https://doi.org/10.1016/j.ijinfomgt.2014.02.002
126. Lambert, D. M., & Cooper, M. C. (2000). Issues in supply chain management. Industrial Marketing Management, 29(1), 65–83. https://doi.org/10.1016/S0019-8501(99)00113-3
127. Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. Communications of the ACM, 21(7), 558–565. https://doi.org/10.1145/359545.359563
128. Lanctot, M., Lockhart, E., Lespiau, J B., Zambaldi, V., Upadhyay, S., Pires, B. A. O., Yang, S., Tuyls, K., Pérolat, J., & Graepel, T. (2019). OpenSpiel: A framework for reinforcement learning in games. arXiv. https://doi.org/10.48550/arXiv.1908.09453
129. Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. European Journal of Operational Research, 59(3), 345–358. https://doi.org/10.1016/0377-2217(92)90192-C
130. Laporte, G. (2007). What you should know about the vehicle routing problem. Naval Research Logistics, 54(8), 811–819. https://doi.org/10.1002/nav.20261
131. Le, D. N., & Fan, L. (2024). Digital twin in logistics and supply chain management: A systematic literature review and future research agenda. Computers & Industrial Engineering, 190, 109768. https://doi.org/10.1016/j.cie.2023.109768
132. Lee, D. H., & Kwon, H. (2023). A deep reinforcement learning approach to solve the team orienteering problem. In AIAA SCITECH 2023 Forum. https://doi.org/10.2514/6.2023-2662
133. Lee, D., Kim, S., & Cho, H. (2025). Digital twin driven deep reinforcement learning for real time control of automated guided vehicles in intralogistics. International Journal of Production Research. https://doi.org/10.1080/00207543.2025.2543491
134. Lee, E. A. (2008). Cyber physical systems: Design challenges. Proceedings of the 11th IEEE International Symposium on Object Oriented Real Time Distributed Computing (ISORC), 363–369. https://doi.org/10.1109/ISORC.2008.25
135. Lesort, T., Díaz Rodríguez, N., Goudou, J. F., & Filliat, D. (2020). Continual learning for robotics: Definition, frameworks, challenges, and opportunities. Information Fusion, 58, 52–68. https://doi.org/10.1016/j.inffus.2019.12.004
136. Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End to end training of deep visuomotor policies. Journal of Machine Learning Research, 17(39), 1–40. https://doi.org/10.48550/arXiv.1504.00702
137. Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv. https://doi.org/10.48550/arXiv.2005.01643
138. Li, J., Zhang, X., & Chen, H. (2025). Digital twin driven deep reinforcement learning for real time intralogistics optimization. International Journal of Production Research. https://doi.org/10.1080/00207543.2025.2543491
139. Li, M., Long, Y., Li, T., Liang, H., & Chen, C. L. P. (2024). Dynamic event triggered consensus control for input constrained multi agent systems with a designable minimum inter event time. IEEE CAA Journal of Automatica Sinica, 11(3), 649–660. https://doi.org/10.1109/JAS.2023.123582

140. Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50–60. https://doi.org/10.1109/MSP.2020.2975749
141. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2016). Continuous control with deep reinforcement learning. https://doi.org/10.48550/arXiv.1509.02971
142. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2016). Continuous control with deep reinforcement learning. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1509.02971
143. Lim, M. K., Bahr, W., & Leung, S. C. H. (2013). RFID in the warehouse: A literature analysis (1995–2010) of its applications, benefits, challenges and future trends. International Journal of Production Economics, 145(1), 409–430. https://doi.org/10.1016/j.ijpe.2013.05.006
144. Lin, K., Zhao, R., Xu, Z., & Zhou, J. (2018). Efficient large-scale fleet management via multi-agent deep reinforcement learning. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1774–1783). Association for Computing Machinery. https://doi.org/10.1145/3219819.3219993
145. Liu, J., Huang, J., Zhou, Y., Li, Y., & others. (2022). From distributed machine learning to federated learning: A survey. Knowledge and Information Systems, 64(4), 885–917. https://doi.org/10.1007/s10115-022-01664-x
146. Liu, X., Xu, Y., & Yang, S. (2024). Multi agent deep reinforcement learning for multi echelon inventory management. Journal of Supply Chain Management. https://doi.org/10.1177/10591478241305863
147. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, O. P., & Mordatch, I. (2017). Multi agent actor critic for mixed cooperative competitive environments. arXiv. https://doi.org/10.48550/arXiv.1706.02275
148. Lu, Y., Liu, C., Wang, K. I K., Huang, H., & Xu, X. (2020). Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. Robotics and Computer-Integrated Manufacturing, 61, 101837. https://doi.org/10.1016/j.rcim.2019.101837
149. Macal, C. M., & North, M. J. (2010). Tutorial on agent based modelling and simulation. Journal of Simulation, 4(3), 151–162. https://doi.org/10.1057/jos.2010.3
150. MacCarthy, B. L., Blome, C., Olhager, J., Srai, J. S., & Zhao, X. (2016). Supply chain evolution—Theory, concepts and science. International Journal of Operations & Production Management, 36(12), 1696–1718. https://doi.org/10.1108/IJOPM-02-2016-0080
151. Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials, 19(3), 1628–1656. https://doi.org/10.1109/COMST.2017.2682318
152. Mai, T., Zhang, H., & Leung, V. C. M. (2020). Multi agent actor critic reinforcement learning based intelligent resource allocation. In 2020 IEEE Global Communications Conference (GLOBECOM) (pp. 1–6). https://doi.org/10.1109/GLOBECOM42002.2020.9322277
153. Malewicz, G., Austern, M. H., Bik, A. J. C., Dehnert, J. C., Horn, I., Leiser, N., & Czajkowski, G. (2010).
154. Pregel: A system for large scale graph processing. Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, 135–146. https://doi.org/10.1145/1807167.1807184
155. Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials, 19(4), 2322–2358. https://doi.org/10.1109/COMST.2017.2745201
156. Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2019. An Empirical Study of Rich Subgroup Fairness for Machine Learning. In Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT* '19). Association for Computing Machinery, New York, NY, USA, 100–109. https://doi.org/10.1145/3287560.3287592
157. Mili, K. (2025). Adaptive vehicle routing for humanitarian aid in conflict settings: A stochastic and AI based approach. Frontiers in Future Transportation, 6, 1603726. https://doi.org/10.3389/ffutr.2025.1603726
158. Minerva, R., Lee, G. M., & Crespi, N. (2020). Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models. Proceedings of the IEEE, 108(10), 1789–1824. https://doi.org/10.1109/JPROC.2020.2998530

159. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru,
160. T. (2019). Model cards for model reporting. Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*), 220–229. https://doi.org/10.1145/3287560.3287596
161. Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., & Kavukcuoglu, K.
162. (2016). Asynchronous methods for deep reinforcement learning. International Conference on Machine Learning. https://doi.org/10.48550/arXiv.1602.01783
163. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human level control through deep reinforcement learning. Nature, 518(7540), 529–533. https://doi.org/10.1038/nature14236
164. Moshood, T. D., Nawanir, G., Sorooshian, S., Okfalisa, & van Viet, P. (2020). Digital twin driven supply chain and logistics: A review of digital twin applications in supply chain management. Logistics, 4(2), 29. https://doi.org/10.3390/logistics4020029
165. Nazari, M., Oroojlooy, A., Snyder, L. V., & Takáč, M. (2018). Reinforcement learning for solving the vehicle routing problem. arXiv. https://doi.org/10.48550/arXiv.1802.04240
166. Nedic, A., & Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. IEEE Transactions on Automatic Control, 54(1), 48–61. https://doi.org/10.1109/TAC.2008.2009515
167. Negri, E., Fumagalli, L., & Macchi, M. (2017). A review of the roles of digital twin in CPS based production systems. Procedia Manufacturing, 11, 939–948. https://doi.org/10.1016/j.promfg.2017.07.198
168. Ngai, E. W. T., Moon, K. K.-L., Riggins, F. J., & Yi, C. Y. (2008). RFID research: An academic literature review (1995–2005) and future research directions. International Journal of Production Economics, 112(2), 510–520. https://doi.org/10.1016/j.ijpe.2007.05.004
169. Nguyen, L. K. N., Howick, S., & Megiddo, I. (2024). A framework for conceptualising hybrid system dynamics and agent based simulation models. European Journal of Operational Research, 315(3), 1153–1166. https://doi.org/10.1016/j.ejor.2024.01.027
170. Nichol, A., Achiam, J., & Schulman, J. (2018). On first order meta learning algorithms. arXiv. https://doi.org/10.48550/arXiv.1803.02999
171. Ning, B., Liu, Z., Fang, C., Yang, H., & Zhang, J. (2024). A survey on multi agent reinforcement learning. Journal of Artificial Intelligence, 6(1), 1–32. https://doi.org/10.1016/j.jai.2024.02.003
172. Ning, Z., & Xie, L. (2024). A survey on multi agent reinforcement learning and its application. Journal of Automation and Intelligence, 3(2), 73–91. https://doi.org/10.1016/j.jai.2024.02.003
173. Ning, Z., Zhang, Z., Xia, F., Ullah, N., Kong, X., & Hu, X. (2024). A survey on multi agent reinforcement learning and its application. Journal of Artificial Intelligence, 1(1), 1–36. https://doi.org/10.1016/j.jai.2024.02.003
174. Nowzari, C., Garcia, E., & Cortés, J. (2019). Event triggered communication and control of networked systems for multi agent consensus. Automatica, 105, 1–27. https://doi.org/10.1016/j.automatica.2019.03.009
175. Olfati Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi agent systems. Proceedings of the IEEE, 95(1), 215–233. https://doi.org/10.1109/JPROC.2006.887293
176. Oliveira, T., Thomas, M., & Espadanal, M. (2014). Assessing the determinants of cloud computing adoption: An analysis of the manufacturing and services sectors. Information & Management, 51(5), 497–510. https://doi.org/10.1016/j.im.2014.03.006
177. Özdamar, L., & Ertem, M. A. (2015). Models, solutions and enabling technologies in humanitarian logistics. European Journal of Operational Research, 244(1), 55–65. https://doi.org/10.1016/j.ejor.2014.11.030
178. Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191
179. Panetto, H., & Molina, A. (2008). Enterprise integration and interoperability in manufacturing systems: Trends and issues. Computers in Industry, 59(7), 641–646. https://doi.org/10.1016/j.compind.2007.12.010
180. Papazoglou, M. P., & van den Heuvel, W. J. (2007). Service oriented architectures: Approaches, technologies and research issues. The VLDB Journal, 16(3), 389–415. https://doi.org/10.1007/s00778-007-0044-3

181. Pardo, F., Tavakoli, A., Levdik, V., & Kormushev, P. (2018). Time limits in reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.1712.00378
182. Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. Neural Networks, 113, 54–71. https://doi.org/10.1016/j.neunet.2019.01.012
183. Peng, X. B., Andrychowicz, M., Zaremba, W., & Abbeel, P. (2018). Sim to real transfer of robotic control with dynamics randomization. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 3803–3810). IEEE. https://doi.org/10.1109/ICRA.2018.8460528
184. Pillac, V., Gueret, C., & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. European Journal of Operational Research, 225(1), 1–11. https://doi.org/10.1016/j.ejor.2012.08.015
185. Powell, W. B. (2019). A unified framework for stochastic optimization. European Journal of Operational Research, 275(3), 795–821. https://doi.org/10.1016/j.ejor.2018.07.014
186. Psaraftis, H. N., Wen, M., & Kontovas, C. A. (2016). Dynamic vehicle routing problems: Three decades and counting. Networks, 67(1), 3–31. https://doi.org/10.1002/net.21628
187. Qi, J.; Zhou, Q.; Lei, L.; Zheng, K. Federated reinforcement learning: techniques, applications, and open challenges. Intell. Robot. 2021, 1, 18-57. http://dx.doi.org/10.20517/ir.2021.02
188. Qi, Q., & Tao, F. (2018). Digital twin and big data towards smart manufacturing and Industry 4.0: 360 degree comparison. IEEE Access, 6, 3585–3593. https://doi.org/10.1109/ACCESS.2018.2793265
189. Queiroz, M. M., Ivanov, D., Dolgui, A., & Wamba, S. F. (2020). Impacts of epidemic outbreaks on supply chains: Mapping a research agenda amid the COVID-19 pandemic through a structured literature review. Annals of Operations Research, 319, 1159–1196. https://doi.org/10.1007/s10479-020-03685-7
190. Rahimi, F., Møller, C., & Hvam, L. (2016). Business process management and IT management: The missing integration. International Journal of Information Management, 36(1), 142–154. https://doi.org/10.1016/j.ijinfomgt.2015.10.004
191. Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D., & Barnes, P. (2020). Closing the AI accountability gap: Defining an end-to-end framework for internal
192. algorithmic auditing. Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 33– 44. https://doi.org/10.1145/3351095.3372873
193. Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. (2010). Cyber physical systems: The next computing revolution. Proceedings of the 47th Design Automation Conference, 731–736. https://doi.org/10.1145/1837274.1837461
194. Rasheed, A., San, O., & Kvamsdal, T. (2020). Digital twin: Values, challenges and enablers from a modeling perspective. IEEE Access, 8, 21980–22012. https://doi.org/10.1109/ACCESS.2020.2970143
195. Rashid, T., Farquhar, G., Peng, B., & Whiteson, S. (2020). Monotonic value function factorisation for deep multi agent reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.2003.08839
196. Rashid, T., Samvelyan, M., De Witt, C. S., Farquhar, G., Foerster, J., & Whiteson, S. (2018). QMIX: Monotonic value function factorisation for deep multi agent reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.1803.11485
197. Ren, W., & Beard, R. W. (2008). Distributed consensus in multi vehicle cooperative control. Springer. https://doi.org/10.1007/978-1-84800-015-5
198. Roijers, D. M., Vamplew, P., Whiteson, S., & Dazeley, R. (2014). A survey of multi-objective sequential decision-making. arXiv. https://doi.org/10.48550/arXiv.1402.0590
199. Roman, R., Zhou, J., & Lopez, J. (2013). On the features and challenges of security and privacy in distributed Internet of Things. Computer Networks, 57(10), 2266–2279. https://doi.org/10.1016/j.comnet.2012.12.018
200. Roughgarden, T., & Tardos, É. (2002). How bad is selfish routing. Journal of the ACM, 49(2), 236–259. https://doi.org/10.1145/506147.506153
201. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5), 206–215. https://doi.org/10.1038/s42256-0190048-x
202. Samvelyan, M., Rashid, T., de Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G. J., Hung, C. M., Torr, P. H. S., Foerster, J., & Whiteson, S. (2019). The StarCraft multi agent challenge. arXiv. https://doi.org/10.48550/arXiv.1902.04043

203. Santos, R., Costa, A., Rocha, A., & Barbosa, J. (2024). A simulation-based digital twin architecture for enhanced decision making in production systems. Computers & Industrial Engineering, 197, 110616. https://doi.org/10.1016/j.cie.2024.110616
204. Satyanarayanan, M. (2017). The emergence of edge computing. Computer, 50(1), 30–39. https://doi.org/10.1109/MC.2017.9
205. Satyanarayanan, M., Bahl, P., Caceres, R., & Davies, N. (2009). The case for VM based cloudlets in mobile computing. IEEE Pervasive Computing, 8(4), 14–23. https://doi.org/10.1109/MPRV.2009.82
206. Sayed, A. H. (2014). Adaptation, learning, and optimization over networks. Foundations and Trends in Machine Learning, 7(4–5), 311–801. https://doi.org/10.1561/2200000051
207. Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2016). Prioritized experience replay. In Proceedings of the International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1511.05952
208. Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., & Silver, D. (2020). Mastering Atari, Go, chess and shogi by planning with a learned model. Nature, 588(7839), 604–609. https://doi.org/10.1038/s41586-020-03051-4
209. Schulman, J., Levine, S., Moritz, P., Jordan, M. I., & Abbeel, P. (2015). Trust region policy optimization. https://doi.org/10.48550/arXiv.1502.05477
210. Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2016). High dimensional continuous control using generalized advantage estimation. https://doi.org/10.48550/arXiv.1506.02438
211. Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015). High dimensional continuous control using generalized advantage estimation. arXiv. https://doi.org/10.48550/arXiv.1506.02438
212. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. https://doi.org/10.48550/arXiv.1707.06347
213. Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. Communications of the ACM, 63(12), 54–63. https://doi.org/10.1145/3381831
214. Seipolt, A., & Bauernhansl, T. (2024). Reinforcement learning and digital twin-driven optimization of production scheduling. Manufacturing Review, 11, 17. https://doi.org/10.1007/s43926-024-00087-0
215. Shalev Shwartz, S., Shammah, S., & Shashua, A. (2016). Safe, multi agent, reinforcement learning for autonomous driving. arXiv. https://doi.org/10.48550/arXiv.1610.03295
216. Shapiro, A., Dentcheva, D., & Ruszczyński, A. (2014). Lectures on stochastic programming: Modeling and theory (2nd ed.). SIAM. https://doi.org/10.1137/1.9781611973433
217. Sheu, J. B. (2007). An emergency logistics distribution approach for quick response to urgent relief demand in disasters. Transportation Research Part E: Logistics and Transportation Review, 43(6), 687–709. https://doi.org/10.1016/j.tre.2006.04.004
218. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198
219. Shoham, Y., Powers, R., & Grenager, T. (2007). If multi agent learning is the answer, what is the question. Artificial Intelligence, 171(7), 365–377. https://doi.org/10.1016/j.artint.2006.02.006
220. Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. Computer Networks, 76, 146–164. https://doi.org/10.1016/j.comnet.2014.11.008
221. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484–489. https://doi.org/10.1038/nature16961
222. Smith, S. L., Pavone, M., Bullo, F., & Frazzoli, E. (2010). Dynamic vehicle routing with priority classes of stochastic demands. SIAM Journal on Control and Optimization, 48(5), 3224–3245. https://doi.org/10.1137/090749347
223. Son, K., Kim, D., Kang, W., Hostallero, D., & Yi, Y. (2019). QTRAN: Learning to factorize with transformation for cooperative multi agent reinforcement learning. International Conference on Machine Learning. https://doi.org/10.48550/arXiv.1905.05408

224. Spieser, K., Treleaven, K., Zhang, R., Frazzoli, E., Morton, D., & Pavone, M. (2014). Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems. Road Vehicle Automation, 229–245. https://doi.org/10.1007/978-3-319-05990-7_20
225. Stranieri, F., Jorjani, S., & Trivedi, K. (2024). Performance of deep reinforcement learning algorithms in supply chain inventory management. International Journal of Production Research. https://doi.org/10.1080/00207543.2024.2311180
226. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. arXiv. https://doi.org/10.48550/arXiv.1906.02243
227. Sukhbaatar, S., Szlam, A., & Fergus, R. (2016). Learning multi agent communication with backpropagation. Advances in Neural Information Processing Systems. https://doi.org/10.48550/arXiv.1605.07736
228. Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., & Graepel, T. (2017). Value decomposition networks for cooperative multi agent learning. arXiv. https://doi.org/10.48550/arXiv.1706.05296
229. Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. Machine Learning, 3(1), 9–44. https://doi.org/10.1023/A:1022633531479
230. Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction (key concepts widely used in architecture). IEEE Transactions on Neural Networks, 9(5), 1054–1054. https://doi.org/10.1109/TNN.1998.712192
231. Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., & Wiewiora, E. (2009). Fast gradient descent methods for temporal difference learning with linear function approximation. In Proceedings of the 26th International Conference on Machine Learning (pp. 993–1000). https://doi.org/10.1145/1553374.1553501
232. Tako, A. A., & Robinson, S. (2012). The application of discrete event simulation and system dynamics in the logistics and supply chain context. Decision Support Systems, 52(4), 802–815. https://doi.org/10.1016/j.dss.2011.11.015
233. Tan, Q., Li, Z., & Wang, Y. (2025). Defending against backdoor attacks in federated learning for edge intelligence. CMES Computer Modeling in Engineering & Sciences. https://doi.org/10.32604/cmes.2025.063811
234. Tang, C. S. (2006). Perspectives in supply chain risk management. International Journal of Production Economics, 103(2), 451–488. https://doi.org/10.1016/j.ijpe.2005.12.006
235. Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., & Sui, F. (2018). Digital twin driven product design, manufacturing and service with big data. The International Journal of Advanced Manufacturing Technology, 94(9–12), 3563–3576. https://doi.org/10.1007/s00170-017-0233-1
236. Tao, F., Zhang, H., Liu, A., & Nee, A. Y. C. (2019). Digital twin in industry: State of the art. IEEE Transactions on Industrial Informatics, 15(4), 2405–2415. https://doi.org/10.1109/TII.2018.2873186
237. Thomas, P. S., & Brunskill, E. (2016). Data efficient off policy policy evaluation for reinforcement learning. In Proceedings of the 33rd International Conference on Machine Learning (pp. 2139–2148). https://doi.org/10.48550/arXiv.1604.00923
238. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 23–30). IEEE. https://doi.org/10.1109/IROS.2017.8202133
239. Toth, P., & Vigo, D. (2014). Vehicle routing: Problems, methods, and applications (2nd ed.). SIAM. https://doi.org/10.1137/1.9781611973594
240. Truex, S., Baracaldo, N., Anwar, A., Zhou, S., Ludwig, H., & Zhang, R. (2019). A hybrid approach to privacy preserving federated learning. Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, 1–12. https://doi.org/10.1145/3338501.3357370
241. Tuyls, K., & Weiss, G. (2012). Multi agent learning: Basics, challenges, and prospects. AI Magazine, 33(3), 41–52. https://doi.org/10.1609/aimag.v33i3.2426
242. Vamplew, P., Dazeley, R., Berry, A., Issabekov, R., & Dekker, E. (2011). Empirical evaluation methods for multi objective reinforcement learning algorithms. Machine Learning, 84, 51–80. https://doi.org/10.1007/s10994-010-5232-5
243. van der Valk, W., Haijema, R., & Reiner, G. (2022). Supply chains in the era of digital twins: A review. Procedia Computer Science, 200, 227–234. https://doi.org/10.1016/j.procs.2022.08.019

244. Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double Q learning. In Proceedings of the AAAI Conference on Artificial Intelligence, 30(1). https://doi.org/10.1609/aaai.v30i1.10295
245. Van Wassenhove, L. N. (2006). Humanitarian aid logistics: Supply chain management in high gear. Journal of the Operational Research Society, 57(5), 475–489. https://doi.org/10.1057/palgrave.jors.2602125
246. Varghese, B., & Buyya, R. (2018). Next generation cloud computing: New trends and research directions. Future Generation Computer Systems, 79, 849–861. https://doi.org/10.1016/j.future.2017.09.020
247. Vernadat, F. B. (2007). Interoperable enterprise systems: Architectures and methods. Annual Reviews in Control, 31(1), 137–147. https://doi.org/10.1016/j.arcontrol.2007.03.004
248. Villamizar, M., Garcés, O., Castro, H., Salamanca, L., Casallas, R., & Gil, S. (2015). Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud. 2015 10th Computing Colombian Conference (10CCC), 583–590. https://doi.org/10.1109/ColumbianCC.2015.7333476
249. Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R.,
250. Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J., Jaderberg, M., … Silver, D. (2019). Grandmaster level in StarCraft II using multi agent reinforcement learning. Nature, 575(7782), 350–354. https://doi.org/10.1038/s41586-019-1724-z
251. Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. Transportation Research Part C: Emerging Technologies, 43, 3–19. https://doi.org/10.1016/j.trc.2014.01.005
252. Vogels, W. (2009). Eventually consistent. Communications of the ACM, 52(1), 40–44. https://doi.org/10.1145/1435417.1435432
253. Waller, M. A., & Fawcett, S. E. (2013). Data science, predictive analytics, and big data: A revolution that will transform supply chain design and management. Journal of Business Logistics, 34(2), 77–84. https://doi.org/10.1111/jbl.12010
254. Wamba, S. F., Gunasekaran, A., Akter, S., Ren, S. J.-F., Dubey, R., & Childe, S. J. (2017). Big data analytics and firm performance: Effects of dynamic capabilities. Journal of Business Research, 70, 356–365. https://doi.org/10.1016/j.jbusres.2016.08.009
255. Wang, D., Chen, Y., & Lee, D. (2025). Digital twin driven management strategies for logistics operations with unmanned vehicles. Scientific Reports, 15, 96641. https://doi.org/10.1038/s41598-025-96641-z
256. Wang, D., Sun, L., & Szeto, W. Y. (2020). Dynamic holding control to avoid bus bunching: A multi-agent deep reinforcement learning framework. Transportation Research Part C: Emerging Technologies, 116, 102661. https://doi.org/10.1016/j.trc.2020.102661
257. Wang, G., Gunasekaran, A., Ngai, E. W. T., & Papadopoulos, T. (2016). Big data analytics in logistics and supply chain management: Certain investigations for research and applications. International Journal of Production Economics, 176, 98–110. https://doi.org/10.1016/j.ijpe.2016.03.014
258. Wang, J. X., Kurth Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., Hassabis, D., & Botvinick, M. (2016). Learning to reinforcement learn. arXiv. https://doi.org/10.48550/arXiv.1611.05763
259. Wang, J., Zhang, Z., & Wang, Y. (2022). More centralized training, still decentralized execution: Multi agent conditional policy factorization. arXiv. https://doi.org/10.48550/arXiv.2209.12681
260. Wang, L., Deng, T., Shen, Z.-J. M., Hu, H., & Qi, Y. (2022). Digital twin-driven smart supply chain. Engineering Management, 9(1), 56–70. https://doi.org/10.1007/s42524-021-0186-9
261. Wang, X., Chen, X., Wang, Y., & Zhang, H. (2023). Event triggered consensus control of heterogeneous leader follower multi agent systems. Science China Information Sciences, 66, 152202. https://doi.org/10.1007/s11432-022-3683-y
262. Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., & de Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. Proceedings of the 33rd International Conference on Machine Learning. https://doi.org/10.48550/arXiv.1511.06581
263. Watkins, C. J. C. H., & Dayan, P. (1992). Q learning. Machine Learning, 8(3–4), 279–292. https://doi.org/10.1007/BF00992698

264. Wen, J., Zhang, X., Lan, Y., Liu, Q., & Wang, J. (2022). From distributed machine learning to federated learning: A survey. Knowledge and Information Systems, 64(4), 885–917. https://doi.org/10.1007/s10115-02201664-x
265. Wieland, A., & Wallenburg, C. M. (2013). The influence of relational competencies on supply chain resilience: A relational view. International Journal of Physical Distribution & Logistics Management, 43(4), 300–320. https://doi.org/10.1108/IJPDLM-08-2012-0243
266. Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data, 3, 160018. https://doi.org/10.1038/sdata.2016.18
267. Williams, R. J. (1992). Simple statistical gradient following algorithms for connectionist reinforcement learning. Machine Learning, 8(3–4), 229–256. https://doi.org/10.1007/BF00992696
268. Winkelhaus, S., & Grosse, E. H. (2020). Logistics 4.0: A systematic review towards a new logistics system.
269. International Journal of Production Research, 58(1), 18–43. https://doi.org/10.1080/00207543.2019.1612964
270. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386
271. Wurman, P. R., D’Andrea, R., & Mountz, M. (2008). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. AI Magazine, 29(1), 9–20. https://doi.org/10.1609/aimag.v29i1.2082
272. Xia, Q., Ye, W., Tao, Z., Wu, J., & Li, Q. (2021). A survey of federated learning for edge computing: Research problems and solutions. High Confidence Computing, 1(1), 100008. https://doi.org/10.1016/j.hcc.2021.100008
273. Xie, X., Ban, X. (Jeff), Chen, H., Chen, Z., & Xu, S. (2023). Dynamic ridepooling with heterogeneous riders: A deep reinforcement learning approach. Transportation Science, 57(4), 1021–1044. https://doi.org/10.1287/trsc.2022.1188
274. Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology, 10(2), 1–19. https://doi.org/10.1145/3298981
275. Yau, K. L. A., Qadir, J., Khoo, H. L., Ling, M. H., & Komisarczuk, P. (2017). A survey on reinforcement learning models and algorithms for traffic signal control. ACM Computing Surveys, 50(3), Article 34. https://doi.org/10.1145/3068287
276. Yi, W., & Özdamar, L. (2007). A dynamic logistics coordination model for evacuation and support in disaster response activities. European Journal of Operational Research, 179(3), 1177–1193. https://doi.org/10.1016/j.ejor.2005.03.077
277. Zhang, C., Patras, P., & Haddadi, H. (2019). Deep learning in mobile and wireless networking: A survey. IEEE Communications Surveys & Tutorials, 21(3), 2224–2287. https://doi.org/10.1109/COMST.2019.2904897
278. Zhang, K., Yang, Z., & Basar, T. (2021). Multi agent reinforcement learning: A selective overview of theories and algorithms. Handbook of Reinforcement Learning and Control, 321–384. https://doi.org/10.1007/978-3030-60990-0_12
279. Zhang, K., Yang, Z., & Başar, T. (2021). Multi agent reinforcement learning: A selective overview of theories and algorithms. https://doi.org/10.48550/arXiv.1911.10635
280. Zhang, K., Yang, Z., & Başar, T. (2021). Multi agent reinforcement learning: A selective overview of theories and algorithms. In Handbook of Reinforcement Learning and Control (pp. 321–384). Springer. https://doi.org/10.1007/978-3-030-60990-0_12
281. Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. Information Fusion, 42, 146–157. https://doi.org/10.1016/j.inffus.2017.10.006
282. Zhang, R., & Pavone, M. (2016). Control of robotic mobility on demand systems: A queueing theoretical perspective. The International Journal of Robotics Research, 35(1–3), 186–203. https://doi.org/10.1177/0278364915581863
283. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. (2018). Federated learning with non IID data. arXiv. https://doi.org/10.48550/arXiv.1806.00582

284. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., … Sun, M. (2020). Graph neural networks: A review of methods and applications. AI Open, 1, 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

285. Zhu, C., Dastani, M., & Wang, S. (2024). A survey of multi agent deep reinforcement learning with communication. Autonomous Agents and Multi Agent Systems, 38, 4. https://doi.org/10.1007/s10458-02309633-6