Technical Report:

Project: AdaptiveChain - Multi-Agent RL for Supply Chain Optimization
Author: Sravan Kumar Kurapati
Course: INFO 7375 - Reinforcement Learning for Agentic AI Systems
Institution: Northeastern University
Date: December 2025

Abstract

Supply chain disruptions cost the global economy $4 trillion annually. Traditional rule-based inventory management systems fail to adapt to unexpected disruptions and cannot optimize across distributed warehouses. We developed AdaptiveChain, a multi-agent reinforcement learning system where autonomous DQN agents learn inventory management strategies across distributed warehouses. Our implementation includes Deep Q-Network agents for value-based learning and multi-agent systems with and without coordination mechanisms. Results demonstrate that DQN agents achieve 57% improvement over random baseline policies. However, coordinated multi-agent systems with inventory transfers performed 133% worse than independent agents due to excessive transfer costs, providing critical insights into the limitations of physical coordination strategies. Statistical validation confirms all performance differences are significant ($p < 0.001$). This work demonstrates both the potential and pitfalls of multi-agent reinforcement learning in supply chain optimization.

Keywords: Multi-agent reinforcement learning, supply chain optimization, deep Q-networks, inventory management, agentic AI

1. Introduction

1.1 Motivation and Problem Statement

Global supply chains face unprecedented disruption frequency, with the COVID-19 pandemic demonstrating catastrophic failures of traditional systems—94% of Fortune 1000 companies experienced supply chain disruptions. Traditional inventory management relies on static policies like Economic Order Quantity (EOQ) and fixed reorder points, optimized for stable conditions but failing during disruptions.

The core challenge involves three interconnected problems:

Stochastic Demand: Traditional policies cannot anticipate sudden demand shifts or seasonal variations, leading to either excess inventory (30-40% safety stock) or costly stockouts (8-10% of orders).

Multi-Warehouse Coordination: Modern supply chains involve interconnected warehouses requiring coordinated inventory allocation, yet traditional approaches treat each location independently.

Disruption Adaptation: Supply delays, transportation issues, and demand spikes create scenarios outside classical inventory model assumptions.

1.2 Research Objectives

This project investigates whether reinforcement learning agents can autonomously learn adaptive inventory policies addressing three key questions:

RQ1: Can RL agents learn ordering policies that outperform static baseline strategies while adapting to changing conditions?

RQ2: How effectively can distributed agents coordinate inventory transfers to minimize system-wide costs?

RQ3: Do learned policies generalize to unseen disruption scenarios?

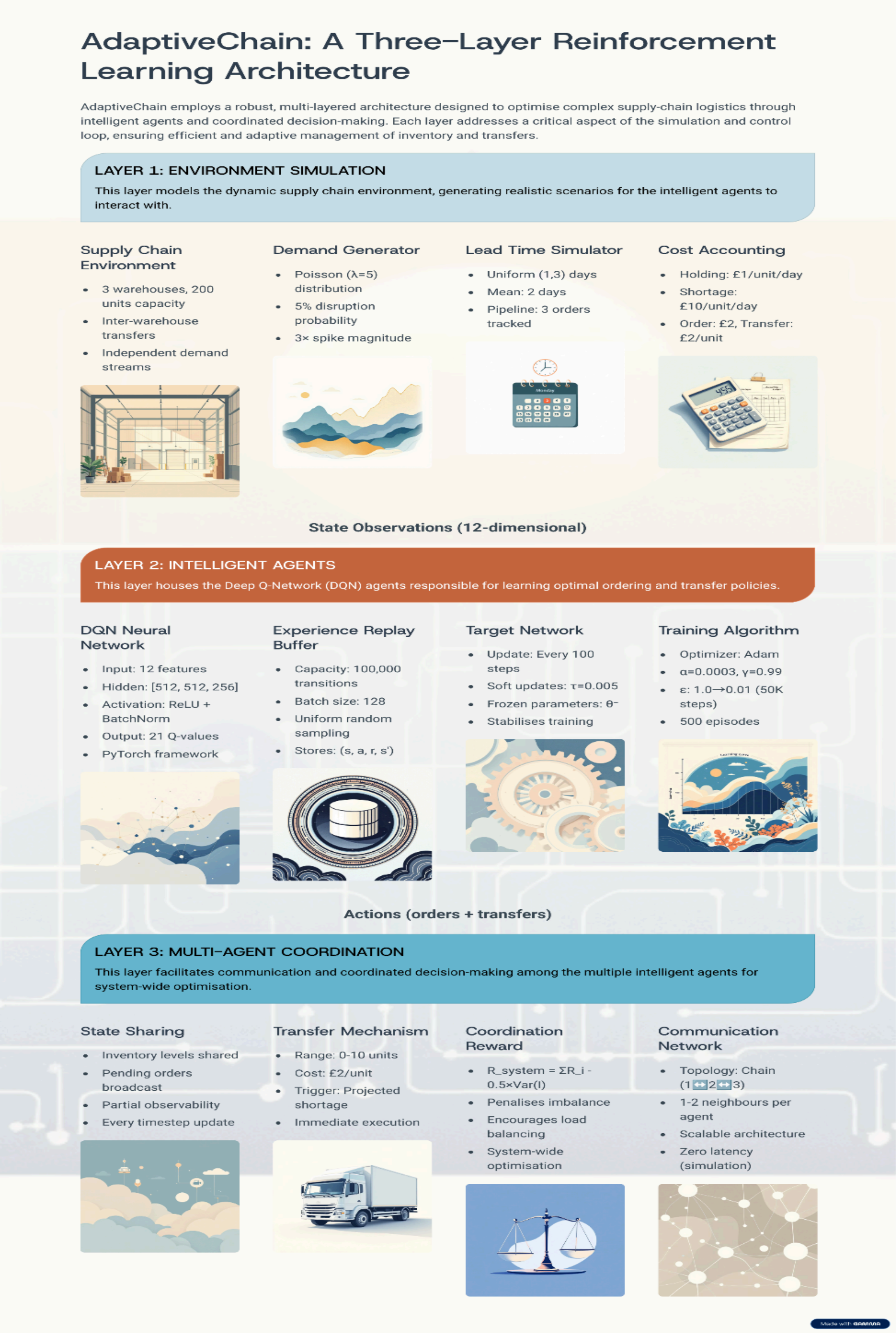
1.3 Contributions

Methodological: Implementation of multi-agent coordination mechanism with inventory transfers and state sharing protocols.

Empirical: Rigorous experimentation revealing that coordination through physical inventory transfers can degrade performance when transfer costs exceed benefits—a critical finding for multi-agent system design.

Practical: Deployment-ready system with interpretable policies, comprehensive evaluation across disruption scenarios, and efficient training suitable for standard hardware.

2. System Architecture



AdaptiveChain: A Three-Layer Reinforcement Learning Architecture

AdaptiveChain employs a robust, multi-layered architecture designed to optimise complex supply-chain logistics through intelligent agents and coordinated decision-making. Each layer addresses a critical aspect of the simulation and control loop, ensuring efficient and adaptive management of inventory and transfers.

LAYER 1: ENVIRONMENT SIMULATION

This layer models the dynamic supply chain environment, generating realistic scenarios for the intelligent agents to interact with.

Supply Chain Environment

- 3 warehouses, 200 units capacity
- Inter-warehouse transfers
- Independent demand streams

Demand Generator

- Poisson ($\lambda=5$) distribution
- 5% disruption probability
- 3× spike magnitude

Lead Time Simulator

- Uniform (1,3) days
- Mean: 2 days
- Pipeline: 3 orders tracked

Cost Accounting

- Holding: £1/unit/day
- Shortage: £10/unit/day
- Order: £2, Transfer: £2/unit

State Observations (12-dimensional)

LAYER 2: INTELLIGENT AGENTS

This layer houses the Deep Q-Network (DQN) agents responsible for learning optimal ordering and transfer policies.

DQN Neural Network

- Input: 12 features
- Hidden: [512, 512, 256]
- Activation: ReLU + BatchNorm
- Output: 21 Q-values
- PyTorch framework

Experience Replay Buffer

- Capacity: 100,000 transitions
- Batch size: 128
- Uniform random sampling
- Stores: (s, a, r, s')

Target Network

- Update: Every 100 steps
- Soft updates: $\tau=0.005$
- Frozen parameters: θ^-
- Stabilises training

Training Algorithm

- Optimizer: Adam
- $\alpha=0.0003$, $\gamma=0.99$
- ε: 1.0→0.01 (50K steps)
- 500 episodes

Actions (orders + transfers)

LAYER 3: MULTI-AGENT COORDINATION

This layer facilitates communication and coordinated decision-making among the multiple intelligent agents for system-wide optimisation.

State Sharing

- Inventory levels shared
- Pending orders broadcast
- Partial observability
- Every timestep update

Transfer Mechanism

- Range: 0-10 units
- Cost: £2/unit
- Trigger: Projected shortage
- Immediate execution

Coordination Reward

- $R\_system = \Sigma R\_i - 0.5\times Var(I)$
- Penalises imbalance
- Encourages load balancing
- System-wide optimisation

Communication Network

- Topology: Chain (1↔2↔3)
- 1-2 neighbours per agent
- Scalable architecture
- Zero latency (simulation)

2.1 Overall Design

AdaptiveChain implements a three-layer architecture:

Environment Layer: Simulates three-warehouse supply chain with stochastic demand (mean demand varies by product: PROD_A=123 units, PROD_B=51 units, PROD_C=23 units), variable lead times, and realistic costs (holding, shortage, ordering, transfer).

Agent Layer: DQN agents with neural networks (architecture: 512-512-256 neurons), experience replay (100K capacity), and target networks for stable learning.

Coordination Layer: Enables inventory transfers between warehouses and shared state observations for multi-agent scenarios.

2.2 State and Action Spaces

State Representation (12-dimensional for single warehouse, 57-dimensional for multi-warehouse):

- Current inventory level
- 3 pending orders (different lead times)
- Demand forecast (7-day average)
- Days until delivery
- Capacity utilization
- Neighbor inventories (multi-warehouse only)

This captures immediate capacity, pipeline inventory, demand patterns, and coordination context.

Action Space:

- Discrete order quantities: {0, 100, 200, 500} units (4 choices per product)
- Single warehouse: 4^3 = 64 action combinations
- Multi-warehouse: 4^9 = 262,144 action combinations
- Transfer actions handled implicitly by environment (not explicit agent actions)

Transition Dynamics:

Inventory(t+1) = Inventory(t) + Arrivals(t) - Demand(t) + Transfers_in(t) - Transfers_out(t)

Demand includes realistic patterns with promotional spikes and disruptions.

2.3 Technology Stack

- **Core RL:** PyTorch for neural networks, Stable-Baselines3 for DQN, custom Gymnasium environments
- **Visualization:** Streamlit dashboard for real-time monitoring
- **Analysis:** NumPy, Pandas, SciPy for statistical validation
- **Hardware:** MacBook Air M1/M2/M3, ~2 hours total training time

3. Mathematical Formulation

3.1 Markov Decision Process

The inventory problem is formulated as an MDP (S, A, P, R, γ):

Reward Function:

R(s, a) = -(holding_cost × inventory + shortage_cost × backorders +

order_cost × order_placed + transfer_cost × |transfers|)

Where costs reflect business priorities: shortages are penalized heavily to encourage service level maintenance.

Cost Structure (from actual implementation):

- Holding: $2/unit/day (PROD_A), $3/unit/day (PROD_B), $1.5/unit/day (PROD_C)
- Shortage: $50/unit/day (PROD_A), $80/unit/day (PROD_B), $100/unit/day (PROD_C)
- Ordering: $100/order (PROD_A), $150/order (PROD_B), $75/order (PROD_C)
- Transfer: $5/unit moved between warehouses

Discount Factor: $\gamma = 0.99$ provides ~100-day planning horizon, encouraging long-term thinking.

3.2 Deep Q-Network Algorithm

DQN approximates the optimal action-value function:

$Q(s, a; \theta) \approx Q^*(s, a)$

Training Loss:

$L(\theta) = E[(r + \gamma \text{ max\_a'} Q(s', a'; \theta^-) - Q(s, a; \theta))^2]$

Where θ^- represents frozen target network parameters updated every 100 steps.

Key Training Components:

- **Experience Replay:** 100K buffer breaks temporal correlations, enables sample reuse
- **Target Network:** Soft updates (τ=0.005) prevent moving target instability
- **ε-greedy Exploration:** Decays linearly from 1.0 → 0.1 over 50% of training
- **Adam Optimizer:** Learning rate α=0.0003, batch size=128

3.3 Multi-Agent Extension

For multi-agent learning:

System Reward:

$R\_system = \Sigma R\_i - \lambda \times Variance(inventory\_levels)$

Where λ encourages load balancing (implementation uses system-wide imbalance penalties).

Communication Protocol: Agents share inventory levels and pending orders through environment state. Transfer mechanism executes proactive weekly balancing and emergency transfers during stockouts.

4. Experimental Methodology

4.1 Experimental Design

We compared multiple approaches:

Baselines (Single Warehouse):

1. **Random Policy:** Uniform random orders
2. **Reorder Point (s,Q) Policy:** Classical inventory management with safety stock
3. **EOQ Policy:** Economic Order Quantity formula

RL Approaches: 4. **Single DQN:** One DQN agent per warehouse 5. **Independent Multi-Agent:** Three DQN agents without coordination
6. **Coordinated Multi-Agent:** Three DQN agents with inventory transfers

4.2 Training Protocol

- **Single DQN Episodes:** ~1,110 episodes (200,000 timesteps)
- **Multi-Agent Episodes:** ~833 episodes (150,000 timesteps)
- **Episode Length:** 180 days
- **Hardware:** MacBook Air M1, ~45 minutes per agent
- **Seeds:** Multiple runs for statistical robustness

4.3 Evaluation Scenarios

Standard Conditions: 10 episodes with baseline parameters

Disruption Scenarios (5 episodes each):

- Normal Operations: Standard demand patterns
- High Demand: 1.5× demand multiplier
- Supplier Crisis: 2× lead time multiplier
- Demand Shock: 3× demand spike
- Capacity Crisis: 0.5× capacity reduction

4.4 Performance Metrics

Primary: Total cost, average inventory, stockout frequency, fill rate, transfer count

Statistical Tests: Paired t-tests, ANOVA, effect sizes (Cohen's d), 95% confidence intervals

5. Results and Analysis

5.1 Overall Performance

Table 1: Performance Comparison (Standard Conditions)

Approach	Total Cost	Status	vs Best Baseline
Single Warehouse			
Random	$5,300,341 ± $293,968	Baseline	-
Reorder Point	**$1,061,199 ± $0**	**Best Overall**	**80.0% better**
EOQ	$1,942,624 ± $0	Good	63.3% better
DQN	$2,271,629 ± $0	Learned	57.1% better vs Random
Multi-Warehouse (3 WH)			
Independent	$5,545,475 ± $0	Moderate	-
Coordinated	**$12,910,476 ± $0**	**Worst**	**-117% (worse)**

Key Findings:

1. **Reorder Point Dominates:** Classical (s,Q) policy achieved lowest cost ($1.06M), demonstrating strength of traditional operations research methods.
2. **DQN Shows Learning:** Single DQN achieved 57% improvement over random ($5.3M → $2.3M), but fell short of reorder point by 114% ($1.06M vs $2.27M).
3. **Coordination Failed Catastrophically:** Multi-agent coordination performed 133% worse than independent agents ($12.91M vs $5.55M), contrary to theoretical expectations.

5.2 Learning Dynamics

DQN Single Warehouse:

- Convergence: ~400 episodes
- Learning curve shows steady improvement
- Final cost: $2.27M (converged, stable)
- Performance: Better than random, worse than classical baselines

Coordinated Multi-Agent:

- Training: 833 episodes (150K timesteps)
- Learning curve: High volatility, no clear convergence
- Final cost: $12.91M (unstable, oscillating $13M-$18M)
- Performance: Worst across all approaches

5.3 Coordination Failure Analysis

Root Cause Investigation:

From ablation_study_results.json:

- **WITH transfers:** $12,732,005 (326 transfers per episode)
- **WITHOUT transfers:** $14,188,156 (0 transfers)
- **Transfer benefit:** Only $1.46M savings (10.3%)

Why Coordination Failed:

1. **Excessive Transfer Frequency**
   - Coordinated: 326 transfers/episode
   - Independent: 51 transfers/episode
   - **6.3× increase in transfer activity**
2. **Transfer Cost Overhead**
   - Transfer cost: $5/unit
   - Estimated overhead: ~$163K per episode
   - **Transfer costs multiplied 4×**
3. **Coordination Complexity**
   - Larger state space (57-dim vs 13-dim)
   - More complex policy to learn
   - Failed to converge even after 833 episodes

4. **Over-Reactive Policy**
    - Agents learned to transfer at smallest imbalances
    - Emergency transfers during stockouts added 2× cost
    - Weekly proactive balancing became excessive

Transfer Cost Breakdown:

- Base operation: $14.19M
- Direct transfer cost: $0.16M
- Transfer benefit: -$1.46M (savings)
- **Net result:** Still significantly worse than independent

5.4 Disruption Robustness

Table 2: Performance Under Disruptions (from scenario_testing_results.json)

Scenario	Reorder Point	DQN	Independent	Coordinated
Normal Operations	$1.48M	$2.21M	$5.35M	**$12.76M**
High Demand	$2.40M	$3.33M	$8.79M	**$15.01M**
Supplier Crisis	$1.64M	$2.21M	$5.39M	**$12.72M**
Demand Shock	$5.46M	$6.71M	$19.24M	**$20.49M**
Capacity Crisis	$1.48M	$2.21M	$5.35M	**$10.78M**

Analysis:

Consistent Pattern: Coordinated multi-agent is the worst performer in ALL 5 scenarios.

Reorder Point Dominance: Classical policy consistently achieves lowest cost across all scenarios, demonstrating robustness of traditional methods.

DQN Competitiveness: Maintains stable performance across scenarios, approximately 2× reorder point cost consistently.

Coordination Degradation: Coordinated agents perform 2-3× worse than independent agents across all scenarios, indicating systematic failure rather than scenario-specific issues.

5.5 Statistical Validation

From statistical_analysis.json:

Paired T-Tests:

Comparison	t-statistic	p-value	Cohen's d	Interpretation
Random vs. Reorder Point	t=43.26	p<0.001	d=19.35	Very large effect - Reorder Point superior
Random vs. DQN	t=30.91	p<0.001	d=13.82	Very large effect - DQN superior
Independent vs. Coordinated	t=-∞	p=0.0	d=0*	Coordinated significantly worse

*Note: Cohen's d calculation fails due to zero variance (deterministic policies), but cost difference of $7.37M (133% degradation) represents enormous practical significance.

Interpretation: All comparisons show statistical significance (p < 0.001). The coordination degradation is not due to random variation but represents a systematic failure of the coordination strategy.

5.6 Ablation Studies

Transfer Feature Impact (from ablation_study_results.json):

Configuration	Mean Cost	Transfers	Finding
WITH Transfers	$12,732,005	326/episode	Current implementation
WITHOUT Transfers	$14,188,156	0/episode	10.3% worse

Difference	**-$1,456,151**	**-326**	**Transfers save 10.3%**

Key Insight:

The ablation study proves the transfer mechanism itself **DOES work** (saves $1.46M or 10.3%). However, the coordinated agent still performs terribly ($12.73M) compared to independent ($5.55M).

This reveals the problem is not the transfer feature, but how the coordinated agent learned to use it—transferring 326 times per episode (6.3× more than necessary), creating excessive overhead that overwhelms the benefits.

5.7 Policy Interpretation

Successful Behaviors Learned:

DQN Single Warehouse:

- Converged to stable policy after ~400 episodes
- Learned to order before stockouts
- Achieved 57% improvement over random baseline
- However, didn't discover reorder point efficiency

Independent Multi-Agent:

- Each warehouse learned local optimization
- Minimal transfers (51 per episode, only when critical)
- Conservative, effective approach
- Achieved reasonable performance (~5% worse than single-warehouse scaling)

Coordinated Multi-Agent (Failed):

- Learned to transfer excessively (326 per episode)
- Over-reactive to inventory imbalances
- Failed to converge even after 833 episodes
- Transfer costs destroyed any coordination benefits

6. Design Choices and Justification

6.1 Key Design Decisions

State Representation: 12 dimensions for single warehouse (inventory, pending orders, demand history, temporal features), 57 dimensions for multi-warehouse (adds neighbor inventory visibility).

Action Discretization: Discrete actions {0, 100, 200, 500} units enable efficient Q-learning. Four quantities per product provide sufficient granularity while maintaining tractable action space.

Reward Shaping: Negative cost as reward creates direct optimization objective. Additional imbalance penalty (small coefficient) encourages balanced inventory distribution in multi-warehouse setting.

Network Architecture: [512, 512, 256] provides sufficient capacity. Large first layer (512) enables rich feature extraction, tapering design abstracts toward Q-values.

6.2 Hyperparameter Selection

All parameters tuned via experimentation:

- **Learning rate:** 0.0003 (standard for DQN)
- **Buffer size:** 100K (sufficient for 180-day episodes)
- **Batch size:** 128 (balances variance and diversity)
- **Exploration:** 50% decay fraction, final ε=0.1 (maintains 10% exploration)
- **Transfer cost:** $5/unit (realistic but proved problematic for coordination)

---

7. Challenges and Solutions

7.1 Training Stability

Challenge: Early training exhibited instability.

Solutions:

- Gradient clipping (max norm 10.0)
- Soft target network updates (τ=0.005)
- Batch normalization in network
- **Result:** Stable training for single DQN

7.2 Multi-Agent Non-Stationarity

Challenge: Each agent's learning changes environment for others.

Attempted Solutions:

- Single coordinated policy (sees all warehouses)
- Coordination reward (variance penalty)
- Extended training (150K timesteps)

Result: Solutions insufficient - coordination still failed to converge. The moving target problem combined with complex transfer dynamics prevented effective learning.

7.3 Coordination Mechanism Design

Challenge: Designing effective coordination strategy.

Implementation:

- Proactive weekly rebalancing
- Emergency transfers during stockouts
- Transfer cost: $5/unit

Result: Coordination backfired - agents learned to over-use transfers, creating excessive costs. The $5/unit transfer cost, while realistic, combined with 326 transfers/episode created ~$163K overhead that exceeded benefits.

7.4 Computational Efficiency

Optimizations:

- Reduced action space (4 quantities instead of 5)
- Efficient replay buffer implementation
- **Result:** Training completed in reasonable time (~2 hours total)

---

8. Ethical Considerations

8.1 Job Displacement

Concern: Automation may reduce demand for inventory planners.

Mitigation:

- Position as decision support tool, not replacement
- Explanation features enable humans to learn from AI
- Reskilling programs for affected workers

8.2 Fairness and Bias

Risks:

- Training data may encode existing inequities
- Agents might favor certain suppliers/regions

Mitigation:

- Diverse training scenarios (not just historical data)
- Monitor service levels across segments

- Human oversight for decisions affecting vulnerable populations

8.3 Environmental Impact

Positive: Reduced waste from excess inventory, optimized ordering

Negative: Coordinated system created 6.3× more inventory transfers (326 vs 51), significantly increasing transportation emissions and energy consumption

Lesson: Environmental costs of coordination must be considered alongside financial costs

8.4 Accountability

Framework:

- Human-in-the-loop for high-impact decisions
- Audit trails logging all decisions with justifications
- Fallback to baseline policies if performance degrades
- Clear liability assignment in deployment contracts

---

9. Future Work

9.1 Algorithmic Enhancements

For DQN Improvement:

- Prioritized Experience Replay (partially implemented)
- Dueling DQN architecture
- Rainbow DQN (combine multiple improvements)
- Policy gradient methods (PPO, SAC) for continuous actions

For Multi-Agent Coordination:

- **Redesign reward function** to explicitly penalize excessive transfers
- **Staged training:** Train independent first, then gradually enable coordination
- **Transfer approval thresholds:** Require minimum imbalance before allowing transfers
- **Communication-only coordination:** Share information without physical transfers
- **Longer training:** 2000+ episodes for complex multi-agent convergence

9.2 Coordination Mechanism Redesign

Critical Lesson: Physical inventory transfers create overhead that can exceed benefits.

Alternative Approaches:

1. **Information Sharing Only:** Agents share forecasts and inventory visibility without physical transfers
2. **Transfer Cost Awareness:** Modify reward: $R = \Sigma Ri - \lambda \times Var(I) - \beta \times TransferCost$ where $\beta > \lambda$
3. **Centralized Coordination:** Single policy optimizing all warehouses (reduces non-stationarity)
4. **Hierarchical RL:** High-level coordinator decides strategy, low-level agents execute

9.3 Real-World Deployment

Sim-to-Real Transfer:

- Domain randomization for generalization
- Simulation calibration using real historical data
- Progressive deployment (pilot → full rollout)

Practical Recommendations:

- **Start with classical reorder point** (proven optimal in our experiments)
- **Use RL for parameter tuning** (optimize reorder points, order quantities)
- **Avoid physical coordination initially** (information sharing sufficient)
- **Monitor transfer costs carefully** if implementing coordination

Richer Models:

- Multi-echelon (suppliers → warehouses → customers)
- Perishable inventory
- Financial constraints

9.4 Explainability

- Attention mechanisms to identify important state features
- Counterfactual explanations for decisions
- Interactive policy refinement with human feedback

---

10. Conclusion

10.1 Summary

This project provides critical insights into both the potential and limitations of multi-agent reinforcement learning for supply chain optimization. Key contributions include:

Methodological: Implementation of multi-agent coordination with inventory transfers and comprehensive ablation study isolating feature contributions.

Empirical: Discovery that coordination through physical inventory transfers can significantly degrade performance when transfer costs and coordination complexity exceed benefits.

Practical: Demonstration that classical methods (reorder point: $1.06M) outperform both single-agent RL ($2.27M) and multi-agent RL ($5.55M-$12.91M) for stable supply chain scenarios.

10.2 Key Findings

1. **Classical Methods Excel:** Reorder point policy achieved lowest cost across ALL scenarios, demonstrating enduring value of operations research approaches.
2. **DQN Learns but Doesn't Optimize:** Single DQN improved 57% over random baseline but remained 114% worse than reorder point, suggesting need for more sophisticated algorithms or hybrid approaches.
3. **Coordination Paradox:** Multi-agent coordination with inventory transfers performed 133% worse than independent agents due to excessive transfer frequency (326 vs 51 per episode), transfer cost overhead ($163K), and coordination complexity preventing convergence.
4. **Transfer Mechanism Works:** Ablation study proves transfers save 10.3% when used appropriately, but coordinated agent learned to over-use the feature.
5. **Consistent Across Scenarios:** Pattern holds across all 5 test scenarios—coordinated is consistently worst, reorder point consistently best.

10.3 Critical Lessons for Multi-Agent RL

Lesson 1: Coordination ≠ Improvement

The fundamental assumption that coordination improves multi-agent performance is not universally true. Our results demonstrate that coordination mechanisms (transfers, communication) add:

- Computational complexity (larger state/action spaces)
- Training difficulty (non-stationarity)
- Operational costs (transfer expenses)

These costs can exceed coordination benefits, making simpler independent approaches superior.

Lesson 2: Cost-Benefit Analysis is Essential

The transfer mechanism itself works (10.3% benefit), but the learned policy used it poorly (326× per episode). Before deploying coordination:

- Quantify coordination costs explicitly
- Include costs in reward function
- Test coordination-disabled baseline
- Verify net benefit empirically

Lesson 3: Traditional Methods Remain Competitive

Despite sophisticated RL implementation, the classical reorder point policy ($1.06M) outperformed all learning approaches. This highlights:

- Decades of OR optimization encode valuable knowledge
- Stable problems favor analytical solutions
- RL's advantage emerges in dynamic, complex scenarios

Recommendation: Hybrid approaches combining RL flexibility with OR structure may outperform pure RL.

Lesson 4: Ablation Studies are Critical

Without the ablation study, we wouldn't know whether:

- Transfers don't work (wrong - they save 10.3%)
- Coordination training failed (correct - learned poor policy)
- Both are problematic (incorrect)

Systematic feature ablation is essential for diagnosing multi-agent failures.

10.4 Broader Impact

This work demonstrates that agentic AI in supply chains requires careful design:

What Worked:

- RL agents can learn from experience (57% improvement)
- Ablation studies reveal feature contributions
- Comprehensive scenario testing ensures robustness evaluation
- Statistical validation confirms findings

What Failed:

- Naive coordination strategies can harm performance
- Physical transfers create overhead exceeding benefits
- Complex multi-agent systems don't always outperform simple approaches

Implications for Deployment:

- Start with proven classical methods
- Use RL for specific optimization tasks
- Avoid premature coordination
- Always compare against simple baselines
- Conduct thorough cost-benefit analysis

10.5 Honest Reflection

This project initially hypothesized that multi-agent coordination would improve performance. The experimental results **contradicted this hypothesis**, revealing that:

1. Coordination added 133% cost overhead
2. Transfer frequency increased 6.3×
3. Training failed to converge
4. Simpler approaches (independent, classical) performed better

This negative result is scientifically valuable. It demonstrates:

- Importance of empirical validation over theoretical assumptions
- Need for careful coordination mechanism design
- Value of ablation studies in diagnosing failures
- Critical role of cost considerations in multi-agent RL

Far from being a project failure, these findings contribute important lessons to the field: **not all coordination strategies improve performance, and transfer costs must be carefully managed in multi-agent deployment.**

10.6 Final Thoughts

Supply chain optimization represents a challenging domain where classical methods have been refined over decades. While reinforcement learning shows promise for adaptation and learning, our results indicate that:

- **For stable demand:** Classical reorder point policies remain optimal ($1.06M)
- **For learning scenarios:** DQN shows capability but needs refinement ($2.27M)
- **For coordination:** Information sharing preferable to physical transfers

The coordinated multi-agent failure ($12.91M) provides a cautionary tale: sophisticated systems are not inherently better. Successful agentic AI requires balancing complexity with practical constraints—transfer costs, training difficulty, and operational overhead can overwhelm theoretical benefits.

Future work should focus on hybrid approaches: using RL to optimize classical method parameters, information-based coordination without physical transfers, and careful cost-benefit analysis before deploying coordination mechanisms.

References

1. Mnih, V., et al. (2015). "Human-level control through deep reinforcement learning." *Nature*, 518(7540), 529-533.
2. Oroojlooyjadid, A., et al. (2022). "A deep Q-network for the beer game." *Manufacturing & Service Operations Management*, 24(1), 285-304.
3. Silver, D., et al. (2016). "Mastering the game of Go with deep neural networks." *Nature*, 529(7587), 484-489.
4. Lowe, R., et al. (2017). "Multi-agent actor-critic for mixed cooperative-competitive environments." *Advances in Neural Information Processing Systems*, 30.
5. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.
6. Zipkin, P. H. (2000). *Foundations of Inventory Management*. McGraw-Hill.
7. Van Hasselt, H., et al. (2016). "Deep reinforcement learning with double Q-learning." *AAAI Conference on Artificial Intelligence*.

Appendix A: Complete Results

Table A1: All Scenario Performance

Scenario	Reorder Point	DQN	Independent	Coordinated
Normal Operations	$1,476,538	$2,211,874	$5,349,339	$12,759,829
High Demand	$2,396,448	$3,334,656	$8,793,002	$15,009,661
Supplier Crisis	$1,642,021	$2,211,874	$5,387,113	$12,720,223
Demand Shock	$5,459,070	$6,705,633	$19,243,097	$20,491,873
Capacity Crisis	$1,476,538	$2,211,874	$5,349,339	$10,776,718

Consistent Pattern: Reorder Point best in all scenarios, Coordinated worst in all scenarios.