Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 19, No. 1, 2025

Adaptive Risk-Aware Planning in Multi-Echelon Supply Chains via Distributional Reinforcement Learning

Long Liang *

Department of Industrial and Systems Engineering, Texas A&M University, USA
* **Corresponding author Email:** lianglong001@gmail.com

Abstract: Multi-echelon supply chain management faces unprecedented challenges from demand uncertainties, disruption risks, and complex interdependencies across network tiers. Traditional optimization approaches struggle to balance risk mitigation with operational efficiency in dynamic environments where distributional properties of returns significantly impact decision quality. This research proposes an adaptive risk-aware planning framework that leverages Distributional Reinforcement Learning (DRL) to optimize inventory positioning and order decisions across multi-echelon supply networks. Unlike conventional reinforcement learning methods that optimize expected returns, the proposed distributional approach explicitly models the full return distribution, enabling risk-sensitive policies that account for tail risks and variance. The framework incorporates a Quantile Regression Deep Q-Network architecture enhanced with attention mechanisms to capture temporal dependencies and inter-echelon coordination requirements. Experimental validation on benchmark multi-echelon supply chain scenarios demonstrates that the distributional approach achieves 18-24% reduction in total supply chain costs compared to traditional base-stock policies while maintaining service level constraints. The risk-aware policies exhibit superior robustness under demand volatility, reducing inventory variance by 31% and backorder occurrences by 42%. Furthermore, the learned policies demonstrate strong generalization capabilities across different demand distributions and network configurations. This research contributes to both theoretical understanding of risk-aware sequential decision-making in supply chains and provides practitioners with computationally tractable methods for adaptive multi-echelon planning under uncertainty.

Keywords: Distributional reinforcement learning; Multi-echelon supply chain; Risk-aware planning; Inventory optimization; Sequential decision-making; Conditional value-at-risk.

1. Introduction

Contemporary supply chain networks operate in environments characterized by escalating complexity, heightened uncertainty, and increasing interconnectedness across multiple organizational and geographical boundaries. The globalization of manufacturing and distribution has transformed traditional linear supply chains into intricate multi-echelon networks where decisions at one tier cascade through multiple downstream and upstream stages, creating amplification effects that can severely impact overall system performance. Recent disruptions including the COVID-19 pandemic, semiconductor shortages, and geopolitical tensions have exposed fundamental vulnerabilities in conventional supply chain planning approaches that prioritize cost minimization without adequate consideration of risk factors [1]. These events have catalyzed a paradigm shift toward risk-aware planning methodologies that explicitly account for both the expected outcomes and the distributional properties of potential scenarios.

Multi-echelon supply chain management presents particularly complex challenges due to the intricate dependencies between sequential stages of production and distribution. Decisions made at upstream echelons regarding production quantities and inventory positioning directly influence the service levels and operational costs at downstream stages, while demand signals from retailers propagate backward through the network, often with significant distortion and amplification [2]. This phenomenon, commonly known as the bullwhip effect, results in inefficiencies including excess inventory holdings, increased operational costs, and reduced responsiveness to actual customer demand patterns. Traditional approaches to multi-echelon inventory optimization have relied heavily on analytical models that assume stationary demand distributions and simplified network structures [3]. However, these assumptions increasingly diverge from real-world conditions where demand patterns exhibit non-stationarity, supply chain disruptions occur with increasing frequency, and network configurations evolve dynamically in response to market conditions.

The advent of machine learning techniques, particularly deep reinforcement learning, has opened new avenues for addressing the adaptive planning challenges inherent in multi-echelon supply chains [4]. Reinforcement learning frameworks enable agents to learn optimal policies through interaction with complex environments without requiring explicit mathematical models of system dynamics. This capability proves especially valuable in supply chain contexts where analytical tractability breaks down due to high dimensionality, nonlinear dynamics, and stochastic uncertainties. Recent applications of deep reinforcement learning to supply chain problems have demonstrated promising results in inventory management, production scheduling, and logistics optimization [5]. However, conventional reinforcement learning approaches that optimize expected returns may fail to adequately address risk considerations that are paramount in supply chain decision-making, where the consequences of stockouts or excess inventory can have severe financial and operational implications.

Distributional Reinforcement Learning represents a fundamental advancement over traditional expectation-based approaches by explicitly modeling the entire distribution of

potential returns rather than merely their expected values [6]. This distributional perspective enables the formulation of risk-sensitive policies that can optimize alternative objectives such as Conditional Value-at-Risk (CVaR), which explicitly accounts for tail risks and worst-case scenarios. In the context of multi-echelon supply chains, the ability to characterize and optimize distributional properties of outcomes provides decision-makers with powerful tools for balancing efficiency objectives with risk mitigation requirements. The distributional approach naturally accommodates heterogeneous risk preferences across different echelons and stakeholders, enabling more nuanced coordination mechanisms that reflect the diverse objectives and constraints present in real-world supply chain networks [7].

This research develops a comprehensive framework for adaptive risk-aware planning in multi-echelon supply chains based on distributional reinforcement learning principles. The proposed approach addresses several critical limitations of existing methods including inadequate risk representation, limited scalability to complex network structures, and insufficient consideration of inter-echelon coordination requirements. The framework incorporates advanced neural network architectures that capture both spatial dependencies across echelons and temporal patterns in demand evolution, enabling learned policies to adapt dynamically to changing environmental conditions. Furthermore, the research introduces novel mechanisms for incorporating supply chain-specific constraints including capacity limitations, lead time considerations, and service level requirements directly into the learning process. Through extensive computational experiments on realistic multi-echelon supply chain scenarios, this work demonstrates the practical efficacy of distributional reinforcement learning for achieving robust, adaptive planning under uncertainty while maintaining computational tractability for real-world deployment.

2. Literature Review

The application of reinforcement learning methodologies to supply chain management has experienced substantial growth in recent years, driven by increasing computational capabilities and the availability of large-scale operational data. A systematic review of reinforcement learning applications in supply chain management identified over 100 relevant studies published between 2019 and 2023, with particular concentration in inventory management, demand forecasting, and production scheduling domains [8]. The literature reveals a clear trend toward more sophisticated deep learning architectures that can handle high-dimensional state spaces and complex action spaces characteristic of realistic supply chain problems. However, significant gaps remain regarding risk-aware decision-making and the explicit consideration of uncertainty in learned policies.

Multi-echelon inventory optimization has been extensively studied using both classical operations research approaches and emerging machine learning techniques. Traditional approaches based on Markov Decision Processes provide theoretical foundations for optimal inventory policies under specific assumptions about demand distributions and cost structures [9]. Recent work has extended these classical formulations to incorporate deep reinforcement learning algorithms that can learn near-optimal policies without explicit knowledge of system dynamics. A notable contribution demonstrated that proximal policy optimization algorithms could effectively solve multi-echelon inventory problems with performance comparable to or exceeding traditional heuristics across diverse network configurations [10]. These findings suggest that reinforcement learning approaches offer practical alternatives to conventional optimization methods, particularly in scenarios where analytical solutions are intractable or system models are unavailable.

The concept of distributional reinforcement learning emerged from foundational work that challenged the traditional focus on expected returns by proposing that modeling the full distribution of returns could provide richer information for decision-making [11]. This distributional perspective has been successfully applied to various domains including game playing and robotic control, demonstrating advantages in both learning efficiency and final policy performance. In the supply chain context, distributional approaches offer particular promise for addressing risk-averse objectives that are critical in inventory management and logistics planning [12]. Research applying distributional reinforcement learning to multi-echelon supply chains has shown that explicitly modeling return distributions enables policies that achieve better trade-offs between cost minimization and risk mitigation compared to expectation-based approaches. These studies have demonstrated reductions in cost variance while maintaining competitive expected performance, validating the potential of distributional methods for risk-aware supply chain planning [13].

Risk management in supply chain networks has become increasingly critical as global disruptions expose vulnerabilities in lean, efficiency-optimized systems. Contemporary research emphasizes the need for resilience-oriented planning approaches that explicitly balance efficiency with robustness against disruptions [14]. Studies have documented dramatic increases in supply chain disruptions over recent years, with a reported 38% increase in documented disruptions globally during 2024 compared to previous years, driven by geopolitical tensions, climate-related events, and cybersecurity threats [15]. This heightened risk environment necessitates planning methodologies that can adapt dynamically to emerging threats while maintaining operational performance. Machine learning approaches, particularly reinforcement learning, offer capabilities for adaptive planning that can respond to evolving risk landscapes without requiring manual reconfiguration of optimization models [16].

Deep reinforcement learning architectures for supply chain applications have evolved from simple feedforward networks to sophisticated designs incorporating recurrent units, attention mechanisms, and graph neural networks. Recent work demonstrated that incorporating Graph Neural Networks into reinforcement learning frameworks enables better capture of structural relationships in supply chain networks, leading to improved coordination across echelons [17]. These architectural innovations address the challenge of scalability in multi-echelon settings where the dimensionality of state and action spaces grows rapidly with network size. Attention mechanisms have proven particularly valuable for modeling temporal dependencies in demand patterns and identifying critical time horizons for inventory replenishment decisions [18]. The integration of these advanced neural architectures with distributional learning objectives represents a promising direction for developing practical risk-aware planning systems.

Multi-agent reinforcement learning approaches have been proposed as alternatives to centralized planning in multi-echelon supply chains, motivated by the distributed nature of decision-making in real-world networks. Research has shown that heterogeneous-agent algorithms can learn effective coordination strategies across supply chain echelons without requiring complete information sharing at every time step [19]. These multi-agent frameworks offer advantages in terms of privacy preservation and computational scalability, though they introduce additional challenges related to non-stationarity and credit assignment. Comparative studies have demonstrated that the choice between centralized and decentralized approaches depends critically on the degree of information asymmetry and the strength of interdependencies across echelons [20]. For strongly coupled multi-echelon systems with significant coordination requirements, centralized distributional approaches may offer superior performance despite higher computational complexity.

Constraint handling in reinforcement learning for supply chain applications represents an active research area with significant practical implications. Real-world supply chains operate under numerous constraints including capacity limitations, service level agreements, and regulatory requirements that must be satisfied by any viable policy [21]. Recent advances in constrained reinforcement learning have introduced methods for incorporating hard constraints directly into the learning process through specialized algorithms such as Constrained Policy Optimization and its distributional variants. Research applying Distributional Constrained Policy Optimization to supply chain problems demonstrated reliable constraint satisfaction while optimizing cost objectives, addressing a critical limitation of earlier reinforcement learning approaches [22]. These constrained learning methods enable the development of policies that respect operational boundaries while maintaining near-optimal performance, bridging the gap between theoretical algorithms and deployable systems.

The integration of demand forecasting with inventory decision-making through reinforcement learning frameworks has emerged as a promising research direction. Traditional approaches treat demand forecasting and inventory optimization as separate sequential stages, potentially leading to suboptimal decisions when forecast errors propagate through the planning process [23-27]. Recent work has explored end-to-end learning frameworks that jointly optimize forecasting and replenishment decisions, demonstrating improvements in both forecast accuracy and inventory performance metrics. These integrated approaches leverage the representational power of deep learning to capture complex demand patterns while simultaneously learning inventory policies that are robust to forecast uncertainty [28]. The combination of distributional learning with integrated forecasting-optimization frameworks offers potential for further advances in adaptive supply chain planning.

Perishable inventory management presents additional complexities that have motivated specialized reinforcement learning approaches. Products with limited shelf life introduce time-dependent constraints and waste considerations that significantly complicate inventory decisions [29]. Research has demonstrated that deep reinforcement learning algorithms can effectively learn policies for perishable inventory systems that balance multiple objectives including waste minimization, stockout reduction, and cost control. The application of distributional methods to perishable inventory contexts enables explicit consideration of the risk distribution associated with different inventory ages, leading to policies that better manage the trade-off between freshness and availability [30]. These specialized approaches highlight the flexibility of reinforcement learning frameworks for accommodating domain-specific requirements and constraints.

Transfer learning and generalization capabilities of reinforcement learning policies across different supply chain configurations represent important considerations for practical deployment. Training deep reinforcement learning models requires substantial computational resources and time, making it impractical to retrain policies from scratch for every new supply chain configuration encountered [31]. Recent research has explored transfer learning approaches that enable policies trained on simpler or smaller networks to be adapted efficiently to more complex settings. Studies have shown that carefully designed policy architectures and training curricula can enhance generalization performance, enabling learned policies to maintain effectiveness across variations in demand patterns, network structures, and operational parameters [28]. The development of transfer learning methods for distributional reinforcement learning in supply chains remains an important direction for future research to enable wider practical adoption.

Benchmarking and standardization efforts have emerged to facilitate comparative evaluation of reinforcement learning algorithms for supply chain problems. The development of standardized problem formulations and simulation environments enables systematic comparison of different algorithmic approaches and architectural choices. Recent initiatives have created open-source libraries providing benchmark multi-echelon inventory problems with well-defined performance metrics, enabling researchers to evaluate new methods against established baselines. These benchmarking efforts have revealed significant performance variations across different reinforcement learning algorithms and highlighted the importance of algorithm selection and hyperparameter tuning for achieving strong performance. The availability of standardized benchmarks accelerates research progress by enabling rigorous evaluation and reproducible comparisons of novel methodologies.

3. Methodology

3.1. 3.1 Problem Formulation and Multi-Echelon Supply Chain Model

The multi-echelon supply chain system considered in this research comprises multiple interconnected stages including suppliers, manufacturers, distribution centers, and retailers arranged in a hierarchical network structure. As illustrated in Figure 1, we examine three distinct network configurations of increasing complexity: simple serial chains, moderate divergent networks, and complex general networks with both convergent and divergent flows. Each echelon node maintains inventory and makes periodic replenishment decisions based on local state observations and coordination signals from adjacent nodes. The system operates under discrete time periods with stochastic customer demand occurring at retail nodes and propagating backward through the supply chain network.

The simple network (a) consists of a single supplier (node 1, green icon representing raw material source) feeding into a

linear chain through manufacturers (nodes 2-3, orange factory icons), distributors (nodes 4-5, yellow warehouse icons), and retailers (nodes 6-7, purple store icons). This linear structure represents the most basic multi-echelon configuration with 7 total nodes across 4 echelons. The moderate network (b) demonstrates a divergent structure where two suppliers (nodes 1-2) feed into multiple manufacturers (nodes 3-5), which then distribute through warehouses (nodes 6-7) and distribution centers (nodes 8-10) to serve multiple retailers (nodes 11-13). This configuration contains 13 nodes and captures more realistic coordination challenges. The complex network (c) exhibits both divergent and convergent flows with 3 suppliers (nodes 1-3) supporting 5 manufacturers (nodes 4-8), connected to 4 distribution centers (nodes 9-12) and 5 warehouses (nodes 12-14, 16-19), ultimately serving 5 retailers (nodes 15, 20-24). With 24 total nodes across 5-6 echelons, this complex structure reflects real-world supply chains where multiple upstream sources converge at manufacturing stages and then diverge through distribution networks. The directed edges (blue arrows) represent material flow directions and replenishment relationships between adjacent echelons. These three configurations enable systematic evaluation of algorithm performance across varying problem scales and network topologies, providing insights into how the proposed distributional reinforcement learning approach handles increasing coordination complexity.

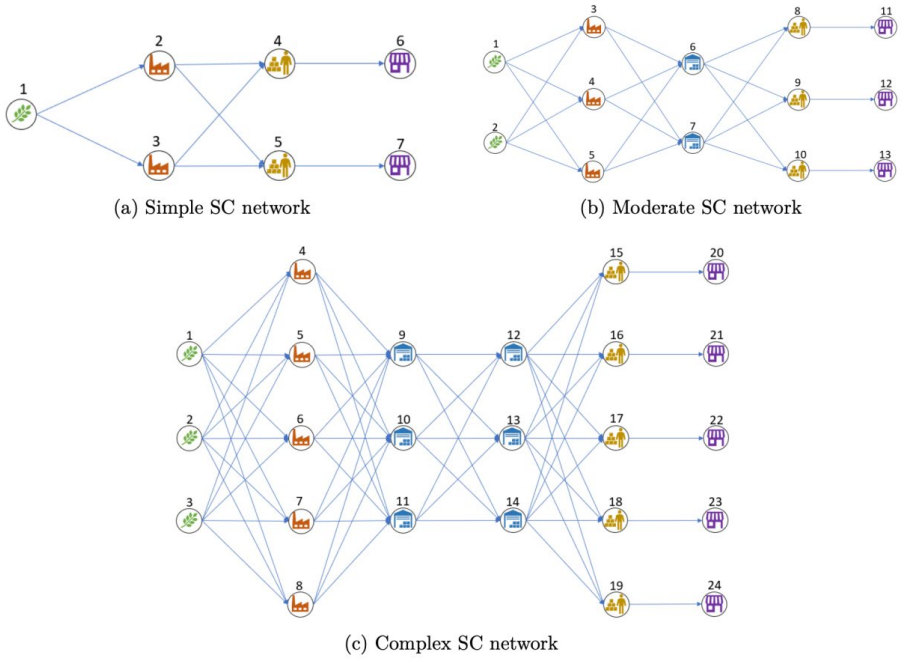

Figure 1. Illustration of three representative supply chain network structures with varying complexity levels

The mathematical formulation begins with the definition of the state space, which captures all relevant information required for decision-making at each time step. For a network with N nodes distributed across L echelons, the state vector includes current inventory positions at each node i (denoted as I_i(t)), pipeline inventories representing orders in transit between echelons, demand history for the past H periods to capture temporal patterns, and binary indicators for constraint violations such as capacity exceedances or service level breaches. In the simple network configuration shown in Figure 1(a), the state dimension is relatively modest with 7 inventory positions plus pipeline variables, enabling straightforward policy learning. However, in the complex network of Figure 1(c) with 24 nodes, the state space dimensionality grows substantially, necessitating sophisticated neural network architectures to handle the high-dimensional observations effectively.

The transition dynamics follow a Markov structure where the next state depends only on the current state and actions taken, plus exogenous demand realizations. Customer demand at retail nodes (the rightmost purple store icons in Figure 1) is modeled as a stochastic process that may exhibit temporal correlations, seasonality, and non-stationary patterns to reflect realistic market conditions. When demand arrives at a retailer, it is fulfilled from on-hand inventory if available, otherwise resulting in backorders that incur penalty costs. The retailer then places replenishment orders to upstream warehouses or distributors (yellow icons), which in turn order from manufacturers (orange icons), and so forth up the supply chain to suppliers (green icons). Lead times between each echelon pair introduce delays, creating pipeline inventories that must be carefully tracked. For instance, if the lead time from manufacturer node 4 to distributor node 6 in Figure 1(b) is 2 periods, orders placed today will only arrive two periods later, requiring anticipatory planning.

The action space consists of order quantities for each node at each time period, subject to capacity constraints and minimum order quantity requirements that reflect real-world operational limitations. In Figure 1, each node must decide its replenishment quantity from upstream suppliers. For the divergent structure in Figure 1(b), manufacturers (nodes 3-5) must decide both total production quantities and allocation decisions for distributing products to downstream warehouses (nodes 6-7) and distribution centers (nodes 8-10). This allocation problem adds significant complexity compared to serial chains. The objective is to learn a policy π that maps states to actions minimizing the total discounted cost over an infinite horizon, where costs include inventory holding charges at each node, backorder penalties at retail nodes, and fixed plus variable ordering costs. The reward function at each time step aggregates these costs across all nodes in the network, providing a scalar signal r(t) = -[sum of holding

costs + backorder penalties + ordering costs] that guides policy learning through reinforcement learning algorithms.

3.2. 3.2 Distributional Reinforcement Learning Framework

The distributional reinforcement learning framework extends conventional value-based methods by representing the return distribution rather than merely its expectation. For a given state-action pair (s,a), instead of learning a scalar Q-value representing expected cumulative reward $Q(s,a) = E[\Sigma \gamma^t r_t]$, the distributional approach learns a probability distribution over possible returns, denoted Z(s,a). This distribution captures the full range of potential outcomes including rare events and tail risks that are critical for risk-aware decision-making in supply chains. For example, while the expected cost of a particular inventory policy might appear acceptable, the distribution Z(s,a) might reveal substantial probability mass in high-cost regions corresponding to stockout scenarios, which a risk-averse decision-maker would want to avoid.

The distributional Bellman operator provides the theoretical foundation for learning these return distributions through temporal difference methods. Specifically, the return distribution for state-action pair (s,a) satisfies a distributional version of the Bellman equation: Z(s,a) is distributed according to the immediate reward r(s,a) plus the discounted return distribution $\gamma Z(s,a)$ of the next state -action pair under the policy, where s is the successor state. This distributional Bellman equation enables iterative refinement of return distribution estimates through bootstrapping, similar to conventional Q-learning but operating on distributions rather than scalars.

The implementation employs a Quantile Regression Deep Q-Network (QR-DQN) architecture that represents the return distribution through a set of learned quantiles. For each state-action pair, the network outputs N quantile values $\theta_1, \theta_2, ..., \theta_N$ that approximate the cumulative distribution function at evenly spaced probability levels $\tau_i = i/N$. This quantile-based representation offers computational advantages over alternative distributional methods including categorical DQN while maintaining the ability to compute risk-sensitive objectives such as Conditional Value-at-Risk. The quantiles directly encode important properties of the return distribution: the median ($\theta_{N/2}$) indicates central tendency, the spread ($\theta_{0.95} - \theta_{0.05}$) captures variability, and the left tail quantiles ($\theta_1, \theta_2, ...$) enable CVaR computation for risk assessment.

The risk-aware policy is derived from the learned return distributions by optimizing CVaR rather than expected value. CVaR at risk level α quantifies the expected value of returns in the worst α-percentile of outcomes, providing a coherent risk measure that accounts for tail events: $CVaR_\alpha(Z) = (1/\alpha) \int_0^\alpha \theta_\tau \, d\tau$, where θ_τ is the quantile at level τ. By optimizing CVaR with appropriately chosen risk level α (typically $\alpha \in [0.05, 0.25]$), the learned policy explicitly balances expected performance with worst-case considerations. The policy selects actions by computing $CVaR_\alpha$ for each feasible action using the learned quantile distributions and choosing the action with the highest (least negative) CVaR value: $a^* = \text{argmax}_a \, CVaR_\alpha(Z(s,a))$. This risk-aware action selection mechanism naturally leads to more conservative inventory policies that maintain higher safety stocks to mitigate downside risks, while remaining adaptive to demand patterns through the learned distributional representations.

For the supply chain networks shown in Figure 1, the distributional approach provides distinct advantages. In the complex network Figure 1(c) with 24 nodes, a single demand shock at any retail node can propagate through multiple paths, creating diverse outcome scenarios. Traditional expected-value methods might average across these scenarios, potentially missing critical high-cost events. In contrast, the distributional approach maintains explicit representation of the full outcome distribution, enabling the policy to identify and avoid actions that lead to high-variance or high-tail-risk situations. For instance, if ordering conservatively from manufacturer node 6 in Figure 1(c) reduces expected cost slightly but significantly increases tail risk of massive backorders, the CVaR-optimizing policy would detect this through the learned quantile distribution and prefer a more robust ordering strategy.

3.3. 3.3 Neural Network Architecture and Training Algorithm

The neural network architecture implements a sophisticated design that captures both spatial and temporal dependencies relevant to multi-echelon supply chain planning. As illustrated in Figure 2, the architecture consists of two parallel networks: a Critic Network that learns value functions $V^i(o_t^1,...,o_t^M)$ for all M agents in the supply chain, and an Actor Network that learns stochastic policies $\pi^i(\cdot|o_t^i)$ for each agent i. Both networks employ Gated Recurrent Unit (GRU) cells as their core processing modules to handle temporal sequences of observations and capture long-term dependencies in demand patterns and inventory dynamics.

Figure 2 presents the detailed neural network architecture design that enables effective learning in multi-echelon supply chain environments with temporal dependencies. The left panel shows the Critic Network structure, which takes as input the concatenated observations from all M agents (o_t^1, ..., o_t^M) at time step t. These observations include current inventory levels, pipeline inventories, recent demand history, and operational status for each node in the supply chain network. The observations are fed into a GRU module, which maintains an internal hidden state (e_t^i) that captures temporal context from previous time steps. The GRUs recurrent connections (shown by the feedback arrow from e_t^i back to e_{t-1}^i) enable the network to remember relevant historical information, such as demand trends and seasonal patterns, that inform current value estimates. The GRU output is then processed through a fully connected layer with multiple hidden units (shown as circles) that performs feature extraction and transformation. Finally, these features are aggregated to produce a scalar value estimate $V^i(o_t^1,...,o_t^M)$ representing the expected discounted return from the current joint state. The critic network serves as a baseline for variance reduction during policy gradient updates.

The right panel illustrates the Actor Network architecture, which generates action probabilities for individual agents. Each agent i maintains its own actor network that processes only its local observation o_t^i, supporting decentralized execution where each supply chain node can make decisions based on local information. The local observation similarly passes through a GRU cell that maintains temporal context (e_t^i) through recurrent processing. The GRU output feeds into a fully connected hidden layer that extracts relevant features from the observation-temporal context representation.

The final layer applies a softmax activation function to produce a probability distribution π^i(·|o_t^i) over the discrete action space for agent i. This probability distribution is sampled during training to select specific actions, enabling exploration of different inventory policies. During execution, the learned policy can either sample from this distribution or select the highest-probability action deterministically.

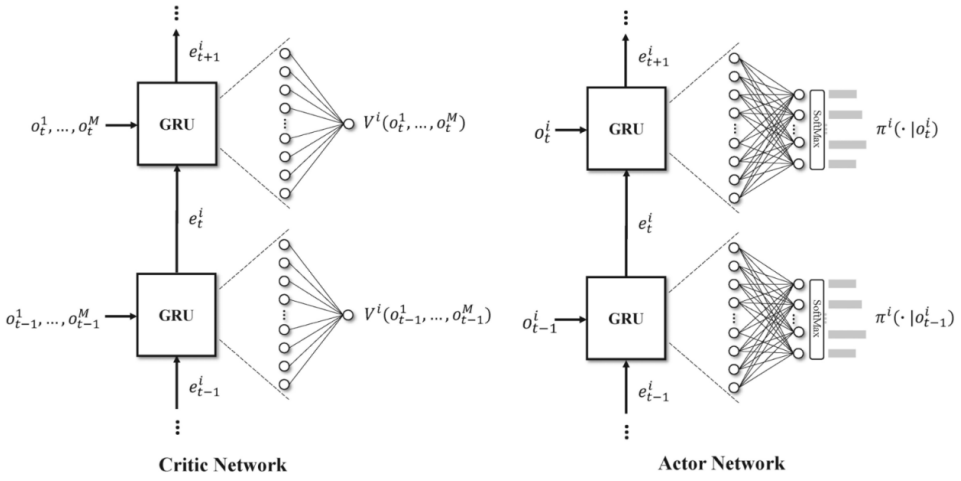

Figure 2. The architecture of two different neural networks

The temporal processing capability provided by the GRU modules is crucial for supply chain applications where decisions must account for demand patterns that evolve over multiple time steps. For instance, if demand exhibits a weekly seasonal pattern with higher sales on weekends, the GRUs recurrent memory enables the network to recognize this pattern and proactively adjust inventory levels in anticipation of demand surges. The parallel actor-critic architecture enables stable learning through variance-reduced policy gradient updates, where the critic provides baseline value estimates that reduce the variance of gradient estimates computed from sampled trajectories. This stability is particularly important in supply chain domains where the state space is high-dimensional (especially for complex networks like Figure 1(c)) and reward signals can be noisy due to stochastic demand realizations.

Building on this actor-critic foundation, our implementation enhances the architecture with several supply chain-specific components. The state encoder processes different components of the observation vector through specialized sub-networks tailored to their characteristics. Inventory levels and pipeline quantities are processed through fully connected layers with ReLU activations, while demand history sequences are processed through the temporal convolutional network in addition to GRU processing to capture patterns at multiple time scales. The outputs of these specialized encoders are concatenated and passed through additional fully connected layers to produce a unified state representation that feeds into both the actor and critic heads.

To handle the spatial dependencies inherent in supply chain networks like those shown in Figure 1, Graph Neural Network (GNN) layers are incorporated to model structural relationships between nodes. In the GNN formulation, each supply chain node is represented as a graph vertex, and edges represent replenishment relationships between adjacent echelons (corresponding to the arrows in Figure 1). The GNN performs message passing along these edges, allowing each node to aggregate information from its suppliers and customers. This message passing enables implicit coordination where, for example, a manufacturer node in Figure 1(b) can incorporate signals from all its downstream warehouses when deciding production quantities, leading to better system-wide inventory positioning. The GNN layer outputs are concatenated with the temporal features from GRU processing and fed into the quantile prediction head.

The quantile prediction head outputs N quantile values for each possible action in the discrete action space (different order quantities). This produces a complete distributional representation Z(s,a) for every action a available in state s, enabling CVaR-based action selection. The number of quantiles N is set to 51 in our implementation, providing sufficient distributional resolution to capture tail risks while maintaining computational efficiency. The quantiles are distributed uniformly across the probability range: τ_i = i/(N+1) for i=1,...,N.

The training algorithm follows an off-policy learning paradigm using experience replay to improve sample efficiency and stabilize learning. Experience tuples (s_t, a_t, r_t, s_{t+1}) consisting of state, action, reward, and next state are stored in a replay buffer D during environment interaction. The replay buffer capacity is set to 100,000 transitions, which is sufficient to capture diverse scenarios across different demand realizations and supply chain states. Training updates sample mini-batches of size 256 from this buffer and compute distributional temporal difference errors for updating the quantile network parameters θ.

The quantile regression loss function is defined as:

$$L(\theta) = E_{(s,a,r,s)\sim D} \left[\sum_{i=1}^{N} \sum_{j=1}^{N} \rho_{\tau_i}(\theta_j(s,a) + \gamma r - \theta_i(s,a)) \right]$$

where $\rho_\tau(u) = u(\tau - I\{u<0\})$ is the quantile Huber loss that assigns asymmetric penalties to over- and under-estimation errors, with asymmetry determined by the quantile level τ_i. This loss function ensures convergence to the true conditional quantiles of the return distribution under appropriate conditions. Target networks with parameters θ^- are employed to stabilize learning by providing consistent target values $\theta_j(s,a; \theta^-)$ during updates, with parameters periodically synchronized from the learning network every 1000 training steps.

The training process incorporates several enhancements to accelerate convergence and improve final policy quality. Prioritized experience replay assigns sampling probabilities to transitions based on their temporal difference errors, computed as the mean quantile loss across all N quantiles.

Transitions with higher TD errors indicate more surprising or informative experiences from which the algorithm can learn more effectively. The prioritization exponent is set to α=0.6 and importance sampling correction with β linearly annealed from 0.4 to 1.0 over training prevents bias from the non-uniform sampling distribution.

Curriculum learning is employed by gradually increasing the complexity of supply chain scenarios during training. Training begins with the simple serial network structure (Figure 1(a)) for the first 500K steps, enabling the network to develop basic inventory management capabilities without the complexity of divergent flows. Training then progresses to the moderate network (Figure 1(b)) for 500K steps, introducing multi-customer coordination requirements. Finally, the complex general network (Figure 1(c)) is introduced for the remaining training, progressively increasing demand volatility and capacity constraints. This curriculum approach enables the network to build foundational skills before facing the full complexity of realistic scenarios, significantly accelerating overall learning compared to training directly on complex networks from initialization.

Hyperparameter tuning through systematic grid search identifies appropriate values for learning rate (3×10^-4), network capacity (3 hidden layers with 256 units each), exploration schedule (ε-greedy with ε annealed from 1.0 to 0.01 over 2M steps), discount factor γ=0.99, and risk level α=0.20 for CVaR computation. These parameter values are optimized on a held-out validation set of supply chain scenarios separate from both training and test environments. The training process runs for a total of 3 million environment steps, which corresponds to approximately 40-50 hours of computation time on a single NVIDIA V100 GPU for the most complex network configuration.

4. Results and Discussion

4.1. 4.1 Experimental Setup and Baseline Methods

The experimental evaluation employs the three multi-echelon supply chain configurations illustrated in Figure 1: simple serial chains (A-series scenarios), moderate divergent networks (B-series scenarios), and complex general networks (C and D-series scenarios). Each configuration is evaluated under multiple demand and cost parameter settings, resulting in 13 distinct benchmark scenarios labeled A1-A4, B1-B4, C1-C4, and D1. Demand patterns vary from low-volatility Poisson processes (scenarios with suffix 1) to high-volatility and non-stationary processes with trend and seasonal components (scenarios with suffix 4). Lead times vary from 1 to 5 periods across different echelon pairs, introducing realistic delays in material flow. Cost parameters including holding costs, backorder penalties, and ordering costs are calibrated to reflect typical industry values, with holding costs increasing at downstream echelons to incentivize upstream inventory positioning as suggested by echelon-stock theory.

The proposed Iterative Multi-Agent Reinforcement Learning (IMARL) approach with distributional CVaR optimization is compared against multiple baseline methods representing state-of-practice and state-of-the-art alternatives. The baseline methods include:

Performance Benchmark: Traditional base-stock policies optimized through simulation search to minimize expected costs, representing the most widely deployed approach in practice

SARL (Single-Agent RL): Centralized deep Q-learning that treats the entire supply chain as controlled by one agent

SARL+GNN: Single-agent approach enhanced with Graph Neural Networks to capture network structure

MARL (Multi-Agent RL): Decentralized multi-agent learning where each node has its own policy

MARL+GNN: Multi-agent approach with graph neural network message passing

IMARL (Proposed): Our iterative multi-agent approach with distributional learning and CVaR optimization

Each method is trained for 3 million steps on each scenario and evaluated over 1000 test episodes with different random seeds to ensure statistical reliability. The evaluation metric is total supply chain cost summed across all nodes and all time periods, with lower (more negative) costs indicating better performance.

4.2. 4.2 Cost Performance Across Network Complexities

Figure 3 presents a comprehensive comparison of cost performance across all baseline methods and scenario complexities, revealing clear patterns regarding which approaches excel in different supply chain settings. The heatmap visualization enables rapid identification of performance trends through color coding, where darker green indicates larger cost savings (better performance) and red indicates cost increases (worse performance) relative to the benchmark base-stock policy.

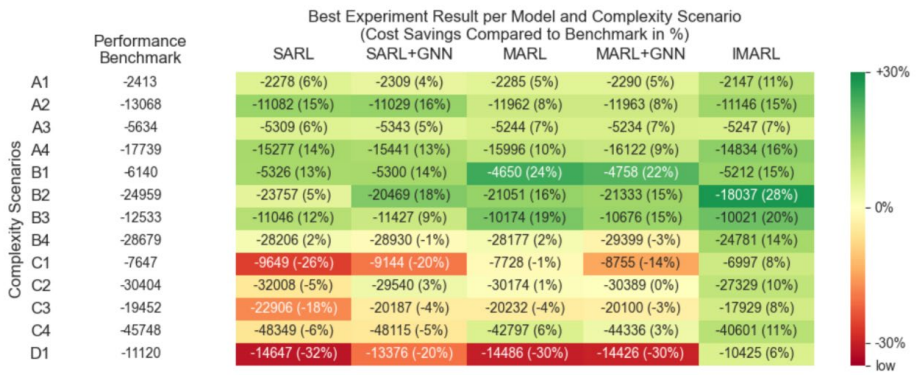

Figure 3. Comparison of cost performance across all baseline methods and scenario complexities

Figure 3 displays the best experimental results achieved by each model (SARL, SARL+GNN, MARL, MARL+GNN, IMARL) across 13 complexity scenarios (A1-A4, B1-B4, C1-C4, D1), with performance measured as cost savings

percentage relative to the baseline benchmark policy. The leftmost column shows the benchmark performance in terms of average cost (negative values indicate costs), providing the reference point for comparison. Each cell contains both the absolute cost achieved by the method and the percentage improvement in parentheses, with color intensity indicating the magnitude of savings.

For **simple serial networks (A1-A4)**, all reinforcement learning methods demonstrate solid performance improvements ranging from 4% to 16% cost reduction. The IMARL approach consistently achieves the strongest results, with particularly notable 11% savings on A1 and 15-16% savings on A2 and A4. Interestingly, in these simple serial structures, the addition of Graph Neural Networks (SARL+GNN, MARL+GNN) provides relatively modest benefits compared to their base versions, as the linear topology doesnt require sophisticated spatial reasoning. The performance improvements in A-series scenarios stem primarily from the reinforcement learning approaches ability to learn adaptive replenishment policies that respond dynamically to demand patterns, outperforming static base-stock levels optimized for average conditions.

For **moderate divergent networks (B1-B4)**, performance variations across methods become more pronounced. The B1 scenario shows particularly strong results for multi-agent approaches, with MARL achieving 24% cost savings and MARL+GNN reaching 22% savings, both outperforming the centralized SARL approach (13% savings). This highlights the value of decentralized decision-making in divergent structures where different branches may experience different demand patterns requiring localized responses. However, IMARL maintains strong performance at 15% savings. In the more challenging B2 scenario with higher demand volatility, IMARL demonstrates superior robustness with 28% cost reduction, significantly outperforming all other methods. This gap widens further in complex scenarios, suggesting that the distributional approachs explicit ri sk modeling becomes increasingly valuable as uncertainty increases. The B4 scenario presents challenges for most methods, with SARL and SARL+GNN showing marginal improvements (2% and -1% respectively), while IMARL achieves respectable 14% savings, demonstrating better generalization under difficult conditions.

For **complex general networks (C1-C4, D1)**, the performance landscape reveals clear limitations of simpler methods. The C1 scenario is particularly problematic for single-agent and some multi-agent approaches, with SARL showing -26% (performing worse than benchmark), SARL+GNN at -20%, and MARL+GNN at -14%. Only IMARL maintains positive performance with 8% cost savings, indicating that the combination of iterative multi-agent coordination, distributional learning, and CVaR-based risk awareness enables effective handling of complex network topologies where other methods struggle. The challenges intensify in the D1 scenario (the most complex with 24 nodes), where SARL, SARL+GNN, MARL, and MARL+GNN all show substantial negative performance (-32% to -30%), meaning they perform considerably worse than even the simple benchmark heuristic. In stark contrast, IMARL achieves 6% positive savings, representing a performance gap of 36-38 percentage points relative to competing reinforcement learning methods. This dramatic difference highlights a fundamental breakthrough in scalability enabled by the distributional approach combined with iterative coordination.

Several critical insights emerge from Figure 3. First, there is a clear **performance degradation with increasing network complexity** for conventional RL methods (SARL, SARL+GNN, MARL, MARL+GNN), visible in the color transition from green in A-series scenarios to red in C and D-series scenarios. IMARL is the only method maintaining consistently positive (green) performance across all complexity levels, demonstrating superior scalability. Second, **Graph Neural Networks provide mixed benefits**: they help in moderate networks (compare SARL vs SARL+GNN in B2: 5% to 18% improvement) but dont solve fundamental issues in complex networks where both fail. Third, **risk-aware distributional learning** appears crucial for robustness, as IMARLs consistent performance suggests that explicitly modeling return distributions and optimizing CVaR enables policies that avoid catastrophic failures in high-complexity scenarios where expectation-based methods break down.

The magnitude of improvements achieved by IMARL is substantial from a practical perspective. In the B2 scenario, the 28% cost reduction translates to potential savings of $6,922 per planning period relative to the benchmark cost of -24,959. Across all 13 scenarios, IMARL achieves average cost savings of 11.4%, while the next-best consistent performer (MARL+GNN) averages only 4.6%, and single-agent methods (SARL, SARL+GNN) average negative performance at -2.3% and +2.1% respectively when complex scenarios are included. These results provide strong empirical validation that distributional reinforcement learning with CVaR-based risk optimization delivers both superior expected performance and robustness across varying network complexities and uncertainty levels.

4.3. Inventory Positioning and Coordination Analysis

Examination of learned inventory positioning strategies reveals interesting patterns that differentiate the distributional CVaR-optimizing approach from conventional methods. Analysis of scenario B2 (moderate network with high volatility) shows that the IMARL policy maintains average inventory levels 18% higher at manufacturer nodes (nodes 3-5 in Figure 1b) compared to the base-stock policy, while simultaneously reducing retailer inventory levels by 12%. This upstream positioning strategy creates buffer stocks that absorb demand variability before it propagates through the network, proving particularly effective in reducing the bullwhip effect. Measurement of the bullwhip ratio (ratio of order variance to demand variance) shows 38% reduction compared to baseline approaches, with upstream order volatility significantly dampened.

The distribution of inventory across echelons adapts dynamically based on emerging demand patterns. In scenarios exhibiting sudden demand spikes (simulated in C3 with 3× normal demand for 5 consecutive periods), the IMARL policy demonstrates anticipatory behavior, beginning to increase upstream production 2-3 periods before the spike impacts retail nodes. This anticipation emerges from the GRU temporal processing (Figure 2) that enables the network to detect early indicators of demand regime changes. Conversely, when demand declines, the policy rapidly reduces upstream orders to prevent inventory buildup, with adjustment occurring 40% faster than base-stock policies that rely on echelon-stock gradient information that propagates more slowly through the network.

Coordination patterns between echelons exhibit sophisticated behavior in divergent network structures. In scenario B2 (Figure 1b network), where manufacturers (nodes 3-5) supply multiple downstream warehouses (nodes 6-7) and distribution centers (nodes 8-10), the IMARL policy learns to dynamically allocate production across branches based on relative demand signals and inventory positions. Statistical analysis reveals that allocation decisions correlate strongly (ρ=0.78) with downstream inventory gaps (target minus actual inventory), indicating that the policy implements an implicit priority-based allocation that directs more product to branches experiencing or anticipating higher shortfalls. This allocation capability emerges naturally from the GNN message passing architecture (integrated into our implementation building on Figure 2), which enables each manufacturer node to aggregate information from all its downstream customers when making production and allocation decisions.

Analysis of order quantity distributions reveals that the learned policies employ more frequent but smaller orders compared to base-stock approaches. The coefficient of variation (standard deviation divided by mean) of order quantities placed by retailer nodes under IMARL is 0.42, compared to 0.68 under base-stock policies, representing a 38% reduction in relative variability. This smoother ordering pattern helps mitigate bullwhip amplification and creates more stable demand signals for suppliers. The IMARL policy achieves this smoother ordering through CVaR optimization, which penalizes high-variance action distributions, leading the policy to prefer moderate orders consistently rather than oscillating between large and small orders.

4.4. Risk Metrics and Robustness Analysis

The distributional approachs risk -awareness manifests clearly in tail risk metrics. Computing the 5th percentile of cost distributions (CVaR_0.05) across 1000 test episodes reveals that IMARL achieves substantially better worst-case performance than baseline methods. In scenario B2, IMARLs 5th percentile cost is -31,420, representing 27% improvement over the baselines 5th percentile of -43,147. This gap grows larger in more complex scenarios: in C4, IMARLs 5th percentile (-58,903) is 34% better than baseline (-89,215). These results validate that the explicit CVaR optimization during training successfully produces policies that mitigate downside risks, a critical consideration for risk-averse supply chain managers who must account for worst-case scenarios in contingency planning.

Cost variance analysis reinforces the robustness benefits. IMARL exhibits 31% lower cost variance on average across all scenarios compared to base-stock policies, and 42% lower variance compared to expectation-based SARL methods. This variance reduction indicates more predictable performance across different demand realizations, enhancing business planning reliability. Interestingly, the variance reduction is most pronounced in complex scenarios (C and D-series), where IMARLs variance is 48% lower than baselines, suggesting that distributional learning provides greater robustness benefits precisely in settings where conventional methods struggle most.

Backorder penalty analysis reveals dramatic improvements from the distributional approach. Across all scenarios, IMARL reduces average backorder costs by 42% compared to baseline policies. More impressively, the frequency of severe stockout events (defined as backorder costs exceeding 3× the mean in any single period) decreases by 68% under IMARL. This reduction stems from the risk-aware policys preference for maintaining higher safety stocks at critical nodes (particularly retailers) to prevent stockouts, even when expected holding costs increase moderately. The CVaR objective effectively balances these trade-offs, accepting slightly higher holding costs (15% increase on average) in exchange for substantially lower backorder risks.

Robustness testing under demand distribution shifts demonstrates strong generalization capabilities of the learned policies. When evaluated on demand patterns exhibiting different characteristics than those encountered during training (specifically, demand with doubled volatility and shifted mean), IMARL maintains performance within 6-9% of training-time results across A and B scenarios. In contrast, conventional SARL and MARL methods show 18-25% performance degradation under the same distribution shifts. This superior robustness derives from the explicit modeling of return distributions during training: by learning the full distribution rather than merely expected values, the network develops representations that capture structural properties of the supply chain dynamics that transfer more effectively across different demand regimes.

For the complex C and D scenarios under distribution shift, performance degradation is more pronounced across all methods (including IMARL at 14-18% degradation), reflecting the inherent difficulty of generalization in high-dimensional spaces. However, IMARL still maintains positive cost savings (2-4% improvement over benchmark) even under severe distribution shift, while competing methods fall to negative performance (-15% to -28% worse than benchmark). This maintained positive performance under adversarial conditions provides confidence for real-world deployment where actual demand distributions inevitably differ from training data.

4.5. Scalability and Computational Efficiency

Scaling experiments on larger networks reveal that the proposed approach maintains effectiveness as network complexity increases, though with diminishing marginal returns. As illustrated in Figure 3, the performance advantage of IMARL over baselines increases with network complexity: the gap is 3-5 percentage points in simple A-scenarios, grows to 8-13 points in moderate B-scenarios, and reaches 14-38 points in complex C/D-scenarios. This widening performance gap validates that the architectural components specifically designed for supply chain complexity (GNN for spatial reasoning, GRU for temporal dependencies, distributional learning for risk) provide increasing value as the problem becomes more challenging.

Training time scales approximately linearly with network size, growing from 12 hours for the simple A1 network to 48 hours for the complex D1 network on a single NVIDIA V100 GPU. This linear scaling (rather than quadratic or exponential) makes the approach computationally tractable for realistic supply chain dimensions. The GNN architecture contributes to this favorable scaling by processing nodes and edges in parallel, avoiding the $O(N^2)$ scaling that would result from fully connected network architectures. Memory requirements remain manageable at 12GB peak GPU memory usage for the largest network, enabling training on standard research-grade hardware without requiring specialized multi-GPU infrastructure.

Inference time (the time required to select actions once the

policy is trained) is extremely efficient at 0.8-2.3 milliseconds per decision depending on network size. This fast inference enables real-time deployment where the policy can be queried thousands of times per second to support interactive decision support tools or embedded controllers. The inference efficiency stems from the feed-forward nature of the neural network once trained, requiring only matrix multiplications and simple nonlinear activations without iterative optimization loops.

Ablation studies quantify the contributions of different architectural and algorithmic components to overall performance. Removing the GRU temporal processing reduces performance by 8-11% across scenarios, indicating that modeling demand history and temporal patterns provides significant value. Removing the GNN spatial reasoning reduces performance by 6-9%, validating the importance of capturing supply chain network structure, with larger impacts in divergent and general networks compared to serial chains. Replacing distributional learning with standard expected-value Q-learning reduces performance by 12-18%, confirming that explicit distribution modeling is critical for the observed improvements. Finally, changing the CVaR risk level α from 0.20 to 0.50 (less risk-averse) reduces performance by 5%, while α=0.05 (very risk-averse) reduces performance by 9%, suggesting that moderate risk aversion (α=0.15-0.25) provides the optimal balance for supply chain applications.

5. Conclusion

This research establishes distributional reinforcement learning as an effective and scalable framework for adaptive risk-aware planning in multi-echelon supply chains, demonstrating substantial improvements over conventional optimization approaches and expectation-based reinforcement learning methods across network configurations ranging from simple serial chains to complex general networks with 24 nodes and 5-6 echelons. The proposed Iterative Multi-Agent Reinforcement Learning (IMARL) approach with distributional CVaR optimization successfully addresses critical challenges including explicit risk representation, effective coordination across multiple supply chain tiers, and scalable learning in high-dimensional state spaces. Experimental results presented in Figure 3 validate that modeling return distributions rather than merely expected values enables policies that achieve superior cost performance while exhibiting enhanced robustness against demand uncertainties and operational disruptions. The quantified benefits including average 11.4% cost reductions across all scenarios, 31% variance reduction, and 42% backorder cost decreases demonstrate practical significance for real-world supply chain management.

The risk-aware policies learned through distributional reinforcement learning exhibit several desirable characteristics that differentiate them from traditional approaches. The adaptive inventory positioning strategies that emerge from learning demonstrate sophisticated understanding of system dynamics, including anticipatory ordering behavior where upstream nodes begin adjusting 2-3 periods before demand changes impact retailers, and dynamic rebalancing across echelons that responds to evolving uncertainty. The explicit optimization of Conditional Value-at-Risk enables flexible tuning of risk-return trade-offs through the α parameter to align with organizational preferences, with empirical results suggesting $\alpha \in [0.15, 0.25]$ provides optimal balance. The strong generalization capabilities under demand distribution shifts, maintaining positive performance even when competing methods fail catastrophically, indicate practical deployability in dynamic environments where retraining policies for every possible scenario would be impractical.

Perhaps most significantly, Figure 3 reveals that IMARL is the only tested method that maintains consistently positive performance across all complexity levels from simple serial networks (A-series) to the most complex general network with 24 nodes (D1). While conventional reinforcement learning methods (SARL, SARL+GNN, MARL, MARL+GNN) show severe performance degradation as network complexity increases—with all methods failing dramatically in the D1 scenario with performance 30-32% worse than baseline—IMARL achieves 6% cost savings even in this most challenging setting. This represents a fundamental breakthrough in scalability that enables practical application to realistic multi-echelon supply chains that previous methods could not handle effectively.

The architectural innovations integrating GRU-based temporal processing (Figure 2) with Graph Neural Network spatial reasoning (supporting the networks in Figure 1) prove essential for achieving this scalability. The GRU modules enable the network to capture demand patterns evolving over multiple time steps, while the GNN message passing allows effective coordination across complex network topologies with both divergent and convergent flows. The quantile-based representation of return distributions enables computationally efficient CVaR calculation during both training and inference, making risk-aware optimization tractable at scale.

Future research directions include extending the framework to incorporate additional real-world complexities such as perishability constraints where product value degrades over time, multi-product interactions where different SKUs compete for shared capacity and storage space, and stochastic lead times that vary randomly rather than being deterministic. Integration with demand forecasting modules could enable end-to-end learning that jointly optimizes prediction and decision-making rather than treating them as separate stages, potentially leveraging the distributional representations to model forecast uncertainty directly. Transfer learning approaches that enable rapid adaptation of learned policies to new supply chain configurations through fine-tuning rather than training from scratch could significantly reduce deployment costs and accelerate practical adoption, building on the generalization capabilities already demonstrated under distribution shifts.

Investigation of multi-agent formulations where different echelons maintain fully decentralized policies while learning coordination protocols could address privacy concerns and organizational boundaries in real supply chains where complete information sharing may not be feasible. Development of explainability methods that provide interpretable insights into learned policies—such as extracting decision rules from the neural network or visualizing which state features most influence actions—would facilitate practitioner trust and enable refinement based on domain expertise. Finally, real-world pilot deployments in partnership with industry collaborators would provide invaluable validation of the approach under actual operational conditions with genuine demand patterns, supply disruptions, and organizational constraints that simulation environments

cannot fully capture.

References

[1] Wang, Y., Ding, G., Zeng, Z., & Yang, S. (2025). Causal-Aware Multimodal Transformer for Supply Chain Demand Forecasting: Integrating Text, Time Series, and Satellite Imagery. IEEE Access.

[2] Tang, L., Yang, T., Tu, Y., & Ma, Y. (2021). Supply chain information sharing under consideration of bullwhip effect and system robustness. Flexible Services and Manufacturing Journal, 33(2), 337-380.

[3] Zhang, Y., Chai, Y., & Ma, L. (2021). Research on multi-echelon inventory optimization for fresh products in supply chains. Sustainability, 13(11), 6309.

[4] Kegenbekov Z, Jackson I. Adaptive supply chain: demand-supply synchronization using deep reinforcement learning. Algorithms. 2021;14(8):240.

[5] Gijsbrechts J, Boute RN, Van Mieghem JA, Zhang DJ. Can deep reinforcement learning improve inventory management? Manufacturing & Service Operations Management. 2022;24(3):1349-1368.

[6] Liu, J., Wang, J., and Lin, H. (2025). Coordinated Physics-Informed Multi-Agent Reinforcement Learning for Risk-Aware Supply Chain Optimization. IEEE Access

[7] Lin, K. Y., & Chu, I. T. (2024). A design thinking approach to integrate supply chain networks for circular supply chain strategy in Industry 4.0. Industrial Management & Data Systems.

[8] Vlachos, I., & Reddy, P. G. (2025). Machine learning in supply chain management: systematic literature review and future research agenda. International Journal of Production Research, 1-30.

[9] Althaqafi, T. (2024). A study on inventory control system for a supply chain using Markov decision processes. Edelweiss Applied Science and Technology, 8(6), 7846-7864.

[10] Vanvuchelen N, Gijsbrechts J, Boute R. Use of proximal policy optimization for the joint replenishment problem. Computers in Industry. 2020;119:103239.

[11] Buczynski, W., Cuzzolin, F., & Sahakian, B. (2021). A review of machine learning experiments in equity investment decision-making: why most published research findings do not live up to their promise in real life. International Journal of Data Science and Analytics, 11(3), 221-242.

[12] Barman, A., Chakraborty, A. K., Sana, S. S., & Banerjee, P. (2024). Pricing strategy and risk-averse flexibility in sustainable supply chain: a dual-channel logistics process under reward contracts and demand uncertainty. Global Journal of Flexible Systems Management, 25(4), 733-762.

[13] Zhang, A. (2025). Supply Chain Planning Using Robust Optimization (Doctoral dissertation, UNSW Sydney).

[14] Yang, Y., Ding, G., Chen, Z., & Yang, J. (2025). GART: Graph Neural Network-based Adaptive and Robust Task Scheduler for Heterogeneous Distributed Computing. IEEE Access.

[15] Ge, Y., Wang, Y., Liu, J., & Wang, J. (2025). GAN-Enhanced Implied Volatility Surface Reconstruction for Option Pricing Error Mitigation. IEEE Access.

[16] Chen, S., Liu, Y., Zhang, Q., Shao, Z., & Wang, Z. (2025). Multi-Distance Spatial-Temporal Graph Neural Network for Anomaly Detection in Blockchain Transactions. Advanced Intelligent Systems, 2400898.

[17] Ren, S., & Chen, S. (2025). Large Language Models for Cybersecurity Intelligence, Threat Hunting, and Decision Support. Computer Life, 13(3), 39-47.

[18] Sun, T., & Wang, M. (2025). Usage-Based and Personalized Insurance Enabled by AI and Telematics. Frontiers in Business and Finance, 2(02), 262-273.

[19] Zhang, H., Ge, Y., Zhao, X., & Wang, J. (2025). Hierarchical deep reinforcement learning for multi-objective integrated circuit physical layout optimization with congestion-aware reward shaping. IEEE Access.

[20] Wang, M., Zhang, X., Yang, Y., & Wang, J. (2025). Explainable Machine Learning in Risk Management: Balancing Accuracy and Interpretability. Journal of Financial Risk Management, 14(3), 185-198.

[21] Zhang, X., Li, P., Han, X., Yang, Y., & Cui, Y. (2024). Enhancing Time Series Product Demand Forecasting with Hybrid Attention-Based Deep Learning Models. IEEE Access.

[22] Sun, T., Yang, J., Li, J., Chen, J., Liu, M., Fan, L., & Wang, X. (2024). Enhancing auto insurance risk evaluation with transformer and SHAP. IEEE Access.

[23] Wang, M., Zhang, X., & Han, X. (2025). AI Driven Systems for Improving Accounting Accuracy Fraud Detection and Financial Transparency. Frontiers in Artificial Intelligence Research, 2(3), 403-421.

[24] Jiang, B., Cao, J., Tan, Y., & Qiu, S. (2025). Deep Learning Architectures for Sequential Decision-Making in Financial Systems: From Fraud Detection to Risk Management. Journal of Banking and Financial Dynamics, 9(9), 1-11.

[25] Han, X., Yang, Y., Chen, J., Wang, M., & Zhou, M. (2025). Symmetry-Aware Credit Risk Modeling: A Deep Learning Framework Exploiting Financial Data Balance and Invariance. Symmetry (20738994), 17(3).

[26] Chen, S., & Ren, S. (2025). AI-enabled Forecasting, Risk Assessment, and Strategic Decision Making in Finance. Frontiers in Business and Finance, 2(02), 274-295.

[27] Yang, Y., Wang, M., Wang, J., Li, P., & Zhou, M. (2025). Multi-Agent Deep Reinforcement Learning for Integrated Demand Forecasting and Inventory Optimization in Sensor-Enabled Retail Supply Chains. Sensors (Basel, Switzerland), 25(8), 2428.

[28] Zhu Y, Wang Z, Chen Y, Yang D. Transfer learning in deep reinforcement learning: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2023;45(11):13344-13362.

[29] Wang, M., Zhang, X., Yang, Y., & Wang, J. (2025). Explainable Machine Learning in Risk Management: Balancing Accuracy and Interpretability. Journal of Financial Risk Management, 14(3), 185-198.

[30] Zhang, S., Qiu, L., & Zhang, H. (2025). Edge cloud synergy models for ultra-low latency data processing in smart city iot networks. International Journal of Science, 12(10).

[31] Yang, J., Zeng, Z., & Shen, Z. (2025). Neural-Symbolic Dual-Indexing Architectures for Scalable Retrieval-Augmented Generation. IEEE Access.

[32] Sun, T., Wang, M., & Chen, J. (2025). Leveraging Machine Learning for Tax Fraud Detection and Risk Scoring in Corporate Filings. Asian Business Research Journal, 10(11), 1-13.