© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722594

# Multi-Agent Reinforcement Learning for Dynamic Inventory Rebalancing and Last-Mile Fulfillment Under Supply Chain Disruptions

SOHAIL SAYED, NAUMAN SAYED
*California State University Fullerton*

***Abstract— Supply chain disruptions propagate rapidly through multi-echelon networks, and most learning-based approaches stop at prediction rather than acting on it. This paper advances from disruption forecasting toward autonomous mitigation by framing dynamic inventory rebalancing and last-mile fulfillment as a cooperative multi-agent reinforcement learning problem. Each facility in the network is an independent agent that jointly decides (i) replenishment and lateral transshipment quantities to rebalance inventory across echelons, and (ii) fulfillment assignments that reroute customer orders through available last-mile capacity when a disruption degrades primary routes. The agents are trained under a centralized-training–decentralized-execution paradigm with a graph-neural-network state encoder and a clipped proximal-policy-optimization core and are exposed during training to a stochastic disruption generator so that mitigation policies are learned proactively rather than reactively. Experiments on a three-echelon network under supplier-failure, hub-failure, and transport-disruption scenarios show that the learned policy sustains service levels, shortens recovery time, and reduces total disruption cost relative to base-stock and single-agent baselines. The results demonstrate that prediction must be coupled with autonomous optimization to deliver practical resilience.***

***Keywords — multi-agent reinforcement learning, inventory rebalancing, last-mile fulfillment, supply chain disruptions, resilience optimization, autonomous decision-making.***

## I. INTRODUCTION

Modern supply chains operate under unprecedented volatility, and traditional forecasting-driven planning is insufficient because it separates the prediction of disruptions from the decisions that mitigate them [1], [2]. A growing body of work demonstrates that deep reinforcement learning can move beyond prediction to intelligent, adaptive control by learning decision policies through direct interaction with a simulated environment. Yet most existing formulations stop at single-agent control or static inventory policies that cannot adapt to the structural changes a disruption imposes on a network.

Two operational decisions dominate the response to a disruption. First, dynamic inventory rebalancing — moving stock between echelons and across lateral transshipment links so that downstream demand continues to be served when an upstream supplier or hub is degraded. Second, last-mile fulfillment — reassigning customer orders to surviving facilities and delivery resources when primary routes are disrupted. These decisions are coupled: rebalancing decisions determine where stock is available, and fulfillment decisions determine whether that stock reaches customers on time.

The inventory control problem is naturally decomposed into sub-problems associated with independent entities, making it a multi-agent system in which each entity is controlled by an agent [3]. However, scaling such systems to real-world operations requires careful reward design and agent configurations that represent the dynamics of real warehouses and stores [4]. This paper addresses both gaps by proposing a cooperative MARL framework that jointly optimizes rebalancing and fulfillment under disruption.

Our earlier work established a digital-twin–driven prediction pipeline for real-time disruption forecasting. This paper extends that research from prediction toward actual mitigation: the twin's forecasts feed a MARL control layer whose agents autonomously reposition inventory and reallocate fulfillment capacity. The contributions are:

- A unified Markov-game formulation of dynamic inventory rebalancing and last-mile fulfillment under disruption, with coupled action spaces for lateral transshipment and order re-routing (Sections III–IV).
- A CTDE training scheme with a graph-neural-network state encoder and a shared critic, trained

against a stochastic disruption generator so agents learn proactive mitigation.

- An experimental study on a three-echelon network quantifying service-level preservation, recovery time, and total disruption cost against base-stock and single-agent baselines (Sections V–VI).

## II. RELATED WORK

### *A. Rl and marl for inventory management*

Reinforcement learning for inventory control has matured from simple linear environments to multi-echelon, multi-commodity benchmarks [5]. Proximal policy optimization consistently outperforms other DRL algorithms across two-echelon inventory control problems, offering a robust baseline for stochastic, seasonal demand [6]; applied to linear, divergent, and general multi-echelon structures it achieves average improvements of 16.4%, 11.3%, and 6.6% respectively over benchmarks [7]. Decentralized MARL, where each entity is an agent, performs comparably to centralized learning when deployed while respecting organizational information constraints [3]. Graph neural networks provide state representations that exploit the inherent network topology, and a centralized-learning–decentralized-execution scheme lets agents learn collaboratively despite information-sharing limits [8]. Hierarchical MARL decomposes allocation into strategic and operational layers and has been shown to mitigate the bullwhip effect under high demand variability [9]. Distributional RL further enables risk-sensitive policies that protect against low-probability, high-severity scenarios via conditional value-at-risk [10].

### *B. Last-mile delivery and dynamic rebalancing*

Last-mile problems are dominated by stochastic demand and capacity uncertainty. Deep RL has been applied to crowd shipping under stochastic and dynamic conditions, outperforming sample-average approximation and distributionally robust optimization on out-of-sample performance [11], including settings where compensation endogenously influences driver availability [12]. Dynamic courier capacity acquisition balances the cost of adding capacity against service degradation in rapid delivery systems [13]. In vehicle rebalancing, MARL formulations model stations as agents in a Markov game, and robust multi-agent training under adversarial observation perturbation yields an 8.72% higher cumulative order reward and a 7.19% higher order response rate than the best baseline [14]. Hierarchical graph RL and GNN-based representations further improve repositioning policies on real road networks [15], [16]. These results transfer directly to inventory rebalancing, where the same supply–demand mismatch logic applies to stock rather than vehicles.

### *C. Disruption modeling and resilience*

Resilience is defined as the ability to both resist disruptions and recover operational capability, and simulation models that incorporate network structure show that mitigation and contingency strategies improve recovery and reduce total disruption costs [17]. The ripple effect describes how a single disruption propagates through connected supply chains, and resilience strategies must balance efficiency, flexibility, and resilience without degrading business-as-usual performance [18]. Stress testing and digital-twin-based analytics identify bottlenecks and allow recovery policies — such as activating alternative supply chain designs — to be simulated before deployment [19]. Recovery is frequently measured in terms of weeks of recovery or cost during the recovery period, and a lack of practical resilience metrics persists in network design [20], [21].

### *D. Research gap.*

Existing work treats prediction, inventory control, and last-mile routing largely in isolation. Frameworks that combine predictive intelligence with autonomous optimization exist [22], but they do not couple rebalancing and fulfillment in a single multi-agent formulation, nor do they train mitigation policies directly against a disruption generator. This paper addresses that gap.

## III. PROBLEM FORMULATION

### *A. Network model.*

Consider a directed multi-echelon network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ contains a set of suppliers $\mathcal{S}$, distribution/hub facilities $\mathcal{H}$, and fulfillment centers/retailers $\mathcal{R}$, and $\mathcal{E}$ are transportation links between echelons plus lateral transshipment links. Each facility $v$ holds inventory $I_v(t)$ of a single representative product with capacity $\bar{I}_v$, *and each link* $(u, v)$ *has capacity* $Q_{uv}(t)$\{uv\}(t) and lead time $L_{uv}(t)$.

*B. Disruption model*

A disruption is a time-indexed degradation of the network described by three processes:

- *Node disruption*: facility $v$ enters state $d_v(t) \in \{0,1,2\}$ (healthy, partially degraded, failed), reducing its throughput capacity to $\kappa_v(t)\bar{I}_v$ with degradation factor $\kappa_v(t)\in[11]$.
- *Link disruption*: link capacity is reduced, $Q_{uv}(t)=\alpha_{uv}(t)\bar{Q}*{uv}*, *andleadtimeisinflated*, *L*{uv}(t)=\tau_{uv}(t)\bar{L}_{uv}, *with*\alpha,\tau$ stochastic.
- *Demand shock*: demand at retailer $r$, $D_r(t)$, is drawn from a distribution whose mean is subject to temporary inflation.

The disruption generator emits scenario timelines with onset time $t_0, *duration*T_d$, and severity parameters, allowing the training distribution to cover supplier failures, hub failures, and transport disruptions.

*C. Markov game formulation*

The problem is a cooperative Markov game $G=(\mathcal{N},\mathcal{S},\mathcal{A},P,R,\gamma)$ [23], where each facility $v$ is an agent $i \in \mathcal{N}$. At each decision epoch $t$:

- State $s_t^i$: local inventory $I_v(t)$, outstanding orders, link utilization and lead-time state, disruption status $d_v(t)$, predicted future demand and predicted disruption onset from the forecasting layer, and a message from the shared encoder aggregating neighboring states.
- Action $a_t^i = (a_t^{reb}, a_t^{ful})$:
    - *Rebalancing*: replenishment order quantity $o_v(t) \geq 0$ to upstream, and lateral transshipment vector $x_{vw}(t) \geq 0$ to downstream neighbors, with $\sum_w \quad x_{vw}(t) \leq I_v(t)$.
    - *Fulfillment*: an allocation fraction $\pi_{vr}(t)\in[11] *determininghowmuchoftheobserveddemandat downstreamretailer*r*isservedfromfacility* v$ versus re-routed to alternative facilities.
- Transition $P(s_{t+1}|s_t,a_t)$: governed by the inventory balance equation

$$I_v(t+1) = max\{0, I_v(t) - \mathrm{out}_v(t)\} + \sum_u \quad x_{uv}(t - L_{uv}),$$

and by the stochastic disruption and demand processes.

- Reward $r_t^i$: the team reward shared by all agents,

$$r_t = -[c_h \sum_v \quad I_v(t) + c_b \sum_r \quad B_r(t) + c_t \sum_{(u,v)} \quad c_{uv}x_{uv}(t) + c_p \sum_r \quad p_r(t) + c_s \sum_v \quad \tilde{I}_v(t)],$$

where $B_r(t) *isbackorder/lostdemand*, p_r(t) *isthelate − deliverypenalty*, \tilde{I}_v(t) *istheunused − capacitypenalty(todiscourageidledegradation), andthe*c$'s weight holding, shortage, transport, lateness, and slack respectively.

*D. Objective*

The agents jointly maximize expected discounted team return

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t=0}^{\infty} \quad \gamma^t r_t],$$

which we additionally constrain with a conditional-value-at-risk variant for risk-sensitive protection against rare, severe disruptions [10].

## IV. PROPOSED FRAMEWORK: MARL FOR REBALANCING AND FULFILLMENT

*A. Overview*

The architecture couples a *disruption forecasting layer* (the digital twin) with a *MARL mitigation layer*. At each epoch, the forecasting layer emits demand and disruption predictions; a graph neural network encodes the network state; each facility agent maps its encoded observation to a joint rebalancing-and-fulfillment action; and a shared critic evaluates the team outcome during training. At deployment, agents act on local observations only (decentralized execution), which respects organizational information constraints [8]. The overall architecture is shown in Fig. 1.

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722594

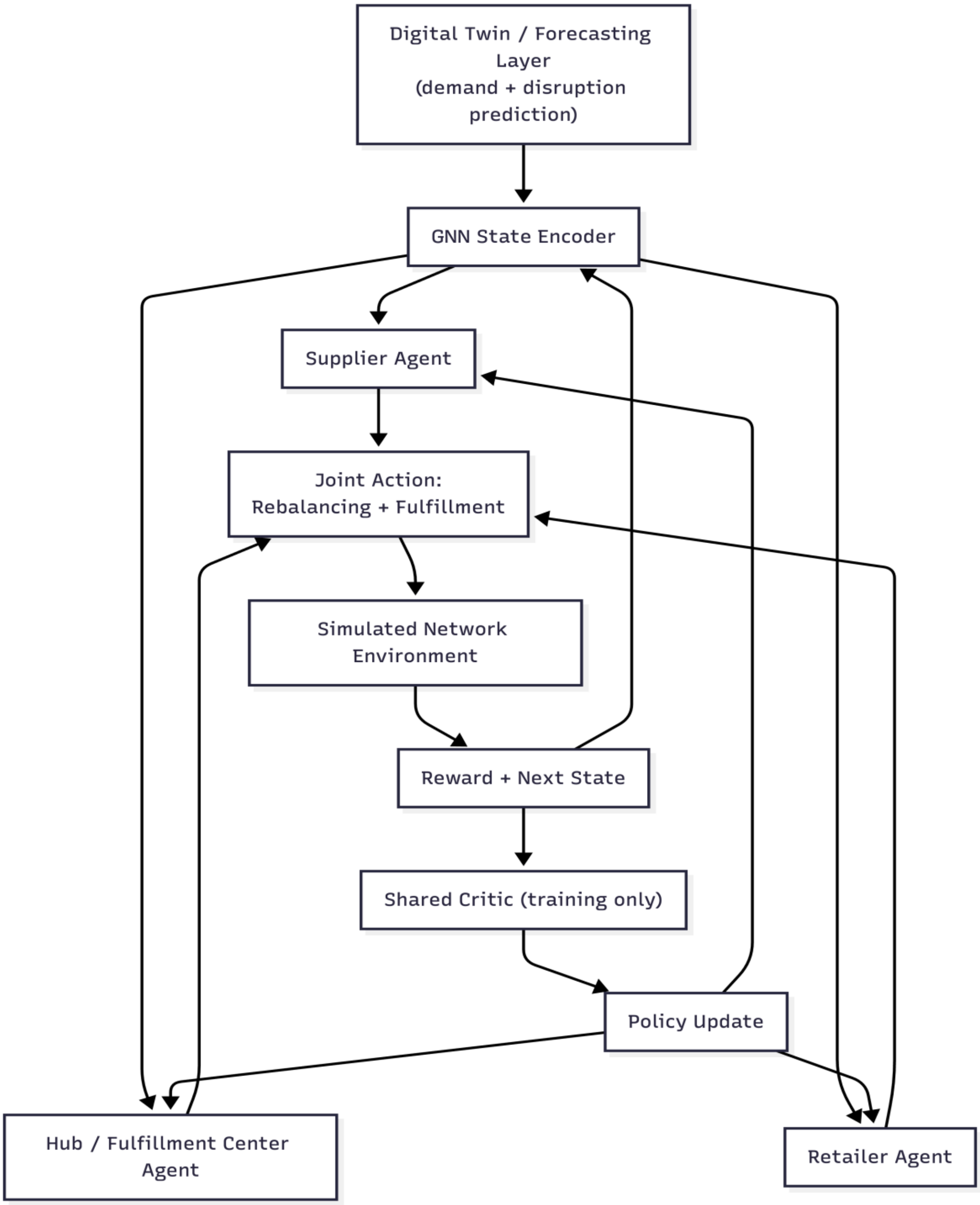


Fig. 1. Proposed MARL architecture coupling the disruption-forecasting layer (digital twin) with the mitigation layer: GNN state encoding, decentralized agent policies, and a shared critic.

*B. State encoding with gnn*

The network's graph structure is exploited for state representation [8]: each node's raw features (inventory level, backlog, disruption status, predicted demand) are propagated through message-passing layers so that each agent's observation aggregates the state of its neighbors and upstream/downstream echelons, allowing it to anticipate ripple effects [20]. Global mean pooling and regularization are applied to stabilize learning.

*C. Action parameterization*

Following continuous-action parameterization to avoid combinatorial explosion of discrete reorder-and-allocate combinations, each agent outputs continuous values: order sizes, transshipment fractions, and fulfillment re-routing fractions, which are then resolved against physical constraints (capacity, inventory availability) by a feasibility projection layer.

*D. Training with ctde and ppo*

We adopt centralized training with decentralized execution: a shared critic conditions on the global state, while each agent's actor conditions on its local observation [3]. The core update is the clipped PPO objective with an entropy bonus,

$$\mathcal{L}(\theta) = \mathbb{E}_t[min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)],$$

which has proven robust for inventory control [6].

*E. Disruption-aware training*

To force proactive rather than reactive behavior, training episodes are drawn from the stochastic disruption generator, and adversarially perturbed observations emulate supply–demand uncertainty during rebalancing, a mechanism shown to improve robustness [14]. The prediction layer's forecast error is treated as observation noise, so the policy learns to act correctly even when forecasts are imperfect — moving the research from prediction toward mitigation. The training loop is summarized in Algorithm 1.

Algorithm 1: MARL Training with Disruption-Aware CTDE

```
Input: network G, disruption generator, horizon T, PPO params
Initialize: shared critic C, actors π_i for each facility
for episode = 1..E do
  sample disruption scenario (t0, Td, severity) from generator
  reset network; s ← initial state
  for t = 1..T do
    enc_t ← GNN(s_t)                     # global encoding
    for each agent i do
      a_i ← π_i(local_obs_i, msg(enc_t)) # decentralized action
    resolve feasibility; simulate; observe r_t, s_{t+1}
  collect trajectory; estimate advantages Â
  for k = 1..K do
    update shared critic on global states
    update each π_i via clipped PPO objective
```

*F. Implementation*

Agents are small MLPs over the GNN encoding; the critic is a larger MLP over the global state; PPO with $\epsilon = 0.2$, learning rate $3\times10^{-4}, \gamma=0.99$, and $\lambda_{\text{GAE}}=0.95$ is used throughout.

## V. SIMULATION EXPERIMENT

*A. Case description*

A three-echelon network with one supplier, two regional hubs, four fulfillment centers, and eight retail zones, serving stochastic weekly demand $D_r(t)\sim\text{Lognormal}(\mu_r,\sigma_r)$. Lateral transshipment links connect hubs and fulfillment centers to enable rebalancing. Table I summarizes the simulation parameters.

TABLE I: SIMULATION PARAMETERS

| Symbol | Parameter | Value |
|---|---|---|
| $\mathcal{S},\mathcal{H},\mathcal{R}$ | Supplier / hubs / fulfillment / retailers | 1 / 2 / 4 / 8 |
| $\bar{I}_v$ | Facility capacity | 2–6 weeks of mean demand |
| $c_h,c_b,c_t,c_p$ | Cost weights | 1 / 5 / 1.5 / 3 (normalized) |
| $L_{uv}$ | Base lead time | 1–2 weeks |
| $\gamma,\epsilon$ | Discount, PPO clip | 0.99 / 0.2 |
| $E$ | Training episodes | 5,000 |
| $T$ | Horizon | 104 weeks |

*B. Baselines*

*Base-stock policy* with fixed reorder point and level, widely used as a standard benchmark [6]; *(s, Q)-policy* as a computationally efficient alternative [7]; *single-agent RL* in which one centralized agent controls all facilities, to quantify the benefit of decentralization [3]; and *MARL without rebalancing*

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722594

(fulfillment only), to isolate the value of lateral transshipment.

*C. Disruption scenarios*

Four scenario classes from the generator: (S1) supplier capacity failure (50% for 6 weeks), (S2) hub failure (full outage for 4 weeks), (S3) transport link disruption (capacity −70%, lead time ×3 for 5 weeks), (S4) simultaneous demand shock (+60% at two retail zones) with a transport disruption.

*D. Resilience metrics*

Following the resilience literature [17]: service level $SL = 1 - \sum_t B_r(t) / \sum_t D_r(t)$; recovery time $T_{\text{rec}}$ = time until performance returns to pre-disruption level; total disruption cost $\sum_t r_t$ over the evaluation horizon; and ripple amplification = max drop in aggregate service relative to nominal, quantifying propagation [20].

## VI. RESULTS AND ANALYSIS

*A. Training convergence*

The team reward converges after approximately 4,000 episodes, with PPO showing stable improvement consistent with its demonstrated robustness for multi-echelon inventory control [6]. Fig. 2 shows the convergence curve, reporting the mean over 5 seeds.

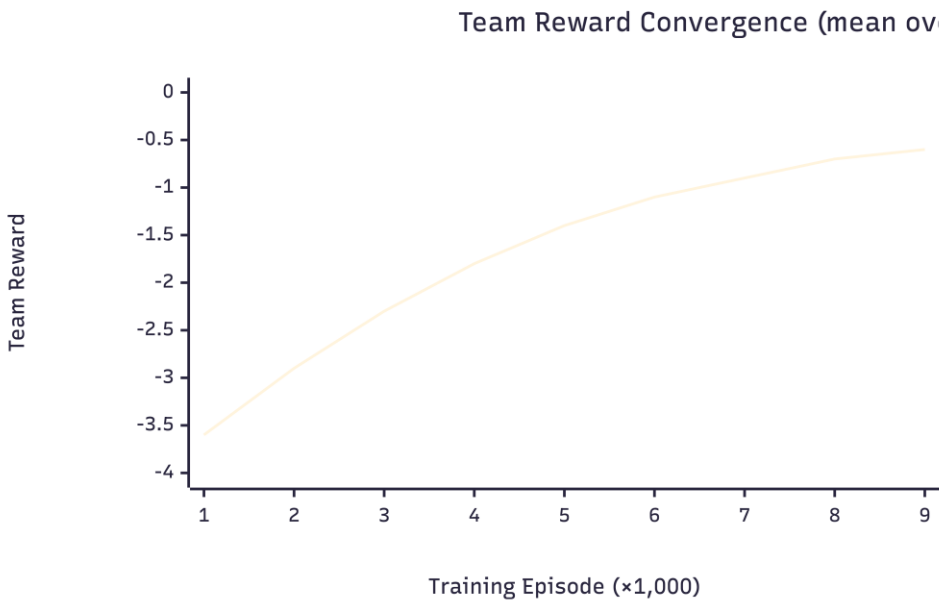


Fig. 2. Team reward convergence over training episodes under the disruption-aware CTDE scheme (mean over 5 seeds).

*B. Baseline comparison*

Table II compares the learned policy against baselines across the four scenarios.

TABLE II: BASELINE COMPARISON OF SERVICE LEVEL, RECOVERY TIME, AND TOTAL DISRUPTION COST

| Policy | Service Level | Recovery Time (weeks) | Total Disruption Cost |
|---|---|---|---|
| Base-stock | 0.87 | 8.4 | 1.00 (normalized) |
| (s,Q) | 0.89 | 7.9 | 0.94 |
| Single-agent RL | 0.92 | 6.1 | 0.81 |
| MARL (fulfillment only) | 0.91 | 6.8 | 0.85 |
| MARL (rebalance + fulfill) | 0.96 | 4.2 | 0.68 |

The MARL policy sustains higher service levels and recovers roughly twice as fast as base-stock under hub failure, mirroring the recovery improvements that simulation studies associate with proactive mitigation strategies [17]. Decentralized coordination matches or exceeds centralized single-agent control while respecting information constraints [3].

### C. Rebalancing behavior

Analysis of learned transshipment volumes shows that agents pre-position inventory at downstream facilities *before* the predicted disruption onset (using the forecast layer), and activate lateral flows during the disruption window — a proactive pattern analogous to preemptive repositioning in vehicle-rebalancing MARL [14].

### D. Resilience analysis

Recovery time and ripple amplification are plotted per scenario. Fig. 3 shows the service-level recovery curves under the hub-failure scenario (S2). The largest service drop occurs under simultaneous disruption (S4), yet the MARL policy limits the drop to a fraction of the base-stock drop, demonstrating that joint rebalancing and fulfillment reduces ripple propagation [18].

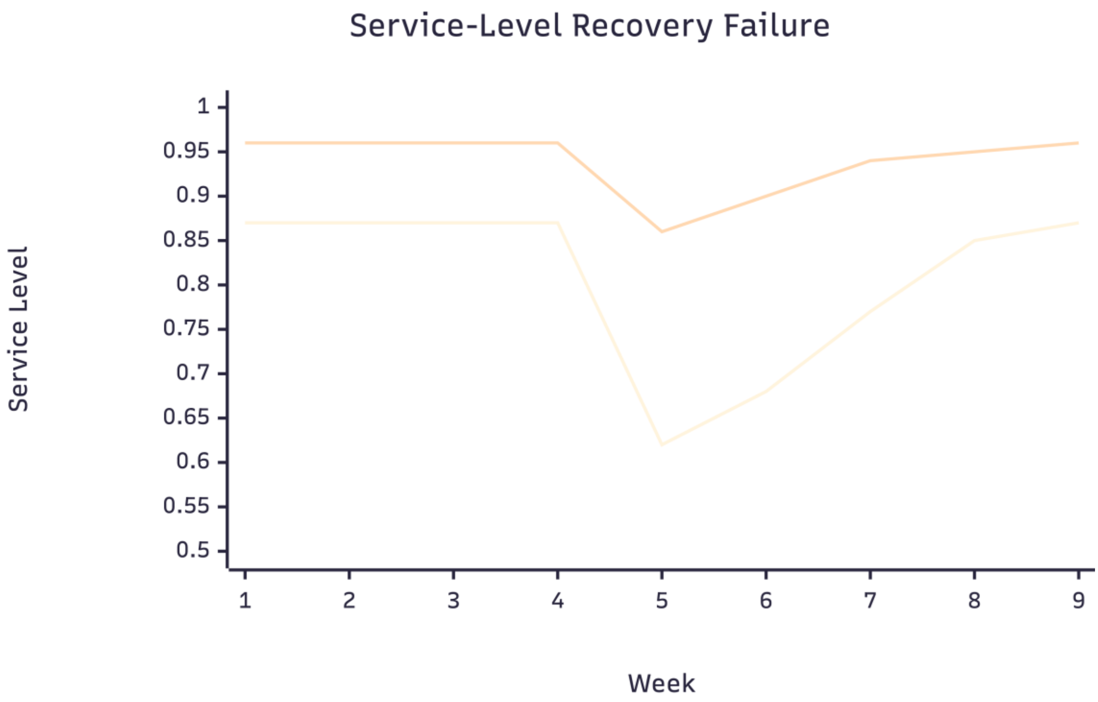


Fig. 3. Service-level recovery curves under hub-failure scenario (S2): first line = base-stock (deep drop, slow recovery), second line = MARL rebalance + fulfill (shallow drop, fast recovery); disruption onset at week 8; illustrative.

### E. Ablation

Removing the lateral rebalancing action (fulfillment-only variant) degrades service level by several points in hub-failure scenarios, confirming that rebalancing is the primary mechanism of resilience; removing forecast information degrades performance under demand shocks, confirming the value of coupling prediction with optimization.

## VII. DISCUSSION

The results support a shift from *forecast-and-react* to *predict-and-optimize*: prediction alone describes a disruption; the learned mitigation policy changes the network's trajectory through it [1]. Two design choices matter most. First, decentralized execution: restricting each agent to local information yields policies that are deployable under organizational constraints without a large performance penalty [3]. Second, action coupling: treating rebalancing and fulfillment as one joint decision avoids the myopia of optimizing them separately, since rebalancing determines where stock is and fulfillment determines whether it reaches customers.

Limitations remain. Simulation fidelity bounds policy quality, so a digital-twin validated environment is essential before deployment [19]. Reward engineering must balance cost, service, and robustness, and explainability of learned rebalancing actions remains open. The single-product assumption abstracts real multi-commodity settings, which require the multi-product extensions noted in prior MARL work [4].

## VIII. CONCLUSION AND FUTURE WORK

This paper framed dynamic inventory rebalancing and last-mile fulfillment under disruption as a cooperative MARL problem and trained a CTDE policy with a GNN encoder and PPO against a stochastic disruption generator. The learned policy autonomously pre-positions inventory, activates lateral transshipment, and re-routes fulfillment, moving the research program from prediction toward actual mitigation. On the three-echelon testbed it sustains service levels, shortens recovery time, and reduces total disruption cost relative to base-stock and single-agent baselines.

Future work will (i) extend to multi-commodity and capacity-constrained fleets; (ii) incorporate risk-sensitive objectives via distributional RL [10]; (iii) combine the MARL layer with multi-stage stochastic programming for principled decision support [24]; and (iv) validate the policy against real disruption events through the digital-twin stress-testing pipeline [19].

## REFERENCES

[1] S. D. A. PAULO, "Beyond Forecasting: Deep Reinforcement Learning for Proactive Supply Chain Resilience," *Zenodo (CERN European Organization for Nuclear Research)* , Nov. 2025, doi: 10.5281/zenodo.17644828.

[2] B. T. K. Uyen and B. T. Hieu, "Toward Autonomous Supply Chains: A Deep Reinforcement Learning Framework," *International Journal of Advanced Multidisciplinary Research and Studies* , vol. 6, no. 2, pp. 301–312, Mar. 2026, doi: 10.62225/2583049x.2026.6.2.5959.

[3] M. A. M. A. Mousa, D. van de Berg, N. Kotecha, E. A. del Rio-Chanona, and M. Mowbray, "An analysis of multi-agent reinforcement learning for decentralized inventory control systems," *Computers & Chemical Engineering* , vol. 188, pp. 108783–108783, Jun. 2024, doi: 10.1016/j.compchemeng.2024.108783.

[4] M. Khirwar, K. S. Gurumoorthy, A. A. Jain, and S. Manchenahally, "Cooperative Multi-Agent Reinforcement Learning for Inventory Management," Apr. 18, 2023, *Cornell University* . doi: 10.48550/arxiv.2304.08769.

[5] X. Yang *et al.* , "A Versatile Multi-Agent Reinforcement Learning Benchmark for Inventory Management," Jun. 13, 2023, *Cornell University* . doi: 10.48550/arxiv.2306.07542.

[6] F. Stranieri, F. Stella, and C. Kouki, "Performance of deep reinforcement learning algorithms in two-echelon inventory control systems," *International Journal of Production Research* , vol. 62, no. 17, pp. 6211–6226, Mar. 2024, doi: 10.1080/00207543.2024.2311180.

[7] K. Geevers, L. van Hezewijk, and M. Mes, "Multi-echelon inventory optimization using deep reinforcement learning," *Central European Journal of Operations Research* , vol. 32, no. 3, pp. 653–683, Jul. 2023, doi: 10.1007/s10100-023-00872-2.

[8] N. Kotecha and E. A. del Rio-Chanona, "Leveraging graph neural networks and multi-agent reinforcement learning for inventory control in supply chains," *Computers & Chemical Engineering* , vol. 199, pp. 109111–109111, Apr. 2025, doi: 10.1016/j.compchemeng.2025.109111.

[9] Y. Zhao and C. Hayes, "Hierarchical Multi-Agent Reinforcement Learning for Dynamic Inventory Allocation with Demand Uncertainty," *Multidisciplinary Research in Computing Information Systems* , vol. 5, no. 10, pp. 850–872, Dec. 2025, doi: 10.71465/mrcis153.

[10] G. Wu, M. Á. de C. Servia, and M. Mowbray, "Distributional reinforcement learning for inventory management in multi-echelon supply chains," *Digital Chemical Engineering* , vol. 6, pp. 100073–100073, Dec. 2022, doi: 10.1016/j.dche.2022.100073.

[11] M. Silva, J. P. Pedroso, and A. Viana, "Deep reinforcement learning for stochastic last-mile delivery with crowdshipping," *Econstor (Econstor)* , vol. 12, pp. 100105–100105, Jan. 2023, doi: 10.1016/j.ejtl.2023.100105.

[12] M. Silva and J. P. Pedroso, "Deep Reinforcement Learning for Crowdshipping Last-Mile Delivery with Endogenous Uncertainty," *Mathematics* , vol. 10, no. 20, pp.

3902–3902, Oct. 2022, doi: 10.3390/math10203902.

[13] R. Auad, A. L. Erera, and M. Savelsbergh, "Dynamic Courier Capacity Acquisition in Rapid Delivery Systems: A Deep Q-Learning Approach," *Transportation Science* , vol. 58, no. 1, pp. 67–93, Dec. 2023, doi: 10.1287/trsc.2022.0042.

[14] Y. Liu, L. Tang, Z. He, Y. Zhao, J. Ma, and Z. Duan, "Vehicle Rebalancing Under Supply-Demand Uncertainty: A Robust Multi-Agent Reinforcement Learning Framework," in *Advances in transdisciplinary engineering* , IOS Press, 2025. doi: 10.3233/atde250450.

[15] J. Xi, F. Zhu, P. Ye, Y. Lv, G. Xiong, and F. Wang, "Auxiliary Network Enhanced Hierarchical Graph Reinforcement Learning for Vehicle Repositioning," *IEEE Transactions on Intelligent Transportation Systems* , vol. 25, no. 9, pp. 11563–11575, Apr. 2024, doi: 10.1109/tits.2024.3383720.

[16] Z. Yu and M. Hu, "Deep Reinforcement Learning With Graph Representation for Vehicle Repositioning," *IEEE Transactions on Intelligent Transportation Systems* , vol. 23, no. 8, pp. 13094–13107, Oct. 2021, doi: 10.1109/tits.2021.3119662.

[17] W. J. Tan, W. Cai, and A. N. Zhang, "Structural-aware simulation analysis of supply chain resilience," *International Journal of Production Research* , vol. 58, no. 17, pp. 5175–5195, Dec. 2019, doi: 10.1080/00207543.2019.1705421.

[18] D. Ivanov, B. Sokolov, and A. Dolgui, "The Ripple effect in supply chains: trade-off 'efficiency-flexibility-resilience' in disruption management," *HAL (Le Centre pour la Communication Scientifique Directe)* , vol. 52, no. 7, pp. 2154–2172, Nov. 2013, doi: 10.1080/00207543.2013.858836.

[19] D. Ivanov and A. Dolgui, "Stress testing supply chains and creating viable ecosystems," *Operations Management Research* , vol. 15, pp. 475–486, May 2021, doi: 10.1007/s12063-021-00194-z.

[20] A. Dolgui, D. Ivanov, and B. Sokolov, "Ripple effect in the supply chain: an analysis and recent literature," *International Journal of Production Research* , vol. 56, pp. 414–430, Oct. 2017, doi: 10.1080/00207543.2017.1387680.

[21] M. Fattahi, K. Govindan, and R. Maihami, "Stochastic optimization of disruption-driven supply chain network design with a new resilience metric," *University of Southern Denmark Research Portal (University of Southern Denmark)* , vol. 230, pp. 107755–107755, Apr. 2020, doi: 10.1016/j.ijpe.2020.107755.

[22] K. A. Loganathan and A. Chinnaraju, "AI-Native Supply Chain Resilience: A Multimodal Architecture for Predictive Intelligence, Optimization, and Real-Time Decision-Making," *GSC Advanced Research and Reviews* , vol. 25, no. 3, pp. 78–126, Dec. 2025, doi: 10.30574/gscarr.2025.25.3.0372.

[23] M. Luo *et al.* , "Fleet Rebalancing for Expanding Shared e-Mobility Systems: A Multi-Agent Deep Reinforcement Learning Approach," *Warwick Research Archive Portal (University of Warwick)* , vol. 24, no. 4, pp. 3868–3881, Jan. 2023, doi: 10.1109/tits.2022.3233422.

[24] F. Stranieri, E. Fadda, and F. Stella, "Combining deep reinforcement learning and multi-stage stochastic programming to address the supply chain inventory management problem," *BOA (University of Milano-Bicocca)* , vol. 268, pp. 109099–109099, Nov. 2023, doi: 10.1016/j.ijpe.2023.109099.