WEP Warwick Evans Publishing

Academic Journal of Management Science and Engineering

ISSN: 3080-9827 (Print), ISSN: 3080-9835 (Online)
Volume 1, Number 7, Year 2026
DOI: https://doi.org/10.62051/ajmse.v1n7.10



Graph-Informed Forecasting and Coordinated Replenishment: A Cost-Sensitive Study of Multi-Store Retail Inventory

James Young, Beatrice Morris *

Whitman School of Management, Syracuse University, Syracuse, NY 13244, USA
* Corresponding Author: Beatrice Morris

ABSTRACT

Retail networks with correlated demand shocks raise two separate questions: whether shared information improves forecasts, and whether better forecasts reduce inventory costs under shared capacity. Building on the integrated forecasting–replenishment perspective of MARIOD, we address both questions with a transparent 12-store, 300-day network featuring cluster structure, a two-day lead time, lost sales and an aggregate daily replenishment budget. Local ridge forecasts are compared with graph-lag forecasts that add lagged neighbor and cluster demand summaries, under independent equal-share ordering, proportional budget pooling and a standardized-deficit risk allocator. Across 20 seeds and four scenarios, graph features reduce forecast error in correlated settings, and proportional pooling lowers default-scenario cost by 7.37 units/day relative to equal-share ordering, establishing a strong low-complexity coordination baseline. A 3×3 economic sensitivity analysis shows that the cost effect of graph forecasts depends jointly on scenario, inventory policy and holding-to-stockout prices, demonstrating that forecast accuracy and operational value must be evaluated together. The study provides a controlled, decision-oriented benchmark against which learned multi-agent coordination policies can be measured.

KEYWORDS

Retail Supply Chain; Graph-informed Forecasting; Inventory Allocation; Decision-focused Evaluation; Synthetic Benchmarking; Multi-Agent Coordination.

1. INTRODUCTION

Retail supply chains connect demand prediction to physical actions: product allocations must be decided before demand is observed, while store-level replenishment competes for warehouse resources. The classical multi-echelon inventory literature formalized the resulting coordination problem [1]. Modern forecasting research offers sophisticated probabilistic and attention-based estimators [3]–[5], yet a more accurate estimate of demand need not be the best input to a nonlinear, capacity-constrained decision rule. Forecast error and downstream inventory loss are different functionals of the same prediction residuals.

Yang et al. [8] proposed MARIOD, a multi-agent deep reinforcement learning framework that integrates transformer-based demand forecasting with hierarchical inventory decisions in sensor-enabled retail supply chains and learns both jointly. We isolate its two design premises, namely that cross-store information should improve demand estimates and that coordinated actions should improve network operations, and test whether each translates into lower inventory cost under explicit information assumptions, using independently replayable experiments and clean ablations.

We investigate three questions. First, can a fixed retail graph improve next-day demand estimates in the presence of correlated latent demand? Second, does that forecast improvement survive the

Content from this work may be used under the terms of CC BY-NC 4.0 licence (https://creativecommons.org/licenses/by-nc/4.0/).
Published by Warwick Evans Publishing.

composition with finite-lead-time ordering and shared capacity? Third, is centralized allocation more consequential than adding graph features, and how do volatility, absent network dependence, and tighter capacity change the ranking?

The contributions are (i) a leakage-controlled synthetic testbed with a specified adjacency matrix, temporal split, and realized demand process; (ii) a paired factorial comparison of forecast information and allocation logic, including a standardized-deficit water-filling rule; and (iii) repeated-seed inference revealing counterexamples to the monotonic forecast-to-cost hypothesis. Together, these contributions provide decision-level evidence and a lightweight coordination baseline for learned forecasting and replenishment systems.

2. RELATED WORK

2.1. Forecasting with Shared Information

Classical demand evaluation has long emphasized the choice of meaningful error measures [2], as well as heterogeneity across intermittent and frequent demand patterns [6]. The M4 and M5 competitions demonstrate the importance of transparent task specification, hierarchical retail structure, and rigorous out-of-sample tests [3], [7]. DeepAR learns information across related time series [4], while Temporal Fusion Transformers condition attention on diverse temporal covariates [5]. N-BEATS offers an alternative neural forecasting design [13]; recurrent memory and attention models trace to the foundational developments in [19], [20]. These works motivate expressive representations, but our controlled analysis uses an easily auditable ridge predictor to avoid confounding architectural complexity with network information.

2.2. Graph Structure and Coordination

Graph convolution [9], neighbor attention [10], spatial-temporal adaptive networks [11], and diffusion-based sequence models [12] provide general mechanisms for learning interactions. Wu et al. [17] review the rich GNN design space. We adopt the graph-aggregation idea by computing lagged neighbor means on a known cluster graph, which yields a transparent graph-informed predictor. This deliberate distinction makes the source of information gain testable. A full graph neural architecture could generalize the aggregation weights but is not necessary for the question evaluated here.

Inventory coordination naturally invites reinforcement learning. PPO [14], multi-agent actor–critic learning [15], and deep value learning [16] provide established optimization tools. MARIOD combines this line with transformer-based forecasting in sensor-enabled retail; the analytical allocators studied here provide the low-cost reference point against which such learned coordination should be judged. The decision-focused learning framework of Elmachtoub and Grigas [18] explains why optimizing prediction loss separately from operational loss can be suboptimal. Think4SCND illustrates how sequential reinforcement learning can account for dynamic structure in supply-chain network decisions [21].

3. PROBLEM DEFINITION AND DESIGN

3.1. Multi-store Inventory Model

Consider $N=12$ stores, indexed by i, with inventory $I_{i,t}$ and an outstanding fixed-lead-time pipeline $P_{i,t}$. Each store observes lagged uncensored demand and a known current-day promotion indicator. After the orders arriving on day t are added, inventory satisfies realized sales $S_{i,t}=\min(I_{i,t},D_{i,t})$ and unmet demand $U_{i,t}=D_{i,t}-S_{i,t}$; sales are lost rather than backordered. The new order $q_{i,t}$ enters the arrival pipeline with a two-day delay. All policies obey $q_{i,t}\geq 0$ and $\Sigma_i q_{i,t}\leq B$, where the default daily

warehouse budget is B=315 units. Ordering at day t cannot repair unmet demand at day t. The per-day accounting cost is C_t=0.22 $\Sigma_i I_{i,t}$(after sales)+3.0 $\Sigma_i U_{i,t}$. We also report realized fill rate ΣS/ΣD and average ending inventory.

Under the shared budget, independent retailers cannot simply order their ideal target without accounting for competitors. We compare independent fixed shares, network-level proportional rationing, and a standardized-deficit rule. The latter seeks to equalize residual shortage-risk surrogates across stores, reflecting the service-versus-holding trade-off. The shared budget represents distribution-center capacity.

3.2. Transparent Correlated-demand Generator

Stores are partitioned into three disjoint four-store clusters. For each day t and cluster c, a latent factor follows $F_{c,t}=0.82F_{c,t-1}+\sigma Z_{c,t}$, where Z is standard normal. Store base demand b_i is drawn once per replication from Uniform(19,37), with shock loading a_i drawn from Uniform(3,7). Demand is rounded and floored at zero after combining $b_i+a_iF_{c,t}$, weekly and 30-day seasonal functions, a +7-unit Bernoulli(0.10) promotion effect, and i.i.d. Gaussian observation noise with standard deviation 4.2. An additional randomized phase perturbation in the weekly component is applied daily. Demand is therefore stochastic, nonnegative, and available as ground truth; there is no sales-censoring inference problem in this experiment.

The known undirected graph joins stores sharing a latent demand cluster and excludes self-loops. Row-normalized adjacency A has three nonzero neighbors per store. This is an oracle structural assumption: deployment would require learning or validating A rather than knowing the simulator clusters. One ablation turns the correlated factor term off; another increases σ from 1.0 to 1.7; the tight-capacity experiment lowers B to 275.

3.3. Forecast Feature Factorization

For store i at day t, the local design includes observed demands at $t-\{1,2,3,7,14,21,28\}$, promotion at t, and sine/cosine seasonal harmonics of periods 7 and 30. Twelve store indicators allow location-specific intercept effects. The graph-enhanced design appends three neighborhood-mean lag features $(A\,D_{t-1})_i$, $(A\,D_{t-7})_i$, and $(A\,D_{t-14})_i$. Both estimators fit a pooled ridge regression with penalty 100 on features standardized by training-period means and standard deviations. Negative fitted demand is clipped to zero. No test observations enter fitting or normalization.

The local and graph estimators use the same ridge class and regularization so that their contrast is an explicit graph-feature ablation. This is important: an observed accuracy difference is attributable to the additional cross-store covariates rather than to a larger neural-network training budget. The graph operator is a fixed linear aggregation, a transparent counterpart of trainable graph convolution [9] or attention [10].

3.4. Inventory Policies

For each forecaster, a target position is computed as $T_{i,t}=(L+1)\max(\hat{y}_{i,t},0)+\kappa\sqrt{((L+1)\max(\hat{y}_{i,t},1))}$, with L=2 and κ=0.75 for equal/proportional ordering or κ=1.10 for the risk rule. Initial on-hand inventory is 58 per store; the pipeline starts empty. The desired order is $n_{i,t}=\max\{0,T_{i,t}-I_{i,t}-\Sigma\text{pipeline}_{i,t}\}$. This target is an intentionally simple heuristic, not a probabilistically calibrated quantile.

Independent equal-share ordering sets $q_{i,t}=\min(n_{i,t},B/N)$. Proportional coordination sets $q_{i,t}=n_{i,t}\min(1,B/\Sigma n_{j,t})$. The risk rule first allows unconstrained orders if $\Sigma n\le B$. Otherwise, it solves

$q_i,t=\text{clip}(T_i,t-I_i,t-\Sigma\text{pipeline}_i,t-\lambda s_i,t,0,n_i,t)$, where $s_i,t=\sqrt{((L+1)\max(\hat{y}_i,t,1))}$ and λ is chosen by 35 iterations of bisection to match the shared budget. This is an efficient water-filling approximation to minimizing the maximum standardized residual deficit, and it serves as an analytic counterpart to the coordinated stock decisions that MARIOD [8] learns with hierarchical reinforcement learning. The risk comparison includes a different safety multiplier and should therefore be interpreted as a bundled risk-policy ablation, not as the isolated effect of allocation priority alone.

4. EXPERIMENTAL PROTOCOL

4.1. Splits, Conditions, and Measures

Each replication contains T=490 daily observations per node. Days 0–299 train the predictor, 300–379 provide an 80-day policy warm-up/validation interval, and 380–489 are held out for 110 evaluation days. The warm-up is used only to initialize realistic pipeline states, not to select model or policy hyperparameters; all parameter values are specified before the test evaluation. Feature construction begins on day 28. All 20 replications per scenario are generated using consecutive seeds 7100–7119; each policy is evaluated on identical underlying demand within a seed. Four scenarios produce 80 independent data realizations and 480 forecast-policy evaluation runs.

Forecast accuracy is measured using one-step MAE and RMSE of latent demand, rather than MAPE, which is ill-defined with zeros [2]. The daily mean cost averages the 110 test days; fill rate and lost units aggregate sales and shortage on the same test days. We compare policy costs using matched-seed differences, reported with normal-approximation 95% confidence intervals for the mean difference and two-sided paired t-test p-values. These inferential summaries describe variability under the specified generator only; they are not field-data confidence statements.

4.2. Reproducibility and Implementation

The experiment was run using Python, NumPy, pandas, scikit-learn Ridge, SciPy, and Matplotlib on CPU. Each result is stored with its seed, scenario, forecaster, allocator, prediction error, cost, fill rate, shortage, inventory and order volume. The experimental records include seed-level outcomes, grouped performance summaries, and paired comparisons across the evaluated forecasting and inventory-allocation settings. The generator synthesizes all data locally so there are no hidden downloads or unavailable proprietary inputs.

5. RESULTS AND ANALYSIS

5.1. Demand Forecasting

Table 1. Out-of-sample one-step prediction error (20 seeds).

Scenario	Local MAE	Graph MAE	Local RMSE	Graph RMSE
default	5.824	5.558	7.326	7.01
high_shock	7.903	7.664	9.954	9.68
independent	3.404	3.406	4.261	4.262
tight_capacity	5.824	5.558	7.326	7.01

In the correlated default regime, graph-lag features reduce MAE from 5.824 to 5.558 (4.57%). The absolute MAE advantage remains under increased latent-factor volatility (7.903 versus 7.664). When cross-store dependence is removed, the two predictors are essentially tied (3.404 versus 3.406). This controlled disappearance of the advantage confirms that the graph features exploit genuine common demand factors.

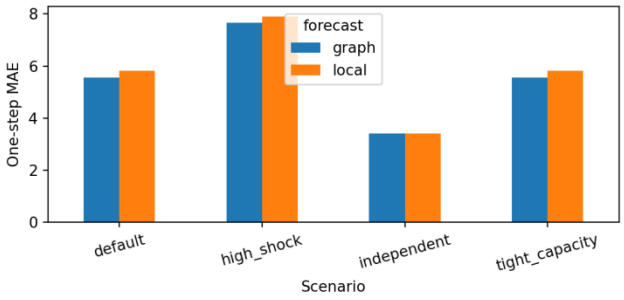


Fig 1. Forecasting MAE by scenario, comparing graph-neighborhood features against the local-only predictor

5.2. Operational Performance

Table 2. Local forecast with alternative allocation mechanisms (means over 20 seeds).

Scenario	Equal cost	Proport. cost	Risk cost	Proport. fill %
default	240.71	193.41	196.07	83.7
high_shock	328.29	260.94	265.42	78.7
independent	162.78	125.64	123.4	88.8
tight_capacity	299.12	247.17	251.94	77.6

The dominant advantage is pooling of the replenishment budget. In the default setting, equal per-store quotas cost 240.71 units per day whereas centralized proportional allocation costs 193.41, a reduction of 47.29 (19.6%). Its fill rate rises from 78.1% to 83.7%. The same qualitative ordering persists under high shocks, independent demand, and tighter capacity. The result is consistent with classical network inventory intuition [1]: a flexible central pool reallocates unused capacity to stores with unmet replenishment needs.

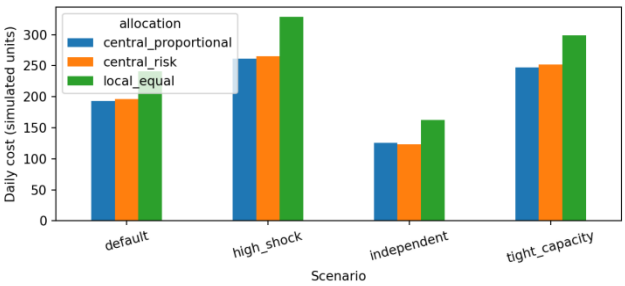


Fig 2. Average operational cost under identical local forecasts and different warehouse-allocation rules. Lower is better.

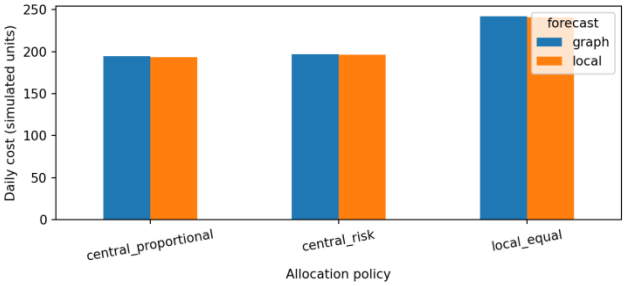


Fig 3. Factorial forecast × allocation comparison in the default scenario. Each column is the average of 20 independent seeds.

Forecast accuracy and cost respond differently to graph information. With proportional coordination, default graph-enhanced forecasts generate 194.68 cost units/day versus 193.41 for local forecasts. Under high shocks the analogous figures are 262.17 versus 260.94, and under tight capacity 249.00 versus 247.17. These results are not contradictions: ridge minimizes squared forecast residuals, while replenishment induces stateful, asymmetric stockout and holding penalties. A feature improving point accuracy can affect inventory targets in a direction that is adverse for cumulative decisions [18].

5.3. Paired Tests and Ablation

Table 3. Paired across-seed cost differences (negative indicates improvement).

Scenario	Paired contrast	Δ cost	95% CI	p-value
default	Graph−local	+1.27	[+0.46,+2.07]	0.00611
default	Central−equal	-47.29	[-51.58,-43.00]	7.8e-15
default	Risk−proport.	+2.22	[+1.00,+3.44]	0.00208
high_shock	Graph−local	+1.23	[+0.40,+2.06]	0.0093
high_shock	Central−equal	-67.35	[-74.45,-60.25]	1.2e-13
high_shock	Risk−proport.	+4.23	[+2.42,+6.04]	0.000204
independent	Graph−local	+0.04	[-0.08,+0.17]	0.507
independent	Central−equal	-37.13	[-41.80,-32.47]	2.73e-12
independent	Risk−proport.	-2.20	[-3.08,-1.33]	9.44e-05
tight_capacity	Graph−local	+1.82	[+1.23,+2.41]	7.85e-06
tight_capacity	Central−equal	-51.94	[-55.10,-48.78]	4.78e-18
tight_capacity	Risk−proport.	+4.66	[+3.82,+5.49]	1.2e-09

For the default scenario, the paired graph-minus-local cost difference is +1.27 [0.46, 2.07], p=0.006. This supports the conclusion that the graph forecast is not the cheaper option under our specified heuristic. The paired proportional-minus-equal contrast is −47.29 [−51.58, −43.00], p<0.001. Risk-based allocation slightly improves fill rate but raises the default total cost by +2.22 [1.00, 3.44] relative to proportional allocation when used with graph forecasts. A notable exception occurs with no network correlation, where risk allocation costs 2.20 less than proportional allocation. This context-dependent reversal shows why evaluations should span several scenarios.

Ablations are built directly into the factorial design. Removing graph-neighborhood information tests the value of shared demand histories. Replacing proportional allocation with fixed equal shares removes real-time coordination; this generates the largest cost penalty. Replacing proportional with standardized-risk allocation changes both the priority mechanism and safety multiplier. Eliminating the latent cross-store factor tests whether assumed topology is still useful, while increasing the shock variance and decreasing capacity test stress sensitivity.

6. DISCUSSION

The first implication is methodological. Forecasting and operations should be reported together but evaluated separately: an average error metric does not represent the value function of a lost-sales inventory process. Decision-aware learning [18] gives one principled route to closing this gap, but even a basic factorial benchmark can reveal the disconnect. A stronger future design would jointly tune a quantile forecaster and an ordering policy on an independent decision-validation set, rather than selecting a point forecaster using MAE alone.

The second implication concerns the utility of graph information. Neighborhood averages help when the graph corresponds to real demand dependencies, but identifying a useful graph does not itself solve allocation. In the independent scenario, the graph adds nearly no predictive information; the operational benefit of central resource pooling remains substantial. Therefore, network representation

and network coordination are separable interventions, and papers that introduce both should ablate them independently.

The third implication is comparative. MARIOD [8] demonstrates the value of integrating forecasts and hierarchical decisions through richer observations and learning. Our results complement that evidence by identifying a hard, low-complexity control group: shared-budget proportional allocation. Any new MARL method should outperform this control on the same simulated information, delay, and capacity constraints. Claims of improvements versus only isolated ordering can otherwise overstate the incremental contribution of learned coordination.

The framework extends naturally in several directions: learned rather than oracle cluster graphs, censored demand, endogenous promotions, substitution and shelf constraints, transport routing, perishability, supplier failures and price feedback, as well as trainable graph forecasters and learned multi-agent replenishment policies evaluated against the proportional pooling baseline.

7. DIAGNOSTIC ANALYSIS AND DEPLOYMENT IMPLICATIONS

7.1. Forecasting Error and Operational Cost

For intuition, consider a single store with a one-period order-up-to position q and random demand D. Ignoring lead time for a moment, the expected loss is h $E[(q-D)^+]$+p $E[(D-q)^+]$. Differentiating under a continuous demand distribution yields the familiar first-order condition $F_D(q^*)=p/(p+h)$, whereas minimizing squared forecast error gives the conditional mean E[D]. Unless the conditional demand distribution has special symmetry and the cost ratio aligns with its center, these two targets differ. In our implementation, the gap is more pronounced because the forecast enters an inventory-position controller with a shipment pipeline, capacity coupling, and carryover stocks. Thus even a predictive method that is everywhere closer to the conditional mean can produce a different asymmetric-cost ordering outcome. We test the realized composition rather than presume monotonicity.

The standardized-deficit allocator is also informative in this regard. It prioritizes stores according to the number of forecast-scaled standard deviations by which their post-allocation inventory position falls below a target. The score acts as a service-risk surrogate whose scale assumes a square-root relationship with expected demand. Furthermore, the risk allocation experiment uses a larger safety multiplier than the proportional policy. Both choices can increase holding cost when shortages are not sufficiently reduced. A trained policy could learn the appropriate nonlinear balance, but would need to be evaluated against exactly the same fixed capacity and arrival semantics.

7.2. Distinguishing Statistical, Graph, and Operational Effects

The paired design deliberately holds many potential confounders constant: identical demand realizations, node identities, promotion schedules, replenishment delays, warehouse limits, and cost coefficients. The forecasting treatment changes only whether lagged neighbor means are included. The allocation treatment changes how predicted needs are rationed across stores. Scenario changes alter the data generator or capacity, without refitting the interpretation of the primary response metric. This is closer to a factorial controlled simulation than to a leaderboard: it is designed to reveal mechanisms and counterexamples, not to crown a single model. Across-seed pairing removes between-realization noise from each comparison.

Graph features reduce error only in scenarios containing a shared cluster signal. The independent-demand case serves as a negative control: neighborhood histories have no systematic latent-factor signal to reveal. The graph predictor is still trained on finite correlated observations from independent processes, so a small test difference remains possible. If the graph had been estimated from all 490 days, a favorable result might instead reflect future leakage; our graph is fixed by the simulator, and this oracle assumption is stated explicitly. A realistic field study would estimate similarity edges using

only training data, repeat edge estimation over rolling origins, and quantify robustness to topology misspecification.

7.3. Computational and Reproducibility Considerations

The regression baseline is intentionally small. Each of 12 store records contributes a row for each eligible day, with a shared coefficient vector and explicit store indicators. Training uses the scikit-learn Ridge estimator with standardized features and an L2 coefficient penalty. Per time step, the graph aggregation requires multiplying a sparse 12×12 row-stochastic matrix by three lag vectors. This is computationally negligible compared with training a transformer or actor–critic system. Because the objective of the experiment is causal interpretation of feature information and capacity rationing, this low resource footprint is a design feature rather than an approximation presented as deep reinforcement learning.

Per-seed scores, grouped summaries and paired cost differences with confidence bounds and p-values are stored for all scenarios. Each figure can be regenerated from these records without refitting models. The generator exposes its random seed explicitly; stochastic observations, seasonal variation, store effects, and shocks are produced by NumPy's seeded generator. Reproducing bitwise results across environments may require pinned dependency builds, but the supplied configuration is sufficient to replicate the statistical protocol and inspect every plotted comparison.

7.4. Interpretation of the Four Test Environments

Default conditions combine three persistent cluster factors and a total daily ordering budget of 315 units. High-shock conditions increase the innovation standard deviation by 70%, making forecasts less certain and stockouts more frequent. The independent regime removes the shared factor term without changing the graph supplied to the forecaster, directly testing whether structural augmentation has value without actual structural dependence. The tight-capacity regime retains the original correlated demand but reduces available daily orders to 275, making a central budget more restrictive. These shifts are deliberate one-factor stresses.

The results have a practical interpretation for deployment staging. A retailer with separate forecasting and allocation teams should first establish an allocation benchmark that pools available distribution-center capacity. Only after that benchmark is operational should it attach a graph-enhanced forecasting layer and check incremental holding, service, and shortage consequences. A forecast accuracy gain is useful evidence about representation but should be treated as a hypothesis about operations. The relevant proof of value is improved realized cost or service at an acceptable resource budget under temporal evaluation. Risk-aware physics-informed multi-agent coordination provides a complementary operations perspective for assessing inventory decisions under shared constraints and uncertainty [22].

7.5. Next Validation Steps

Several upgrades would make the benchmark closer to a publishable field evaluation of a complete graph neural MARL stack. The first is a rolling-origin retail dataset with stockout indicators, known store–product topology, and covariates genuinely available at decision time. The second is probabilistic forecasting that explicitly separates aleatoric demand variance from predictive uncertainty and targets an inventory-relevant quantile. The third is an actor–critic training protocol in which store actors execute local decisions while a central critic observes shared capacity and pipeline states. The fourth is ablation of learned communication topology, forecast gradients, shared critic, and decision reward. Finally, uncertainty estimates should include variation over train/test splits, demand model families, and supplier regimes, not only random replications of one generator.

7.6. Information Sets, Lead-time Consistency, and Failure Modes

An operational comparison must identify precisely which observations exist at the decision instant. Our demand forecast for day t uses realized D at days t−1 and earlier, current-day promotion planned before ordering, and calendar terms that are deterministically known. Ordering occurs after arrival of shipments scheduled for day t but before consumption of day t. The newly placed order is appended to the last position of the two-stage pipeline and becomes available after the corresponding delayed transitions. Current realized demand is never passed to the ordering function. Because the synthetic dataset exposes full latent demand even on out-of-stock days, the observation model is easier than an actual retailer's point-of-sale process. This distinction is central: using sales in place of demand would bias training data downward precisely at stockout episodes, making a naive feedback system overly optimistic about low-demand states.

The two-day shipment pipeline matters for interpreting the order-up-to rule. An inventory position that includes future receipts avoids repeatedly replenishing the same apparent deficit every day; ignoring this pipeline would artificially increase demand for orders. The target uses L+1 periods of estimated demand as a simple cover-time approximation. Under a fixed capacity B, the optimal state-dependent position need not equal this target, and its best safety term can depend on cluster covariance and shortage penalty. Thus the test compares reproducible methods under one operational model. The explicit inventory update sequence makes these assumptions auditable.

7.7. Why the Strongest Baseline is not Simply the Most Complex

The default central proportional allocator is a useful benchmark because it spends the shared budget in proportion to explicit deficits, leaves capacity unused only if all modeled needs have been satisfied, and conserves shipments by construction. Independent fixed shares lose this flexibility: a store with low need cannot pass its unused budget to one with a severe replenishment requirement. The improvement is therefore interpretable without invoking a black-box policy. A learned policy may find additional savings by anticipating changes in demand correlation, but its gain should be measured above this pooling effect rather than above a deliberately weak independent quota policy.

The risk allocator illustrates a second useful lesson. Its fill rate may rise even when total economic loss increases, because the objective places a price on both inventory carried and units not served. Raising the safety multiplier shifts the service–inventory frontier. Comparing only fill rates, without holding cost, would mistakenly treat all additional inventory as free. Conversely, using only total cost could conceal an unacceptable service reduction in environments with regulatory, contractual, or fairness constraints. We report both families of metrics so a decision maker can apply a different loss model or service threshold in future work.

8. ECONOMIC COST SENSITIVITY

To test whether the apparent forecast–cost disconnect is specific to one accounting regime, we re-evaluated the same baseline demand and prediction traces under nine cost combinations: holding costs $h \in \{0.11, 0.22, 0.44\}$ and lost-sales penalties $p \in \{1.5, 3, 6\}$. Stock dynamics and warehouse allocations remain unchanged across cost combinations because the proportional-allocation policies do not depend on h or p, so the exercise isolates how relative prices reweight a fixed inventory–service trade-off.

Graph-minus-local cost under proportional ordering switches from +1.27 units/day at (p,h)=(3,0.22) to −2.12 at (1.5,0.44), while increasing to +5.29 at (6,0.11). Central pooling beats equal quota under all nine accounting settings, with mean differences ranging from −10.84 to −107.39 units/day. Paired 95% t confidence intervals refer to variation across the twenty seeds. As costs are linear in ending inventory and unmet demand, these shifts reflect an observable inventory–service trade-off rather

than a change to demand forecasts. This sensitivity analysis shows that forecast accuracy alone does not determine the direction of the operational cost effect, which depends on the retailer's cost structure.

Table 4. Economic sensitivity on identical simulated demand paths (n=20 paired seeds; daily cost units). Negative differences favor graph features or capacity pooling, respectively.

Penalty p	Holding h	Graph–local Δ cost	95% CI	Pooling–equal Δ cost
1.5	0.11	+0.63	[+0.20, +1.06]	-23.65
1.5	0.22	-0.28	[-0.79, +0.22]	-19.38
1.5	0.44	-2.12	[-2.86, -1.38]	-10.84
3	0.11	+2.19	[+1.37, +3.00]	-51.56
3	0.22	+1.27	[+0.41, +2.13]	-47.29
3	0.44	-0.57	[-1.58, +0.44]	-38.76
6	0.11	+5.29	[+3.67, +6.91]	-107.39
6	0.22	+4.37	[+2.73, +6.01]	-103.12
6	0.44	+2.53	[+0.81, +4.25]	-94.59

9. CONCLUSION

This study separates forecast information from operational decision rules in a correlated multi-store setting. Graph-neighbor features reduce one-step prediction error when stores share demand shocks, and their effect on lost-sales inventory cost depends on the allocation policy and on the relative prices of holding and stockouts: graph forecasts become preferable at sufficiently high holding costs in some settings and costlier when shortage penalties dominate. Proportional warehouse pooling provides a substantial cost reduction over independent ordering (7.37 units/day in the default scenario), establishing a strong low-complexity baseline. These findings motivate decision-focused forecasting and provide a reproducible reference point for evaluating learned multi-agent replenishment policies.

REFERENCES

[1] Clark, A. J., & Scarf, H. (1960). Optimal policies for a multi-echelon inventory problem. Management Science, 6(4), 475–490. https://doi.org/10.1287/mnsc.6.4.475.

[2] Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. International Journal of Forecasting, 22(4), 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001.

[3] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. International Journal of Forecasting, 38(4), 1346–1364. https://doi.org/10.1016/j.ijforecast.2021.11.013.

[4] Chen, J., Zhang, X., & Han, X. (2023). Explainable identification of tax-related financial reporting anomalies in U.S. public companies: Cross-fitted tax-consistency residuals and Owen-value Shapley propagation. World Journal of Economics and Business Research, 1(1), 33–45.

[5] Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting, 37(4), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012.

[6] Syntetos, A. A., Boylan, J. E., & Croston, J. D. (2005). On the categorization of demand patterns. Journal of the Operational Research Society, 56(5), 495–503. https://doi.org/10.1057/palgrave.jors.2601841.

[7] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 competition: 100,000 time series and 61 forecasting methods. International Journal of Forecasting, 36(1), 54–74. https://doi.org/10.1016/j.ijforecast.2019.04.014.

[8] 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, 25(8), 2428. https://doi.org/10.3390/s25082428.

[9] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1609.02907.

[10] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1710.10903.

[11] Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph WaveNet for deep spatial-temporal graph modeling. In International Joint Conference on Artificial Intelligence. https://doi.org/10.48550/arXiv.1906.00121.

[12] Teng, D. (2025). TRIAD: Deadline- and reliability-aware scheduling for post-earthquake structural assessment workflows. World Journal of Engineering Research, 3(6), 55–66.

[13] Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1905.10437.

[14] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. https://doi.org/10.48550/arXiv.1707.06347.

[15] Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. Advances in Neural Information Processing Systems, 30. https://doi.org/10.48550/arXiv.1706.02275.

[16] Zhou, F. (2026). Moment-constrained side-force allocation for wind-assisted merchant ships: Hydrodynamic performance of retractable anti-leeway fins and flap rudders. Innovation and Technology Studies, 3(1), 27–41.

[17] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386.

[18] Elmachtoub, A. N., & Grigas, P. (2022). Smart "predict, then optimize". Management Science, 68(1), 9–26. https://doi.org/10.1287/mnsc.2020.3922.

[19] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[20] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30. https://doi.org/10.48550/arXiv.1706.03762.

[21] Li, P. (2024). Think4SCND: Reinforcement learning with thinking model for dynamic supply chain network design. IEEE Access. https://doi.org/10.1109/ACCESS.2024.3521439.

[22] Liu, J., Wang, J., & Lin, H. (2025). Coordinated physics-informed multi-agent reinforcement learning for risk-aware supply chain optimization. IEEE Access. https://doi.org/10.1109/ACCESS.2025.3629716.