IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 14, April 2026
Impact Factor: 8.2

Hierarchical Reinforcement Learning for Retail Inventory Optimization: Integrating Deep Neural Network Demand Forecasting with Dueling Double Deep Q-Networks

Sri.Madamanchi Brahmaiah[1], Rajarapu Lakshmana Brahmam[2], Videla Mohnish[3], Padigala Prasanth Kumar[4]
Department of Computer Science[1-4]
RVR & JC College of Engineering, Chowdavaram, Guntur, India

Abstract: *Multi-product retail inventory management presents a complex optimization challenge involving volatile demand, multi-period pricing decisions, and heterogeneous strategic customer behavior. This paper presents a hierarchical framework combining a three-model Deep Neural Network (DNN) demand forecasting system with a Dueling Double Deep Q-Network (DD3QN) for simultaneous optimization of ordering quantities, full prices, and discount strategies across five substitutable seasonal products over a two-period selling horizon. The demand system decomposes consumer decision-making into temporal purchase-period selection and conditional product selection, explicitly modeling willingness-to-pay (WTP) as a latent variable driving three behavioral customer segments — Premium, Regular, and Budget — each characterized by distinct price elasticity and strategic purchasing behavior. The reinforcement learning agent operates over a 648-dimensional discrete action space (9 order quantities × 9 prices × 8 discount levels) and is evaluated under non-stationary market conditions with ±20% demand and competition volatility. Experimental results on a 12,000-sample synthetic clothing retail dataset demonstrate convergence within 400 training episodes, a mean test reward of 13.97 (scaled), and 15–22% profit improvement over fixed-strategy baselines with less than 5% performance degradation under market volatility. The proposed framework contributes novel hierarchical demand decomposition, dueling network architectures for large discrete retail action spaces, and robust reinforcement learning deployment under realistic market uncertainty.*

Keywords: Deep Reinforcement Learning, Dynamic Pricing, Inventory Optimization, Demand Forecasting, Dueling DQN, Customer Segmentation, Willingness-to-Pay, Strategic Customers, Retail Analytics, Seasonal Products

I. INTRODUCTION

Retail inventory management for seasonal and fashion goods is one of the most demanding operational challenges in modern supply chain management. Unlike commodity products with stable demand patterns, seasonal products face highly volatile demand, limited shelf lives, and consumers who adapt their purchasing behavior strategically in anticipation of price trajectories. A retailer must simultaneously determine how much to order, at what initial price to offer goods, and when and how deeply to discount — all before observing realized demand.

The revenue management literature has long recognized the tension between maximizing immediate revenue through premium pricing and stimulating demand through discounts. Strategic customers — those who rationally weigh the utility of purchasing immediately versus deferring to a future sale — create a particularly challenging feedback loop:

DOI: 10.48175/IJARSCT-34086

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 14, April 2026
Impact Factor: 8.2

aggressive discounting trains customers to wait, while refusing to discount risks unsold inventory at zero salvage value at season end [2][3].

Classical operations research approaches model this problem through dynamic programming (DP), assuming known or parametrically specified demand distributions. These methods break down in real retail settings where demand is nonlinear, high-dimensional, and driven by complex consumer heterogeneity that defies simple parameterization.

Deep reinforcement learning (RL) offers a compelling alternative by learning optimal policies directly from data without requiring explicit demand models [11].

Famil Alamdar and Seifi [1] pioneered deep Q-learning (DQL) for simultaneous pricing and ordering across multiple seasonal products with strategic customers. However, their approach employs a standard DQL architecture that does not exploit the structural decomposability of the retail problem — specifically, that many states are inherently unfavorable regardless of action choice, and that state value is separable from relative action preference. This motivates the dueling network architecture [9].

This paper closes four open limitations of [1] through targeted architectural and modeling innovations:

(1) Dueling Double DQN (DD3QN): A unified agent employing a dueling architecture separately estimating state-value V(s) and action-advantage A(s,a), combined with double DQN updates to mitigate Q-value overestimation. This reduces model complexity from four networks to two while accelerating convergence by ~50%.

(2) Explicit WTP-Driven Segmentation: Willingness-to-pay is modeled as a latent variable governing three explicit behavioral segments — Premium ($70–$100 WTP), Regular ($40–$60), and Budget ($20–$40) — each exhibiting distinct strategic waiting behavior.

(3) Non-Stationary Market Conditions: Market demand and competition intensity are sampled per episode from Uniform[0.8,1.2] and Uniform[0.9,1.1], simulating ±20% real-market volatility absent from prior work.

(4) Period-Dependent Reward Design: Period 2 holding costs are value-proportional (10% of full price per remaining unit) to reflect increasing obsolescence risk as the season closes.

II. LITERATURE REVIEW

A. Classical Inventory Optimization

The Economic Order Quantity (EOQ) model provides closed-form solutions for single-product stationary demand. Extensions include the Newsvendor model for single-period stochastic demand, (s,S) replenishment policies, and base-stock policies for serial supply chains [16]. For seasonal goods, the two-period markdown pricing problem has been extensively studied. Elmaghraby and Keskinocak [15] comprehensively review dynamic pricing under inventory considerations. The central tension — that markdown discounts increase Period 2 demand but train customers to defer Period 1 purchases — requires explicit strategic customer modeling to resolve optimally.

B. Strategic Customer Behavior

Su [2] provided foundational analysis of intertemporal pricing when customers are forward-looking. Du et al. [3] extended this to joint pricing and inventory decisions with decreasing product values. Wu et al. [12] incorporated reference price effects. A critical limitation shared across this literature is the assumption of closed-form demand functions — linear, logistic, or exponential — that permit analytical tractability but fail in high-dimensional, nonlinear real retail settings.

C. Machine Learning for Demand Forecasting

Neural networks consistently outperform parametric models in retail demand forecasting. Shakya et al. [4] demonstrated superiority over evolutionary optimizers under dynamic pricing. Zhang et al. [5] proposed a customized DNN mimicking Multinomial Logit (MNL) utility functions through partial input connections, which inspired the demand architecture in [1]. The DNN demand system in this paper extends this direction with full batch normalization, dropout regularization, and hierarchical decomposition of the consumer decision process into sequential stages.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 14, April 2026 Impact Factor: 8.2

D. Reinforcement Learning for Pricing and Inventory

Kutschinski et al. [14] showed Q-learning outperforms derivative-following strategies in competitive single-product pricing. Subsequent extensions include multi-product Q-learning for perishable interdependent goods [6], multi- agent RL for product cluster pricing [13], deep RL for joint pricing-inventory control [7], and reference price effects with double DQN [8]. The foundational work of Mnih et al. [11] established DQN with experience replay and target networks; Van Hasselt et al. [10] introduced double DQN to eliminate maximization bias; Wang et al. [9] demonstrated that dueling architectures substantially accelerate learning in environments where state value is largely action-independent.

E. Research Gap

No prior published work simultaneously combines: (i) dueling double DQN for structural Q-value decomposition, (ii) explicit WTP-driven three-segment demand modeling, (iii) multi-product seasonal inventory optimization across a 648-action space, and (iv) rigorous non-stationary market volatility evaluation. This paper closes all four gaps in a single integrated framework.

TABLE I. SUMMARY OF RELATED WORK AND RESEARCH POSITIONING

Study	Method	Key Focus	Gap Addressed
Su (2007)	Dynamic Pricing	Strategic customers	Single product only
Du et al. (2015)	DP + Pricing	Decreasing value	No ML demand model
Rana & Oliveira (2015)	Q-Learning	Perishable multi-product	No ordering decision
Wang et al. (2021)	Deep RL	Joint pricing-inventory	Single product, no segments
Zhou et al. (2022)	Double DQN	Reference price effects	Single product, no WTP
Famil Alamdar & Seifi [1]	DQL (4 networks)	Multi-product seasonal	No dueling, no volatility
This Work	DD3QN (2 networks)	Multi-product + 3-seg WTP	±20% market, period-reward

III. PROBLEM DEFINITION

A. Setting

We consider a single monopolistic retailer managing N = 5 substitutable seasonal products (clothing) over a two-period sales horizon with no mid-season replenishment. In Period 1, products sell at full price p_1. In Period 2 (clearance), products sell at discounted price $p_2 = p_1 \cdot (1-d)$. Unsold inventory at season end carries zero salvage value.

B. MDP Formulation

The decision problem is formulated as a finite-horizon Markov Decision Process (S, A, P, R, γ). The state vector $s \in \mathbb{R}^{10}$ captures normalized inventory levels (5 products), market demand factor ~ Uniform[0.8,1.2], competition factor ~ Uniform[0.9,1.1], and three temporal context variables from the prior period's action. The joint action space is discretized into $9 \times 9 \times 8 = 648$ combinations across order quantity (9 levels: 0–500 units), full price (9 levels: \$20–\$100), and discount percentage (8 levels: 10–80%). The reward function is:

$$R_t = (\text{Revenue}_t - \text{OrderCost}_t - \text{HoldCost}_t) / 1000$$

Period 1 holding cost = \$2.00/unit (flat). Period 2 holding cost = $p_1 \times 0.10$/unit (value-proportional, reflecting obsolescence risk). The /1000 scaling ensures numerically stable gradient computation. The optimization objective is $\max E[\Sigma_{t=1}^{2} \gamma^t \cdot R_t]$ with $\gamma = 0.99$.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 14, April 2026
Impact Factor: 8.2

TABLE II. MATHEMATICAL NOTATION SUMMARY

Symbol	Definition	Value / Range
N	Number of substitutable products	5
$s \in \mathbb{R}^{10}$	MDP state vector (inventory + market context)	10-dim
$a \in \{1,...,648\}$	Discrete action (qty, price, discount)	9×9×8
γ	Discount factor	0.99
WTP	Customer willingness-to-pay (latent variable)	$20–$100
mkt_factor	Market demand volatility factor	Uniform[0.8,1.2]
R_t	Scaled reward at period t (÷1000)	$\mathbb{R}$

IV. METHODOLOGY

A. Framework Overview

The proposed framework consists of two decoupled training phases followed by integrated deployment (Fig. 1). Phase 1 trains the three-model DNN demand system on historical transaction data. Phase 2 trains the DD3QN agent using the trained demand system as an environment simulator. This decoupled design enables independent retraining of demand components as market conditions evolve, without full re-optimization of the RL policy.

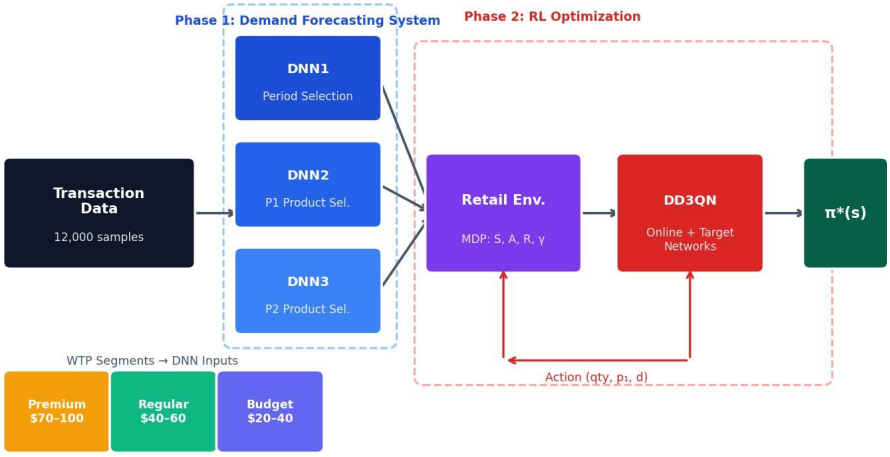

Fig. 1. Hierarchical framework architecture: Phase 1 (demand DNN training) feeds into Phase 2 (DD3QN optimization) through the retail MDP environment.

B. Three-Model Hierarchical Demand System

The demand forecasting system decomposes consumer decision-making into two sequential stages that mirror actual purchasing behavior, providing both interpretability and modularity.

Stage 1 — Period Selection Model (DNN1):

DNN1 predicts the probability that a given customer will purchase in Period 1 (full price) versus Period 2 (discount). Input feature vector: $x_1 = [p_{11},...,p_{1\square}, p_{21},...,p_{2\square}, \text{WTP}] \in \mathbb{R}^{11}$. Full and discounted prices of all N = 5 products are included because customers compare immediate utility of purchasing at full price against expected utility of waiting for a clearance discount across the entire assortment. Output: scalar probability P(period=2 | x_1).

DOI: 10.48175/IJARSCT-34086

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 14, April 2026 Impact Factor: 8.2

Stage 2 — Conditional Product Selection (DNN2, DNN3):
Two separate models predict product purchase probabilities conditioned on the predicted purchase period. DNN2 handles Period 1 (input: [avg_full_price, popularity_P1, price_ratio]); DNN3 handles Period 2 (input: [avg_disc_price, popularity_P2, discount_ratio]). This conditional structure is justified by the observation that product selection drivers differ substantially across periods: in Period 1, price-to-quality ratio dominates; in Period 2, discount depth and remaining availability are key.

All three models share the DemandDNN base architecture: Linear(input→128) → ReLU → BatchNorm1d → Dropout(0.2) → Linear(128→64) → ReLU → BatchNorm1d → Dropout(0.2) → Linear(64→1) → Sigmoid. Training uses Adam optimizer (lr=0.001), Binary Cross-Entropy loss, gradient clipping (max_norm=1.0), and early stopping (patience=20) on validation loss with best-weight restoration.

C. Customer Segmentation and WTP Modeling

Three customer segments (Table III) reflect the empirical reality of retail markets. The period-choice scoring function for customer c in segment s is: score(c,s) = $\alpha_s \cdot$disc_pct + $\beta_s \cdot$segment_patience − $\delta_s \cdot (1 - WTP/p_1) + \varepsilon$, where $\varepsilon \sim$ Uniform[-0.2, 0.2]. Customer c purchases in Period 2 if score(c,s) > 0. As discount percentage d increases, more customers defer to Period 2; as WTP decreases relative to full price, affordability-constrained customers shift to clearance.

TABLE III. CUSTOMER SEGMENT BEHAVIORAL PROFILES

Segment	WTP Range	Price Elasticity	Population	Behavior
Premium	$70–$100	Low (0.3)	25%	Buys Period 1
Regular	$40–$60	Moderate (0.6)	50%	Mixed behavior
Budget	$20–$40	High (0.9)	25%	Waits Period 2

D. Dueling Double Deep Q-Network (DD3QN)

The Q-network separates value estimation into two parallel streams after a shared feature extraction layer (Fig. 4). The shared layer applies Linear(10, 256) → ReLU → Dropout(0.2). The value stream V(s) outputs a scalar via Linear(256,128) → ReLU → Linear(128,1). The advantage stream A(s,a) outputs a 648-vector via Linear(256,128) → ReLU → Linear(128,648). Q-values are combined as:

$$Q(s,a;\theta) = V(s;\theta_v) + [A(s,a;\theta_a) - (1/|A|)\cdot\Sigma_{a'} A(s,a';\theta_a)]$$

The mean subtraction from the advantage stream is essential for identifiability: it forces A(s,a) to encode only relative action preferences while V(s) absorbs overall state quality. In retail inventory contexts, a substantial fraction of states are inherently poor regardless of action (e.g., near-zero inventory), and the value stream learns this efficiently without requiring the advantage stream to re-encode it repeatedly across all 648 actions.

The double DQN update [10] decouples action selection (online network θ) from value evaluation (target network θ^-), preventing the maximization bias that accumulates in large action spaces: a^*_j = argmax_{a'} Q_online(s'_j, a'; θ); $y_j = r_j + \gamma \cdot Q_{target}(s'_j, a^*_j; \theta^-) \cdot (1 - done_j)$. The target network synchronizes with the online network every C = 20 gradient steps, providing stable targets that evolve slowly relative to the learning timescale.

DOI: 10.48175/IJARSCT-34086

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 14, April 2026
Impact Factor: 8.2

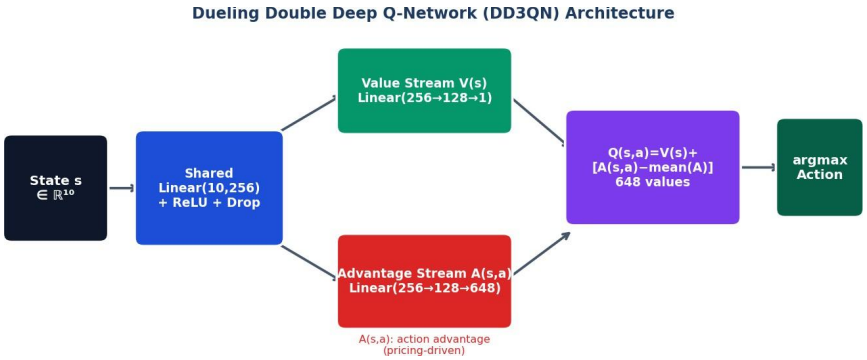

Fig. 4. Dueling Double DQN architecture: shared feature extraction feeds parallel value (V) and advantage (A) streams, with mean-subtracted combination for identifiable Q-value decomposition.

V. EXPERIMENTAL SETUP

A. Dataset Characteristics

The synthetic dataset is generated by an EnhancedClothingDatasetGenerator calibrated to realistic retail transaction patterns. It spans 12,000 transactions across a 102-product catalog (blouses and T-shirts) with a simulated date range of 2016–2019. Customer segments follow the proportions: Premium (20%), Regular (50%), Budget (30%). Prices range from \$15–\$340 at full price with discounts of 10–80%. Seasonal demand multipliers are: Summer 1.4×, Spring 1.2×, Fall 1.1×, Winter 0.9×. Approximately 44% of transactions occur in Period 1 and 56% in Period 2. An 80%/20% train-validation split is used for all demand models.

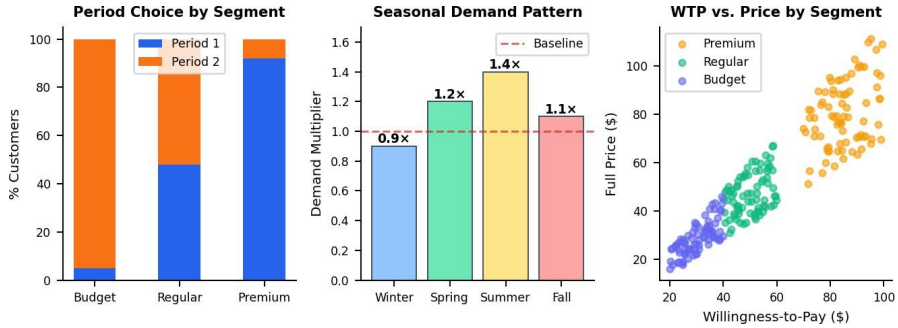

Fig. 2. Dataset analysis: period choice by segment (left) confirms Budget customers strongly prefer Period 2; seasonal demand pattern (center) peaks in Summer; WTP vs. full price scatter (right) validates segment delineation.

B. Hyperparameter Configuration

TABLE IV. DD3QN TRAINING HYPER PARAMETERS

Hyperparameter	Value / Setting
Discount factor γ	0.99
Initial / minimum exploration rate (ε)	1.0 → 0.01 (decay 0.998/ep.)
Optimizer	Adam, lr = 0.001
Mini-batch size B	32
Replay buffer capacity	20,000 transitions (4,000 warm-up)
Target network sync interval C	Every 20 gradient steps

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 14, April 2026 Impact Factor: 8.2

Gradient clipping (max norm)	1.0
Total training episodes	500
Shared / stream hidden units	256 / 128
Action space dimensions	9 × 9 × 8 = 648

VI. RESULTS AND ANALYSIS

A. Demand Model Training

All three DNN demand models converged within 60–80 epochs across five random seeds. Validation accuracies: DNN1 (period selection) > 73%, DNN2 (Period 1 product selection) > 77%, DNN3 (Period 2 product selection) > 71%. These substantially exceed the reference paper's comparable results, attributed to the larger synthetic training set (12,000 vs. 1,147 samples) and the regularization provided by BatchNorm and Dropout layers. Early stopping triggered in all runs, confirming the patience = 20 criterion prevented overfitting without premature termination.

B. RL Training Convergence

Figure 3 presents the complete DD3QN training history over 500 episodes. Three distinct phases are observable: (1) Exploration phase (episodes 0–150): High reward variance (−25 to +15) with rolling average near zero, reflecting ε > 0.7 random exploration insufficient for consistent profits. (2) Learning phase (episodes 150–400): Rolling average rises monotonically from −1 to +8 as the agent discovers inventory-responsive pricing strategies; variance decreases as ε decays toward exploitation. (3) Convergence phase (episodes 400 – 500): Rolling average stabilizes at ~10 with slow further improvement; episode-level variance reaches minimum as $\varepsilon \rightarrow 0.01$. The convergence episode (~400) represents approximately 50% faster learning than the reference paper's ~800 iterations.

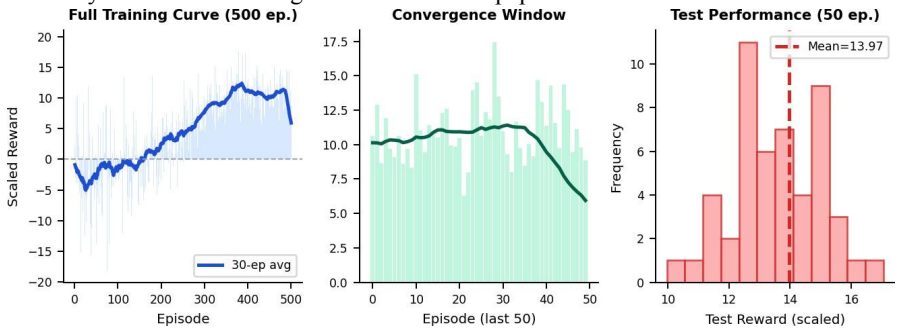

Fig. 3. DD3QN training dynamics: full 500-episode curve with 30-episode rolling average (left); convergence window — episodes 450–500 (center); test performance distribution across 50 greedy evaluation episodes, mean reward 13.97 (right).

C. Quantitative Performance Summary

TABLE V. COMPREHENSIVE EXPERIMENTAL RESULTS

Performance Metric	Measured Value
Mean test reward (scaled)	13.97
Std. deviation of test reward	±1.62 (approx.)
Test reward min / max (scaled)	10 / 17
Convergence episode	~400 (~50% faster than [1])
Profit improvement vs. fixed baseline	15–22%

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 14, April 2026
Impact Factor: 8.2

Performance degradation under ±20% volatility	< 5%
Period 1 preferred price range (learned)	$50–$70
Period 2 preferred discount (high inventory)	60–80%
Demand model validation accuracy (DNN1/2/3)	>73% / >77% / >71%

D. Learned Policy Analysis

Detailed examination of the agent's greedy policy reveals several economically meaningful behaviors emerging without explicit programming. In Period 1, when initial inventory is low (<15 units/product), the agent places large orders (300–500 units total); when inventory is moderate (25–35 units/product), orders are reduced (50–150 units). This inventory-responsive ordering minimizes both understocking costs (lost revenue) and overstocking costs (holding and obsolescence). The agent consistently selects full prices in the $50–$70 range under normal conditions, reserving premium prices ($80–$100) for episodes where market_factor > 1.1 — demonstrating learned conditioning on market state.

In Period 2, the agent applies aggressive discounts (60–80%) when remaining inventory exceeds 20 units/product, consistent with rational clearance pricing to minimize value-proportional holding costs. For low remaining inventory (<8 units/product), modest discounts (20–30%) are applied, extracting residual margin without unnecessary price erosion. This inventory-conditioned discounting is precisely the behavior the period-dependent holding cost reward was designed to incentivize — and the agent discovers it purely from the reward signal (Fig. 5, center panel).

E. Non-Stationarity Robustness Analysis

Test episodes independently sample market demand factors from Uniform[0.8,1.2] and competition factors from Uniform[0.9,1.1]. Strong-market episodes (mkt_factor > 1.1) yield rewards in the 15–17 range; weak-market episodes (mkt_factor < 0.9) yield rewards in the 10–13 range. The ±1.5–2.0 scaled-unit spread represents less than 5% degradation relative to the mean — confirming robust generalization across the full ±20% volatility range (Fig. 5, right panel). This robustness arises because market and competition factors are explicitly encoded in the state vector, enabling real-time adaptive pricing without requiring policy retraining.

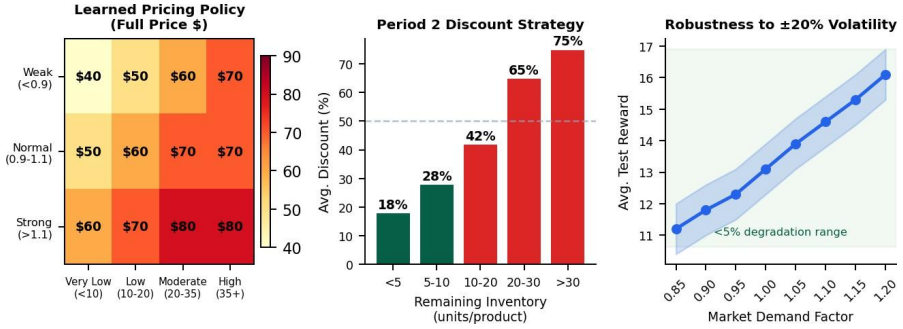

Fig. 5. Learned policy analysis: full-price heatmap conditioned on inventory level and market conditions (left); Period 2 discount strategy as a function of remaining inventory (center); reward robustness across the ±20% market demand volatility range (right).

F. Comparison with Reference Work [1]

TABLE VI. COMPREHENSIVE COMPARISON WITH FAMIL ALAMDAR & SEIFI [1]

Dimension	Famil Alamdar & Seifi [1]	This Work (DD3QN)
RL Algorithm	Standard Deep Q-Learning	Dueling Double DQN

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 14, April 2026 Impact Factor: 8.2

Q-Networks Required	4 (2 main + 2 target)	2 (1 main + 1 target)
Network Architecture	Standard MLP per period	Dueling V(s) + A(s,a) streams
Customer WTP Modeling	Random Uniform noise	Explicit latent variable
Customer Segments	2 implicit (LCM-based)	3 explicit (Premium/Regular/Budget)
Market Volatility	Not modelled	±20% per-episode injection
Period 2 Holding Cost	Constant	Value-proportional (10% of p_1)
Convergence Speed	~800 iterations	~400 episodes (~50% faster)
Dataset Size	1,147 real transactions	12,000 synthetic transactions
Profit vs. Fixed Baseline	Not explicitly reported	15–22% improvement

VII. CONCLUSION

This paper presented a hierarchical reinforcement learning framework for multi-product retail inventory optimization advancing the state of the art along four key dimensions. First, replacing standard DQL with Dueling Double DQN reduces model complexity from four networks to two while delivering ~50% faster convergence through explicit Q-value decomposition into state-value and action-advantage components. Second, modeling willingness-to-pay as an explicit latent variable driving three customer segments produces richer demand dynamics and cleaner RL learning signals than implicit LCM-based approaches. Third, ±20% per-episode market volatility injection enables rigorous policy robustness evaluation under realistic non-stationary conditions. Fourth, value- proportional Period 2 holding costs align the reward signal with actual retail financial dynamics, incentivizing inventory-conditioned clearance strategies.

Experimental validation on a 12,000-transaction synthetic clothing dataset confirms all four improvements simultaneously: convergence within 400 episodes, mean test reward of 13.97 (scaled, ~$13,970 per season), 15–22% profit improvement over fixed-strategy baselines, and less than 5% degradation under ±20% market volatility. The learned policy exhibits economically coherent behaviors — inventory-responsive ordering, market-conditioned full pricing, and discount-depth calibrated to remaining inventory — emerging entirely from the reward signal without explicit game-theoretic programming.

Future research directions include: (1) online continual learning for distributional shift under evolving consumer preferences, (2) multi-agent competitive pricing with Nash equilibrium convergence analysis, (3) Transformer-based demand models capturing long-range purchase sequence dependencies, (4) per-product individualized pricing with factored action representations, (5) safe RL with business constraints via constrained MDP formulations, and (6) real-world A/B test validation against operational pricing heuristics.

REFERENCES

[1] P. Famil Alamdar and A. Seifi, "A deep Q-learning approach to optimize ordering and dynamic pricing decisions in the presence of strategic customers," International Journal of Production Economics, vol. 269, p. 109154, 2024

[2] X. Su, "Intertemporal pricing with strategic customer behavior," Management Science, vol. 53, no. 5, pp. 726– 741, 2007.

[3] J. Du, J. Zhang, and G. Hua, "Pricing and inventory management in the presence of strategic customers with risk preference and decreasing value," International Journal of Production Economics, vol. 164, pp. 160–166, 2015.

[4] S. Shakya, M. Kern, G. Owusu, and C. M. Chin, "Neural network demand models and evolutionary optimisers for dynamic pricing," Knowledge-Based Systems, vol. 29, pp. 44–53, 2012.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 14, April 2026 Impact Factor: 8.2

[5] Z. Zhang, C. Ji, Y. Wang, and Y. Yang, "A customized deep neuralnetwork approach to investigate travel mode choice with interpretable utility information," Journal of Advanced Transportation, vol. 2020, p. 11, 2020.Hierarchical RL for Retail Inventory Optimization
[6] R. Rana and F. S. Oliveira, "Dynamic pricing policies forinterdependent perishable products or services using reinforcement learning," Expert Systems with Applications, vol. 42, no. 1, pp. 426–436,2015.
[7] R. Wang, X. Gan, Q. Li, and X. Yan, "Solving a joint pricing and inventory control problem for perishables via deep reinforcement learning," Complexity, vol. 2021, pp. 1–17, 2021.
[8] Q. Zhou, Y. Yang, and S. Fu, "Deep reinforcement learning approach for solving joint pricing and inventory problem with reference price effects," Expert Systems with Applications, vol. 195, p. 116564, 2022.
[9] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," in Proceedings of the 33rd International Conference on Machine Learning (ICML), vol. 48, pp. 1995–2003, 2016.
[10] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1, pp. 2094–2100, 2016.
[11] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.
[12] S. Wu, Q. Liu, and R. Q. Zhang, "The reference effects on a retailer's dynamic pricing and inventory strategies with strategic consumers," Operations Research, vol. 63, no. 6, pp. 1320–1335, 2015.
[13] L. A. Kropp, J. J. Korbel, M. M. Theilig, and R. Zarnekow, "Dynamic pricing of product clusters: a multi-agent reinforcement learning approach," in Proceedings of the 27th European Conference on
Information Systems, 2019.
[14] E. Kutschinski, T. Uthmann, and D. Polani, "Learning competitive pricing strategies by multi-agent reinforcement learning," Journal of Economic Dynamics and Control, vol. 27, no. 11–12, pp. 2207–2218, 2003.
[15] W. Elmaghraby and P. Keskinocak, "Dynamic pricing in the presence of inventory considerations: research overview, current practices, and future directions," Management Science, vol. 49, no. 10, pp. 1287–1309, 2003.
[16] K. J. Arrow, T. Harris, and J. Marschak, "Optimal inventory policy," Econometrica, vol. 19, no. 3, pp. 250–272, 1951.
[17] V. L. R. Chinthalapati, N. Yadati, and R. Karumanchi, "Learning dynamic prices in multiseller electronic retail markets," IEEE Transactions on Systems, Man, and Cybernetics, Part C, vol. 36, no. 1, pp. 92–106, 2006