From Heuristics to Reinforcement Learning: Integrated Operational–Financial Control of Supply Chains Under Demand Disruption

BADAKHSHAN, Ali, BADAKHSHAN, Ehsan <http://orcid.org/0000-0002-5298-764X>, SAAD, Sameh <http://orcid.org/0000-0002-9019-9636> and BAHADORI, Ramin <http://orcid.org/0000-0001-6439-7033>

applied sciences

MDPI

Article

From Heuristics to Reinforcement Learning: Integrated Operational–Financial Control of Supply Chains Under Demand Disruption

Ali Badakhshan [1], Ehsan Badakhshan [2,*], Sameh Saad [2] and Ramin Bahadori [2]

[1] Department of Engineering, Durham University, Durham DH1 3LE, UK
[2] Sheffield Business School, Sheffield Hallam University, Sheffield S1 1WB, UK
* Correspondence: e.badakhshan@shu.ac.uk

Featured Application

The proposed framework can be applied as a decision-support layer for multi-echelon supply chains where service performance and working capital efficiency must be managed jointly, such as consumer goods or industrial manufacturing. The framework can be embedded in planning or digital-twin systems to generate coordinated ordering, production, and cash-collection decisions using routinely available data. Adaptive heuristics can serve as transparent baselines in stable periods, while learning-based control can be used during demand disruption to contain backlog, service shortfalls, and working capital stress.

Abstract

Supply chain control requires balancing operational performance and financial efficiency when decisions are made using delayed and imperfect demand information. Although fixed heuristics, adaptive policies, and reinforcement learning approaches have been proposed, their relative effectiveness and robustness under temporary informational mismatch remain unclear. This study addresses this gap by developing an integrated simulation–reinforcement learning framework that jointly captures operational and financial dynamics in supply chains, which enables adaptive optimisation of working capital policies under uncertainty. A unified simulation framework is developed for a multi-echelon supply chain that jointly models service levels, backlog, customer retention, and working capital exposure through the cash conversion cycle. Five classes of controllers are evaluated: fixed-threshold heuristics, adaptive threshold policies optimised using stochastic and evolutionary search, and a reinforcement learning controller based on proximal policy optimisation. Performance is assessed under stationary demand and under demand disruptions. The results reveal a clear hierarchy of performance. Fixed heuristics provide transparent and stable baselines but suffer from structural rigidity. Adaptive threshold policies substantially improve coordination, with evolutionary search yielding the strongest performance among structured approaches. The reinforcement learning controller achieves the best overall outcomes by learning a nonlinear state–action mapping that sharply reduces backlog and service shortfalls while maintaining comparable working capital exposure. These gains arise from improved coordination across operational and financial decisions rather than single-metric optimisation. Practically, adaptive heuristics offer robust baselines, while learning-based controllers are most valuable in more volatile environments.

Keywords: supply chain resilience; working capital management; reinforcement learning (RL); simulation; demand disruption

Academic Editor: Vittorio Solina

Received: 30 April 2026
Revised: 3 June 2026
Accepted: 4 June 2026
Published: 5 June 2026

1. Introduction

Working capital management is a critical component of supply chain (SC) performance because it concerns the efficient control of short-term assets and liabilities, including inventory, cash, accounts receivable, and accounts payable. The way these elements are governed shapes liquidity, the cost of capital tied up in operations, and the operational flexibility of firms embedded in interconnected networks. Working capital is thus not merely a firm-level financial indicator; it underpins the capacity of SCs to sustain material flows, absorb uncertainty, and respond to changing market conditions. Effective management strengthens both financial stability and operational responsiveness, whereas poor management constrains liquidity and undermines overall SC performance [1].

Traditional approaches to working capital management have focused on optimising components within single firms, e.g., reducing local inventory, accelerating receivables, or extending payables. While such measures can improve firm-level performance in the short term, they often yield suboptimal SC outcomes when pursued in isolation. For instance, inventory cuts by one actor may elevate shortage risks elsewhere, and aggressive payment-term extensions can shift liquidity pressures to upstream or downstream partners. Accordingly, recent research argues for an SC-oriented perspective in which firms coordinate financial and operational decisions across tiers to reduce capital costs and enhance collective resilience [1,2]. Such integration is important because operational decisions, such as inventory replenishment and production planning, directly influence financial outcomes through inventory holding costs, receivables, payables, and cash flow movements, while financial decisions, including payment terms and liquidity availability, can also affect operational performance across SC partners. Consequently, decisions taken by one actor may generate both operational and financial effects throughout the network, requiring coordinated management of material and financial flows. This reframes working capital from a local optimisation task into an inter-organisational coordination problem.

Achieving such coordination is difficult because SCs are inherently dynamic, uncertain, and disruption-prone. Volatile demand, fluctuating lead times, and operational shocks readily create excess stock or stockouts, which erode working capital efficiency and profitability. Simultaneously, poor synchronisation in financial flows, often caused by misaligned payment terms and delays in receivables, can intensify liquidity risks even when operational indicators look satisfactory [3]. As a result, working capital management becomes a complex decision problem involving simultaneous interactions between physical and financial flows over time.

This interdependence also illustrates why existing inventory-focused heuristics or RL-based inventory control methods are insufficient for working capital management. For example, an inventory controller may reduce replenishment quantities to minimise holding costs, but this can increase stockouts, reduce customer retention, and delay future cash inflows. Conversely, a policy that maintains high inventory availability may improve service levels but tie up excessive capital in stock and increase working capital exposure. Similarly, extending payment terms may improve one firm's short-term cash position but transfer liquidity pressure to upstream suppliers, potentially weakening supply reliability. These examples show that policies optimised for inventory or operational metrics alone may generate undesirable financial consequences, while financially attractive policies may undermine operational performance. Therefore, working capital management requires an integrated operational–financial perspective that jointly evaluates inventory dynamics, service performance, receivables, payables, and cash flow movements.

Within this context, simulation has long served as a key method for analysing SC behaviour under stochastic and dynamic conditions. Techniques such as discrete event simulation (DES), agent-based modelling (ABM), and system dynamics (SD) allow researchers

and practitioners to replicate interactions among SC entities and assess the implications of alternative operational and financial policies [4]. By explicitly modelling product, information, and capital flows, simulation offers a controlled experimental environment for testing inventory policies and financial arrangements across varying conditions [5]. Nevertheless, simulation alone does not prescribe optimal decisions.

To address this limitation, many simulation-based SC studies incorporate optimisation mechanisms. A common approach is to use metaheuristics, e.g., genetic algorithms, to search for near-optimal decision rules within complex simulation environments [6]. While effective, these methods often depend on expert-defined search structures, careful parameter tuning, and extensive computation, especially when decision spaces are large and multidimensional. Limitations become more pronounced when SC decisions must be revised continually in response to evolving system states. Under such conditions, there is a need for approaches that can learn and adapt policies dynamically rather than relying solely on predefined search procedures.

Integrating simulation and reinforcement learning (RL) offers a promising way to meet this need. In a simulation–RL framework, the simulation model serves as the environment in which the RL agent learns. The simulation reproduces the SC's response to actions, such as replenishment, production adjustments, or credit-related decisions, and provides performance feedback through reward signals tied to predefined objectives. The agent then updates its policy iteratively to improve performance. This closed-loop structure enables flexible simulation-based optimisation, which allows policies to adjust continually to stochastic, complex environments where analytical solutions are infeasible [7]. Crucially, reward functions can be designed to reflect working capital objectives directly, e.g., reduced capital tied up in operations, thereby aligning RL with both financial and operational performance targets.

Although simulation–RL integrations have proven effective in operational areas such as inventory control (e.g., [8]), their use in working capital management remains limited.

Existing simulation–RL studies primarily focus on inventory-related decisions and provide limited representation of financial elements such as receivables, payables, and cash movements. Consequently, current approaches do not fully capture the operational–financial interaction that determines working capital performance.

Furthermore, previous studies on working capital management generally rely on static or predefined decision rules and do not adapt policies dynamically in response to evolving SC conditions. This highlights the need for adaptive decision-making approaches that can jointly model operational and financial dynamics while continuously optimising working capital policies under stochastic and changing SC conditions.

To address these needs, we develop an integrated simulation–RL framework for optimised working capital management in SCs. The developed framework aims to answer two research questions: (1) To what extent can an RL agent trained within an integrated simulation environment learn working capital policies that adapt dynamically to stochastic and evolving SC conditions? and (2) Does the proposed simulation–RL approach outperform metaheuristic optimisation methods in optimising working capital decisions within SCs? By answering these research questions, the study demonstrates the potential of adaptive, learning-based decision-making for managing the intertwined operational and financial drivers of working capital and assesses whether such an approach offers advantages over established optimisation techniques. To answer the first question, the framework uses a DES model that trains an RL agent to continually optimise working capital decisions in response to evolving SC conditions. To answer the second question, we compare the performance of the proposed simulation–RL approach with genetic algorithms (GAs) and cross-entropy methods (CEMs).

The incremental contribution of this study lies in extending existing simulation–RL research beyond inventory-centred SC control and extending simulation-based working capital optimisation beyond static or predefined policy search. First, the proposed DES environment jointly represents operational flows and financial flows, including inventory movements and trade credit dynamics, represented through receivables, payables, and cash flow timing. Second, the RL formulation embeds working capital objectives into the learning process through a reward structure that reflects both operational performance and financial exposure. Third, the study evaluates the proposed RL controller against fixed heuristic and adaptive metaheuristic-based controllers, thereby distinguishing learning-based policy adaptation from predefined and search-optimised decision rules. Finally, performance is assessed under both stable demand and demand disruption conditions, allowing the robustness of integrated operational–financial control policies to be examined under temporary informational mismatch.

The remainder of the paper is structured as follows. Section 2 reviews the literature on working capital management in SCs and integrated simulation–RL approaches for SC decision-making. Section 3 presents the proposed framework and its main components. Section 4 reports and discusses the results under no-disruption and demand disruption scenarios. Section 5 concludes by outlining the contributions, limitations, and avenues for future research.

2. Literature Review

The literature review is structured around two key SC research domains central to this study: working capital management in SCs and the use of integrated simulation–RL approaches for SC decision-making.

2.1. Working Capital Management in SCs

Working capital management plays a central role in SC performance because it directly affects liquidity, profitability, and operational continuity. The cash conversion cycle (CCC), defined as days inventory outstanding (DIO) plus days sales outstanding (DSO) minus days payable outstanding (DPO), remains the most widely used measure of WCM efficiency. Shorter CCC values generally reflect faster recovery of cash invested in operations and stronger liquidity [9].

A consistent theme across empirical research is the strong relationship between CCC and firm performance. Studies show that CCC affects profitability, returns, and market valuation. For example, Piao et al. [10] find an inverted U-shaped relationship between CCC and ROA, suggesting that moderate CCC levels are beneficial but excessive ones reduce profitability. Similarly, Banerjee et al. [11] show that when CCC exceeds a threshold level, firm stock performance deteriorates. Other studies, such as Oh et al. [12] and Thomya et al. [13], demonstrate negative associations between CCC and return on investment, while Pei et al. [14] highlight the trade-off between improved liquidity (via longer CCC) and increased default risk. These findings collectively underscore that CCC is financially consequential, and that firms must balance inventory, receivables, and payables carefully to avoid liquidity stress and performance deterioration.

Several studies examine working capital management in the context of SC disruption. Hofmann et al. [15] show that disruptions typically raise DIO and compel firms to adjust DSO and DPO to stabilise the overall CCC. Building on this operational impact, Ivanov [16] demonstrates through DES that, during disruptions, shorter payment terms upstream in the SC can enhance resilience by improving cash flow reliability. Trade credit decisions also become more consequential in such settings: Wu et al. [17] find that extending credit to customers can temporarily boost sales, albeit at the cost of reduced firm value. Finally,

bargaining power shapes firms' ability to manage the CCC in turbulent conditions; Carnes et al. [18] show that cash-rich firms can negotiate shorter CCCs, unless relational ties limit their ability to exercise this advantage.

Recent industry research emphasises the growing strain caused by extended buyer payment terms, particularly for small suppliers. The Hackett Group [19] reports that payment terms have continued to lengthen across many sectors, increasing DPO and placing greater liquidity pressure on upstream suppliers. In response to these mounting liquidity pressures and broader working capital management challenges, Industry 4.0 technologies are increasingly shaping working capital practices. Fang [20] shows that blockchain adoption enhances transparency and reduces CCC across SCs. Badakhshan and Ivanov [21] demonstrate that integrating digital twin and blockchain technologies can improve working capital management during disruptions. Samuels [22] notes that AI, blockchain, and IoT can meaningfully improve visibility and reduce delays, although their effectiveness depends on integration quality and organisational readiness.

Overall, the existing literature provides useful insights into how the CCC behaves and how individual working capital levers affect performance. However, much of this work remains largely descriptive. Most studies do not offer guidance on how firms should optimally manage working capital under uncertainty, nor do they treat working capital management as a dynamic process requiring continual adjustment as conditions change. This highlights a need for modelling approaches capable of identifying optimal working capital policies and adapting those policies as the SC environment evolves.

To address this gap, we develop an integrated simulation–RL framework for optimised working capital management in SCs. The proposed framework enables the learning of optimal policies and their refinement over time, which offers a more realistic and adaptive approach to managing working capital in complex and uncertain SC settings.

2.2. Integrated Simulation–RL for SC Decision-Making

Integrated simulation–RL approaches provide a powerful methodology for addressing the complexity, uncertainty, and interdependence characteristic of SC systems. In these approaches, a simulation model serves as the environment within which the RL agent interacts, making decisions, observing state transitions, and receiving feedback through reward signals. This setup allows the agent to learn high-quality decision policies through repeated experimentation in a controlled yet realistic representation of the SC. In SC contexts, simulations can capture the stochastic behaviour of processes such as inventory replenishment, production scheduling, transportation, or even cash flow timing. Reward functions can then be designed to reflect operational goals such as cost minimisation and service-level performance, or financial metrics such as liquidity and working capital efficiency. This closed-loop structure offers a flexible form of simulation-based optimisation capable of handling environments where analytical optimisation becomes intractable [23].

Early applications of integrated simulation–RL predominantly focused on inventory management settings, particularly the well-known Beer Game environment. Chaharsooghi et al. [24] applied Q-learning in a multi-echelon Beer Game simulation, allowing RL agents to dynamically adjust ordering decisions and significantly reduce the bullwhip effect compared with classical heuristics. Mortazavi et al. [25] extended these ideas by incorporating value-at-risk into an RL-based ordering system for multi-tier SCs, thus capturing cost variability and risk exposure. Preil and Krapp [26] advanced this line of research by integrating simulation with multi-armed bandit algorithms to optimise base-stock levels in stochastic multi-echelon settings. Collectively, these studies established the value of treating SC operational decisions as Markov decision processes (MDPs) that can be solved through learning agents interacting with simulations.

Advances in computational capability and deep learning techniques have driven the adoption of deep RL (DRL) in more recent studies. DRL allows for approximating value functions or policies in high-dimensional, continuous, or partially observable SC environments, enabling the solution of problems that were previously computationally prohibitive. Fuji et al. [27] applied a deep multi-agent RL approach to an extended Beer Game scenario. By embedding neural networks within the learning architecture and incorporating evolutionary mechanisms, the authors enabled agents to co-evolve strategies, achieving superior performance in cost and service levels compared with traditional approaches. Similarly, Oroojlooyjadid et al. [28] trained deep Q-networks (DQNs) to play the Beer Game, showing that learned policies not only matched but in some cases exceeded human decision makers and demonstrated strong generalisation properties across changes in cost parameters and demand distributions.

Recent studies have further expanded RL-based SC research in terms of problem scope, algorithmic design, and system complexity. Wang et al. [29] proposed a robust RL framework for risk-averse SC management, showing that robust RL can support SC operations when environments deviate from training conditions. Bussieweke et al. [30] integrated system dynamics and RL to optimise recovery policies under SC disruptions, demonstrating the relevance of RL for disruption response and ripple effect mitigation. Kotecha and del Rio Chanona [31] developed a multi-agent RL framework using graph neural networks for inventory control in SCs, showing how graph-based state representations can support decentralised inventory decision-making. Liu et al. [32] applied multi-agent DRL to decentralised multi-echelon inventory management, using centralised training with decentralised execution to coordinate inventory decisions across SC actors. Hu et al. [33] also investigated multi-agent DRL for decentralised multi-echelon inventory optimisation under stochastic demand, further demonstrating the suitability of RL for adaptive inventory control in uncertain SC environments.

Extending this trajectory, Zhou et al. [34] proposed a multi-agent DRL approach for ordering and inventory allocation in a decentralised two-echelon dual-channel SC. In another contribution, Zhou et al. [35] developed an uncertainty-aware multi-agent RL approach for joint inventory–transportation decisions, incorporating lead-time estimation into the decision process. Collectively, these studies indicate that RL-based SC research is moving beyond simple inventory settings toward decentralised, multi-agent, and structurally complex decision environments. However, their focus remains largely operational, with limited attention to the joint modelling of inventory, receivables, payables, and cash flow movements for working capital optimisation.

Beyond inventory optimisation, integrated simulation–RL methods have been applied to a range of other SC decision domains. Studies have demonstrated the usefulness of RL-driven strategies in dynamic delivery routing [36], production scheduling under uncertain processing times [37], and supplier selection in competitive procurement environments [38]. Badakhshan et al. [23] document rapid growth in DES–RL models applied to multi-echelon planning, disruption management, and real-time control in SCs. Across these applications, simulation models capture the stochastic and interactive nature of the system, while RL agents learn responsive policies that improve adaptively as the environment varies. This line of research has collectively demonstrated that integrating simulation with RL offers a promising paradigm for modelling SC problems as Markov decision processes, in which decisions must be updated sequentially in response to evolving system conditions.

Despite these advances, applications of simulation–RL to working capital management remain limited. Existing RL-based SC studies primarily focus on inventory dynamics and related operational outcomes, including inventory control, service levels, disruption response, and recovery decisions. While these studies demonstrate the value of RL for adaptive oper-

ational decision-making, they generally do not capture how inventory-oriented policies influence receivables, payables, cash conversion, and liquidity exposure. Consequently, a policy that performs well from an operational perspective may still be financially inefficient if it ties up excessive capital, delays cash inflows, or shifts liquidity pressure across SC partners. This limitation is important because working capital efficiency depends not only on inventory management but also on the timing and magnitude of cash movements. Current integrated simulation–RL research does not yet capture the full operational–financial interaction that drives working capital management performance. This highlights the need for frameworks that simulate both operational and financial elements of the SC and pair them with RL-based policy learning.

The hierarchy of controllers evaluated in this study is selected to reflect increasing levels of policy adaptability and learning capability. Fixed heuristic controllers provide transparent and interpretable baselines that represent commonly used rule-based decision structures. GA and CEM are selected as adaptive optimisation baselines because they are established simulation-optimisation methods for searching policy parameters in complex stochastic systems without requiring analytical gradients [6,39]. GA represents an evolutionary population-based search approach [40], while CEM represents a probabilistic sampling-based optimisation approach [41]. PPO is selected as the learning-based controller because it is a stable policy-gradient RL algorithm suitable for sequential decision problems with stochastic state transitions and multidimensional control actions [42]. This comparison allows the study to distinguish between fixed rules, simulation-optimised adaptive policies, and learning-based sequential control.

To address this gap, this study develops a DES environment that jointly represents inventory flows, receivables, payables, and cash flow movements. Within this integrated environment, an RL agent is trained to optimise working capital policies dynamically in response to stochastic supply chain conditions. This approach allows the agent to learn policies that reduce capital tied up in operations, an objective that has been largely overlooked in existing simulation–RL applications.

3. Materials and Methods

3.1. Integrated DES-RL Framework

Effective management of working capital in multi-echelon SCs requires decision-making approaches that can adapt to stochastic demand patterns, nonlinear system interactions, and interdependent financial constraints. To address these challenges, this study develops an integrated framework that combines deep RL (DRL) with a DES model, which enables a data-driven and dynamically responsive approach to system-wide working capital optimisation.

The proposed framework is illustrated in Figure 1, which presents the interaction loop between the DRL agent and the DES environment. As shown, the DES model serves as a high-fidelity digital representation of the SC, capturing operational processes such as production and order fulfilment as well as financial activities, including cash and credit payments.

Within this simulated environment, the DRL agent observes the evolving state of the system and selects actions that govern production rates, replenishment order quantities, and financial policy parameters (i.e., cash-collection policies). The DES model processes these actions and updates the system state through event-driven operational and financial transitions. The resulting outcomes are fed back to the agent in two ways: the state vector, which includes inventories, backlogs, cash levels, receivables, payables, and the demand from the previous period is used for decision-making, while the reward is computed from a weighted combination of CCCs, backlogs, service shortfall, and retention rate,

thereby completing the iterative learning cycle. This closed-loop interaction, depicted in Figure 1, enables the agent to learn policies that account for delayed effects, cross-echelon interdependencies, and nonlinear system responses.

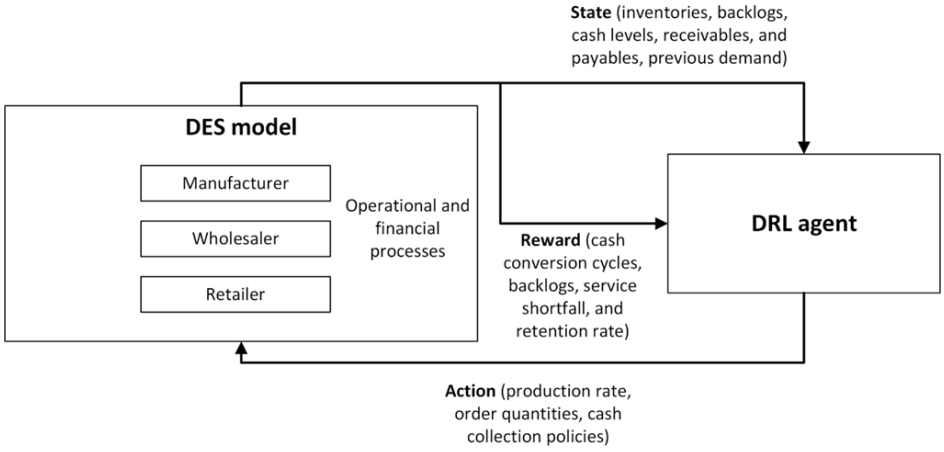

Figure 1. Integrated DES-RL framework.

By coupling a high-fidelity simulation with an adaptive learning mechanism, the integrated DES-RL framework enables the discovery of system-wide working capital policies that are robust to uncertainty and complex operational dynamics. The simulation environment exposes the agent to a broad range of scenarios, including demand disruptions, thereby supporting the development of generalizable decision rules.

3.2. SC Configuration and Simulation Model

In this study, we consider a single-product, three-echelon serial SC comprising a manufacturer, a wholesaler, and a retailer. This structure reflects the configuration commonly observed in fast-moving consumer good (FMCG) SCs. The distribution lead time between each upstream–downstream stage is fixed at one week. Since end customers collect their orders directly from the retailer, no additional lead time is incurred at the final stage of the chain. The manufacturer operates under a weekly production capacity constraint of 44,000 units.

A consolidated summary of the main simulation parameters is provided in Supplementary Table S1 to improve reproducibility. The simulation horizon is 52 weekly decision periods. The maximum weekly retailer order, wholesaler order, and manufacturer production quantities are 22,000, 33,000, and 44,000 units, respectively. Initial inventory levels are set to 17,000, 18,000, and 19,000 units for the retailer, wholesaler, and manufacturer, while initial cash balances are 90,000, 72,000, and 110,000 monetary units. The trade credit period is four weeks for each echelon. Unit selling prices are 30, 25, 20, and 12 for the retailer, wholesaler, manufacturer, and supplier-facing purchase price, respectively. Customer demand is generated from a latent autoregressive process with base demand 9000, linear and quadratic demand coefficients of 10,000, autoregressive coefficient 0.90, and driver noise standard deviation 0.01. The retention state is initialised at 1.00 and bounded between 0.50 and 1.00, with recovery rate 0.05 and stockout sensitivity 0.25. These values were held fixed across all optimisation methods so that performance differences reflect controller behaviour rather than changes in the underlying DES configuration.

When the retailer lacks sufficient inventory to satisfy demand, unmet quantities are backlogged. These backorders, in turn, reduce the SC's service level, defined as the proportion of demand fulfilled immediately from available stock.

All members of the SC employ a periodic-review inventory policy with a one-week review interval. At the beginning of each week, every stage evaluates its on-hand inventory and work-in-progress (WIP), and subsequently issues replenishment orders to its upstream partner.

The operational logic of each SC node follows a standardised sequence:

1. Products ordered in the previous period are received and added to the available inventory;
2. The existing inventory is used to meet downstream replenishment requests as well as any outstanding backorders;
3. Shipments are dispatched downstream, after which inventory and WIP records are updated and new backorders are created if the available stock is insufficient;
4. A non-negative replenishment order is placed with the upstream node based on inventory needs and policy parameters.

In the financial flow, each SC member is required to pay a percentage of their order value at the time of order placement. This percentage can range from 0% to 100%, with the remaining amount paid after the trade credit period. If the cash payment share is 0%, the entire order value is paid after the trade credit period. Conversely, if the cash payment share is 100%, the full order value must be paid upon order placement.

To verify the DES implementation, additional diagnostic checks were conducted and are summarised in Supplementary Table S3. These checks covered three aspects: model verification, statistical output stability, and behavioural validation. The verification audit tested whether the event logic preserved non-negative inventory, backlog, and cash balances; respected order and production capacity limits; maintained affordability constraints; updated lagged demand consistently; kept retention within bounds; and produced non-negative loss components. Across 300 independent replications, all flow-audit checks achieved a 100% pass rate. Behavioural validation was assessed using a separate demand-responsive base-stock diagnostic policy to test whether the DES could reproduce bullwhip-type variance amplification [43]. The bullwhip ratios were computed as the variance of upstream order or production decisions relative to customer-demand variance. These validation checks were used only to verify the internal consistency and behavioural plausibility of the DES, not for optimisation or controller comparison.

Figure 2 shows the structure of the studied SC. The supplier provides the manufacturer with raw materials. Final products flow from the manufacturer to the wholesaler, from the wholesaler to the retailer, and finally to the customer. Cash flows run in the reverse direction, with each transaction split into an immediate cash component and a deferred credit component, which is recorded as accounts receivable (AR) for the seller and accounts payable (AP) for the buyer.

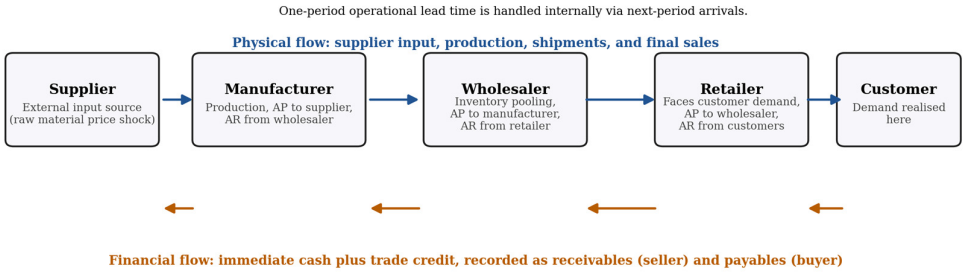

Figure 2. Structure of the studied SC.

3.3. Markov Decision Process Formulation

As illustrated in Figure 3, each decision period begins with the agent observing a fully settled state s_t. Inventory levels in s_t are net of shipments that arrived from the previous period, and cash positions already reflect receivables and payables that matured

at the start of the period. Conditional on this settled state, the agent selects collection policies and replenishment decisions a_t. The environment then realises demand, executes shipments, and records the resulting trade credit transactions. A reward r_t is computed based on service level, customer retention, backlog, inventory, and CCC components. Finally, receivables and payables mature and in-transit shipments are received, which yields the next settled state s_{t+1} for the subsequent period.

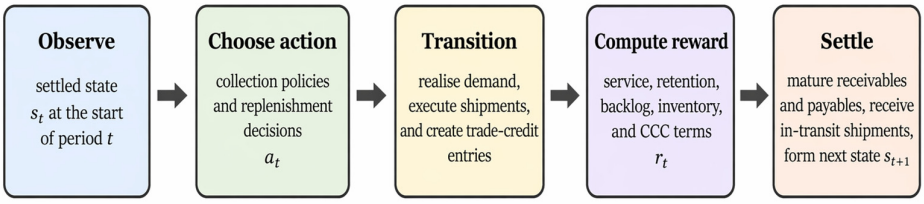

Figure 3. State–action–reward sequence in one decision period.

The state vector comprises three categories of information: physical status variables, financial status variables, and compact predictors of next-period operating conditions. The physical state includes on-hand inventory and backlog at the retailer, wholesaler, and manufacturer. The financial state includes cash balances at each echelon, together with four-bucket receivable and payable profiles that approximate outstanding trade credit obligations. The predictive state includes lagged realised demand, current retention, the current supplier price, and the previous supplier price. Table 1 summarises the full state vector observed at the decision epoch.

Table 1. State variables observed at the decision epoch.

State Block	Variables	Interpretation
Physical state	I_t^r, I_t^w, I_t^m; B_t^r, B_t^w, B_t^m	On-hand inventory and backlog at retailer, wholesaler, and manufacturer.
Cash state	C_t^r, C_t^w, C_t^m	Liquid cash available at each echelon at the decision epoch.
Receivables	$AR_{t,k}^r$, $AR_{t,k}^w$, $AR_{t,k}^m$, k = 0, . . ., 3	Outstanding receivables grouped by time-to-maturity bucket.
Payables	$AP_{t,k}^r$, $AP_{t,k}^w$, $AP_{t,k}^m$, k = 0, . . ., 3	Outstanding payables grouped by time-to-maturity bucket.
Demand/retention	D_{t-1}, R_t	Lagged realised demand and current retention level.
Input-price signal	P_t^s, P_{t-1}^s	Current and previous supplier prices faced by the manufacturer.

The action is a five-dimensional continuous vector. Two elements control the immediate cash-collection fractions applied to wholesaler-to-retailer and manufacturer-to-wholesaler transactions. The remaining three elements determine the retailer order, wholesaler order, and manufacturer production quantities as fractions of their respective capacity limits. Because buyers must pay the immediate cash component at the time of transaction, operational decisions are automatically constrained by affordability as well as by inventory availability. Equation (1) defines the five-dimensional action vector:

$$a_t = [\mu_t^w, \mu_t^m, o_t^r, o_t^w, p_t^m] \quad (1)$$

Demand is driven by a latent autoregressive factor, transformed into a nonlinear baseline mean, then sampled with Gaussian noise, and finally filtered through the current

retention level. In the shock experiment, this baseline is further multiplied by a temporary exogenous shock factor over short windows. This makes the shock setting harder for all methods because the lagged demand observed in the state becomes temporarily less informative about the demand that is about to be realised. Equation (2) summarises this stylised demand process:

$$\mu_t = b_0 + b_1\, z_t + b_2\, {z_t}^2,\ Dt = R_t \cdot max\ (0.3\ \mu_t, \mathcal{N}\ (\mu_t, \sigma_t)) \quad (2)$$

In the shock experiments, the baseline demand process is subjected to temporary multiplicative disturbances. Specifically, the nominal mean is multiplied by a step-specific factor ξ_t. For each episode, up to three non-overlapping shock windows are sampled; each lasts between two and four periods and is buffered away from the episode boundaries so that the policy experiences pre-shock and post-shock settling periods.

$$\xi_t = 1\ outside\ the\ shock\ windows;\ \xi_t = m_j\ for\ s_j \leq t < e_j \quad (3)$$

Within each shock window, the multiplier m_j is sampled either from a negative interval [0.75, 0.90] or from a positive interval [1.10, 1.25]. The realised demand process in the shock experiment therefore becomes

$$D_t = R_t \cdot max\ (0.3\ \mu_t\ \xi_t, \mathcal{N}\ (\mu_t\ \xi_t, \sigma_t \cdot max\ (0.5, \xi_t))) \quad (4)$$

This makes the shock benchmark a transient demand-perturbation test rather than a permanent regime change. The controller still observes only lagged demand, retention, and the financial–operational state, so ξt creates precisely the sort of short-run forecast error that can expose weaknesses in a policy that relies too heavily on the most recent demand signal.

Supplementary Table S1 reports the full set of fixed simulation parameters used in the experiments. These include operational capacities, initial inventories and cash balances, selling prices, trade credit periods, demand-process coefficients, retention parameters, backlog aggregation weights, CCC denominator floors, CCC echelon weights, the CCC penalty threshold, reward weights, episode length, and random-seed handling. The same parameterisation is used across all controllers and both experimental settings unless explicitly stated otherwise.

3.4. Reward and Working Capital Formulation

The one-period reward is the negative of a weighted loss. Four managerial failure modes are penalised: unmet customer demand, deterioration in customer retention, accumulated backlog across echelons, and excessive working capital intensity through the cash conversion cycle. The first three terms capture service quality and operational pressure. The fourth term is what makes the benchmark economically distinctive: a policy is not credited merely for moving inventory, but for doing so while keeping working capital proportional to realised throughput. Equation (5) states the reward construction:

$$L_t = A^*\ S_t + B^*\ \Delta R_t + C^*\ B_t + D^*\ CCC_t, \quad rt = -L_t \quad (5)$$

The reward weights were selected to encode a managerial priority ordering rather than to represent universal economic constants. Service failure and retention deterioration receive the largest penalties because they directly affect customer-facing performance and future demand potential. Backlog receives an intermediate penalty because it reflects operational congestion and delayed fulfilment across the chain. The CCC term receives a lower but persistent penalty because working capital exposure is important, but the aim

is not to minimise CCC at the expense of service and retention. The baseline weights are therefore set to prioritise avoidance of customer-facing deterioration while still discouraging excessive working capital intensity. Because these weights influence the optimisation landscape, supplementary reward-weight sensitivity analyses were conducted to assess whether the main conclusions remain stable under representative perturbations of the service/backlog and CCC terms.

The service-loss component is the normalised fresh shortfall. If F_t denotes the fresh unmet demand in period t and $\overline{D}$ denotes the nominal demand scale, then

$$S_t = F_t / \overline{D} \tag{6}$$

Retention is deliberately modelled as a dynamic state rather than as a static penalty. The stockout rate is $\psi_t = F_t / \max(D_t, 1)$, so the immediate retention damage rises with both the fraction of current demand that goes unmet and the latent-demand condition z_t. Recovery is gradual rather than immediate, which yields the next-period retention law

$$\psi_t = F_t / max(D_t, 1) \tag{7}$$

$$R_{t+1} = clip(R_t - \lambda(0.5 + z_t)\, \psi_t + \rho(1 - R_t), R_{min}, R_{max}) \tag{8}$$

The retention-loss term that enters the reward is then the normalised one-step deterioration

$$\Delta R_t = max(0, R_t - R_{t+1})/(1.5\, \lambda) \tag{9}$$

This construction is important economically. A stockout does not only create an immediate service penalty; it also lowers future effective demand through R_t, which means repeated service failures can damage the system beyond the current period.

Backlog is aggregated across echelons with weights that reflect managerial urgency. Retailer backlog carries the highest weight because it is closest to the customer, while upstream backlog is discounted:

$$B_t = (\omega_r \cdot {B_t}^r + \omega_w \cdot {B_t}^w + \omega_m \cdot {B_t}^m)/\overline{D} \tag{10}$$

The CCC term is defined echelon by echelon. Pre-transition inventory is valued at the relevant upstream cost proxy because it represents capital already tied up before the current decision is executed. Post-transition receivables and payables are then divided by realised sales and cost-of-goods-sold (COGS) denominators, respectively. To prevent numerical explosions in very low-throughput periods, the sales and COGS denominators are stabilised with positive floors. Each echelon CCC is then penalised only above a threshold, which means the reward does not punish every unit of working capital equally; it punishes economically excessive exposure. Equation (10) gives the echelon-wise CCC term:

$${CCC_t}^e = {V_{inv,t}}^{e,pre}/{COGS_t}^e + {AR_t}^{e,post}/{Sales_t}^e - {AP_t}^{e,post}/{COGS_t}^e \tag{11}$$

The aggregate CCC penalty used in the reward is a thresholded weighted sum of the echelon-wise raw CCC values. If τ is the penalty threshold and η_r, η_w, and η_m are the echelon weights, then

$$CCC_t = \eta_r \cdot max({CCC_t}^r - \tau, 0) + \eta_w \cdot max({CCC_t}^w - \tau, 0) + \eta_m \cdot max({CCC_t}^m - \tau, 0) \tag{12}$$

The threshold τ is treated as a design parameter that defines the point at which working capital exposure becomes economically excessive relative to throughput. A thresholded formulation is used because some inventory, receivables, and payables are necessary for

normal operation; the reward should therefore penalise disproportionate exposure rather than every unit of working capital. The baseline value $\tau = 0.30$ was selected to activate the CCC penalty only when echelon-wise exposure moved beyond a moderate operating range. To test whether the conclusions depend on this choice, a supplementary sensitivity analysis varied τ around the baseline specification and compared the resulting PPO performance. This analysis is reported in the Supplementary Materials and is used to verify that the main conclusions are not driven by a single CCC-threshold setting.

The reward therefore penalises working capital exposure only once the echelon-wise CCC terms move into an economically inefficient regime. This avoids punishing every unit of inventory or trade credit equally and instead focuses the optimisation on excessive exposure relative to throughput.

This design creates an intuitive economic interpretation. A policy can increase sales and still be rewarded if it does not allow receivables and inventory to expand disproportionately. Conversely, a policy that carries stock and defers collection without maintaining throughput will see the CCC term increase. This is why some methods in the experiments achieve very low backlog but do not dominate the reward: they gain on service but surrender part of that gain by carrying a less efficient working capital profile.

3.5. Experimental Protocol and Statistical Reporting

All controllers were evaluated under the same DES environment, state representation, action bounds, reward formulation, and episode horizon. The no-disruption benchmark evaluates performance under the baseline latent-demand process, while the disruption benchmark introduces temporary multiplicative demand shocks. In the main disruption experiment, shock windows may be positive or negative across the evaluation distribution, providing an aggregate robustness test under transient demand perturbations.

To account for stochasticity in both training and evaluation, the experiments were conducted using three independent training seeds: 42, 123, and 3000. For each controller, one policy was trained per seed. Each trained policy was then evaluated over 5000 independent stochastic episodes using a fixed evaluation-seed protocol. Performance was summarised using pooled episode-level statistics across the final evaluation episodes. The three training seeds ensure that the reported evaluation sample includes independently trained policies, while the pooled summaries capture variability across stochastic evaluation episodes. The main reported diagnostics are mean episode return, service shortfall, backlog, CCC loss, retention, inventory, and cash. Standard deviations and 95% confidence intervals were computed to distinguish systematic performance differences from simulation noise.

Additional supplementary analyses were conducted to address robustness of the experimental specification. First, the CCC penalty threshold was varied around the baseline value. Second, the reward-weight specification was perturbed by increasing the relative emphasis on service/backlog terms and on the CCC term. Third, because the main disruption setting can pool different shock directions, a controlled shock-direction analysis was conducted using saved trained PPO policies. In this analysis, the same PPO policies were re-evaluated under episodes containing only positive demand shocks and episodes containing only negative demand shocks. This controlled evaluation avoids ambiguity from mixed-shock episodes and tests whether PPO's disruption performance is driven by one specific shock direction.

3.6. Optimisation Approaches

All five methods control the same environment and produce the same five action components. The substantive difference lies in the structure imposed on the policy and the mechanism used to search for good parameter values. The benchmark was intentionally

built as a ladder of modelling commitments. The fixed-threshold methods impose the strongest structure and are easiest to interpret. The adaptive threshold methods relax that structure by allowing targets to respond to state signals. PPO removes the threshold template entirely and learns a direct nonlinear mapping from state to action.

Formally, every controller defines a state-to-action mapping $a_t = \pi_\theta(x_t)$, where x_t is the settled state and $a_t = [\mu_t^w, \mu_t^m, o_t^r, o_t^w, p_t^m]$ collects the two cash-collection decisions and the three capacity-scaled replenishment decisions. The search-based approaches differ in the structure imposed on π_θ and in the way the parameter vector θ is optimised; PPO differs because it learns π_θ directly as a nonlinear policy.

$$a_t = \pi_\theta(x_t) = [\mu_t{}^w, \mu_t{}^m, o_t{}^r, o_t{}^w, p_t{}^m] \tag{13}$$

3.6.1. Simple CEM

Simple CEM uses a five-parameter policy. Two parameters define the cash-collection fractions applied to wholesaler and manufacturer sales, while three parameters define fixed-target inventory positions for the retailer, wholesaler, and manufacturer. Let the echelon inventory position be the difference between on-hand inventory and backlog. If the target exceeds the current inventory position, the policy orders or produces enough to close the gap, subject to capacity and affordability constraints. Because the targets are fixed, this controller cannot respond explicitly to lagged demand or retention pressure.

$$IP_t{}^e = I_t{}^e - B_t{}^e \tag{14}$$

$$o_t{}^r = clip((T_r - IP_t{}^r)/O_{max}{}^r, 0, 1) \tag{15}$$

$$\theta_i{}^{(g)} \sim \mathcal{N}(\mu^{(g)}, diag(\sigma^{2(g)})) \tag{16}$$

The Simple CEM parameter vector is $\theta = [\mu_w, \mu_m, T_r, T_w, T_m]$. Here T_r, T_w, and T_m are fixed-target inventory positions for the retailer, wholesaler, and manufacturer. The policy is therefore static in the sense that the same targets are used in every state, and any variation in realised actions arises only through current inventory positions, affordability constraints, and capacity limits.

$$\theta = [\mu_w, \mu_m, T_r, T_w, T_m] \tag{17}$$

CEM then optimises this vector by repeatedly sampling a Gaussian population, evaluating each candidate over multiple episodes, retaining the elite fraction, and updating the sampling distribution toward the elite mean and dispersion. In other words, it is a distribution-based optimiser over a deliberately simple and interpretable policy class.

CEM searches over this parameter vector by sampling a population from a Gaussian distribution, evaluating each candidate over multiple episodes, keeping the elite fraction, and updating the Gaussian toward the elite mean and elite standard deviation. In practice, Simple CEM is a disciplined direct-search baseline: it is interpretable, but only as expressive as the fixed-threshold structure it is allowed to optimise.

3.6.2. Simple GA

Simple GA uses the same five-parameter fixed-threshold policy as Simple CEM. The difference lies only in the search mechanism. Instead of maintaining and updating a parametric sampling distribution, the genetic algorithm keeps an explicit population of candidate parameter vectors, selects parents by tournament selection, creates children by blend crossover, and perturbs individual genes by Gaussian mutation.

$$\theta_{child} = \lambda\, \theta_1 + (1 - \lambda)\, \theta_2 \tag{18}$$

$$\theta_{j,new} = \theta_j + \mathbb{1}(u_j < p_{mut}) \cdot \varepsilon_j \quad (19)$$

Simple GA searches over the same fixed-threshold vector θ = [μw, μm, Tr, Tw, Tm], so any performance difference relative to Simple CEM comes from the optimiser rather than from the policy class. Tournament selection, blend crossover, and Gaussian mutation allow the algorithm to preserve multiple promising threshold rules simultaneously and refine them over generations without imposing a single Gaussian search distribution on the population.

The managerial interpretation is straightforward. Simple GA still searches for a static policy, but the population-based search can exploit multiple local improvements at once and can therefore fine-tune a fixed rule more aggressively than a single Gaussian search distribution. In the empirical results, this is why Simple GA generally improves on Simple CEM even though the underlying policy class is unchanged.

3.6.3. Adaptive CEM

Adaptive CEM expands the policy class from five to eleven parameters. Each echelon target is now allowed to vary with lagged demand and retention pressure. This means the retailer, wholesaler, and manufacturer each have a base target, a demand-response coefficient, and a retention-pressure coefficient. The collection-policy parameters remain explicit decision variables. The search procedure is the same CEM logic described above, but it is now operating over a more expressive policy family.

$$T_t^e = \alpha_e + \beta_e D_{t-1} + \gamma_e (1 - R_t) \quad (20)$$

The adaptive threshold vector expands to θ = [μ_w, μ_m, α_r, α_w, α_m, β_r, β_w, β_m, γ_r, γ_w, γ_m]. The α_e parameters define base-stock targets, the β_e coefficients determine how strongly each echelon responds to lagged demand, and the γ_e coefficients determine how strongly it reacts when retention weakens.

$$\theta = [\mu_w, \mu_m, \alpha_r, \alpha_w, \alpha_m, \beta_r, \beta_w, \beta_m, \gamma_r, \gamma_w, \gamma_m] \quad (21)$$

$$T_t^e = \alpha_e + \beta_e D_{t-1} + \gamma_e (1 - R_t) \quad (22)$$

Adaptive CEM therefore retains full interpretability—the decision rule is still a transparent threshold law, but it allows the targets themselves to move with the state. The CEM search then operates over this richer eleven-dimensional parameterisation.

This policy is still fully interpretable. A positive demand coefficient means the echelon raises its target when the lagged signal is high; a positive retention-pressure coefficient means the echelon increases protection stock when retention is already under threat. The weakness is that responsiveness is still linear and threshold-based. Under baseline conditions this can be enough to improve materially on a fixed rule, but under shocks the lagged demand signal can become misleading precisely when the policy most wants a good forecast.

3.6.4. Adaptive GA

Adaptive GA retains the same eleven-parameter adaptive threshold law as Adaptive CEM but changes the optimiser from cross-entropy search to evolutionary population search.

Adaptive GA uses the same eleven-parameter adaptive threshold law as Adaptive CEM but optimises it through evolutionary population updates rather than Gaussian elite updates. Because the adaptive parameter space is richer and more multimodal than the fixed-threshold space, the genetic algorithm can preserve several promising state-

responsive policies simultaneously and recombine them across generations. This is why Adaptive GA often delivers a stronger stationary trade-off between service, backlog, and CCC than the simpler search baselines.

3.6.5. Proximal Policy Optimisation (PPO)

PPO removes the handcrafted threshold map altogether. A neural policy reads the full observed state and outputs the five action components directly. This gives PPO the highest representational flexibility of all methods in the benchmark. Because the objective is the expected cumulative reward, PPO is not tuned to minimise one metric at a time. It learns a compromise over service, backlog, retention, and working capital efficiency as expressed by the reward function.

$$max_{\varphi}\ \mathbb{E}[min(r_t(\varphi)\ \hat{A}_t,\ clip(r_t(\varphi), 1-\varepsilon, 1+\varepsilon)\ \hat{A}_t)] \tag{23}$$

In PPO, the control law is not written as a threshold formula at all. Instead, a neural policy reads the full settled state and directly outputs the five action components, while a value network estimates continuation value to stabilise policy updates.

$$a_t \sim \pi_{\varphi}(\cdot \mid x_t), \qquad V_{\varphi}(x_t) \approx \mathbb{E}[\textstyle\sum_{k\geq 0} \gamma^k\ r_{t+k} \mid x_t] \tag{24}$$

This means PPO can exploit nonlinear interactions among inventories, backlogs, cash balances, receivables, payables, lagged demand, retention, and supplier prices that are inaccessible to the linear threshold maps used by the search-based approaches. The cost of that flexibility is lower coefficient-level interpretability.

The clipped surrogate objective is important because it stabilises learning while still allowing the policy to improve from experience. Relative to the structured methods, PPO sacrifices some interpretability but can discover nonlinear state–action relationships that a threshold policy cannot express.

To partially address the lower transparency of PPO, a post hoc surrogate interpretability analysis was conducted. State–action trajectories were collected from the saved PPO policies, and a separate Random Forest surrogate was trained to approximate each of the five PPO action components from the observed state variables. Surrogate fidelity was evaluated on held-out episodes. Random Forest feature importance was then reported as the top-ten state variables for each PPO action component, rather than as grouped importance scores, so that the interpretation remains directly tied to observable state variables. This analysis was used only for interpretation and did not affect policy training, evaluation, or optimisation. The resulting feature-importance values are therefore interpreted as surrogate-based associations rather than causal explanations of the internal neural policy.

To ensure reproducibility and comparability, the main optimisation settings are summarised in Table 2. All methods use the same DES environment, state representation, action bounds, reward formulation, and episode horizon. The search-based methods are evaluated using the same population size, optimisation horizon, number of stochastic simulations per candidate during training, and final evaluation protocol. Simple CEM and Simple GA optimise a five-parameter fixed-threshold policy, while Adaptive CEM and Adaptive GA optimise an eleven-parameter state-responsive threshold policy. PPO learns a direct nonlinear state–action mapping using the same five action components and action bounds. Each method was trained using the independent random seeds 42, 123, and 3000, and each trained policy was subsequently evaluated over 5000 stochastic episodes.

Table 2. Optimisation and evaluation settings.

Method	Training and Optimisation Settings	Final Evaluation
Simple CEM	Policy: 5-parameter fixed-threshold rule. Seeds: 42, 123, 3000. CEM settings: population = 200; iterations = 150; elite fraction = 0.20. Training fitness: 8 stochastic simulations per candidate.	5000 episodes per trained seed
Simple GA	Policy: 5-parameter fixed-threshold rule. Seeds: 42, 123, 3000. GA settings: population = 200; generations = 150; elitism = 6; crossover probability = 0.90; mutation probability = 0.30; tournament size = 3. Training fitness: 8 stochastic simulations per candidate.	5000 episodes per trained seed
Adaptive CEM	Policy: 11-parameter adaptive threshold rule. Seeds: 42, 123, 3000. CEM settings: population = 200; iterations = 150; elite fraction = 0.20. Training fitness: 8 stochastic simulations per candidate.	5000 episodes per trained seed
Adaptive GA	Policy: 11-parameter adaptive threshold rule. Seeds: 42, 123, 3000. GA settings: population = 200; generations = 150; elitism = 6; crossover probability = 0.90; mutation probability = 0.30; tournament size = 3. Training fitness: 8 stochastic simulations per candidate.	5000 episodes per trained seed
PPO	Policy: neural state–action mapping. Seeds: 42, 123, 3000. Training budget: 5,000,000 timesteps. Settings: learning rate = 5×10^{-5}; rollout length = 2048; batch size = 256; gamma = 0.995; GAE lambda = 0.95; clip range = 0.20; entropy coefficient = 0.002; value coefficient = 0.50; hidden layers = [256, 256].	5000 episodes per trained seed

The training evaluation budget and final evaluation budget serve different roles. During CEM and GA optimisation, each candidate policy is evaluated over eight stochastic simulations to estimate its fitness within the search process. After training, the best policy obtained from each training seed is evaluated over 5000 independent episodes to obtain stable performance estimates. For PPO, training is conducted for 5,000,000 environment timesteps, after which the saved policy from each training seed is evaluated over the same 5000-episode final evaluation protocol.

Further reproducibility and robustness details are provided in the Supplementary Materials. These include the full simulation parameter table, optimisation hyperparameters, DES verification and bullwhip validation outputs, pooled evaluation summaries, CCC-threshold sensitivity, targeted reward-weight sensitivity, controlled shock-direction analysis, representative dynamic diagnostic plots, and PPO surrogate fidelity and top-ten feature-importance results.

To improve methodological transparency, all experimental settings were kept consistent across controllers unless a supplementary sensitivity test explicitly changed one component. The DES parameterisation, including demand-process coefficients, initial inventories and cash balances, capacity limits, trade credit periods, retention parameters, backlog weights, the CCC threshold, and reward weights, is reported in the Supplementary Materials. The optimisation settings for CEM, GA, and PPO are also reported explicitly, including population sizes, iteration or generation counts, PPO learning parameters, training seeds, and final evaluation episodes. This separation ensures that the main experiments compare differences in policy structure and learning capability rather than differences in environment specification, evaluation budget, or reporting procedure.

4. Results

4.1. No-Disruption

The no-disruption scenario evaluates all five controllers under the baseline latent-demand process, absent any exogenous shocks. The primary comparison is therefore not simply the mean return each method achieves, but the operational pathways through which those returns are generated. Improvements in reward can arise from several sources, suppressing service shortfalls, reducing backlog accumulation, protecting customer retention, operating with a lower CCC penalty, or some combination thereof. Reporting multiple diagnostics in parallel makes it possible to distinguish policies that deliver genuine economic improvements from those that merely shift costs or congestion across different parts of the system.

Table 3 shows that Adaptive GA achieves the best mean return in the no-disruption benchmark, while PPO delivers the strongest customer-facing performance. PPO has the lowest service shortfall, the lowest backlog, and the highest retention, but it does not minimise the CCC loss; Adaptive CEM and Adaptive GA achieve lower average CCC penalties. The no-disruption result therefore indicates a trade-off rather than uniform dominance. Adaptive GA provides the strongest overall structured-policy performance under stable demand, whereas PPO uses its nonlinear state–action mapping to protect service and backlog more aggressively while accepting a higher working capital penalty.

Table 3. No-disruption results (mean ± SD).

Method	Average Cash Total	Average Inventory Total	Average Retention	Average CCC Loss	Average Backlog Total	Average Service Shortfall	Mean Return
Simple CEM	8,407,197 ± 128,701	54,589 ± 551	0.9965 ± 0.0053	0.159 ± 0.024	170.29 ± 244.70	18.22 ± 25.33	−10.52 ± 3.22
Simple GA	8,405,438 ± 130,697	54,950 ± 644	0.9969 ± 0.0048	0.165 ± 0.025	113.14 ± 194.36	16.20 ± 23.26	−10.32 ± 2.71
Adaptive CEM	8,415,817 ± 129,184	54,084 ± 211	0.9970 ± 0.0038	0.148 ± 0.016	194.25 ± 193.44	15.44 ± 18.55	−10.14 ± 2.40
Adaptive GA	8,415,911 ± 132,644	54,542 ± 273	0.9977 ± 0.0043	0.156 ± 0.018	83.83 ± 108.43	12.01 ± 23.70	−9.47 ± 2.03
PPO	8,409,099 ± 133,780	55,869 ± 1920	0.9989 ± 0.0027	0.178 ± 0.038	8.88 ± 14.04	5.30 ± 12.79	−9.61 ± 1.97

Values are reported as mean ± standard deviation over 15,000 evaluation episodes, corresponding to three independently trained policies and 5000 evaluation episodes per trained seed. Full pooled confidence intervals are provided in the Supplementary Materials.

Figure 4 complements Table 3 by summarising the pooled episode-return distributions and the metric-wise rankings across the five methods in the no-disruption benchmark. The return distribution shows the strongest stationary performance for Adaptive GA, while the rank heatmap shows that PPO ranks first on service shortfall, backlog, retention, and the corresponding service- and retention-related losses. The combined evidence indicates that no-disruption performance is governed by a trade-off: Adaptive GA provides the best overall structured-policy return, whereas PPO provides the strongest customer-facing and operational robustness. Detailed pooled confidence intervals are reported in Supplementary Table S4.

4.2. Demand Disruption

The demand disruption scenario retains the same settled-state logic, reward structure, and policy classes as the no-disruption benchmark, but introduces short-lived multiplicative shocks to customer demand. These shocks are not observed directly by the policy: the agent still observes only the lagged demand signal, retention, and the financial–operational state.

The experiment therefore tests not only robustness, but a specific managerial capability: how effectively each controller contains service and working capital damage when realised demand temporarily deviates from what the lagged signal would suggest.

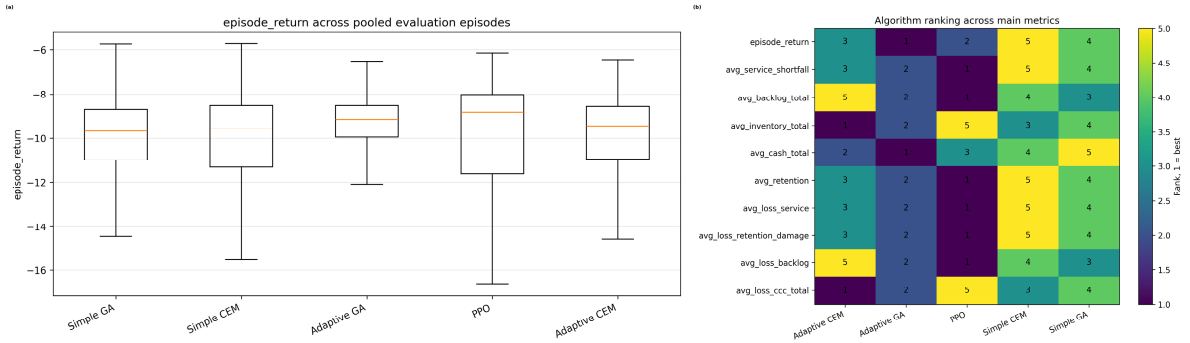

Figure 4. No-disruption summary plots: (**a**) pooled episode-return boxplot; (**b**) rank heatmap across the main performance metrics.

The shock process takes the form of a temporary multiplicative perturbation to the nominal demand mean rather than a permanent change in the latent autoregressive law. In each episode, up to three non-overlapping shock windows are sampled, each lasting between two and four periods and buffered away from the episode boundaries. Within a shock window, the multiplier m_j is drawn from either [0.75, 0.90] to generate a negative shock or [1.10, 1.25] to generate a positive shock:

$$\xi_t = m_j \text{ for } s_j \leq t < e_j, \qquad \text{with } m_j \in [0.75, 0.90] \text{ or } m_j \in [1.10, 1.25]; \qquad \xi_t = 1 \text{ otherwise} \tag{25}$$

Realised demand is thus generated from the shocked mean. Because policies observe only lagged demand and not the contemporaneous shock, the disruption benchmark measures how gracefully each controller degrades when the most recent demand signal becomes temporarily misleading. Table 4 reports the resulting performance under demand disruption $\mu_t \xi_t \zeta_t$.

Table 4. Demand disruption results (mean ± SD).

Method	Average Cash Total	Average Inventory Total	Average Retention	Average CCC Loss	Average Backlog Total	Average Service Shortfall	Mean Return
Simple CEM	8,205,667 ± 175,799	63,500 ± 1105	0.9804 ± 0.0185	0.361 ± 0.053	621.65 ± 545.11	106.01 ± 94.04	−29.20 ± 10.65
Simple GA	8,199,368 ± 174,282	63,108 ± 1556	0.9787 ± 0.0192	0.356 ± 0.057	656.38 ± 599.64	114.68 ± 97.21	−29.57 ± 11.07
Adaptive CEM	8,211,331 ± 169,942	61,050 ± 644	0.9769 ± 0.0168	0.324 ± 0.035	419.48 ± 317.26	123.61 ± 86.18	−26.78 ± 8.02
Adaptive GA	8,228,635 ± 172,763	59,649 ± 1780	0.9756 ± 0.0236	0.289 ± 0.036	356.09 ± 300.85	134.42 ± 125.67	−25.09 ± 8.10
PPO	8,314,267 ± 206,826	62,655 ± 1609	0.9960 ± 0.0078	0.314 ± 0.038	248.68 ± 283.60	24.55 ± 48.78	−19.27 ± 3.36

All controllers deteriorate when realised demand diverges from the lagged signal, but the extent of deterioration differs substantially. The fixed-threshold methods incur the weakest disruption performance, with high service shortfall and backlog. Adaptive GA and Adaptive CEM improve the mean return relative to the fixed baselines, confirming the value of state-dependent threshold adjustment under disruption. PPO remains clearly strongest overall: it achieves the best mean return, the lowest service shortfall, the lowest backlog, and the highest retention. However, PPO does not dominate every individual metric; Adaptive GA has the lowest average CCC loss under disruption. The disruption results

therefore show that PPO's advantage comes from a superior overall balance, especially in service and backlog containment, rather than from uniformly minimising working capital exposure.

Figure 5 provides an aggregate view of controller robustness under temporary demand disruption. The episode-return distribution shows that PPO achieves the most favourable disrupted performance, with a tighter and higher-return interquartile range than the other controllers. The rank heatmap confirms that PPO ranks first on overall return, service shortfall, backlog, cash, retention, and the associated service- and backlog-related losses, while Adaptive GA remains the best performer on the CCC-loss metric. This supports the interpretation from Table 4 that PPO provides the strongest disruption robustness overall, but does not dominate every individual metric.

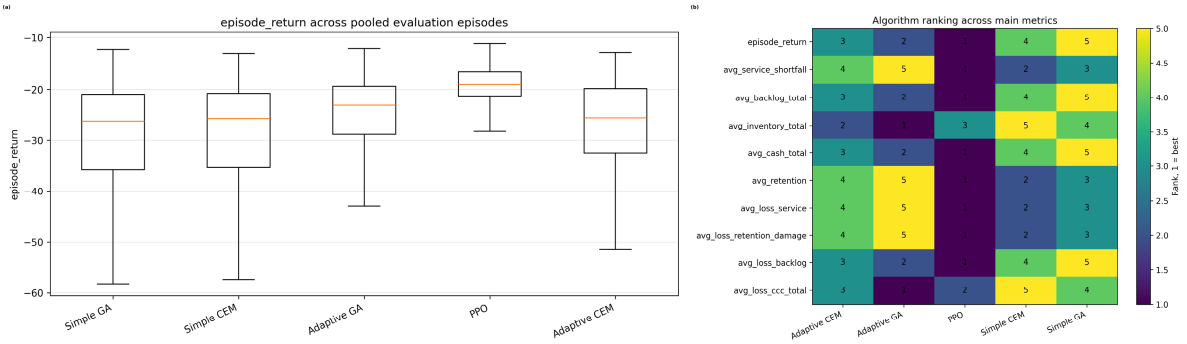

Figure 5. Demand disruption summary plots: (**a**) pooled episode-return boxplot; (**b**) rank heatmap across the main performance metrics.

4.3. Supplementary Robustness and Interpretability Checks

Supplementary analyses were used to verify that the main conclusions are not artefacts of a single simulation specification. The DES verification and bullwhip validation outputs are reported concisely in Supplementary Table S3. The controlled shock-direction analysis re-evaluates the saved PPO policies under episodes containing only positive demand shocks and only negative demand shocks; these results are reported in Supplementary Table S5. Positive shocks produce materially higher service shortfall and backlog than negative shocks, while negative shocks generate higher CCC-loss exposure, confirming that the pooled disruption result is not driven by a single shock direction.

The CCC-threshold and targeted reward-weight sensitivity analyses are reported in Supplementary Tables S6 and S7. The CCC-threshold sensitivity analysis was conducted over three PPO seeds, while the targeted reward-weight sensitivity analysis was conducted on one PPO seed as a behavioural sensitivity check. These experiments focus on CCC loss, inventory, backlog, and retention rather than on absolute return values because changing reward weights or the CCC threshold changes the reward scale itself. They are therefore interpreted as behavioural sensitivity checks rather than direct return comparisons across reward specifications. PPO interpretability is assessed through Random Forest surrogate models trained to approximate each PPO action component from observed state variables. The surrogate fidelity and top-ten feature-importance results are reported in Supplementary Tables S8 and S9. Representative dynamic diagnostics for the no-disruption and disruption settings are provided in Supplementary Figures S1–S12. These results are interpreted as post hoc surrogate associations rather than causal explanations of the neural policy.

5. Concluding Discussion

This study examined alternative control architectures for a multi-echelon SC operating under delayed demand information, comparing fixed heuristic policies, adaptive parametric policies, and a reinforcement learning controller under both no-disruption and demand disruption conditions. The key finding is that the relative value of each controller depends strongly on the operating environment. In the no-disruption setting, Adaptive GA achieved the strongest mean return, showing that a well-tuned adaptive parametric policy can be highly effective when demand evolves within a stable stochastic regime. Under demand disruption, however, PPO showed a clear advantage, achieving the strongest overall return while substantially reducing service shortfall, backlog, and retention loss. This suggests that PPO's main value lies not in uniformly dominating all metrics under all conditions, but in providing greater resilience when lagged demand information becomes temporarily misleading.

The results suggest a clear hierarchy of practical usefulness. Simple CEM and Simple GA serve as transparent fixed-rule baselines. Their value lies precisely in their interpretability: managers can readily explain their logic, implement them with low overhead, and stress-test their behaviour under alternative scenarios. This confirms findings from recent inventory management studies showing that fixed-threshold or base-stock rules remain attractive in practice due to their simplicity and governance advantages, even though they underperform in dynamic environments [44,45]. However, the results also reaffirm a structural limitation of such policies: because their targets are static, they cannot react to changes in the observed operating state, leading to persistent backlog accumulation and service shortfalls as demand conditions evolve.

Allowing protection targets to depend on the state of the system addresses this structural weakness. Both Adaptive CEM and Adaptive GA improve performance by conditioning decisions on lagged demand and retention pressure, consistent with recent work on adaptive and digitally enabled SC control showing that state dependence improves demand–supply coordination under delayed information [46,47]. Within this class, Adaptive GA emerges as the strongest stationary performer. Evolutionary search identifies more effective state-contingent mappings than uniform or manually specified threshold adjustments, yielding better coordination across echelons while preserving operational and financial stability.

The demand disruption results reveal an important boundary of parametric adaptation that has received limited attention in the existing literature. When realised demand temporarily deviates from otherwise informative lagged signals, adaptive threshold-based policies respond indirectly and with delay. Because adjustments are triggered only after backlog or retention has already deteriorated, backlog and working capital stress escalate during disruption windows. This finding refines recent claims about the robustness of adaptive heuristics by showing that state dependence alone is insufficient when informational mismatch is short-lived rather than structural.

The RL controller adds a further layer of flexibility by learning the state–action mapping directly, but its advantage is most pronounced under disruption rather than in the stationary benchmark. Its strong disruption performance aligns with recent studies demonstrating that policy-gradient methods such as PPO can outperform classical and adaptive heuristics in complex, stochastic SC environments by internalising delayed feedback and nonlinear dynamics [23,48]. In this respect, the results corroborate the growing consensus that learned policies are particularly effective when system behaviour is difficult to specify analytically and when short-term informational mismatch requires rapid coordination across multiple decision levers.

Crucially, PPO is not the best method because it minimises the cash conversion cycle in isolation. Rather, it achieves a strong overall balance across the reward components. In the stationary setting, PPO sharply reduces service shortfall and backlog while keeping CCC exposure within a controlled range, although Adaptive CEM and Adaptive GA achieve lower average CCC-loss values. This explicit demonstration of coordinated operational–financial control extends much of the existing reinforcement learning literature, which often embeds financial considerations only as aggregated cost terms. The results show that learned policies can stabilise working capital dynamics while simultaneously improving service and backlog outcomes, rather than optimising one objective in isolation.

The disruption benchmark provides the clearest evidence of PPO's practical value. While all controllers deteriorate when demand signals become temporarily misleading, PPO maintains greater stability in service shortfall, backlog, retention, and overall return. Its CCC loss is not the lowest in the disruption benchmark; Adaptive GA achieves the lowest average CCC-loss term. The PPO advantage should therefore be interpreted as a superior operational–financial trade-off rather than uniform dominance across all individual metrics. This behaviour reflects PPO's ability to redistribute stress dynamically across operational and financial levers without inducing severe service or backlog instability.

The supplementary robustness checks reinforce this interpretation. The controlled shock-direction analysis shows that the PPO result is not an artefact of pooling positive and negative disruptions, while the CCC-threshold and reward-weight analyses show how CCC loss, inventory, backlog, and retention respond to reasonable reward-design perturbations. The surrogate analysis further indicates that PPO decisions can be approximated by interpretable state-dependent relationships, with the top-ranked features aligning with operational and financial state variables such as lagged demand, inventories, cash, payables, and receivables. The surrogate should not be read as a causal explanation of the neural policy, but it provides useful evidence that the learned actions are not arbitrary.

From a managerial perspective, the results indicate that interpretable threshold-based policies remain valuable when transparency, ease of implementation, and governance considerations are paramount. These rules are well suited to relatively stable environments where decision logic must be simple, auditable, and easy to communicate across the organisation. Adaptive parametric policies provide substantial improvements when moderate responsiveness is required, allowing managers to handle variability without fully abandoning familiar control structures. However, when operating conditions become sufficiently volatile that fixed or linearly adaptive rules cannot respond adequately to temporary informational mismatches, direct policy learning becomes increasingly attractive as it enables coordinated adjustment across service levels, backlogs, and working capital exposure.

This study has several limitations that should be acknowledged. First, it focuses on demand uncertainty and temporary demand shocks and does not incorporate other important sources of uncertainty in SCs such as stochastic lead times, supplier availability and reliability, transportation disruption, contractual frictions, or strategic decentralised decision-making. Second, the informational structure is deliberately restricted to lagged demand and observed operational–financial state variables; although this reflects many practical decision environments, results may differ if real-time point-of-sale information, early-warning indicators, or richer forecasting signals are available. Third, the disruption analysis is based on stylised temporary multiplicative shocks rather than empirically calibrated disruption processes. The controlled positive- and negative-shock analysis is therefore best interpreted as a robustness check rather than a calibrated representation of a specific industry disruption. Fourth, although PPO is examined through Random Forest surrogate models, this provides post hoc interpretability rather than a direct causal explanation of the neural policy. Finally, while the policy classes span a meaningful

hierarchy from fixed heuristics to adaptive parametric policies and reinforcement learning, the set is not exhaustive; other hybrid, decentralised, or forecast-augmented controllers may offer different trade-offs between transparency and performance.

Future work could extend the proposed framework along several dimensions. A first step is to introduce additional sources of operational complexity, including stochastic lead times and supplier-side disruptions, to assess whether the observed hierarchy of controllers persists in richer and more realistic environments. Another promising direction is the exploration of hybrid control architectures that combine interpretable parametric structures with learning-based components, potentially improving the trade-off between transparency and performance. For example, residual RL could be used to augment threshold-based policies with learned corrective adjustments. Alternatively, forecast-augmented control could update policy parameters online using machine learning demand predictions while retaining familiar heuristic structures. Further research could also examine robustness to deeper forms of non-stationarity, such as permanent demand shifts or evolving customer behaviour. Finally, empirical validation using real operational data or high-fidelity digital twin implementations would be valuable for assessing practical deployability and quantifying achievable operational and financial gains in real-world settings.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/app16115712/s1, Table S1: Main DES parameterisation; Table S2: Optimisation and evaluation settings; Table S3: DES verification and bullwhip validation summary; Table S4: Pooled evaluation summaries for selected main metrics; Table S5: Controlled positive- and negative-shock PPO evaluation; Table S6: Off-baseline CCC-threshold sensitivity summaries excluding return values; Table S7: Off-baseline reward-weight sensitivity summaries excluding return values; Table S8: Random Forest surrogate fidelity for PPO action components; Table S9: Top ten Random Forest surrogate feature importances for each PPO action component; Figure S1: No-disruption demand tracking: observed lagged demand and realised current demand; Figure S2: No-disruption Simple CEM dynamics: placed decisions and raw CCC terms; Figure S3: No-disruption Simple GA dynamics: placed decisions and raw CCC terms; Figure S4: No-disruption Adaptive CEM dynamics: placed decisions and raw CCC terms; Figure S5: No-disruption Adaptive GA dynamics: placed decisions and raw CCC terms; Figure S6: No-disruption PPO dynamics: placed decisions and raw CCC terms; Figure S7: No-disruption PPO cash-collection policies; Figure S8: Demand-disruption demand tracking: lagged demand and realised demand under temporary shocks; Figure S9: Demand-disruption Adaptive CEM dynamics: placed decisions and raw CCC terms; Figure S10: Demand-disruption Adaptive GA dynamics: placed decisions and raw CCC terms; Figure S11: Demand-disruption PPO dynamics: placed decisions and raw CCC terms; Figure S12: Demand-disruption PPO cash-collection policies; Figure S13: Top-ten Random Forest feature importances by PPO action component.

Author Contributions: Conceptualization, A.B. and E.B.; methodology, A.B., E.B., S.S. and R.B.; software, A.B. and E.B.; validation, A.B. and E.B.; formal analysis, A.B. and E.B.; investigation, A.B. and E.B.; resources A.B. and E.B.; data curation, A.B. and E.B.; writing—original draft preparation, A.B. and E.B.; writing—review and editing, A.B. and E.B., S.S. and R.B.; visualisation, A.B. and E.B.; supervision, E.B., S.S. and R.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets presented in this article are not readily available because they are part of an ongoing study. Requests to access the datasets should be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Huynh, N.; Le, Q.N. From chain to capital: Supply chain risks and working capital management. *Econ. Lett.* **2025**, *247*, 112100. [CrossRef]
2. Zheng, M.; Wang, R.; Ye, J.; Li, T. How does supply chain finance enhance firms' supply chain resilience? *Int. Rev. Econ. Finance* **2025**, *102*, 104231. [CrossRef]
3. Haralambides, H.; Gujar, G. The "new normal", global uncertainty and key challenges in building reliable and resilient supply chains. *Marit. Econ. Logist.* **2023**, *25*, 623–638. [CrossRef]
4. Korder, B.; Maheut, J.; Konle, M. Simulation methods and digital strategies for supply chains facing disruptions: Insights from a systematic literature review. *Sustainability* **2024**, *16*, 5957. [CrossRef]
5. Azizian, M.; Sepehri, M.M.; Mirzapour Al-e-Hashem, S.M.J. Simulation-based models of multi-tier financial supply chain management problem: Application in the pharmacy sector. *Mathematics* **2023**, *11*, 4188. [CrossRef]
6. Badakhshan, E.; Humphreys, P.; Maguire, L.; McIvor, R. Using simulation-based system dynamics and genetic algorithms to reduce the cash flow bullwhip in the supply chain. *Int. J. Prod. Res.* **2020**, *58*, 5253–5279. [CrossRef]
7. Badakhshan, E.; Mustafee, N.; Bahadori, R. Application of simulation and machine learning in supply chain management: A synthesis of the literature using the Sim-ML literature classification framework. *Comput. Ind. Eng.* **2024**, *198*, 110649. [CrossRef]
8. Gijsbrechts, J.; Boute, R.N.; Van Mieghem, J.A.; Zhang, D. Can deep reinforcement learning improve inventory management? Performance on dual sourcing, lost sales and multi-echelon problems. *Manuf. Serv. Oper. Manag.* **2022**, *24*, 1349–1368. [CrossRef]
9. Badakhshan, E.; Ball, P. Applying digital twins for inventory and cash management in supply chains under physical and financial disruptions. *Int. J. Prod. Res.* **2023**, *61*, 5094–5116. [CrossRef]
10. Piao, Z.; Yang, K.; Su, N.; Zheng, Z. Network working capital management, supply chain concentration, and corporate performance of focal companies. *Oper. Manag. Res.* **2024**, *17*, 982–995. [CrossRef]
11. Banerjee, A.; Kundu, S.; Sivasankaran, N. Asymmetric impact of working capital efficiency on market performance of Indian firms. *Glob. Bus. Rev.* **2024**, *25*, 705–723. [CrossRef]
12. Oh, K.; Jeong, E.; Yoo, H. Effects of working capital management on small and medium-sized enterprises' profitability from the continuity of supply chain relationships. *Glob. Bus. Finance Rev.* **2023**, *28*, 51–66. [CrossRef]
13. Thomya, W.; Rangsungnoen, G.; Ritsri, U.; Nonthapot, S.; Saenchaiyathon, K. The relationship between supply chain finance and firm performance: Evidence from Thai listed firms. *Asian Econ. Finance Rev.* **2023**, *13*, 547–588. [CrossRef]
14. Pei, Q.; Chan, H.K.; Zhang, T.; Li, Y. Benefits of the implementation of supply chain finance. *Ann. Oper. Res.* **2023**, *331*, 251–283. [CrossRef]
15. Hofmann, E.; Töyli, J.; Solakivi, T. Working capital behavior of firms during an economic downturn: An analysis of the financial crisis era. *Int. J. Financ. Stud.* **2022**, *10*, 55. [CrossRef]
16. Ivanov, D. Cash flow dynamics in the supply chain during and after disruptions. *Transp. Res. Part E* **2024**, *185*, 103526. [CrossRef]
17. Wu, L.-C.; Eng, T.-Y.; Wang, C.-W. Working capital management under supply chain disruption: The role of government response during economic uncertainty. *J. Gen. Manag.* **2024**, *50*, 65–77. [CrossRef]
18. Carnes, C.M.; Cavanaugh, J.; David, P.; O'Brien, J. Cash creates value for supply chain systems, but who appropriates that value? *J. Bus. Res.* **2023**, *161*, 113834. [CrossRef]
19. The Hackett Group. The Hackett Group® 2025 Working Capital Survey: Payables Rebound, but Receivables and Inventory Lag. 18 August 2025. Available online: https://www.thehackettgroup.com/2025-working-capital-survey-payables-rebound-receivables-inventory-lag (accessed on 18 August 2025).
20. Fang, X. Blockchain applications and supply chain performance: Evidence from Chinese firms. *Technol. Anal. Strateg. Manag.* **2025**, *37*, 1724–1739. [CrossRef]
21. Badakhshan, E.; Ivanov, D. Integrating digital twin and blockchain for responsive working capital management in supply chains facing financial disruptions. *Int. J. Prod. Res.* **2025**, *63*, 7800–7834. [CrossRef]
22. Samuels, A. Digital transformation in supply chains: Improving resilience and sustainability through AI, blockchain, and IoT. *Front. Sustain.* **2025**, *6*, 1584580. [CrossRef]
23. Badakhshan, A.; Badakhshan, E.; Saad, S.M.; Bahadori, R. Integrating simulation and reinforcement learning for optimized working capital management in supply chains. *Procedia Comput. Sci.* **2026**, *277*, 263–270. [CrossRef]
24. Chaharsooghi, S.K.; Heydari, J.; Zegordi, S.H. A reinforcement learning model for supply chain ordering management: An application to the beer game. *Decis. Support Syst.* **2008**, *45*, 949–959. [CrossRef]
25. Mortazavi, A.; Arshadi Khamseh, A.; Azimi, P. Designing of an intelligent self-adaptive model for supply chain ordering management system. *Eng. Appl. Artif. Intell.* **2015**, *37*, 207–220. [CrossRef]
26. Preil, D.; Krapp, M. Bandit-based inventory optimisation: Reinforcement learning in multi-echelon supply chains. *Int. J. Prod. Econ.* **2022**, *252*, 108578. [CrossRef]

27. Fuji, T.; Ito, K.; Matsumoto, K.; Yano, K. Deep multi-agent reinforcement learning using DNN-weight evolution to optimize supply chain performance. In Proceedings of the 51th Hawaii International Conference on System Sciences (HICSS 2018), Hawaii, HI, USA, 3–6 January 2018.
28. Oroojlooyjadid, A.; Nazari, M.R.; Snyder, L.V.; Takáč, M. A deep Q-network for the beer game: Deep reinforcement learning for inventory optimization. *Manuf. Serv. Oper. Manag.* **2022**, *24*, 285–304. [CrossRef]
29. Wang, J.; Swartz, C.L.E.; Huang, K. Risk-averse supply chain management via robust reinforcement learning. *Comput. Chem. Eng.* **2025**, *192*, 108912. [CrossRef]
30. Bussieweke, F.; Mula, J.; Campuzano-Bolarín, F. Optimisation of recovery policies in the era of supply chain disruptions: A system dynamics and reinforcement learning approach. *Int. J. Prod. Res.* **2025**, *63*, 1649–1673. [CrossRef]
31. Kotecha, N.; del Rio Chanona, A. Leveraging graph neural networks and multi-agent reinforcement learning for inventory control in supply chains. *Comput. Chem. Eng.* **2025**, *199*, 109111. [CrossRef]
32. Liu, X.; Hu, M.; Peng, Y.; Yang, Y. Multi-agent deep reinforcement learning for multi-echelon inventory management. *Prod. Oper. Manag.* **2025**, *34*, 1836–1856. [CrossRef]
33. Hu, J.; Xia, L.; Huang, T.; Wu, H. A multi-agent deep reinforcement learning approach for multi-echelon inventory optimization and its application to the beer game. *Transp. Res. Part E Logist. Transp. Rev.* **2025**, *203*, 104367. [CrossRef]
34. Zhou, Q.; Yang, Y.; Ma, F.; Cheng, T.C.E. Multi-agent deep reinforcement learning for ordering and inventory allocation in a decentralized two-echelon dual-channel supply chain. *Int. J. Prod. Econ.* **2026**, *299*, 110067. [CrossRef]
35. Zhou, X.; Feng, L.; Zhu, A.; Shi, H. Uncertainty-aware joint inventory-transportation decisions in supply chain: A diffusion model-based multi-agent reinforcement learning approach with lead times estimation. *Comput. Chem. Eng.* **2026**, *207*, 109567. [CrossRef]
36. Zou, G.; Tang, J.; Yilmaz, L.; Kong, X. Online food ordering delivery strategies based on deep reinforcement learning. *Appl. Intell.* **2022**, *52*, 6853–6865. [CrossRef]
37. Serrano-Ruiz, J.C.; Mula, J.; Poler, R. Development of a multidimensional conceptual model for job shop smart manufacturing scheduling from the Industry 4.0 perspective. *J. Manuf. Syst.* **2022**, *63*, 185–202. [CrossRef]
38. Lee, Y.S.; Sikora, R. Application of adaptive strategy for supply chain agent. *Inf. Syst. e-Bus. Manag.* **2019**, *17*, 117–157. [CrossRef]
39. Fu, M.C. (Ed.) *Handbook of Simulation Optimization*; Springer: New York, NY, USA, 2015.
40. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley: Reading, MA, USA, 1989.
41. Rubinstein, R.Y.; Kroese, D.P. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte Carlo Simulation and Machine Learning*; Springer: New York, NY, USA, 2004.
42. Vanvuchelen, N.; Gijsbrechts, J.; Boute, R.N. Use of proximal policy optimization for the joint replenishment problem. *Comput. Ind.* **2020**, *119*, 103239. [CrossRef]
43. Sterman, J.D. *System Dynamics: Systems Thinking and Modeling for a Complex World*; Massachusetts Institute of Technology: Cambridge, MA, USA, 2002.
44. Pasupuleti, V.; Thuraka, B.; Kodete, C.S.; Malisetty, S. Enhancing supply chain agility and sustainability through machine learning: Optimization techniques for logistics and inventory management. *Logistics* **2024**, *8*, 73. [CrossRef]
45. Cuartas, C.; Aguilar, J. Hybrid algorithm based on reinforcement learning for smart inventory management. *J. Intell. Manuf.* **2023**, *34*, 123–149. [CrossRef]
46. Badakhshan, E.; Ivanov, D. Integrating simulation and decision trees through blockchain-enabled data sharing to prevent the cash flow bullwhip effect in supply chains. *Ann. Oper. Res.* **2025**, 1–48. [CrossRef]
47. Ivanov, D. Conceptual and formal models for design, adaptation, and control of digital twins in supply chain ecosystems. *Omega* **2025**, *137*, 103356. [CrossRef]
48. Kegenbekov, Z.; Jackson, I. Adaptive supply chain: Demand–supply synchronization using deep reinforcement learning. *Algorithms* **2021**, *14*, 240. [CrossRef]